An improved DNA binding protein prediction using Evolutionary, Structure and Physico-chemical feature combination into Chou's general PseAAC

Danial Chakma[1,1,], Samirul Alam khan[1,1,], Md. Abul Basar[1,1,]

[a]*Department of Computer Science and Engineering,*
Bangladesh University of Engineering and Technology,
Dhaka, Bangladesh

Abstract

Proteins that can bind and interact with DNA molecule are called DNA-binding proteins (DNA-BPs). These DNA-BPs play important roles in various cellular biological processes, such as DNA replication, recombination, repair & modification, and gene expression & transcription. Profound knowledge of this type of proteins would not only enhance our understanding of protein functions in such biological processes but also help developing drugs for various diseases and cancers. However, correct identification and detection of DNA-BPs using experimental methods such as ChIP-Sequencing is both extremely expensive as well as time consuming, which presents the need for faster and accurate computational methods, specifically machine learning based methods for identification of DNA-BPs. In this paper, we focus on building a new computational model to identify DNA-BPs in an efficient and accurate way. Our method combine meaningful information from the position specific scoring matrix(pssm), structural & physico-chemical, and primary protein sequence based features. After feature extraction, we have grouped features according to their methodology of feature extraction and we have employed ExtraTree model a variant of Random Forest (RF) to rank the features for recursive feature selection to find optimal feature set. Afterwards, we have trained and tested both PDB1075 and PDB186 dataset with both SVM and ExtraTree classifiers. Our proposed method named **Extralocal-DPP** demonstrates superior performance compared to the state-of-the-art predictors on both standard benchmark training dataset PDB1075 and test dataset PDB186. The tenfold cross-validation accuracy, recall, and specificity are 97.5%, 96.72%, and 98.36% respectively for the former dataset while corresponding performance metrics are 82%, 88%, and 78% respectively for the later dataset. The source code of Extralocal-DPP, along with relevant dataset and detailed experimental results, can be found at `https://github.com/DanialChakma/Extralocal-DPP`.

Keywords: DBP, Bioinformatics, Random Forest, SVM, Extra Tree.

1. INTRODUCTION

A DNA Binding protein is a protein that can bind and interact with a DNA. This class of protein is composed of DNA binding domains that includes transcription factors, nucleases and histones. The transcription factors regulate the process of transcription, while nucleases can split DNA molecules. Histones, on the other hand, are involved in chromosome packaging in the cell nuclei. Protein DNA binding interaction is shown in Fig. 1: a transcription factor is bound to a DNA (left), while the restriction enzyme EcoRV is interacting with its target DNA (right). Thus, DNA-binding protein perform two main function: firstly, they organize and compact DNA and secondly, they regulate and affect various cellular processes like transcription, DNA replication & recombination, repair and modification. Therefore, the DNA-BPs can potentially be used for drug development in treating genetic diseases and cancers Gurova (2009) and Leung (1013). This is why developing efficient and highly accurate methods to identify DNA-BPs is a very important research

*Corresponding authors

Email addresses: `danial08cse@gmail.com` (Danial Chakma), `samir@email.com` (Samirul Alam khan), `basar@email.com` (Md. Abul Basar)

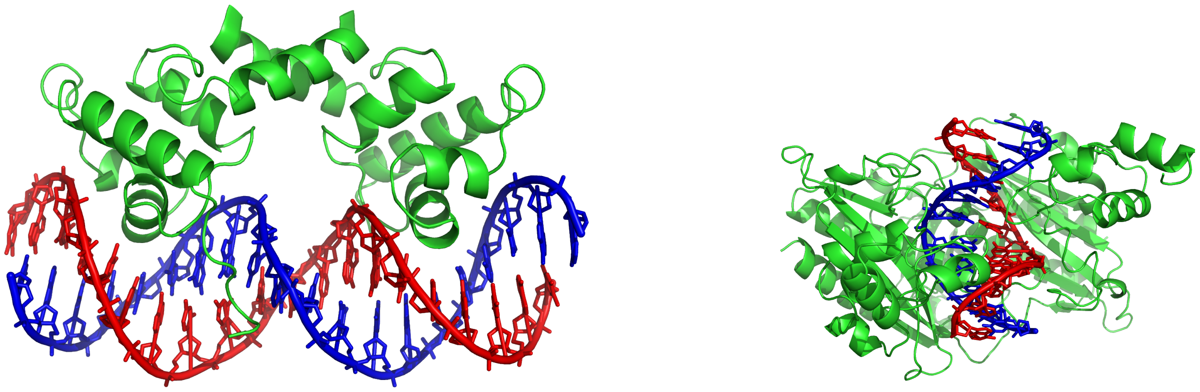


Figure 1: DNA-binding proteins bound to respective target DNAs. (**Left**) The lambda repressor helix-turn-helix transcription factor bound to its DNA target. Created from PDB 1LMB. Image source: Zephyris (2018b). (**Right**) The restriction enzyme EcoRV in a complex with its substrate DNA. Created from PDB 1RVA. Image source: Zephyris (2018a).

problem in the field of molecular biology. Traditionally, the DNA-BPs have been identified through different experimental methods. These include filter binding assays Helwa (2010), genetic analysis Freeman (1995), X-ray crystallography Chia-Cheng Chou (2003), chromatin immunoprecipitation on microarrays Buck (2004) etc. However, as these experimental methods are costly and time consuming, researchers have started to rely on computational methods to identify DNA-BPs. These methods can largely be categorized into two groups: structure based methods and sequence based methods. Structure-based methods depend on the structural information of the protein sequences. These include high-resolution 3D structure, accessible surface area, torsion angles, structure motifs etc. Stawiski (2003) did the pioneering work in identifying DNA-BPs using structural information. They extracted features from the detailed atomic structure of the protein and then employed a three-layer artificial neural network (ANN). Ahmad (2004), on the other hand, used a two-layer neural network with features calculated solely from bulk electrostatic properties. Szilgyi (2006) subsequently proposed a fast and efficient method to predict DNA-BPs from only the amino acid sequences and low-resolution, C^{α}-only protein models. Their predictor is available as a web-server called DNABIND. Gao (2008) proposed DBD-Hunter that applies structural alignment and evaluation of a statistical potential to identify DNABPs. Gao (2009) subsequently proposed DBD-Threader, for the prediction of DNA-binding domains and associated DNA-binding protein residues. While this method uses a template library composed of DNA-protein complex structures, it requires only the target protein's sequence for its classification. This independence from structural information makes the predictor very useful, while its performance remains comparable with DBDHunter. Examples of other structure-based methods can be found in Zhao (2010), Nimrod (2010), Zhou (2011) and Szabov (2012). Structure-based predictors are applicable only when the structural information of a candidate protein is known. While the post-genomic era witnesses a rapid growth in sequence known proteins, the structure of many of these proteins still remain undiscovered. The predictors that solely rely on structural information of proteins are thus limited in their use. Sequence based methods, on the other hand, attempt to identify the DNA-BPs from the amino acid sequence by extracting various discriminating features. Some predictors may additionally rely on some structural features for improved prediction accuracy when the protein structure is known. Examples of prominent sequence based predictors of DNA-BPs can be found in Kumar (2007), Kumar (2009), Lin (2011), Zhao (2010), Dong (2015), Liu (2015), Xu (2015), Motion (2015), Waris (2016), Zhou (2016), Paz (2016), Wei (2017) and Chowdhury (2017). Kumar (2007) used evolutionary information from the Position Specific Scoring Matrix (PSSM) for protein representation. The PSSM profile of each protein was generated from PSI-BLAST Altschul (1997) by searching the non-redundant (nr) protein database using three iterations with e-value cutoff set to 0.001. They applied Support Vector Machine (SVM) Boser (1992) as the learner. Available as a webtool called DNAbinder, the performance of their predictor depends on the

Cite this article as: Danial et al. An improved DNA binding protein prediction using Evolutionary, Structure and Physico-chemical feature combination into Chou's general PseAAC. Journal of Theoretical Biology (2019), `https://doi.org/`

quality of PSSM profiles, which is heavily dependent on the database being searched for homology information. To eliminate this dependency, DNA-Prot was proposed by another group Kumar (2009). This predictor used features such as frequency of amino acid residues and groups, predicted secondary structure (PredSS) information from PSIPRED McGuffin (2000), physico-chemical properties from AAIndex database Kawashima (2007). To reduce the feature vector size, they applied Correlation-based feature subset selection method (CFSS). Lin (2011) incorporated the Grey model Julong (1989) parameters in the general form of Chou's PseAAC Chou (2011) for protein sequence representation. They then trained their model, iDNA-Prot, using Random Forest (RF) Breiman (2001). Lou (2014) introduced a predictor called DBPPred, where amino acid composition, PSSM scores, PredSS and predicted relative solvent accessibility (PredRSA) were used as features. They then used Random Forest to rank the features, followed by a wrapper method. They used Gaussian Nave Bayes (GNB) as the final classifier. They compared their predictor with prior ones using an independent dataset called PDB186, comprising equal number of DNA-binding and non DNA-binding proteins. This dataset has subsequently been used in performance evaluation of many other predictors. Liu (2015) used amino acid distance-pair coupling information into Chou's general form of PseAAC Chou (2011). To reduce the dimension of the feature vector and to speed up the prediction process, they also used amino acid reduced alphabet profile Peterson (2009). They then applied SVM with RBF kernel to produce the prediction tool called iDNA-Prot—dis. To train and assess their predictor using cross-validation, they prepared a stringent balanced dataset of 1075 protein samples. This benchmark dataset has subsequently been referred to as PDB1075 and has been widely used in literature for cross-validation. We have also used this dataset in our work and provide a detailed description of the dataset later in the paper. In addition to preparation of the benchmark dataset, a key contribution of Liu et al.'s work was reimplementation of major earlier predictors and measuring their cross-validation performance using this benchmark dataset. This paved the way for subsequent predictors to be compared with prior art in an apple for apple comparison.

In 2015, Liu (2015) presented another predictor called iDNAPro-PseAAC. They used profile-based representation of the protein sequence and then used PseAAC with the 3rd order sequence order effect. Dong (2015) used Auto-Cross Covariance (ACC) transformation with amino acid k-mer compositions and physicochemical properties. They then used SVM to train the predictor, widely known as Kmer1 + ACC. Wei et al. proposed Local-DPP Wei (2017), where local pseudo position specific scoring matrix (Local Pse-PSSM) features have been used. In this approach, the locally conserved protein information is captured by fragmenting the PSSMs into several equally sized sub-PSSMs. Finally, all the local features are fed into the Random Forest algorithm to learn the classification model.

Very recently, Chowdhury et al. developed iDNAProt-ES Chowdhury (2017), that utilizes both the evolutionary profile and structural information of proteins to identify their DNA-binding functionality. From the PSSM profile, they extracted features like amino acid composition Chou (1995), Dubchak features Dubchak (1997), bigram, auto-covariance, segmented distribution etc. They extracted several structural features using SPIDER2 Yang (2017). Recursive feature elimination was then used to extract an optimal set of features, followed by SVM with linear kernel to learn the model. Their proposed method significantly outperforms the existing state-of-the-art predictors on standard benchmark dataset in cross-validation testing.

While significant amount of work has been done in this field, there is still room for improvement in different ways. Firstly, the prediction performance could be improved further. Secondly, many of the existing predictors use feature extraction techniques that are time consuming, some use sophisticated prediction models. To improve performance, we therefore propose a DNA binding protein predictor that extracts evolutionary, predicted structure, and physicochemical features from the protein sequence, that has a fast and simple prediction model and that outperforms the existing predictors.

2. MATERIAL AND METHODS

We have followed Chou's Chou (2011) guideline to establish a useful tool for any protein attribute prediction problem. These steps can be summarized as follows:

1. Prepare or obtain a stringent benchmark dataset to train and test the predictor.

2. Represent the protein samples through a feature vector that is expressive enough so that the downstream processes can extract and utilize intrinsic information relevant to the attribute to be predicted.
3. Develop a powerful algorithm for the prediction process.
4. Objectively evaluate the predictor.
5. Make the predictor and source code publicly available.

Next, we describe our methodology in accordance with these 5 step rules.

