Robust ensemble of handcrafted and learned approaches for DNA-binding proteins

Loris Nanni
University of Padua, Padua, Italy, and
Sheryl Brahnam
Information Technology and Cybersecurity, Missouri State University, Springfield, Missouri, USA

Received 5 March 2021
Revised 6 April 2021
Accepted 6 April 2021

Abstract

Purpose – Automatic DNA-binding protein (DNA-BP) classification is now an essential proteomic technology. Unfortunately, many systems reported in the literature are tested on only one or two datasets/tasks. The purpose of this study is to create the most optimal and universal system for DNA-BP classification, one that performs competitively across several DNA-BP classification tasks.

Design/methodology/approach – Efficient DNA-BP classifier systems require the discovery of powerful protein representations and feature extraction methods. Experiments were performed that combined and compared descriptors extracted from state-of-the-art matrix/image protein representations. These descriptors were trained on separate support vector machines (SVMs) and evaluated. Convolutional neural networks with different parameter settings were fine-tuned on two matrix representations of proteins. Decisions were fused with the SVMs using the weighted sum rule and evaluated to experimentally derive the most powerful general-purpose DNA-BP classifier system.

Findings – The best ensemble proposed here produced comparable, if not superior, classification results on a broad and fair comparison with the literature across four different datasets representing a variety of DNA-BP classification tasks, thereby demonstrating both the power and generalizability of the proposed system.

Originality/value – Most DNA-BP methods proposed in the literature are only validated on one (rarely two) datasets/tasks. In this work, the authors report the performance of our general-purpose DNA-BP system on four datasets representing different DNA-BP classification tasks. The excellent results of the proposed best classifier system demonstrate the power of the proposed approach. These results can now be used for baseline comparisons by other researchers in the field.

Keywords Support vector machines, Convolutional neural networks, Pseudo amino acid composition, Heterogeneous ensembles, Protein representations

Paper type Research paper

1. Introduction

All living things rely on DNA-binding proteins (DNA-BPs). They are vital components in eukaryotic and prokaryotic proteomes that regulate and affect cellular processes, such as transcription and DNA replication, repair and recombination [1]. Because anywhere from 6–7% of all eukaryotic proteins bind DNA, methods for differentiating DNA-BPs from non-DNA-BPs have been the focus of much recent scientific research. This interest has resulted in hundreds of thousands of protein sequences available to scientists [2]. The development of

The authors gratefully acknowledge the support of NVIDIA Corporation for the "NVIDIA Hardware Donation Grant" of a Titan X used in this research.

Applied Computing and Informatics
Emerald Publishing Limited
e-ISSN: 2210-8327
p-ISSN: 2634-1964
DOI 10.1108/ACI-03-2021-0051

automatic machine learning (ML) methods that quickly and accurately identify these proteins has now become a critical proteomic technology.

The key to building automatic DNA-BP classification systems is finding powerful protein representations. Protein representations are generally classified into two categories: sequence-based models and structure-based models. The structural model relies on information obtained from the high-resolution three-dimensional (3D) structure of a protein's sequence. Representations based on the structural model enhance DNA-BP prediction because of the close connection between structural features and protein function. To illustrate the advantages of using structural-based models for DNA-BP classification, see the literature review provided in [3]. A significant problem with structural-based models is the so-called "structure knowledge gap" [4], where a massive number of protein sequences have a limited number of known structures.

In contrast to the structural model, the sequence model of protein representation is based on extracting features directly from the amino acid composition (AAC) of a protein. A revolutionary protein representation that expanded AAC is Chou's pseudo AAC (PseAAC) [5, 6], which preserves vital information embedded in a protein's sequence (e.g. the sequential order represented as a series of rank-different correlation factors along the protein chain). There are many variants of PseAAC. Most relevant here, however, is the work of Cai and Lin [7], who classified DNA-BP, rRNA-BP and RNA-BP by representing proteins as a 40-dimensional PseAAC variant. Much later, in Liu *et al.* [8], this same PseAAC variant was combined with a physicochemical distance transformation [9] using a reduced alphabet to decrease computational complexity and improve prediction. In Nanni and Lumini [10], sets of reduced alphabets were taken directly from AAC via a genetic algorithm, and a multi-classifier was developed for DNA-BP identification that combined these sets with a method based on grouped weight [11]. Another protein representation approach is that which includes the normalized occurrence frequencies (represented as a vector of length 20^n) of a given n-peptide: dipeptide [12–15], tripeptide [16] and tetrapeptide [17].