2.1. ***Benchmark dataset***

As mentioned in the Introduction section, Liu et al. Liu (2014) prepared a stringent balanced dataset of 1075 protein samples. This dataset is widely known as PDB1075 and has been extensively used in literature for cross-validation. As described in their paper, the DNA-binding proteins were extracted from Protein Data Bank (PDB), December, 2013 version, by searching the mmCIF keyword of 'DNA binding protein' through the advanced search interface. The resulting proteins were filtered further as follows. Proteins shorter than 50 residues were excluded. Proteins containing the residue 'X' were removed because they contained unknown residue. Less than 25% sequence similarity between any protein pair was ensured by using PISCES Wang (2005). A set of 525 DNA-binding proteins was thus obtained. The negative set of 550 proteins was prepared by randomly selecting from other proteins in PDB. The same strict filtering criteria, as mentioned above, was also applied to this negative set. Thus the benchmark dataset had a total of 525+550=1,075 protein samples. We have also used another smaller benchmark dataset for independent testing. Lou et al. Lou (2014) prepared this dataset of 93 DNAbinding and 93 non DNA-binding proteins. This dataset is widely known as the PDB186 dataset. All the sequences in this set are guaranteed to be no smaller than 60 residues and they do not contain any X character. Pairwise sequence identity of no more than 25% was ensured in this dataset using BLASTCLUST Altschul (1997).

2.2. ***Protein Sample Representation***

With the explosive growth of biological sequences in the postgenomic era, one of the most important, albeit difficult, problems in computational biology is how to express a biological sequence with a discrete model, yet capture considerable amount of sequence-order information. In a discrete model, each protein is represented by a fixed length feature vector that is independent of the protein sequence length. This model is preferred because all the existing machine-learning algorithms can only handle feature vectors but not sequence samples, as elucidated in a comprehensive review Chou (2015). However, a vector defined in a discrete model may completely lose all the sequence-pattern information. To overcome this, the concept of Pseudo Amino Acid Composition, or PseAAC in short, was proposed by Chou (2001a). Since then, PseAAC has been widely used in nearly all the areas of computational proteomics (see, for example, Behbahani (2016); Khan (2017); Krishnan (2018); Meher (2017); Mei (2018); Song (2018); Yu (2017) as well as a long list of references cited in Chou (2017)). Because of its wide adoption, several open access softwares such as propy Cao et al. Cao (2013) and PseAAC-General Du (2014) were established. Encouraged by the successes of using PseAAC to deal with protein/peptide sequences, the concept of PseKNC (Pseudo K-tuple Nucleotide Composition) Chen (2014) was developed for generating various feature vectors for DNA/RNA sequences that have proven very useful as well (Chen (2018); Lin (2014); Liu (2017a)). Particularly, recently a very powerful web-server called Pse-in-One Liu (2015) and its updated version Pse-in-One 2.0 Liu (2017a) have been established that can be used to generate any desired feature vectors for protein/peptide and DNA/RNA sequences according to the need of users' studies. In the current study, we are to use Chou's general PseAAC to represent protein samples and develop a model for predicting DNA-BPs. Let a protein sequence P of length L be written as:

$$P = R_1\ R_2\ R_3\ R_4 \ldots R_L \tag{1}$$

where R_i indicates amino acid residues from 20 letter alphabet set Σ. Positive samples or DNA binding proteins are given +1 as labels and negative samples or non-dna binding protein are given -1 as labels. The PseAAC of the protein can be represented as follows:

$$P = [\Psi_1\ \Psi_2\ \ldots\ \Psi_u\ \ldots\ \Psi_\Omega\ ]^{\mathrm{T}} \tag{2}$$

Here, the classical Amino Acid Composition (AAC) is represented by subscripts $1 \leq u \leq 20$ and the subsequent features express sequence order information through one or more different schemes. The sequence order related features that we have extracted can largely be divided into two categories: position independent and position specific. Among the position independent features are Dipeptides (Dip), Tripeptides and nGapped-Dipeptides (nGDip). These features do not depend on any specific position in the amino acid sequence. These features have widely been used in the literature of protein attribute prediction. The position specific features, on the other hand, were used only in Rahman (2018). We will describes different feature extraction techniques from the protein sequences briefly below.

2.2.1. ***PSSM Features Extraction***

PSSM stores PSI-BLAST generated evolutionary information of protein sequences. A given protein sequence $\mathbf{S}$ is expressed as $S_1S_2\cdots S_L$, where $S_i(1 \leq i \leq L)$ is the amino acid (residue) that appears at the i^{th} position of the sequence $\mathbf{S}$, and L is the length of $\mathbf{S}$. The size of the PSSM matrix $\mathbf{P}_{pssm}$ is $L \times 20$ (L rows and 20 columns) formulated as follows:

$$\mathbf{P}_{pssm} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,19} & p_{1,20} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,19} & p_{2,20} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ p_{m,1} & p_{m,2} & \cdots & p_{m,19} & p_{m,20} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ p_{L,1} & p_{L,2} & \cdots & p_{L,19} & p_{L,20} \end{bmatrix} \tag{3}$$

where L is the length of the primary amino acid sequence and $p_{m,n}$ is the score of the amino acid residue R_m at the m^{th} location of primary protein sequence changing to amino acid residue R_n during the evolutionary process. Then, we normalized the PSSM matrix using logistic function and standardization method using following two equation.

$$e_{m,n} = \frac{1}{1 + e^{-s*p_{m,n}}}, \; 0 \leq e_{m,n} \leq 1 \tag{4}$$

$$\begin{aligned} s_{m,n} &= \frac{p_{m,n} - \bar{p_m}}{\sqrt{\frac{1}{20}\sum_{k=1}^{20}(p_{m,k} - \bar{p_m})^2}} \\ \bar{p_m} &= \frac{1}{20}\sum_{j=1}^{20} p_{m,j} \end{aligned} \tag{5}$$

where, s is a user defined scaling parameter. Each of original PSSM elements $p_{m,n}$ is transformed into $e_{m,m}$ by the logistic fucntion defined by Eq. 4 and is transformed into $s_{m,n}$ defined by Eq. 5. We denote these two transformed matrices as $\mathbf{P}_{sig}$, $\mathbf{P}_{\mathbf{std}}$ respectively. We used best performing value s(=6) for searching best values of the performance matrices.

Segmented Composition(S-PSSM). Segmented composition is the sum of a residue type of the particular segment of pssm matrix along the rows. Is defined in the following equation and graphically depicted in Fig. 2a.

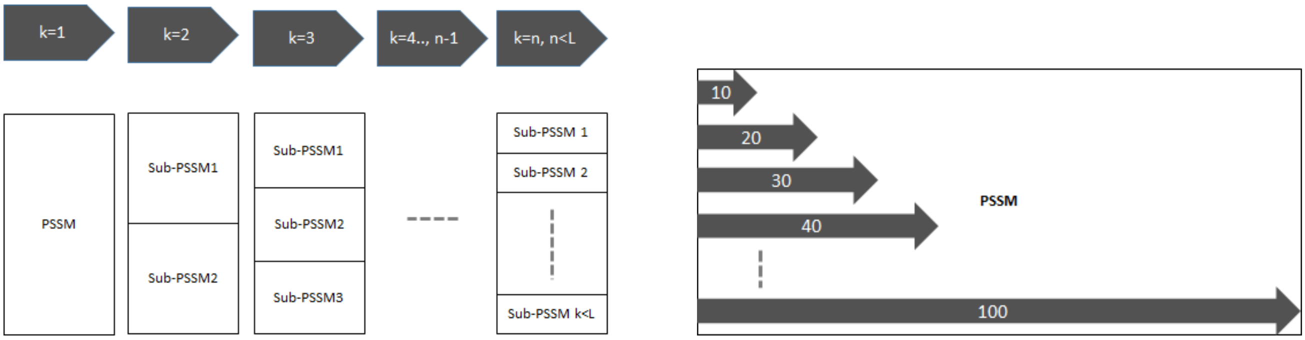
$$\begin{gathered} f_{\lambda,m,n} = \sum_{m=(\lambda-1)*\lfloor \frac{L}{k} \rfloor}^{\lambda*\lfloor \frac{L}{k} \rfloor} p_{m,n}, \\ 1 \leq \lambda \leq k; 1 \leq k \leq L; \\ 1 \leq m \leq L; 1 \leq n \leq 20. \end{gathered} \tag{6}$$

where, $p_{m,n} \in \{\mathbf{P}_{sig}, \mathbf{P}_{\mathbf{std}}\}$, and k is the number of segment of the original PSSM matrix. Each of the λ sub-matrices contains $\lfloor \frac{L}{k} \rfloor$ rows except the last (k^{th}) sub-matrix which contain $L - \lfloor \frac{L}{k} \rfloor$ rows if length L is not divisible by k. The number of features extracted from this technique is $k \times 20 \times 20$. We used k(=18) value, therefore, total number of features extracted is $18 \times 400 = 7200$. Since, we used both sigmoid normalized and standardized pssm matrix $(\mathbf{P}_{sig}, \mathbf{P}_{std})$, total number of S-PSSM features is $2 \times 7200 = 14,400$.

Cumulative Percentile PSSM(CP-PSSM). Cummulative Percentile feature is the same as the Segmented PSSM except we are now segmenting original PSSM matrix with percentile $10\%, 20\%, \cdots, 100\%$. We then sum up each of this segment to calculate feature set. Procedure is graphically depicted in Fig. 2b. Therefore, number of features from this technique is counted up $10 \times 400 = 4000$. As, we considered both forward and reversed way of percentile, so total true number of features from one pssm matrix is counted up to $2 \times 4000 = 8000$. Since, we utilized both normalized and standardized pssm matrix, total number of features reached upto $16,000$ from this techniques.

2.2.2. ***Protein Secondary Structure Features***

We used SPIDER2 to extract predicted secondary structure features. SPIDER2 is a freely available python secondary structure predictor that provides information on accessible surface area, torsion angles, structure motifs in each amino acid residue position. We then extract a novel set of features from the information provided by SPIDER2 as SPD file. The feature extraction is enumerated here in details:



(a) **Segmented feature extraction.** (b) **Cummulative percentage feature extraction.**

Figure 2: In Fig. 2a, PSSM is segmented along the row axis. Parameter k represent number of fragment sub-matrices, e.g. k=2 represent PSSM matrix is divided into two equal sub-matrices by rows, k=3 represent PSSM is divided into three equal sub-matrices by rows and so forth. In Fig. 2b, PSSM is segmented according to percentage of the total rows(Sequence length) of the PSSM matrix. For reverse cummulative percentage feature case, we just need to reverse the order of the rows of PSSM. Here, PSSM matrix is represented by rectangle whose width represent number of rows and height represent 20 standard amino acid column.

Secondary Structure Occurence. There are three types of structural motifs in proteins: -helix (H), -sheet (E) and random coil (C). Secondary Structure Occurrence is the count or frequency of each type present in amino-acid residue positions.

$$\begin{aligned}\mathrm{SSO(i)} &= \sum_{j=1}^{L} SM_{i,j};\ 1 \le i \le 3.\\ SM_{i,j} &= \begin{cases} 1, & if\ \mathrm{SS}_j = \mu_i. \\ 0, & else. \end{cases}\end{aligned} \tag{7}$$

where, μ_i one of three structural motifs and SS_j is the structural motif of the protein sequence at position j.

Secondary Structure Composition. This feature is the normalized secondary structure(motif) occurrences present in the primary amino acid protein sequences.

$$\begin{aligned}\mathrm{SSC(i)} &= \frac{1}{L}\sum_{j=1}^{L} SM_{i,j};\ 1 \le i \le 3.\\ SM_{i,j} &= \begin{cases} 1, & if\ \mathrm{SS}_j = \mu_i. \\ 0, & else. \end{cases}\end{aligned} \tag{8}$$

where, μ_i one of three structural motifs and SS_j is the structural motif of the protein sequence at position j.