Recently, sets of features based on Chou's general PseAAC have been proposed in [18] and transformed for deep learning in [19] using an embedding layer common in natural language processing. Convolutional neural networks (CNNs) have also been trained on sequence-based descriptors in [20–26]. In [20], for instance, CNNs were trained on amino acid sequences combined with contextual information, and, in [26], several CNNs and recurrent neural networks (RNNs), as well as combinations of the two, were trained on sequential descriptors and compared.

The inclusion of evolutionary information is yet another effective approach based on the ACC model. Evolutionary information is included in sequence profiles generated by position-specific iterated Basic Local Alignment Search Tool (PSI-BLAST) [27]. Using a simple support vector machine (SVM) classifier called dDNAbinder, Kumar *et al.* [28] were the first to discover that including evolutionary information resulted in superior identification performance. Since the publication of [28], many researchers have shown that these profiles improve DNA-BP prediction [29–33].

Researchers have also succeeded in extracting robust descriptors from the position-specific scoring matrix (PSSM) [34], which describes a protein using a PSI-BLAST similarity search. For example, Nanni *et al.* [35] generated a high-performing ensemble by combining many matrix representations from PSSM, and Warris *et al.* [15] improved the DNA-BP prediction via descriptors extracted from split AAC, dipeptide composition and PSSM. Wang *et al.* [36] were successful using three different feature vectors: a 200-dimension normalized Moreau–Broto autocorrelations vector [37], a 100-dimension PSSM-DCT (PSSM compressed by a discrete cosine transform) vector and a 1,040-dimension PSSM-DWT (PSSM compressed by a discrete wave transform) vector. Wei *et al.* [38] segmented PSSMs into equally sized sub-PSSMs. Pre-PSSM features [39] were extracted from these segments and classified using

random forest (RF) [40]. RF was also used to classify pseudo PSSM (PsePSSM) features in [19, 40].

Finally, proteins can be treated or represented as images in several ways [41–44]. One method is to treat matrix representations of proteins as images and then extract texture features from them [41–43]. In Nanni *et al.* [41], for instance, different sets of handcrafted texture descriptors (e.g. Haralick descriptors, several local binary patterns (LBP) variants, and features based on the Radon feature transform) were extracted from the two-dimensional (2D) distance matrix, which was obtained from the 3D tertiary structure of a protein. Combining these different feature sets produced significant improvement in ensemble classification performance across several datasets representing protein fold recognition, DNA-BP recognition, biological processes and molecular function recognition. In [42], different feature descriptors were extracted by Nanni *et al.*, starting from a wavelet representation of the protein, and in Kavianpour and Vasighi [43], excellent performance was obtained by extracting texture features from cellular automata images of proteins.

In contrast to extracting handcrafted features from matrix representations of proteins, another image-based method is to classify 3D protein shapes from 2D image renderings as was done in [44], where a deep learning approach was applied using a set of multi-view 2D representations of proteins taken from 3D views rendered using JMol, a well-known protein visualization tool. These images were fed into different pre-trained CNNs, and the fusion of the CNNs resulted in improved classification performance.

The goal of this study is to generate the most optimal and universal system for DNA-BP classification by combining several image-based/matrix approaches as well as other descriptors. In the literature, most ML researchers investigating the DNA-BP problem test their systems on only one or occasionally two datasets. The objective here is to produce a system that works across several datasets representing different DNA-BP classification tasks. A high-performing universal system for DNA-BP classification will provide baseline performance for future comparisons. To accomplish this objective, we generate sets of ensembles trained on many powerful protein features and image-based representations to discover which ones work best across the four DNA-BP classification tasks. Experimentally tested are the following ensemble building blocks:

(1) Sets of matrix representations generated from one-dimensional (1D) vector representations;

(2) Sets of different features extracted from these matrix representations and trained on separate SVMs, which are then combined by sum rule;

(3) Different topologies of pre-trained CNNs fine-tuned with the protein matrix representations and with 2D snapshots of the 3D protein shapes rendered by Jmol; and

(4) Different combinations of heterogeneous classifiers (CNNs combined with SVMs).

We use the same ensembles with the same set of parameters across all four DNA-BP datasets for fair comparisons to demonstrate the universality of the final system proposed here. The results section shows that the proposed ensemble achieves competitive performance with the best performing systems across all four datasets, obtaining state-of-the-art results on two.