Accessible Surface Area Composition. The accessible surface area composition is the normalized sum of accessible surface area defined by:

$$\text{ASA-C(i)} = \frac{1}{L}\sum_{j=1}^{L} ASA(j); \tag{9}$$

Torsional Angles Composition. For four diferent types of torsional angles: ϕ, ψ, τ, and θ, we first convert each of them into radians from degree angles and then take sine and cosine of the angles at each residue position. Thus we get a matrix of dimension $L \times 8$. We denote this matrix by T in this section for torsional angles. Torsional angles composition is defined as:

$$\mathrm{TAC(n)} = \frac{1}{L}\sum_{m=1}^{L} T_{m,n}; (1 \le n \le 8). \tag{10}$$

Structural Probability Composition. Structural probabilities for each position of the amino acid residue are given in spd3 file as a matrix of dimension $L \times 3$. We denote it by P. Structural probabilities composition is defined as:

$$\mathrm{SPC(n)} = \frac{1}{L}\sum_{m=1}^{L} P_{m,n}; (1 \le n \le 3). \tag{11}$$

Torsional Angles Bigram. Bigram for the torsional angles is similar to that of PSSM matrix and defined as:

$$\text{TAB(k,l)} = \frac{1}{L}\sum_{i=1}^{L-1} T_{i,k} * T_{i+1,l}; \quad (1 \leq k \leq 8; 1 \leq l \leq 8). \tag{12}$$

Structural Probablities Bigram. Bigram of the structural probabilities is similar to that of PSSM matrix and defined as:

$$\text{SPB(k,l)} = \frac{1}{L}\sum_{i=1}^{L-1} P_{i,k} * P_{i+1,l}; \quad (1 \leq k \leq 3; 1 \leq l \leq 3). \tag{13}$$

Torsional Angles Auto-Covariance. Torsional Angles Auto-Covariance: This feature is also derived from torsional angles and defined as:

$$\text{TA-AC(k,l)} = \frac{1}{L}\sum_{i=1}^{L-k} T_{i,l} * T_{i+k,l}; \quad (1 \leq k \leq DF; 1 \leq l \leq 8). \tag{14}$$

Structural Probability Auto-Covariance. This feature is also derived from structural probabilities and defined as:

$$\text{SP-AC(k,l)} = \frac{1}{L}\sum_{i=1}^{L-k} P_{i,l} * P_{i+k,l}; \quad (1 \leq k \leq DF; 1 \leq l \leq 3). \tag{15}$$

2.2.3. Physico-chemical Features

Dubchak features. Theses features were previously used for protein fold recognition and protein subcellular localization. They group the amino acid residues according to various physicochemical properties polarity, solvability, hydro-phobicity etc. and calculates the composition, transition and distribution of these groupings. The size of this feature vector is 147.

2.2.4. Primary Sequence Based Features

We also used sequence based features extracted from only primary protein chain. Composition based features, amino acid composition(AAC), Dipeptide composition, Tri-peptide composition all grouped togather to form n-gram features. We also employed percentile amino acid composition and dipeptide composition, which we call them togather as p-ngram(percentile n-gram). Total number of n-gram feature accounted to 20+400+8000=8420. While total p-ngram feature accounted to 10*20+10*400=4200.

2.2.5. Quasi-sequence order features

Quasi-sequence order feature was used by Chou for prediction of sub-cellular location. We directly adopted that feature into our DNA-binding protein prediction. We extracted 86 such QSO features extracted from protein sequence and distance matrices.

2.2.6. Summary of Features

To give a clear summary of the features generated and subsequently used in this paper, we tabulated feature extraction method and number of features in Table 1.

Table 1: Summary of features

Group	Feature Name	Total Features
A	pssm	30,400
B	spd3	201
C	ctd	147
D	ngram	8420
E	p-ngram	4200
F	qso	86
Total Features		43,454

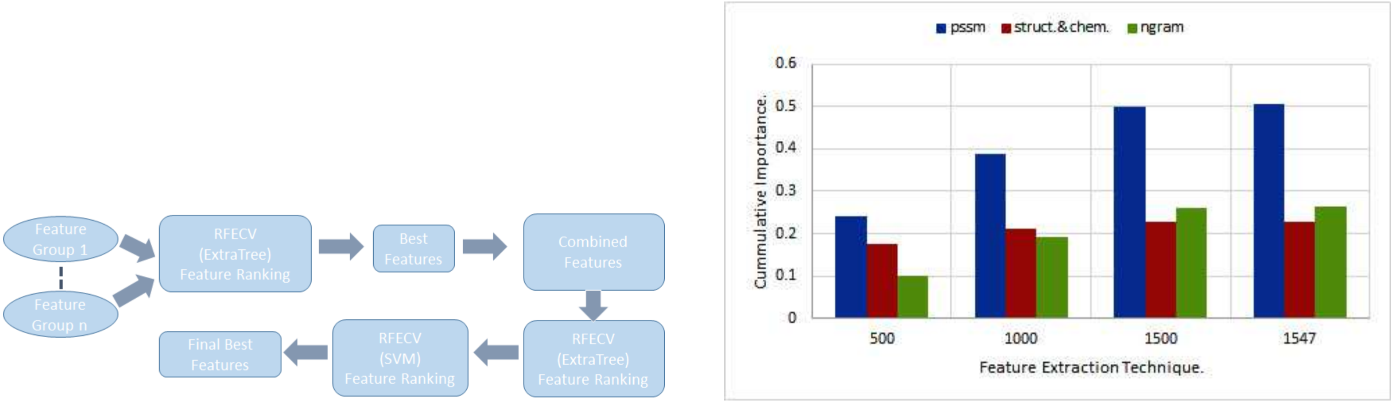
2.3. Feature Selection

2.3.1. Recursive Feature Selection

In recursive feature selection(RFS), we performed RFS on individual group first using Extra Tree Classifier. We avoided combination of all feature group then perform RFS due to large feature space. The steps in RFS are given below:

1. Performed recursive feature selection on individual group first.
2. Combine best features from step 1.
3. Perform recursive feature selection(Coarse grain) on combined best feature set from step 2.
4. Perform recursive feature selection(Fine grain) on combined best feature set from step 3.
5. Collect all the best features from step 4.

In step 1 RFS, step size of 512, 128, 64, 32 and ExtraTree estimator were used sequentially one after another to run RFE algorithm. Likewise, in step 3, step size of 512, 128 and ExtraTree estimator were used to run RFE algorithm. We followed the same procedure, in step 4, but used somewhat fine grained step size of 32, 16 and SVM(support vector machine) estimator was used.



(a) Feature selection process.

(b) Feature importances of different feature extraction techniques.

Figure 3: (**Left**) Recursive feature selection process. (**Right**) Categorized feature importance based on Extra-tree ranking. The aggregate feature importance score is better for each category as number of top ranked feature increased. PSSM: position specific scoring matrix features, Struct.&Chem.:spd3, ctd, quasi-sequence order features. ngram: both ngram and percentile ngram (pngram) features from primary protein sequence.

2.4. *Prediction Algorithms And Classifiers*

Several classification algorithms are used in the experiments in this paper. In this section, we provide very brief description of the algorithms.

Decision Tree: Decision tree Safavian and Landgrebe (1991) classifiers are tree based classifier where attributes are used in a hierarchical manner to find the labels of the samples as leafs of a decision tree.

Random Forest: Random forests Tin(1995) Ho. (1995) is a ensemble of decision trees learned by randomly selected features at each iteration of the algorithm.

Extra Tree Classifier: Extra tree classifiers Pierre G.Pierre Geurts and Wehenkel (2006) are similar to random for in each iteration of the algorithm it also uses sub-samples of the dataset.

SVM: Support Vector machine is a maximum margin classifier that tries to separate two classes in the case of binary classification. It uses a decision rule of the following form.

$$h(\vec{x}) = sign(\sum_j \alpha_j y_j (\vec{x} \cdot \vec{x_j}) - b)$$

Here, x_j is the support vectors that define the maximum margin.

2.5. *Predictor Evaluation*

We have selected following testing methods and performance metrics to evaluate our works.

2.5.1. *Testing methods*

Several testing methods exist that can assess the quality of the learning model while it is being trained as well as after the training has been completed. These include jackknife cross-validation, 10-fold cross-validation test, independent test etc. We briefly describe these techniques below.

Jackknife Cross Validation. In jackknife cross-validation, one sample from the training set is set aside. The remaining part is used to train the predictor. Then the set-aside sample is used to test the model. This is repeated N times, where N is the size of the training set. In each iteration, the testing sample is always different from previous testing samples, so that all samples are considered once as the testing sample. Though this technique executes slowly compared to other testing techniques, it can generate impartial results with small variance. This technique has been used in this paper. Since one sample is left out in each iteration, this technique is also widely known as Leave-one-out cross-validation technique.

Independent Testing. In independent testing, the testing dataset is completely different from training dataset. After the model is completely trained using the training set, independent testing is performed using the testing dataset. The distribution of the testing dataset should be similar to that of the training dataset. Otherwise, output of this testing strategy may be misleading. In this paper, we have used the PDB186 dataset for independent testing.

10-fold Cross Validation. In 10-fold cross-validation technique, training dataset is divided into 10 equal parts. Among these 10 parts, one part is used for testing and other 9 parts are used to train the model. This is repeated 10 times so that each part gets to be used for testing exactly once. We have employed this technique during feature selection step of both group feature selection and recursive feature selection.

2.5.2. Performance metrices

As performance metrics, we have used in this paper accuracy, sensitivity, specificity and Matthew's Correlation Coefficient (MCC). These are well-established performance metrics in the literature. We have also analyzed the Area Under Receiver Operating Characteristic Curve (ROC-Curve). The samples in the dataset can be categorized into two classes: the positive class and the negative class. True positive(tp) is the number of actual positive instances that are predicted as positive, False negative(fn) is the number of actual positive instances that are predicted as negative, False positive(fp) is the number of actual negative instances that are predicted as positive, True negative(tn) is the number of actual negative instances that are predicted as negative. Let P, N, TP, TN, FP, FN respectively denote the number of positives, negatives, true positives, true negatives, false positives and false negatives. Then we can define the relevant performance metrics accuracy(AC), precision(PR), recall(RC), f-measure(F1), and Matthew's Correlation Coefficient(MCC) by the following set of equations:

$$\begin{cases} AC = \dfrac{TP + TN}{P + N} \\ RC = \dfrac{TP}{TP + FN} \\ SP = \dfrac{TN}{TN + FP} \\ MCC = \dfrac{TP * TN - FP * FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}} \end{cases} \tag{16}$$

However, in the form of Eq.16, these metrics lack intuitiveness and is not easy-to-understand for most biologists. In particular, the interpretation of MCC is not at all intuitive in this form, although it is very important in measuring the stability of a prediction method. Therefore, we adopt the formulation based on Chou's symbols Chou (2001) that was recently proposed in Chen (2014) and Xu (2013) as follows: Let $N^+(N^-)$ be the total number of positive (negative) samples in the dataset and $N^+_-(N^-_+)$ be the number of positive (negative) samples that were incorrectly predicted. The relationship between the symbols used in Eq.16 and Chou's symbols just introduced can be given by the following equation:

$$\begin{cases} TP = N^+ - N^+_- \\ TN = N^- - N^-_+ \\ FP = N^-_+ \\ FN = N^+_- \end{cases} \tag{17}$$

And the performance metrics can then be redefined as:

$$\begin{cases} AC = 1 - \dfrac{N^+_- + N^-_+}{N^+ + N^-} \\ RC = 1 - \dfrac{N^+_-}{N^+} \\ SP = 1 - \dfrac{N^-_+}{N^-} \\ MCC = \dfrac{1 - \left(\frac{N^+_-}{N^+} + \frac{N^-_+}{N^-}\right)}{\sqrt{\left(1 + \frac{N^-_+ - N^+_-}{N^+}\right)\left(1 + \frac{N^+_- - N^-_+}{N^-}\right)}} \end{cases} \tag{18}$$

From the definitions in Eq.18, the interpretation of each of the performance metrics is much more intuitive and easier-to-understand. For example, when all the instances of the positive(negative) class are correctly predicted, we have $N^-_+ = 0(N^+_- = 0)$ and thus recall (specificity) of the classifier is 1. On the contrary, if all the positive (negative) instances are incorrectly predicted, then $N^+_- = N^+(N^-_+ = N^-)$. Therefore, recall (specificity) becomes 0. For a perfect classifier, we have $N^-_+ = N^+_- = 0$ and both accuracy and MCC become 1 in this case. On the other hand when all the samples are misclassified (i.e. $N^+_- = N^+$ and $N^-_+ = N^-$), then accuracy and MCC becomes 0 and -1 respectively. For a random predictor, we can expect $N^+_- = \frac{N^+}{2}$ and $N^-_+ = \frac{N^-}{2}$, which results in an accuracy of 0.5 and an MCC of 0.

The advantages of Chou's intuitive metrics have been analyzed and concurred by a series of studies published very recently (see, e.g., Chen (2017);Chen (2018);Cheng (2017a);Cheng (2016);Feng (2017);Jia (2015);Jia (2016);Liu (2016a); Liu (2016b);Liu (2016c);Liu (2017c);Qiu (2017a);Qiu (2017b);Qiu (2016a)). It is important, however, to call out that the set of metrics, as described above, is valid only for the single-label systems (in which each protein only belongs to one functional class). For the multi-label systems (in which a protein might belong to several functional classes), whose existence has become more frequent in systems biology Cheng (2017a,b,c,d,e), systems medicine Cheng (2016, 2017e) and biomedicine Qiu (2016b), a completely different set of metrics as defined in Chou (2013) is needed.

In addition to the performance metrics described above, we have also analyzed the Area Under Receiver Operating Characteristic curve (ROC-Curve). The ROC-Curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. TPR is Sensitivity(Recall), while $(1 - Specificity)$ gives FPR. When ROC-Curve gets close to the left upper corner in the graph, it indicates better performance. In this case, we get higher values for Area Under ROC-Curve (AUROC-Curve).