2. Materials and methods

Since the goal of this study is to produce a universal system for classifying DNA-BP proteins, heterogeneous classifiers are built that combine sets of SVMs and pre-trained CNNs, as illustrated in Figure 1 and detailed in Section 2.1. In ML, representations should produce compact and effective fixed-length descriptors. In our proposed system, DNA-BP proteins are

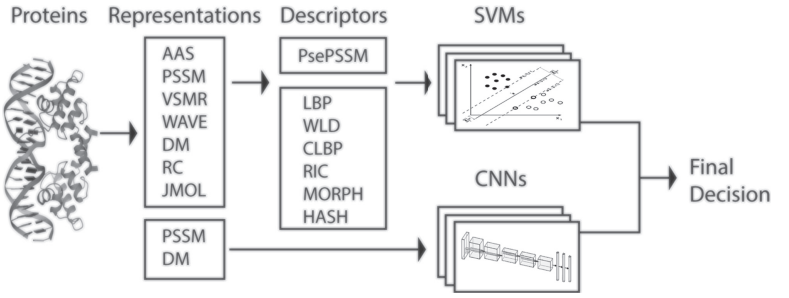
represented in seven ways (primarily matrix/image-based as described in Section 2.2). A set of CNNs with different parameter settings is fine-tuned on two representations. Seven descriptors (Section 2.3) are extracted from these representations and trained on the SVMs. These descriptors are divided into PsePSSM and a set of texture descriptors extracted from the matrix/image-based representations. For some feature extraction methods, the extraction process is applied for every physicochemical property obtained from the amino acid index database [45] available at http://www.genome.jp/dbget/aaindex.html. The amino acid index database currently contains 566 indices and 94 substitution matrices. This low number is not a problem since it is well known that a reduced number of properties is adequate for most protein classification problems. Ignored here are those properties where the amino acids have a value of 0 or 1.

2.1 Classifiers

Descriptors are extracted from the different representations, as illustrated in Figure 1, and trained on separate SVMs. SVM [46] is a popular classifier in the area of bioinformatics. SVMs are binary-class predictors that find the equation of a hyperplane that divides a two-class training set so that all the points belonging to a given class are located on the same side, with the maximum distance separating the two classes and the margin [47]. SVM handles both linear and nonlinear data. Kernel functions are used to project the data onto a higher-dimensional feature space so that they can be separated by a hyperplane in problems that do not have a linear decision boundary. SVM is implemented here with the LibSVM toolbox (http://www.csie.ntu.edu.tw/~cjlin/libsvm/) and are linearly normalized to [0, 1]. Fine-tuning is not performed on SVM: the same SVM parameters (radial basis function with gamma = 0.1 and cost = 1,000) are applied to all extracted descriptors across all four datasets to avoid overfitting.

A set of CNNs is fine-tuned on two matrix/image representations of proteins (Figure 1). CNNs are currently one of the most accurate image classification methods. They are a class of deep feed-forward neural networks composed of interconnected layers of neurons with inputs that have, like most neural networks, learnable weights, biases and activation functions. CNN layers have neurons that are arranged in three dimensions so that every layer transforms a 3D input volume into a 3D output volume of neuron activations. Typically, CNNs are composed of five classes of layers: convolutional (CONV), activation (ACT), pooling (POOL), fully-connected (FC) and classification (CLASS).

Eight CNN topologies pre-trained on ImageNet are tested in the experiments presented here: (1) AlexNet, which was the first winner of the 2012 ImageNet Large Scale Visual



Note(s): Descriptors are extracted from seven protein representations and are of two types: PsePSSM-based and image/matrix-based. These descriptors are trained on separate SVMs. CNNs are trained on PSSM and DM. The decisions of the SVMs and CNNs are combined by sum rule

Figure 1.
Schematic of the proposed approach

Recognition Challenge [48]; (2) GoogleNet [49] and (3) InceptionV3 [50], two CNNs with inception modules that approximates a sparse CNN with a normal dense construction; (4) VGGNet16 and (5) VGGNet19 [51], both of which improve AlexNet by replacing large kernel-sized filters with multiple 3×3 kernel-sized filters; (6) ResNet50 and (7) ResNet101 [52], which are CNN topologies available in MATHLAB with 50 and 101 deep layers, respectively; and (8) DenseNet [53], which is similar to ResNet but interconnects each of the layers. The pre-trained CNN topologies are fine-tuned for all layers using the training data in each dataset and for each of the matrix representations of the proteins and 2D projections.

Fine-tuning is a technique where training is resumed on a pre-trained network with the objective of learning a new classification problem. Since it is not always possible to train a CNN with large batch sizes (BS), a "GPU out of memory" message results in the removal of the CNN. Moreover, CNNs that produce random results on the training data are also eliminated since they fail to converge. If representations are not matrix-based, protein features are resized into a square matrix so that they can be trained on a CNN. Size is recalculated as the maximum size of either the rows or columns, and all empty entries are padded with zeros. The matrix is then resized as needed for input into the other pre-trained CNNs.

Once the SVMs and CNNs are trained, the classifiers are fused, as illustrated in Figure 1, with the final decision obtained by combining the pool of classifiers by the weighted sum rule.

2.2 Protein representations

The protein representations used in this work are predominately matrix-based. In our approach, they are treated as an image, which means that texture descriptors are extracted from the matrices and trained on the SVMs. As noted in Figure 1, PSSM and DM matrix/images are also fed directly into the pre-trained CNNs as other images would be (after padding and resizing as described above).

2.2.1 Amino acid sequence (AAS). Sequential-based models of protein representations start with an amino acid sequence (AAS), defined as $P = (p_1, p_2, \ldots, p_N)$, where $p_i \in \mathscr{A} = [A, C, D, \ldots, Y]$, where $\mathscr{A}$ represents the 20 native amino acids.

2.2.2 Position-specific scoring matrix (PSSM). PSI-BLAST calculates PSSM [34] by taking the PSSM profiles and searching for related proteins or DNA. The parameters considered by PSSM include (1) the position of each amino acid residue in a protein sequence; (2) the probe, which groups sequences of proteins based on functional similarity; (3) the profile matrix of 20 columns corresponding to the 20 amino acids; and (4) consensus, which are those sequences of amino acid residues that are most similar to the alignment residues of probes.

PSSM scores are integers that indicate whether an amino acid occurs more frequently than expected if positive or less frequently if negative.

2.2.3 Variant substitution matrix representation (VSMR). VSMR [35] is a variant of the substitution matrix representation (SMR) [54]. The SMR for protein $P = (p_1, p_2, \ldots, p_N)$ is a $N \times 20$ matrix calculated for VSMR as:

$$\text{VSMR}(i, j) = M(p_i, j)\, i = 1, \ldots N; j = 1, \ldots, 20, \tag{1}$$

where M is a 20×20 substitution matrix, whose element $M_{i,j}$ represents the probability of amino acid i mutating to amino acid j during the evolution process.

In the experiments reported here, 25 substitution matrix properties are randomly selected to create VSMR-based ensembles.

2.2.4 Wavelet (WAVE). Wavelet encoding starts from a protein sequence described by substituting each amino acid with a numeric value corresponding to a physicochemical property c. A decomposition scale produces different results, with high decomposition scales generating excessive redundancy and low decomposition scales discarding too much information.

The method used in this study is described in Li and Li [55]. The Meyer continuous wavelet is applied to the wavelet transform coefficients ($WAVE^c$). A feature set is extracted considering 100 decomposition scales. Twenty-five physicochemical properties are randomly selected to create a WAVE ensemble composed of 25 $WAVE^c$-based predictors.

2.2.5 3D tertiary structure (DM). DM [56] is a heatmap of the inter-residue distances (it considers the distances between atoms and between residues in a protein data bank (PDB) structure). When the size of the heatmap exceeds 250 × 250 pixels, it is resized so that the computational complexity of the feature extraction steps is manageable. DM is treated as a grayscale image from which the texture descriptors described in Section 2.3.2 are extracted.

2.2.6 Reaction center (RC). RC [57] takes a protein's structure and transforms it into two sets of feature maps: one describing the shape of the protein backbone from the torsion angles density of the local distributions of the angles φ and ψ for each amino acid and one extracted from the distances between the amino acid building blocks. These two feature maps are treated as sets of images from which the texture descriptors are extracted. Since the torsion angle densities build a map that is $m \times 19 \times 19$, it is treated as 19 images size $m \times 19$. Since the density of the amino acid distances builds a map that is $m \times m \times 8$, this feature map is treated as eight images size $m \times m$. Different descriptors are extracted from these two sets of images, and they are trained on separate SVMs that are finally combined by sum rule.