2.5.3. Experimental setup and packages

We have conducted experiments using Python language (version 3.5.0 or above) on windows machines with the following configurations:

1. Processor: Intel Core i5-8250U CPU @ 1.60GHz x 4, with 8 logical core.
2. Operating System: Windows 10, 64-bit OS.
3. Memory(RAM): 8 GB RAM.

Random Forest(RF), Extra tree(ET), and SVM etc. machine learning algorithms were used from Scikit learn python packages and to plot graph we used python Matplotlib 3.0 library.

2.6. Predictor Availability

Extralocal-DPP is freely available as python script at `https://github.com/DanialChakma/Extralocal-DPP`. However, user-friendly and publicly accessible web-servers represent the future direction for developing practically more useful predictors, as pointed out in Chou (2009). Such web-based predictors, as given in a series of recent publications Chen (2017), Jia (2016), Feng (2017), Liu (2017c), Chen (2018), Cheng (2017a), Liu (2015), Chen (2016a), Chen (2016b). Liu (2017b), Qiu (2016a) and Zhang (2016), will significantly enhance the impact of theoretical work by attracting usage from the broad experimental scientists. These practically useful web-servers are starting to have increasing impact on medical science Chou (2015), driving medicinal chemistry into an unprecedented revolution Chou (2017). However, due to current financial constraint, we are unable to establish a publicly accessible web server. In future, we will definitely have a publicly accessible web server for wide adoption of **Extralocal-DPP** as DNA-binding protein predictor.

3. RESULT AND DISCUSSION

3.1. Impact of sigmoid scaling factor on performances

To measure the best scaling factor of sigmoid function in Eq. 4, we performed an experiment with varying number of s values ranging from 1 to 10, to see which value gives best performance scores on training set. We used both segmented and cummulative PSSM features to measure 10-fold cross-validation performance metric accuracy, specificity, recall, mathews correlation coefficient(mcc). In this experiment, we used repeated RFECV-ExtraTree with varying number of step from coarse to fine grain to select best features, after that an SMOTE(Synthetic Minority Oversampling Technique) operation to tackle class imbalance problem is performed. Then, another RFECV-ExtraTree feature selection step is performed before feeding into final ensemble classifier ExtraTree. From Fig. 4, we see that for scaling factor 5.5 to 6 the performance score is the highest. As we didn't tested for fractional scaling factor that why we selected 6 as the optimum scaling factor. In subsequent experiment, when we refer to sigmoid normalized pssm features then it means sigmoid normalized with a scaling factor of 6.

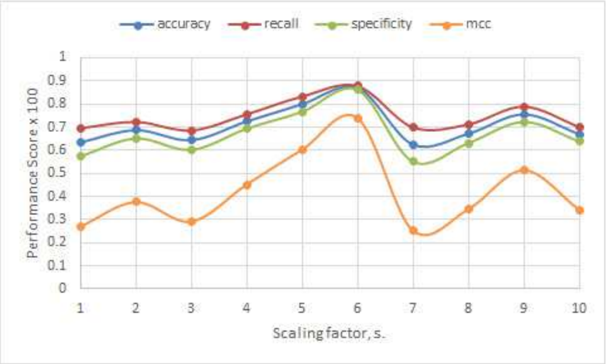


Figure 4: Cross validation (10-fold) Performance Score with varying number of scaling factor values.

3.2. Discriminant Weight Visualization.

To study the discriminant power of the different features, we calculated the discriminant weight vector in the feature space. This vector is also needed during the RFE step and is calculated following the steps used in Guyon (2002). The discriminative weights of top 32 features are shown in Fig. 5. The feature names are encoded as follows:

- A feature starting with the prefix P followed by number and sub-string "_NG_" such as "P10NG_" is a percentile n-gram(pngram) and starting with prefix "NG_" is a ngram features. The integer that follows "P" is the percentile being considered and sub-string that follows "_NG_" is a particular amino-acid composition or dipeptide composition.

- A feature starting with the prefix _SecondaryStr is a secondary structure features, _SolventAccessibility and _Polarity are physico-chemical features.

- A feature starting with the prefix BG_ or "RBG_" or "SBG_" or "RPBG_" or "PBG_" are all PSSM feature. The integer that follows is the particular percentile(or segment) and bigram composition from PSSM matrix.

There are 14 features with positive scores and 18 features with the negative scores. The absolute weights of positive weighted feature set ranges [0.26,2.58] while that of negative weighted feature set ranges [0.04,1.90]. The decrease of importance is gradual as we move to lesser ranked features, and the pattern of the decrease is almost identical for both set of features. The features with positive (negative) scores contribute in prediction of the positive (negative) class.

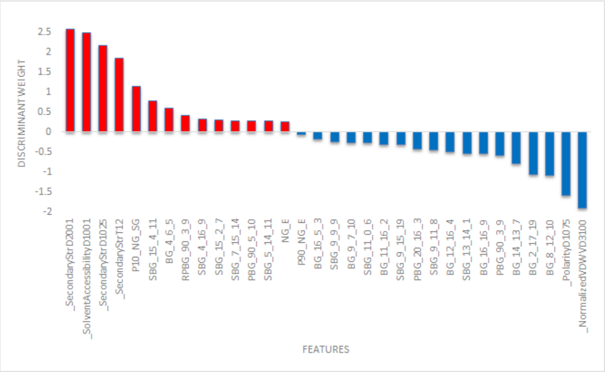


Figure 5: The discriminative weight of top 32 features.

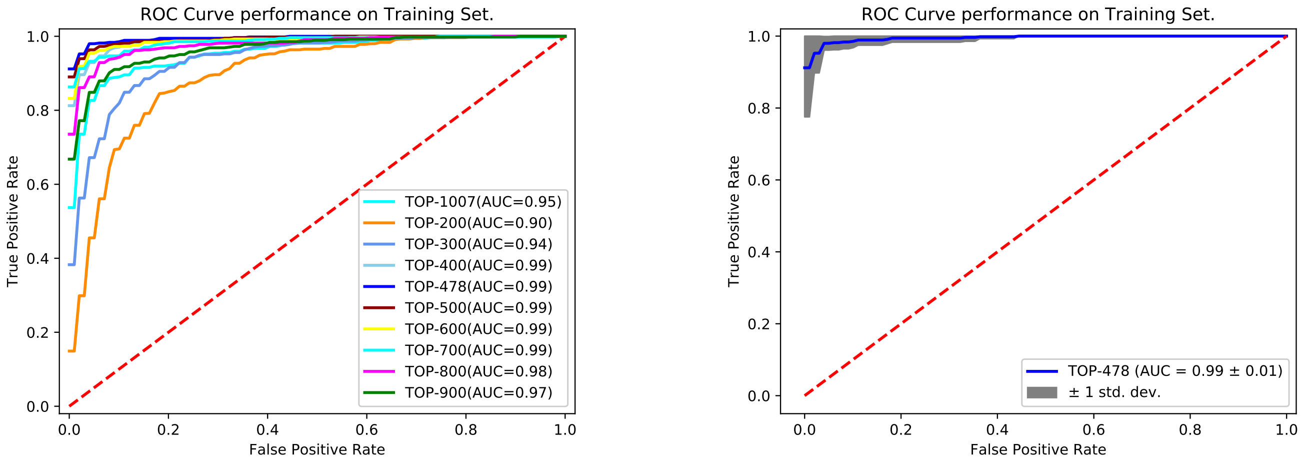
3.3. Train Set Features Impact on Model.

We have run several experiments varying the number of features and analyzed the impact on the classification model. The analysis was done in terms of the various performance metrics discussed in the earlier section. In this process, we were able to identify the right number of features for our model. The ROC curve for a number of SVM models of different number of features is shown Fig. 6. The close a ROC curve is to the top left corner of the graph, the better is the performance of the corresponding model. Therefore, from the curves of Fig. 6, it is clear that the performance with 400, 478, 500, and 600 features is much better compared to the other feature subsets.

This same conclusion can be made from Fig. 7. In Fig. 7, we plot the area under ROC curve, accuracy, sensitivity, specificity and MCC of models that are created with varying number of top-ranked features. We first explore a large feature space, albeit with coarse granularity. That means, the number of features that are added (removed) between experiments is large. As an example, Fig. 7a is generated by starting with a model with 50 top-ranked features. The coarse grain SVM-RFECV feature ranking was used in this case. Then 25 next ranked features were added in each iteration. Based on the curves, in Fig. 7a, 7b, the feature space range [400, 600] seems promising. Therefore, more models are generated in this space, however the change of features in each step becomes finer: 10 features. We thus examine 200 models whose performances are recorded in Fig. 7c. Moving on this way, we keep zooming in the interesting terrain of the feature space and increase our thoroughness in investigating the narrowed-down spaces. From this Figure, the range [475,525] seems most interesting. So, this space is investigated, with single increase steps, yielding 50 different models. Based on these models (in Fig. 7d), the model built with **478** features was chosen to be our final classifier. Among the features, there were 102 n-grams, 126 nGDip and 61 PSN features.

3.4. Impact of feature extraction techniques

To analyze the contribution of the different feature extraction techniques in building the model, we have performed some experiments with the top number of features from different subset. The subset are PSSM: both segmented and percentile pssm features, Struct.&Chem.: spd3, ctd, and quasi-sequence order features, ngram: both n-gram and p-ngram from primary protein sequence, COMB: combination of all features. We have trained three different SVM models using each of these three subsets of top features. In another model, we have trained with the combined top feature set of these group. In Fig. 8a, the accuracy, sensitivity, specificity and MCC values from these four models are compared. While



(a)

(b)

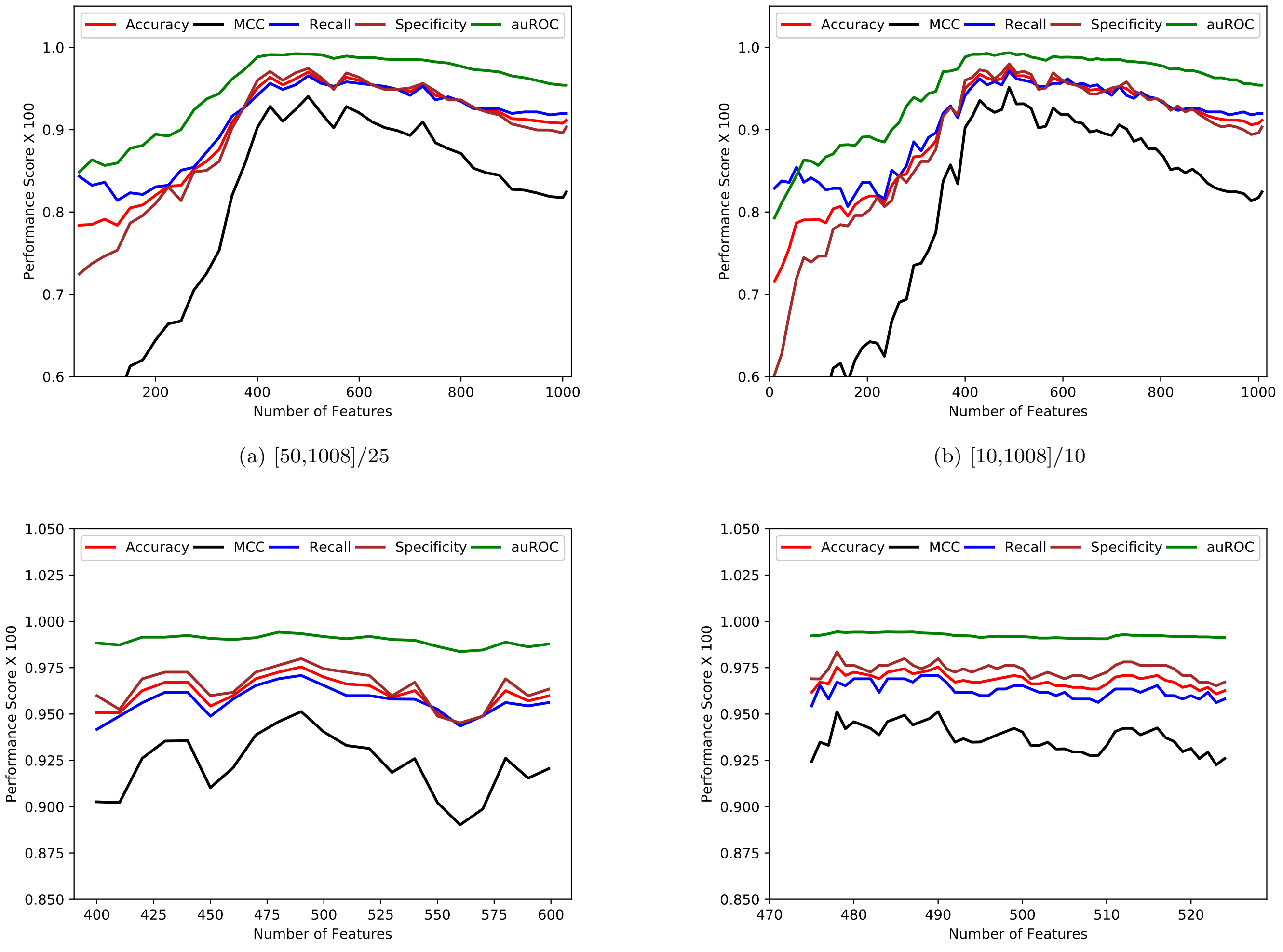
Figure 6: (**Left**) ROC-Curves of prediction models with varying number of features, generated by 10-fold cross validation on the PDB1075 dataset.(**Right**) Optimized ROC-Curve that maximizes area under curve(AUC) scores with 478 features.

combination of all performs slightly better than both Struct.&Chem. and ngram, the PSSM feature extraction technique is a clear winner over the all of these subset.

The size of the feature vectors in the above comparison was widely different. Therefore, we conducted another experiment where we trained 3 different models using top 150 features of the 3 individual feature extraction techniques. We compare the performance of these models to the combined model in Fig. 8b. In this case, the superiority of pssm feature space is now replaced by the combined feature space by a little margin, accuracy by (82-78) 4%, recall by (83-80) 3%, specipicity by (82-75) by 7%, mcc by (65-56) 9%. PSSM, Struct.&Chem., and ngram achieves accuracy values of 78%, 79% and 73%, respectively. When the combined feature space is used instead, the accuracy increases to 82%. Similarly the MCC increases from respective individual values of 0.56, 0.58, 0.47 to 0.65 for the combined feature space. Another observation is worth noting from these experiments. The ngram only classifier is biased towards the negative class by a considerable amount (81-65=16%), in exact opposite, model build on pssm features is biased toward positive class by little margin (80-75=5%).In between these two extreme, model constructed with Struct.&Chem. features is almost balanced (recall=78.88, specipicity=79.56). Its ability to predict the both positive class(indicated by recall score), and negative class(indicated by specipicity) is, in fact, slightly lower than that of the model constructed on the combined feature space, resulting in a superior, although by small margin, overall accuracy and MCC scores of the later. We ran one more experiment to check whether this is indeed the case. In this experiment, we used combination of two feature spaces, leaving the other feature space out. We chose the percentage of top 256 features from each group to construct the model. We compared the performance of the four generated models with that of the model created using the combination of all three feature spaces. The results are shown in Fig. 8c. The composition of each combination is tabulated in Table 2. It is clear from Fig. 8c, among the 2 feature space composition, PSSM and Struct.&Chem. composition is the best. Nevertheless, adding ngram features clearly add some values as seen in COMB_B even though by small margin. One interesting thing to notice is that, within two feature space composition of PSSM and Struct.&Chem. If we increase percentage of PSSM features then overall performance

Table 2: Feature constructed by using different combination of feature group.

ID	PSSM	Struct.&Chem.	n-gram
Comb1	65%	35%	_
Comb2	75%	25%	_
Comb3	67%	_	33%
Comb4	_	67%	33%
Comb_A	54%	35%	11%
Comb_B	34%	33%	33%



(a) [50,1008]/25

(b) [10,1008]/10

(c) [400,600]/10

(d) [475,525]/1

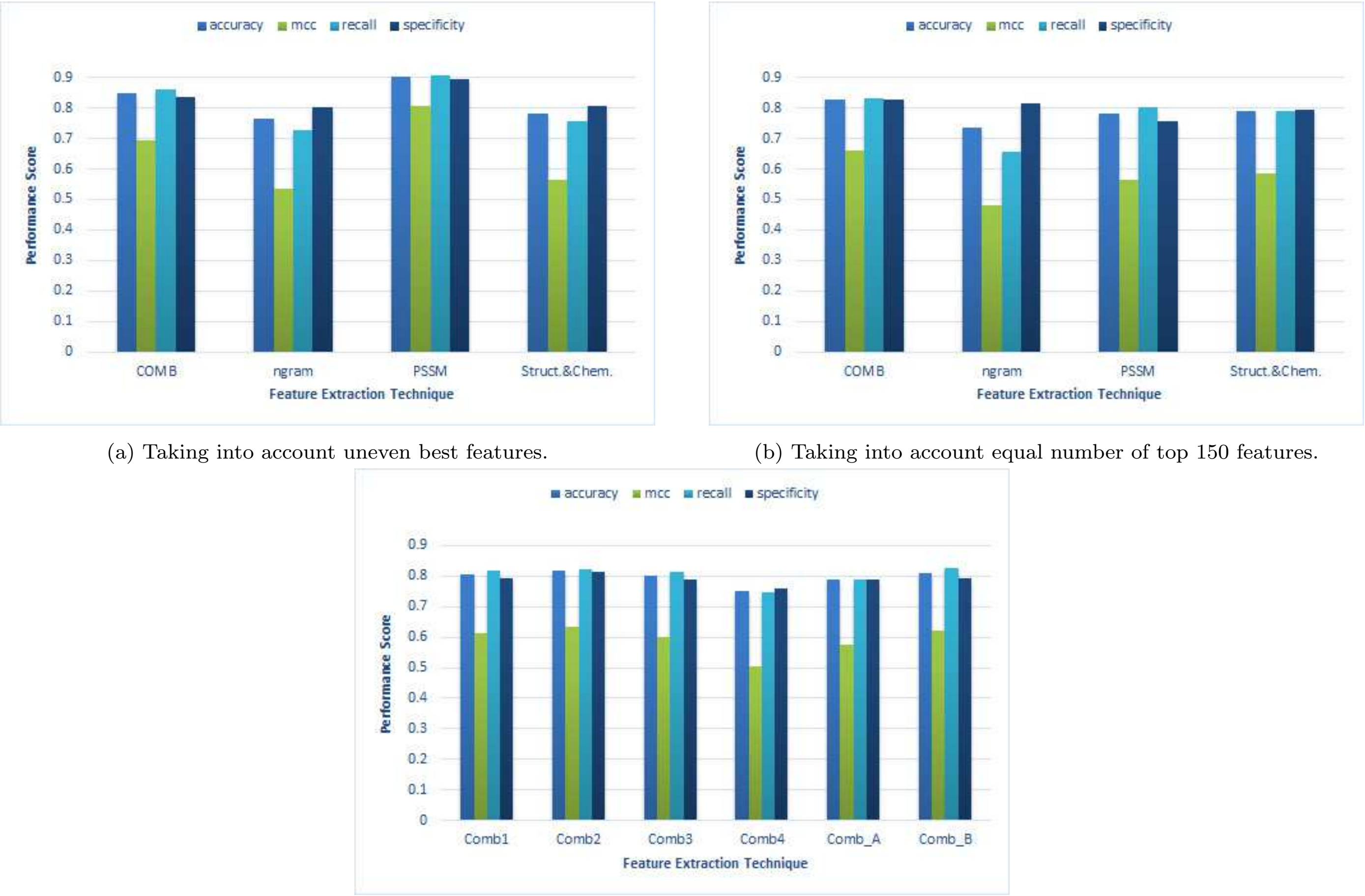
Figure 7: Area under ROC curve (auROC), accuracy, sensitivity, specificity and MCC of models with varying number of features, generated by 10-fold cross validation on the PDB1075 dataset. The [x, y]/z style annotation of each sub-figure means that, the experiment started with x top-ranked features. Then a model was trained with z more features and the performance scores were recomputed. This process continued until the feature count became y.

also increases as well. This proves, superiority of PSSM feature space. However, this trend does not hold in composition of three features spaces, in fact it is opposite which infer that, either PSSM and ngram or PSSM and Struct.&Chem. features are incompatible with each other. As PSSM and ngram composition achieves comparatively better performances, so it is Struct.&Chem. and ngram that are incompatible with each other.

3.5. Test set features impact on model's performance.

In earlier experiment, we have investigated training features impact on model's 10 fold-cross validation performances using the same training feature set(PDB1075). In this sub-section, however, we will investigate model's performances on test set(PDB186). This time we will train model using top train set features and test model performances using the same feature from test set. Note that, we never used test set(PDB186) to find or construct best features and we used ensemble method ET(ExtraTree) as the final predictor instead of SVM. When comparing performances with other methods on PDB186, we are reffering this experimental result. Experimenting this way, we were able to identify the right number of best features for our model that best perform against benchmark test set(PDB186).

In Fig. 7, we plot the area under ROC curve, accu-



(a) Taking into account uneven best features.

(b) Taking into account equal number of top 150 features.

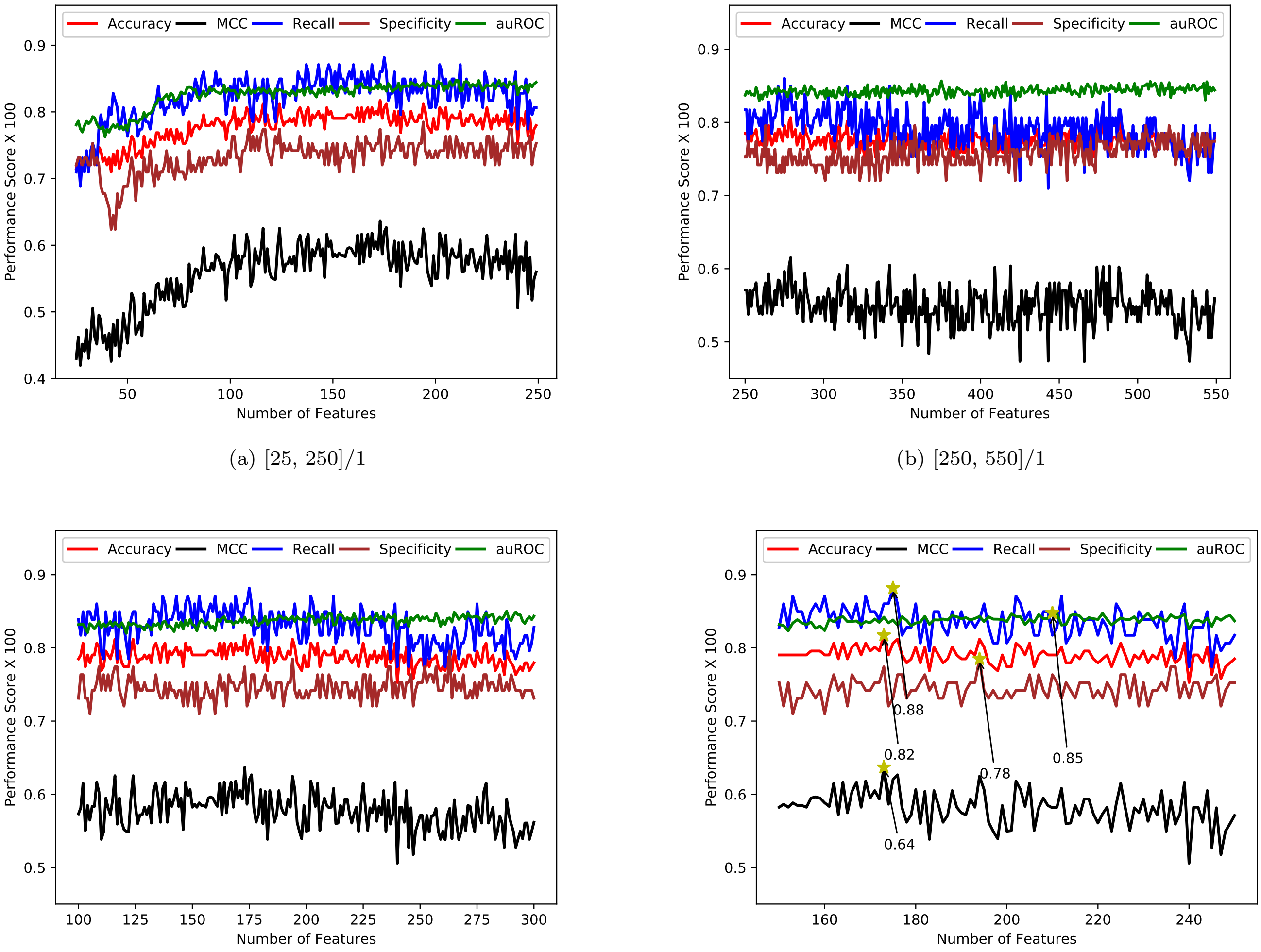
(c) Taking into account percentwise best features from top 256 features.