2.2.7 2D projection. 2D projection [44] takes a 3D visualization of a protein's PDB code from Jmol [58], a molecular visualization software tool. Several multi-view projections are produced by uniformly rotating the protein's 3D structure around its central X, Y and Z viewing axes to produce 125 2D images. Jmol can generate many different protein visualizations. In this study, we use Trace, Ribbons, Rockets and Strands. Trace images illustrate the secondary structures inside a molecule with a smooth curve passing through the middle points between successive atoms in the alpha carbons of a peptide chain or the phosphorus atoms of nucleic acids. Ribbons are like Trace, except that they display a line that connects the main atoms in the backbone as a solid flat ribbon. Rockets place cylinders in stretches where there are alpha helixes and planks for beta stretches, with both ending with an arrowhead. Strands are like Rockets but display the backbones as a series of thin lines so that the molecular structure is represented by parallel longitudinal threads.

2.3 Methods for protein descriptor extraction

In this section, we describe the different approaches used to extract descriptors from the protein representations introduced in Section 2.2.

2.3.1 PsePSSM (PP). The idea behind PsePSSM [59, 60] is to retain information about the AAS by considering the PseAAC. Given an input matrix $\mathrm{Mat} \in \mathfrak{R}^{N \times 20}$, the PsePSSM descriptor is a vector $PP \in \mathfrak{R}^{320}$ defined as:

$$PP(k) = \begin{cases} \dfrac{1}{N}\displaystyle\sum_{i}^{N} E(i,j) & k = 1, \ \ldots, 20 \\ \dfrac{1}{N - \mathrm{lag}}\displaystyle\sum_{i=1}^{N-\mathrm{lag}} \left[E(i,j) - E(i+\mathrm{lag},j)\right]^2 & \begin{array}{l} j = 1, \ \ldots, 20, \ \mathrm{lag} = 1, \ \ldots, 15 \\ k = 20 + j + 20 \cdot (\mathrm{lag} - 1) \end{array} \end{cases}, \tag{2}$$

where k is a linear index used to scan the cells of Mat, lag is the distance between one residue and its neighbors, N is the length of the sequence and $E \in \mathfrak{R}^{N \times 20}$ is the normalized version of Mat, defined as:

$$(i,j) = \frac{\mathrm{Mat}(i,j) - \frac{1}{20}\sum_{v=1}^{20}\mathrm{Mat}(i,v)}{\sqrt{\frac{1}{20}\sum_{u=1}^{20}\left(\mathrm{Mat}(i,u) - \frac{1}{20}\sum_{v=1}^{20}\mathrm{Mat}(i,v)\right)^2}} i = 1, \ldots, N; j = 1, \ldots, 20. \quad (3)$$

2.3.2 Texture descriptors. Many studies (see [41–43]) have demonstrated the benefit of treating a protein matrix representation as an image so that powerful texture descriptors can be extracted from it. Once these texture descriptors are extracted from the protein image, they can be trained on the classifiers, with the set of classifiers combined by sum rule for a final score.

The texture descriptors examined in this work are well regarded in the literature and include the following:

(1) Uniform LBP (LBP) [61] using two setting configurations (R, P), with R the radius and P the number of neighbors: (1, 8) and (2, 16);

(2) Weber law descriptor (WLD) [62] computed within a 3×3 block with the following parameter configurations: BETA = 5, ALPHA = 3 and number of neighbors = 8;

(3) Completed LBP (CLBP) [63] with two configurations (R, P): (1, 8) and (2, 16);

(4) Multiscale rotation invariant co-occurrence of Adjacent LBP (RIC) [64] with $R \in \{1, 2, 4\}$;

(5) A set of morphological features (MORPH) [65] composed of such measures as the aspect ratio, number of objects, area, perimeter, eccentricity and additional measures extracted from a segmented version of the image; and

(6) Default values of the heterogeneous auto-similarities of characteristics features (HASH) [66], a variant of LBP that models linear and nonlinear feature dependencies.

3. Results

3.1 Datasets

Ensembles generated from the methods described above are evaluated across four benchmark datasets taken from PDB1075 [8], PDB594 [67], PDB676 [68], PDB186 [67] (which is from the Protein Databank located at http://www.rcsb.org/pdb/home/home.do) and the dataset in [69]. Protein sequences in these datasets with less than 50 amino acids or that contain the character "X" were removed. Also deleted were all sequences having more than 25% similarity with another sequence. The PDB1075 dataset has 525 DNA-BPs and 550 DNA-non-BPs; the PDB594 dataset [67] has 297 DNA-BPs and 297 DNA-non-BPs; and the PDB186 dataset, which was designed as an independent testing dataset derived from [67], contains 93 DNA-BPs and 93 DNA-non-BPs.