Figure 8: Performance of different feature extraction technique. The results are obtained from ten-fold cross validation on the PDB1075 dataset. PSSM: both segmented and cummulative position specific features. Struct.&Chem.: spd3, ctd and quasi-sequence order features. ngram: both ngram and pngram features. COMB: combination of all features.

racy, sensitivity, specificity and MCC of models that are created with varying number of top-ranked features. We first explore a large feature space sorted by feature importances, with fine granularity. That means, the number of features that are added (removed) between experiments is small, specifically 1. As an example, Fig. 9a is generated by starting with a model with 25 top-ranked features. The fine grained ET-RFECV feature ranking was used in this case. Then 1 next ranked features were added in each iteration until 250 top ranked features is reached. Similar set up is used in Fig. 9b, albeit different feature range. Based on the two curves, feature range [100,301] seems promising as can be seen from Fig. 9c. This narrow range zoom in figure shows that optimum feature range lie between [150,250]. Therefore, we plotted this zoom-in feature range in Fig. 9d to find out the optimum number of top features. From this figure, it is clear that optimum number of feature is 173 which achieves accuracy 81.72(nearly 82), recall 86.02, specificity 77.42, auROC 84.21, MCC 63.68. Therefore, 173 number of features was chosen to be our final classifier.

3.6. *Impact of features on model's ROC performances*

In previous experiment, we have analyzed training set features impact on model's 10 fold-cross validation roc performances using the same training feature set (PDB1075). However, in this sub-section, we will analyze trained model's performance on test set (PDB186) while model is trained on top number of training features. Feature selection process is performed described in earlier section, another RFECV-ExtraTree is applied with finer grain step followed by SMOTE (syntactic minority oversampling technique) operation and final ranking and prediction is



(a) [25, 250]/1

(b) [250, 550]/1

(c) [100, 301]/1

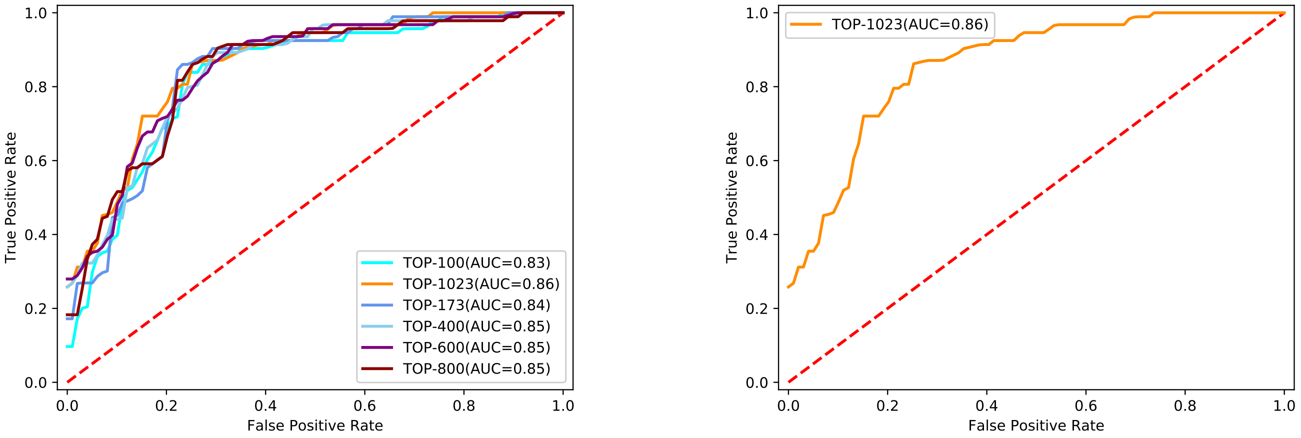
(d) [150, 250]/1

Figure 9: Area under ROC curve (auROC), accuracy, sensitivity, specificity and MCC of models with varying number of top features, generated by RFE on the PDB1075 dataset and tested against PDB186 dataset. The [x, y]/z style annotation of each sub-figure means that, the experiment started with x top-ranked features. Then a model was trained with z more features and the performance scores were recomputed. This process continued until the feature count became y.

performed using RFECV-ExtraTree and ExtraTree classifier respectively. As mentioned earlier, the more of a ROC curve is to the upper left corner of the unit square graph, the better is the area under curve score. The Fig. 10 demonstrate roc performances of test set on different number of top-ranked training features. From this Figure, unlike 10-fold roc performances, increasing top-ranked number of training features also gradually increases area under curve(AUC) performance score for the features from 100 to 1023, and maximized for the 1023^{th} top-ranked features at 0.86 which is graphically depicted in Fig. 10b. This trend, however, may not be the case for the top ranked features well beyond 1023. Therefore, within the above mentioned range of features, we can safely conclude that roc performance score(AUC) is best for the first 1023^{th} top-ranked features.

3.7. *Comparison With Other Methods*

We have also compared the performance of our method on both benchmark training dataset and test dataset with other state-of-the-art methods. We have considered nine of the previous methods to compare with our method. They are: DNAProt Krishna Kumar (2009), DNABinder Kumar (2007), Kmer1+ACC Dong (2015), iDNAProt Lin (2011), iDNAPro-PseAAC Bin Liu (2015), HMMBinder Rianon Zaman (2017), LocalDPP Wei (2017), iDNAProt-ES Chowdhury (2017) and DPP-PseAAC Md. Saifur Rahman (2018). We did not



(a) ROC-Curve performance with different number of top feature set.

(b) Singled-out Optimized ROC-Curve that maximizes area under curve(AUC) score.

Figure 10: ROC-Curve Performance on Test set(PDB186). First, model trained with varying number of top features from PDB1075 and then tested with the same features set from PDB186. (**Left**) ROC-Curves of prediction models with varying number of features, on the PDB186 dataset.(**Right**) Optimized ROC-Curve that maximizes area under curve(AUC) scores with 1023 features.

Table 3: **Performance Comparison with state-of-art methods on Train Set.**

Methods	***ACC(%)***	***SP(%)***	***RC(%)***	***MCC(%)***	***auROC(%)***
DNAProt	72.55	59.8	83	0.44	78.9
DNAbinder	73.58	80.4	66	0.47	81.5
Kmerl+ACC	75.23	73.8	77	0.5	82.8
iDNAProt	75.4	64.7	84	0.5	76.1
iDNAPro-PseAAC	76.76	74.5	84	0.53	
HMMBinder	86.33	85.5	87	0.72	90.2
iDNAProt-ES	90.18	90	90	0.8	94.1
Local-DPP	79.2	74.5	84	0.59	
DPP-PseAAC	95.91	97.6	94	0.92	98.8
GFS	70.21	79.7	61	0.41	75.1
Extra-LocalDPP	**98.99**	**99.45**	**98.54**	**97.99**	**99.65**

Table 4: **Performance Comparison with state-of-art methods on Test Set.**

Methods	***ACC(%)***	***SP(%)***	***RC(%)***	***MCC(%)***	***auROC(%)***
DNAProt	61.8	53.8	70	0.24	24
DNAbinder	60.8	64.5	57	0.22	21.6
Kmerl+ACC	70.96	59.1	83	0.43	43.1
iDNAProt	67.2	66.7	68	0.34	34.4
iDNAPro-PseAAC	69.89	62.4	77	0.4	40.2
HMMBinder	69.02	76.3	61	0.39	63.2
iDNAProt-ES	80.64	80	81	0.61	84.3
Local-DPP	79	65.6	92	0.63	-
DPP-PseAAC	77.42	70.9	83	0.55	79.8
GFS	**82.26**	69.9	**95**	0.67	**82.3**
Extra-LocalDPP	**82**	**78**	88	64	**85**

run the other predictors since we are working on the same datasets. We have taken the results as reported in the literature. Along with jacknife test, we have also conducted experiment to measure 10-fold cross-validation performances. As discussed earlier, we have varied number of features to identify the best model. The model with 478 top-ranked features demonstrated the best performance. While comparing with the state-of-art predictor, Extra-localDPP will actually refer to the predictor with these 478 top-ranked features. The 10-fold cross-validation accuracy, recall, specipicity, MCC and area under ROC curve scores of the model respectively were 97.54%, 96.72%, 98.36%, 95.13, and 99.44%. Subsequently, we have compared the performance of our predictor Extralocal-DPP with prominent prediction tools from literature, using jacknife cross-validation approach. The results are recorded in Table 3, the best values having been highlighted in bold faced font. The results for DNAbinder, DNA-Prot, iDNA-Prot, iDNA-Prot—dis were collected from Liu (2014). For the other predictors, the cross validation results with the same benchmark dataset was available in the respective research papers. Extralocal-DPP demonstrates superiority over all the earlier predictors in terms of each of the performance metrics used. Since PDB1075 is a stringent dataset which guarantees that pairwise sequence similarity is no more than 25%, any concerns of overestimation in jackknife approach is mitigated by Chou (2011). Next we compare performance of Extralocal-DPP with state-of-the-art using independent testing approach. The PDB186 dataset is used in this case. However, if there is significant sequence similarity between proteins of the training set and that of the testing set, then the independent test results will be over estimated. To avoid this, proteins of PDB1075 that had more than 25% sequence identity to any protein in the PDB186 dataset were removed using BLASTCLUST Altschul (1997). The prediction model was then rebuilt using this reduced PDB1075 dataset. This protocol was introduced by Liu (2014) and has subsequntly been followed in independent testing of other DNA-BP predictors. The reduced PDB1075 contained 487 positive samples, 548 negative samples; the total size of the training set became 1035. The independent test results of Extralocal-DPP and state-of-the-art predictors are recorded in Table 4. The results for DNABIND, DNAbinder, DNA-Threader, DNA-Prot, iDNA-Prot and DBPPred were obtained from Lou (2014). As the newer predictors had adopted this dataset for independent testing, the test results for these predictors were obtained from the respective research papers. From the results, we can see that Extralocal-DPP performs better than all prior predictors, except for Local-DPP. If Local-DPP is left out of the comparison, then Extralocal-DPP has the best accuracy, sensitivity, MCC and area under ROC curve. DBD-Threader has the best specificity, but its sensitivity is extremely poor. DBPPred also has better specificity than our method. But ours outperforms DBPPred in terms of sensitivity. The accuracy and MCC values are similar for both approaches, albeit DPP-PseAAC has a slight edge. Now, let us compare DPP-PseAAC with Local-DPP method. Local-DPP has the highest sensitivity among all the methods, a commendable score of 92.5%. Its specificity, however, is only 65.60%. So, it is skewed considerably towards the positive class. Extralocal-DPP has a better specificity and is more balanced in its predictive performance in contrast. To summerize, **Extralocal-DPP** shows best performance in each of the performance metrics in the jackknife cross-validation testing. In case of independent testing, its performance is also commendable, remaining behind of only Local-DPP.

3.8. Summary of Result

To summerize, **Extralocal-DPP** shows best performance in each of the performance metrics in the jackknife cross-validation testing. In case of independent testing, its performance is also commendable, remaining behind of only Local-DPP.

3.9. Discussion

In this section, we present brief discussion on several aspects relevant to our work.

3.9.1. Distinction between Extralocal-DPP and state-of-the-art methods

To differentiate between **Extralocal-DPP** and prior art, Table 5, 7 shows the different steps taken in building these prediction models. As can be seen, the novelty in **Extralocal-DPP** lies in the addition of sigmoid normalized segmented and percentile pssm features with optimum scaling factor into Chou's general PseAAC. The combination of ExtraTree for feature ranking followed by recursive feature elimination with cross-validation using both ExtraTree and SVM is also a new approach in this prediction problem. Another distinguishing factor is that we explored a large feature space, comprising 43,454 features and then selected 278 features for training the model. we have witnessed that even the final selected feature set's size in most of the earlier works is larger than the number of features used. The most recent predictors, iDNAProt-ES Chowdhury (2017) and Local-DPP Wei (2017) respectively used 86 and 120 features. Although, above-mentioned methods

Table 5: **Structure based predictors at a glance.**

Methods	*Feature Extraction Techniques*	*Feature selection*	*Classifier*
Stawiski (2003)	Analysis of electrostatic patches, surface clefts, Conservation analysis of the sequence.	12 features	ANN (3 layers)
Ahmad (2004)	Bulk electrostatic properties.		ANN (2 layers)
DNABIND Szilgyi (2006)	Proportion of certain amino acid residues, Spatial asymmetry of amino acid residues, Dipole moment of the entire molecule.	10 features	LR
DBD-Hunter Gao (2008)	Library of DNA protein complex structures, Structural alignment, Evaluation of a statistical potential, Matching score thresholding.		
DBD-Threader Gao (2009)	Library of DNA protein complex structures, Target proteinś sequence, Matching score thresholding		

Table 6: **Primary sequence based predictors at a glance.**

Methods	*Feature Extraction Techniques*	*Feature selection*	*Classifier*
DNAbinder Kumar (2007).	PSSM.	400 features.	SVM.
DNA-Prot Kumar (2009).	Frequency of amino acid/amino acid groups, hydrophobic, hydrophlic, neutral residues, PredSS from PSIPRED, Amino acid physico-chemical properties, Split sliding 10 residue windows.	CFSS(20 features).	RF.
iDNA-Prot Lin (2011).	AAC, coefficients of the second order Grey differential equation with one variable.	23 features.	RF.
DBPPred Lou (2014)	AAC, PredSS, PredRSA Auto-correlation coefficients of PSSM. Percentile values of PSSM scores.	RF filter GNB Wrapper (56 features).	GNB.
iDNA-Prot—dis Liu (2014)	Amino acid distance-pair coupling Amino acid reduced alphabet profile.	602 features.	SVM (RBF).
iDNAPro-PseAAC Liu (2015)	Profile-based protein representation. PseAAC (= 3).	23 features.	SVM (RBF).
Kmer1+ACC Dong (2015)	ACC, kmer composition, Physico-chemical properties.	SVM.	
Local-DPP Wei (2017).	Local Pse-PSSM.	120 features.	RF.
iDNAProt-ES Chowdhury (2017).	AAC, bigram, auto-covariance from PSSM, Dubchak features, Sructural features from SPIDER2.	SVM-RFE (86 features).	SVM (Linear).
DPP-PseAAC Rahman (2018).	AAC, dipeptide and tripeptide comp., Gapped dipeptide composition, Position specific features.	RF filter SVM-RFE (289 features).	SVM (Linear).

Table 7: **Fusion based predictor at a glance.**

Methods	*Feature Extraction Techniques*	*Feature selection*	*Classifier*
Extralocal-DPP.	PSSM, structure & primary sequence based feature.	278 features.	SVM & ExtraTree.

use PSSM features, extraction of which take time compared to primary sequence based n-gram, it is important to note that they are quite effective in describing protein adequately. Our approach, therefore, combine different feature extraction technique to extract features from evolutionary, predicted structure and primary sequences, then use RFECV to select effective feature and finally use SVM & ExtraTree model to predict the class/category of protein sequence. Additionally, if the target protein does not have enough homologous sequences in the database, the generated PSSM cannot describe the protein adequately Lin (2013), in constrast, if target protein does have plenty of homologous sequences in the database then over-estimation may occur in prediction model which may even lead to degradation of performance for the unseen protein sequences(generalization errors). Therefore, any prediction model dependent on PSSM information should be careful in choosing protein database for pssm feature extraction and must strike a balance in doing this. To mitigate this problem, we used non-redundant protein database (nrdb90), which is almost representative for all the known proteins.

3.9.2. ***Reported errors in earlier predictors***

In the independent testing, we have not compared Extralocal-DPP with iDNAProt-ES Chowdhury (2017), which was the best predictor so far in terms of both jackknife and independent testing. Extralocal-DPP outperformed it in the jackknife cross-validation test. And we found a flaw in the independent testing of iDNAProt-ES. As discussed earlier, the protocol followed by Liu (2014) was to eliminate the sequences in PDB1075 that had more than 25% pairwise similarity with the independent test set (PDB186), and then retrain the predictor with this reduced set. This protocol was followed by subsequent authors as well. As reported in Rahman (2018), unfortunately, this important step was missed in the independent testing of iDNAProt-ES, authors were notified about the error in their independent test process through private communication channel. Therefore, the performance scores reported for that tool are over estimations. As such, we have excluded it in our independent test comparisons. Another minor error is observed in the MCC score of independent test of Local-DPP Wei (2017). Since we know P and N values of PDB186 (93 each), the TP, TN, FP, FN values can easily be computed from the accuracy, sensitivity and specificity data. When we plug these values into the equation of MCC, we get an MCC of 0.602. However, the reported value in Wei (2017) is 0.625, which is over estimated.

3.9.3. ***Unavailability of BLASTCLUST in standalone BLAST***

The reduced dataset was created by Liu (2014) using BLUSTCLUST tools Altschul (1997). In protein prediction task, subsequent authors who have followed the same protocol or steps, nevertheless, did not make reduced dataset publicly available. So, we needed to follow the same steps to generate this reduced training set. However, we were not able to find the BLUSTCLUST tool in the up-to-date version of standalone BLAST software downloadable from NCBI (2018). Also, some discussion forums suggested that it was deprecated Anon. (2018). While we found an older version (version 2.2.14) from NCBI that contained BLUSTCLUST, we could not make it work. For example, we tried to check how many clusters are there in the PDB186 data set with a 25% cut off, but all proteins showed up in single cluster, which seemed wrong. As such, we reached out to Rahman (2018) and they kindly shared their reduced PDB1075 dataset with us which they collected from Wei (2017).

3.9.4. ***Jackknife cross-validation vs. independent testing***

We have shown that **Extralocal-DPP** has the best performance in terms of jackknife testing. It, however, ranked 2^{nd} in terms of independent testing. Moreover, there was even a fall in the performance scores when compared to jackknife testing score. The jackknife cross-validation results should be trusted as PDB1075 has less than 25% pairwise sequence similarity. We think, the lesser performance in the independent test can easily be explained by the protocol that was used. As discussed earlier, the PDB1075 was reduced in size to eliminate sequence similarity of this set with sequences in the PDB186. This eliminated 40 samples from the training set. More importantly, 38 of these samples were positive samples. Therefore, data imbalance was introduced, resulting in a model that is inferior to the original model, even though we partially did handled data imbalance problems by SMOTE as seen in literature. By following Chou's argument that sheds doubt on the objectivity of the independent testing Chou (2011) prioritization has been given to jackknife cross-validation performance over independent test results:

"*The way of how to select the independent proteins to test the predictor could be quite arbitrary unless the number of independent proteins is sufficiently large. This kind of arbitrariness might result in completely different conclusions. For instance, a predictor achieving a higher success rate than the other predictor for a given independent testing dataset might fail to keep so when tested by another independent testing dataset. Accordingly, the independent dataset test is not a fairly objective test method although it was often used to demonstrate the practical application of a predictor.*"

4. CONCLUSION

In summary, we presented both our assigned paper result and reproduced result with proper experimentation after solving some technical hurdles. Performance of Machine Learning Computational model largely depends on Generalization capacity of the model, Large dataset without noise, outlier and redundancy i.e. pure representative dataset. Well feature extraction method that will capture key pattern and sequence order information. In future work, on the one hand, we will incorporate a novel feature extraction method or combine better feature extraction techniques that will not only capture sequence order information but also rely on only representative biological sequence. On the other hand, we will use a powerful deep learning model to increase generalization capacity of the model.

Acknowledgement

We thank Assistant Professor Md. Saifur Rahman for providing us guidance and reduced PDB1075 dataset. This dataset was used to retrain our classifier before independent testing. We are also grateful for the constructive comments of the anonymous reviewers and the Editor. Our special thanks go to the anonymous reviewers for providing us with a number of relevant references and for suggesting some useful texts to be added in the manuscript.

References

Ahmad, S., S.A., 2004. Moment-based prediction of dna-binding proteins. J. Mol. Biol. 341 (1), 65–71.

Altschul, S.F., M.T.S.A.Z.J.Z.Z.M.W.L.D., 1997. Gapped blast and psiblast: a new generation of protein database search programs. Nucleic Acids Res. 25 (17), 3389–3402.

Anon., 2018. Question: Blastclust standalone download address?. URL: https://www.biostars.org/p/92324/.

Behbahani, M., M.H.N.M., 2016. Analysis and comparison of lignin peroxidases between fungi and bacteria using three different modes of chous general pseudo amino acid composition. J. Theor. Biol. 411, 1–5.

Bin Liu, Shanyi Wang, X.W., 2015. Dna binding protein identification by combining pseudo amino acid composition and profile-based protein representation. Scientific reports. 5, 5–15479.

Boser, B.E., G.I.V.V., 1992. A training algorithm for optimal margin classifiers., pp. 470–475.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Buck, M.J., L.J., 2004. Chip-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. Genomics 83 (3), 349–360.

Cao, D.-S., X.Q.S.L.Y.Z., 2013. propy: a tool to generate various modes of chous pseaac. J. Mol. Graph. Modell. 29 (7), 960–962.

Chen, W., D.H.F.P.L.H.C.K.C., 2016a. Iacp: a sequence-based tool for identifying anticancer peptides. Oncotarget 7(13), 16895.

Chen, W., F.P.Y.H.D.H.L.H.C.K.C., 2017. Irna-ai: identifying the adenosine to inosine editing sites in rna sequences. Oncotarget 8 (3), 4208.

Chen, W., F.P.Y.H.D.H.L.H.C.K.C., 2018. Irna-3typea: identifying 3-types of modification at rnas adenosine sites. Mol. Ther. Nucleic Acids. .

Chen, W., L.T.Y.J.D.C.L.H.C.K.C., 2014. Pseknc: a flexible web server for generating pseudo k-tuple nucleotide composition. Anal. Biochem. 456, 53–60.

Chen, W., T.H.Y.J.L.H.C.K.C., 2016b. Irna-pseu: identifying rna pseudouridine sites. Mol. Ther. Nucleic Acids 5 .

Cheng, X., X.X.C.K.C., 2017a. Ploc-meuk: predict subcellular localization of multi-label eukaryotic proteins by extracting the key go information into general pseaac. Genomics .

Cheng, X., X.X.C.K.C., 2017b. Ploc-mplant: predict subcellular localization of multi-location plant proteins by incorporating the optimal go information into general pseaac. Mol. Biosyst. 13 (9), 1722–1727.

Cheng, X., X.X.C.K.C., 2017c. Ploc-mvirus: predict subcellular localization of multi-location virus proteins via incorporating the optimal go information into general pseaac. Gene 628, 315–321.

Cheng, X., Z.S.G.L.W.Z.X.X.C.K.C., 2017d. Ploc-manimal: predict subcellular localization of animal proteins with both single and multiple sites. Bioinformatics 33 (22), 3524–3531.

Cheng, X., Z.S.G.X.X.C.K.C., 2016. Iatc-misf: a multi-label classifier for predicting the classes of anatomical therapeutic chemicals. Bioinformatics 33 (3), 341–346.

Cheng, X., Z.S.G.X.X.C.K.C., 2017e. Iatc-mhyb: a hybrid multi-label classifier for predicting the classification of anatomical therapeutic chemicals. Oncotarget 8(35), 58494.

Chia-Cheng Chou, Ting-Wan Lin, C.Y.C.A.H.J.W., 2003. Crystal structure of the hyperthermophilic archaeal dna-binding protein sso10b2 at a resolution of 1.85 angstroms. Journal of bacteriology 185(14), 4066–4073.

Chou, K.C., 1995. A novel approach to predicting protein structural classes in a (201)-d amino acid composition space. Proteins Struct. Funct. Bioinf. 21 (4), 319–344.

Chou, K.C., 2001. Prediction of protein cellular attributes using pseudo-amino acid composition. Proteins Struct. Funct. Bioinf. 43 (3), 246–255.

Chou, K.C., 2011. Some remarks on protein attribute prediction and pseudo amino acid composition. J. Theor. Biol. 273 (1), 236–247.

Chou, K.C., 2013. Some remarks on predicting multi-label attributes in molecular biosystems. Mol. Biosyst. 9 (6), 1092–1100.

Chou, K.C., 2015. Impacts of bioinformatics to medicinal chemistry. Med. Chem. (Los Angeles) 11 (3), 218–234.

Chou, K.C., 2017. An unprecedented revolution in medicinal chemistry driven by the progress of biological science. Curr. Top. Med. Chem. 17 (21), 2337–2358.

Chou, K.-C., S.H.B., 2009. Recent advances in developing web-servers for predicting protein attributes. Nat. Sci. (Irvine). 1 (02), 63.

Chowdhury, S.Y., S.S.D.A., 2017. idnaprot-es: identification of dna-binding proteins using evolutionary and structural features. Sci. Rep. 7 (1), 14938.

Dong, Q., W.S.W.K.L.X.L.B., 2015. Identification of dna-binding proteins by auto-cross covariance transformation., pp. pp. 470–475.

Du, P., G.S.J.Y., 2014. Pseaac-general: fast building various modes of general form of chous pseudo-amino acid composition for large-scale protein datasets. Int. J. Mol. Sci. 15 (3), 3495–3506.

Dubchak, I., M.I.K.S.H., 1997. Protein folding class predictor for scop: approach based on global descriptors., in: ISMB, pp. 104–107.