The four datasets used in this paper are the following:

(1) IND1: training takes place on the PDB1075 dataset and testing on the independent PDB186 dataset;

(2) IND2: training takes place on the PDB594 dataset and testing on the independent PDB186 dataset;

(3) IND3: training takes place on the PDB1075 dataset and testing on PDB676 [68], an independent dataset that contains 338 DNA-BPs and 338 non-DNA-BPs. PDB676 was obtained (1) by searching for all DNA-binding and non-DNA-binding sequences from

PDB, (2) by removing sequences that had less than 60 amino acids or contained the character "X" and (3) by deleting all sequences having more than 25% similarity with another sequence. Filtered as well were all sequences already present in the PDB186 dataset; and

(4) IND4 [69]: training takes place on 2104 proteins and testing on 296 proteins. This new non-redundant gold-standard dataset was created following Chou's five-step rule [70]. First, 1,200 non-DNA-BPs and 1,200 DNA-BPs were collected from PDB. Second, all sequences having more than 25% similarity as well as any chain less than 50 residues in length were deleted. Also removed were any proteins containing an "X" residue. These two steps produced 1,052 proteins in each class. A total of 148 chains were chosen from each class to form the negative independent validation subset.

3.2 Performance indicators

Two performance indicators are reported in the experiments presented below: classification accuracy and area under the ROC curve (AUC). Accuracy is the ratio between the number of correctly classified samples and the total number of samples. The ROC curve is a graphical plot of the sensitivity of a binary classifier vs false positives (1 − specificity). AUC [71] is a scalar measure of the probability that the classifier will assign a lower score to a randomly picked positive pattern over a randomly picked negative pattern. With multiclass datasets, the one-versus-all area under ROC curve [72] is used. AUC is considered one of the most reliable performance indicators [73]. For this reason, the internal comparisons are evaluated using AUC. Accuracy is reported so that the system proposed here can be compared to others in the literature that do not report the AUC performance indicator. It should be noted that before each fusion, scores are normalized to mean 0 and standard deviation 1.

3.3 Experiments

In Table 1, performance is reported (with accuracy at the top and AUC at the bottom of each cell) on the set of texture (TXT) descriptors described in Section 2.3.2 (namely, LBP, WLD, CLBP, RIC, MORPH and HASH) extracted from the matrix protein representations detailed in Section 2.2 (PSSM, VSMR, WAVE, DM, RC). In Table 2, the performance of PsePSSM (PP) as a matrix-based descriptor is reported.

In Tables 1 and 2, the column labeled FUS is the fusion by sum rule of PSSM, VSMR, WAVE, DM and RC. The column labeled FUS_noPDB reports the performance of the fusion of methods not based on PDB, i.e. PSSM, VSMR and WAVE.

From the results reported above, the following conclusions can be drawn:

(1) With TXT, FUS outperforms the standalone methods on IND1, IND3 and IND4;

Table 1.
TXT for describing the different matrix protein representations (accuracy top and AUC bottom)

TXT	PSSM	VSMR	WAVE	DM	RC	FUS_noPDB	FUS
IND1	83.87%	80.65%	*86.02%*	82.26%	73.66%	80.11%	82.80%
	96.25%	93.50%	94.10%	93.20%	83.19%	88.35%	*96.51%*
IND2	66.67%	64.52%	61.29%	61.29%	*67.74%*	60.75%	65.59%
	68.46%	70.01%	67.65%	66.61%	*72.33%*	65.94%	69.94%
IND3	68.20%	73.67%	68.20%	71.30%	64.94%	72.78%	*76.04%*
	74.72%	82.11%	72.92%	79.02%	70.73%	82.24%	*83.52%*
IND4	63.51%	66.89%	59.80%	*70.27%*	60.47%	64.86%	69.59%
	69.04%	73.68%	67.13%	75.48%	69.94%	73.02%	*77.08%*

(2) PP obtains best performances, except on IND4, where TXT outperforms PP;

(3) The fusion of the different features descriptors extracted from the same matrix representation enhances performance, with FUS the average best method considering all the datasets; and

(4) Although the representations related to the PDB protein format boosts performance, the enhancement in performance using this representation is not exceptional.

In Table 3, we report the performance obtained by the deep learning ensembles labeled eCNN and Proj, as well as the performance obtained by combining, via weighted sum rule, the different methods tested in this work.