Feng, P., D.H.Y.H.C.W.L.H.C.K.C., 2017. Irna-psecoll: identifying the occurrence sites of different rna modifications by incorporating collective effects of nucleotides into pseknc. Mol. Therapy Nucleic Acids 7, 155–163.

Freeman, K., G.M.S.D., 1995. Molecular and genetic analysis of the toxic effect of rap1 overexpression in yeast. Genetics 141 (4), 1253–1262.

Gao, M., S.J., 2008. Dbd-hunter: a knowledge-based method for the prediction of dnaprotein interactions. Nucleic Acids Res. 36 (12), 3978–3992.

Gao, M., S.J., 2009. A threading-based method for the prediction of dna-binding proteins with application to the human genome. PLoS Comput. Biol. 5 (11), e1000567.

Gurova, K., 2009. New hopes from old drugs: revisiting dna-binding small molecules as anticancer agents. Future Oncol 5 (10), 1685–1704.

Guyon, I., W.J.B.S.V.V., 2002. Gene selection for cancer classification using support vector machines. Machine Learning 46, 389422.

Helwa, R., H.J., 2010. Analysis of dnaprotein interactions: from nitrocellulose filter binding assays to microarray studies. Anal. Bioanal. Chem. 398 (6), 2551–2561.

Ho., T.K., 1995. Random decision forests., in: Document analysis and recognition, 1995., IEEE, proceedings of the third international conference on, pp. 278–282.

Jia, J., L.Z.X.X.L.B.C.K.C., 2015. Ippi-esml: an ensemble classifier for identifying the interactions of proteins by incorporating their physicochemical properties and wavelet transforms into pseaac. J. Theor. Biol. 377, 47–56.

Jia, J., L.Z.X.X.L.B.C.K.C., 2016. Icar-psecp: identify carbonylation sites in proteins by monte carlo sampling and incorporating sequence coupled effects into general pseaac. Oncotarget 7 (23), 34558.

Julong, D., 1989. Introduction to grey system theory. J. Grey system 1 (1), 1–24.

Kawashima, S., P.P.P.M.K.A.K.T.K.M., 2007. Aaindex: Amino acid index database, progress report 2008. Nucleic Acids Res. 36b (suppl_1), D202–D205.

Khan, M., H.M.K.S.I.N., 2017. Unb-dpc: identify mycobacterial membrane protein types by incorporating un-biased dipeptide composition into chous general pseaac. J. Theor. Biol. 415, 13–19.

Krishna Kumar, Ganesan Pugalenthi, P.S., 2009. Dna-prot: identification of dna binding proteins from protein sequence information using random forest. Journal of Biomolecular Structure and Dynamics. 26, 679–686.

Krishnan, S., 2018. Using chous general pseaac to analyze the evolutionary relationship of receptor associated proteins (rap) with various folding patterns of protein domains. J. Theor. Biol. 445, 62–74.

Kumar, M., G.M.R.G., 2007. Identification of dnabinding proteins using support vector machines and evolutionary profiles. BMC Bioinf. 8 (1), 463.

Kumar, K.K., P.G.S.P., 2009. Dnaprot: identification of dna binding proteins from protein sequence information using random forest. J. Biomol. Struct. Dyn. 26 (6), 679–686.

Leung, C.-H., C.D.H.M.V.Y.M.D.L., 1013. Dna-binding small molecules as inhibitors of transcription factors. Med Res Rev 33(4), 823–846.

Lin, H., C.W.D.H., 2013. Acalpred: a sequence-based tool for discriminating between acidic and alkaline enzymes. PLoS ONE 8 (10), e75726.

Lin, H., D.E.Z.D.H.C.W.C.K.C., 2014. Ipro54-pseknc: a sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition. Nucleic Acids Res. 42 (21), 1296112972.

Lin, W.-Z., F.J.A.X.X.C.K.C., 2011. Idna-prot: identification of dna binding proteins using random forest with grey model. PLoS ONE 6 (9), e24756.

Liu, B., L.R.C.K.C., 2016a. idhs-el: identifying dnase i hypersensitive sites by fusing three different modes of pseudo nucleotide composition into an ensemble learning framework. Bioinformatics 32 (16), 2411–2418.

Liu, B., W.H.C.K.C., 2017a. Pse-in-one 2.0: an improved package of web servers for generating various modes of pseudo components of dna, rna, and protein sequences. Nat. Sci. (Irvine) 9 (04), 67.

Liu, B., W.S.L.R.C.K.C., 2016b. irspot-el: identify recombination spots with an ensemble learning approach. Bioinformatics 33 (1), 35–41.

Liu, B., X.J.F.S.X.R.Z.J.W.X., 2015. Psedna-pro: Dna-binding protein identification by combining chous pseaac and physicochemical distance transformation. Mol. Inform. 34, 8–17.

Liu, B., X.J.L.X.X.R.Z.J.W.X.C.K.C., 2014. idna-prot—dis: identifying dna-binding proteins by incorporating amino acid distance-pairs and reduced alphabet profile into the general pseudo amino acid composition. PLoS ONE 9, e106691.

Liu, Z., X.X.Y.D.J.J.J.Q.W.R.C.K.C., 2016c. prnam-pc: predicting n6-methyladenosine sites in rna sequences via physicalchemical properties. Anal. Biochem. 497, 60–67.

Liu, B., Y.F.C.K.C., 2017b. 2l-pirna: a two-layer ensemble classifier for identifying piwi-interacting rnas and their function. Molecular Therapy-Nucleic Acids 7, 267–277.

Liu, B., Y.F.H.D.S.C.K.C., 2017c. ipromoter-2l: a two-layer predictor for identifying promoters and their types by multi-window-based pseknc. Bioinformatics 34 (1), 33–40.

Lou, W., W.X.C.F.C.Y.J.B.Z.H., 2014. Sequence based prediction of dna-binding proteins based on hybrid feature selection using random forest and gaussian naive bayes. PLoS ONE 9 (1), e86703.

McGuffin, L.J., B.K.J.D., 2000. The psipred protein structure prediction server. ACM 16 (4), 404–405.

Md. Saifur Rahman, Swakkhar Shatabda, S.S.M.K.M.S.R., 2018. Dpp-pseaac: A dna-binding protein prediction model using chous general pseaac. Journal of Theoretical Biology. 452, 22–34.

Meher, P.K., S.T.S.V.R.A., 2017. Predicting antimicrobial peptides with improved accuracy by incorporating the compositional, physico-chemical and structural features into chous general pseaac. Sci. Rep. 7, 42362.

Mei, J., Z.J., 2018. Prediction of hiv-1 and hiv-2 proteins by using chous pseudo amino acid compositions and different classifiers. Sci. Rep. 8 (1), 2359.

Motion, G.B., H.A.H.E.J.S., 2015. Dna-binding protein prediction using plant specific support vector machines: validation and application of a new genome annotation tool. Nucleic Acids Res. 43 (22), e158–e158.

Nimrod, G., S.M.S.A.L.C.B.T.N., 2010. idbps: a web server for the identification of dna binding proteins. Bioinformatics 26 (5), 692–693.

Paz, I., K.E.B.B.M.G.Y., 2016. Bindup: a web server for non-homology-based prediction of dna and rna binding proteins. Nucleic Acids Res. 44 (W1), W568–W574.

Peterson, E.L., K.J.T.J.P.R., 2009. Reduced amino acid alphabets exhibit an improved sensitivity and selectivity in fold assignment. Bioinformatics 25 (11), 1356–1362.

Pierre Geurts, D.E., Wehenkel, L., 2006. Extremely randomized trees. Machine learning 63 (1), 3–42.

Qiu, W.-R., J.S.Y.X.Z.C.X.X.C.K.C., 2017a. irnam5c-psednc: identifying rna 5-methylcytosine sites by incorporating physical-chemical properties into pseudo dinucleotide composition. Oncotarget 8 (25), 41178.

Qiu, W.-R., S.B.Q.X.X.X.D.C.K.C., 2017b. iphos-pseevo: identifying human phosphorylated proteins by incorporating evolutionary information into general pseaac via grey system theory. Mol Inf. 36, (56).

Qiu, W.-R., S.B.Q.X.X.X.Z.C.C.K.C., 2016a. ihyd-psecp: identify hydroxyproline and hydroxylysine in proteins by incorporating sequence-coupled effects into general pseaac. Oncotarget 7 (28), 44310.

Qiu, W.-R., S.B.Q.X.X.X.Z.C.C.K.C., 2016b. iptm-mlys: identifying multiple lysine ptm sites and their different types. Bioinformatics 32 (20), 3116–3123.

Rahman, M.S., R.M.K.M.R.M., 2018. isgpt: an optimized model to identify sub-golgi protein types using svm and random forest based feature selection. Artif. Intell. Med. 84, 90–100.

Rianon Zaman, Shahana Yasmin Chowdhury, M.A.R.A.S.A.D.S.S., 2017. Hmmbinder: Dna-binding protein prediction using hmm profile based features. BioMed research international,2017. .

Safavian, S., Landgrebe, D., 1991. A survey of decision tree classifier methodology. IEEE Transactions on Systems, Man, and Cybernetics 21, 660–674. doi:10.1109/21.97458.

Song, J., W.Y.L.F.A.T.R.N., 2018. iprot-sub: a comprehensive package for accurately mapping and predicting protease-specific substrates and cleavage sites. Brief. Bioinf. 42 (21), 1296112972.

Stawiski, E.W., G.L.M.G.Y., 2003. Cannotating nucleic acid-binding function based on protein structure. J. Mol. Biol. 326 (4), 1065–1079.

Szabov, A., K.O.e..F.T.J., 2012. Prediction of dna-binding propensity of proteins by the ball-histogram method using automatic template search. Proteome Sci. 9, S1. BioMed Central. .

Szilgyi, A., S.J., 2006. Efficient prediction of nucleic acid binding function from low-resolution protein structures. J. Mol. Biol. 358 (3), 922–933.

Wang, G., D.R., 2005. Pisces: recent improvements to a pdb sequence culling server. Nucleic Acids Res. 33, W94–W98.

Waris, M., A.K.K.M.H.M., 2016. Identification of dna binding proteins using evolutionary profiles position specific scoring matrix. Neurocomputing 199, 154–162.

Wei, L., T.J.Z.Q., 2017. Local-dpp: an improved dna-binding protein prediction method by exploring local evolutionary information. Inf. Sci. (Ny) 384, 135–144.

Xu, Y., S.X.J.W.L.Y.D.N.Y.C.K.C., 2013. isno-aapair: incorporating amino acid pairwise coupling into pseaac for predicting cysteine s-nitrosylation sites in proteins. PeerJ 1, e171.

Xu, R., Z.J.L.B.H.Y.Z.Q.W.X.C.K.C., 2015. Identification of dna-binding proteins by incorporating evolutionary information into pseudo amino acid composition via the top-n-gram approach. J. Biomol. Struct. Dyn. 33 (8), 1720–1730.

Yang, Y., H.R.P.K.L.J.D.A.S.A.W.J.S.A.Z.Y., 2017. Spider2: a package to predict secondary structure, accessible surface area, and main-chain torsional angles by deep neural networks., in: Prediction of Protein Secondary Structure., pp. 55–63.

Yu, B., L.L.L.S.Z.Y.Q.W.W.X.W.M.T.B., 2017. Prediction of protein structural class for low-similarity sequences using chous pseudo amino acid composition and wavelet denoising. J. Mol. Graph. Modell. 76, 260–273.

Zephyris, 2018a. The english language wikipedia - transferred from en.wikipedia to commons., cc by-sa 3.0. URL: https://commons.wikimedia.org/w/index.php?curid=2426900.

Zephyris, 2018b. The english language wikipedia, cc by-sa 3.0. URL: https://commons.wikimedia.org/w/index.php?curid=2426895.

Zhang, C.-J., T.H.L.W.C.L.H.C.W.C.K.C., 2016. iori-human:identify human origin of replication by incorporating dinucleotide physicochemical properties into pseudo nucleotide composition. Oncotarget 7 (43), 69783–69793.

Zhao, H., Y.Y.Z.Y., 2010. Structure-based prediction of dna-binding proteins by structural alignment and a volume-fraction corrected dfire-based energy function. Bioinformatics 26 (15), 1857–1863.

Zhou, J., L.Q.X.R.G.L.W.H., 2016. Cnnsite: Prediction of dna-binding residues in proteins using convolutional neural network with sequence features., pp. 78–85.

Zhou, W., Y.H., 2011. Prediction of dna-binding protein based on statistical and geometric features and support vector machines. Proteome Sci. 9, S1. BioMed Central. .