The method eCNN is the ensemble built using the following CNN topologies: AlexNet, Vgg16, Vgg19, ResNet50, ResNet101 and DenseNet, along with different combinations of learning rates (LR) ({0.001, 0.0001}) and BS ({10, 30, 50, 70}). For inputs, only two different matrix representations (DM and PSSM) are reported to reduce computation time, and, for each matrix representation, a different set of CNNs is trained. The 80 CNNs (five different CNN topologies $\times$ 2 LR $\times$ 4 BS $\times$ 2 matrix protein representations) are combined by the sum rule.

Proj is the ensemble built by combining, via the sum rule, three CNN topologies: GoogleNet, Inceptionv3 and ResNet50, coupled with the four 2D projections (Ribbons, Rockets, Strands, Trace). Only three CNN topologies are examined due to computational issues. No selection strategy is performed. For each 2D projection, a different CNN is trained, resulting in the fusion of 12 CNNs by the sum rule (due to computational issues, we combine Proj only with BS = 30 and LR = 0.001).

With the weighted sum rules, values were not determined by running a parameter selection process or by overfitting them but rather by testing a set of reasonable values (0.50 and 0.25). As noted in Table 3, the performance of PP(FUS) + 0.50 $\times$ TXT(FUS) and PP(FUS) + 0.25 $\times$ TXT(FUS) produce similar results.

In Tables 4 and 5, the best ensembles presented here are compared with the literature for IND1, IND2 and IND4. So far, IND3 is used only in [68], where an AUC of 89.78% is reported; our best method proposed here produces a similar AUC of 88.68%. As is evident in Tables 4 and 5, our proposed ensemble is similar in performance (across all four datasets) to the best-reported methods trained on each of the individual datasets. Notice that our proposed method achieves state-of-the-art performance on IND1 and IND2.

Most methods in the literature are only validated on one or two datasets/tasks while we report performance on four. This shows the generalizability of our best classifier. The excellent results of our approach can now be used for baseline comparisons by other researchers in the field.

PP	PSSM	VSMR	WAVE	DM	RC	FUS_noPDB	FUS
IND1	82.26%	82.26%	87.10%	77.42%	79.03%	82.26%	*84.41%*
	92.37%	92.67%	93.41%	85.92%	90.87%	96.67%	*96.90%*
IND2	69.89%	66.67%	55.38%	55.91%	60.22%	71.51%	*69.89%*
	76.64%	74.05%	58.12%	62.64%	66.66%	*79.49%*	77.80%
IND3	71.75%	75.74%	63.17%	66.27%	68.20%	77.51%	*78.55%*
	79.63%	83.63%	67.92%	76.25%	75.32%	86.15%	*86.83%*
IND4	65.20%	62.84%	55.74%	64.53%	63.85%	64.86%	*68.58%*
	71.95%	67.57%	57.87%	76.25%	69.43%	72.86%	*74.66%*

Table 2. PP for describing the different matrix protein representations (accuracy top and AUC bottom)

Table 3.
CNN for describing the different matrix protein representations (accuracy top and AUC bottom)

	PP(FUS)+ TXT(FUS)	PP(FUS) + 0.5 × TXT(FUS)	PP(FUS) + 0.25 × TXT(FUS)	eCNN	PP(FUS) + 0.25 × TXT(FUS) + eCNN	PP(FUS) + eCNN	Proj	PP(FUS)+eCNN+0.25 × Proj
IND1	83.87%	*84.95%*	*84.95%*	83.33%	84.41%	*84.95%*	79.03%	83.87%
	96.39%	96.76%	*96.90%*	92.23%	95.88%	95.51%	88.14%	96.09%
IND2	70.43%	69.35%	69.89%	66.13%	*72.04%*	70.43%	67.20%	71.51%
	76.41%	77.13%	77.59%	76.30%	78.01%	*79.10%*	73.46%	78.98%
IND3	77.37%	76.78%	77.22%	75.89%	77.37%	77.66%	72.34%	*79.59%*
	86.08%	86.71%	86.80%	85.55%	88.15%	88.21%	81.84%	*88.68%*
IND4	69.59%	70.27%	69.59%	72.30%	73.65%	*74.22%*	71.96%	73.65%
	75.47%	75.63%	75.54%	79.50%	79.37%	79.56%	76.77%	*80.21%*

IND1	Accuracy	AUC
PP(FUS) + eCNN + 0.25 × Proj	83.67%	*96.09%*
PP(FUS) + eCNN	84.95%	95.51%
iDNA-Prot|dis [8]	72.0%	
PseDNA-Pro [9]	–	
iDNAPro-PseAAC [29]	71.5%	
Kmer1+ACC [74]	71.0%	
Local-DPP [38]	79.0%	
[36]	76.3%	
[31]	80.6%	
[75]	80.65%	88.03%
[68]	*86.55%*	88.78%
[76]	84.31%	
[77]	80.00%	87.40%
IND2		
PP(FUS) + eCNN + 0.25 × Proj	*71.51%*	78.98%
PP(FUS)+eCNN	70.43%	*79.1%*
iDNA-Prot [8]	67.2%	–
DNA-Prot [28]	61.8%	–
DNAbinder [78]	60.8%	60.7%
DNABIND [79]	67.7%	69.4%
DBD-Threader [80]	59.7%	–
[67]	*76.9%*	*79.1%*

Table 4. Comparison with the literature on IND1 and IND2 (note: results on IND2 are taken from [67])

IND4	Accuracy	Server
PP(FUS) + eCNN + 0.25 × Proj	73.65	Here
PP(FUS) + eCNN	74.22	Here
iDNA-Prot	62.16	iDNA-Prot server at http://www.jci-bioinfo.cn/iDNA-Prot/
PseDNA-Pro	67.23	PseDNA-Pro server at http://bioinformatics.hitsz.edu.cn/PseDNA-Pro/
iDNAPro-PseAAC	66.22	iDNAPro-PseAAC server at http://bioinformatics.hitsz.edu.cn/ iDNAPro-PseAAC/
iDNA-Prot|dis	68.24	iDNA-Prot|dis server at http://bioinformatics.hitsz.edu.cn/iDNA-Prot_dis/
Local-DPP	48.65	Local-DPP server at http://server.malab.cn/Local-DPP/Index.html
PSFM-DBT	68.58	PSFM-DBT server at http://bioinformatics.hitsz.edu.cn/PSFM-DBT/
HMMBinder	50.34	Standalone version of HMMBinder [81]
IKP-DBPPred	58.11	IKP-DBPPred server at http://server.malab.cn/IKP-DBPPred/index.jsp
iDNAProt-ES (trained on PDB1075)	71.62	Implementation of iDNAProt-ES on PDB1075 (so it is an unfair comparison since a different training set is used)
iDNAProt-ES (trained on *Str*)	68.58	Implementation of iDNAProt-ES on *Str* (i.e. the training set of the official testing protocol)
DPP-PseAAC	61.15	DPP-PseAAC server at http://77.68.43.135:8080/DPP-PseAAC/
TargetDBP	76.69	Standalone version of TargetDBP [69]

Table 5. Comparison with the literature on IND4 (note: results from the literature are taken from [69])

4. Conclusions

The purpose of this study was to generate a powerful general-purpose heterogeneous ensemble for DNA-binding proteins. Experiments performed across four DNA-binding datasets show that the PsePSSM descriptor extracted from a set of matrix representations of proteins and trained on SVMs that are combined with two sets of CNNs trained on the matrix representations and the 2D projections of 3D renderings of proteins using Jmol, significantly boost classification performance. Our best ensemble obtained state-of-the-art performance across all four datasets, demonstrating generalizability.

Future studies will focus on combining other classification approaches, including ensembles made with AdaBoost and Rotation Forest. These methods, however, require enormous computational resources during the training phase.

Instead of implementing Chou's recommendation of developing a web server for identifying DNA-BP with our proposed classification system, we are sharing the MATLAB code used in this study so that the public can freely implement and set up any number of servers using our best ensemble. Sharing our source code will also allow other researchers to extend and compare our work with their novel approaches.

References

1. Luscombel NM, Austin SE, Berman HM, Thornton JM. An overview of the structures of protein–DNA complexes. Genome Biol. 2000; 11. doi: 10.1186/gb-2000-1-1-reviews001.
2. Wu CH, *et al.* The Universal Protein Resource (UniProt): an expanding universe of protein information. Nucleic Acids Res. 2006; 34(suppl_1): D187-91. doi: 10.1093/nar/gkj161.
3. Xiong Y, Zhu X, Dai H, Wei DQ. Survey of computational approaches for prediction of dna-binding residues on protein surfaces. In: Huang T. (ed). Computational systems Biology: methods in molecular Biology, 1754. New York, NY: Humana Press; 2018.
4. Schwede T. Protein modeling: what happened to the "protein structure gap"?. Structure. 2013; 21(9): 1531-1540. doi: 10.1016/j.str.2013.08.007. (in eng).
5. Chou KC. Prediction of protein cellular attributes using pseudo-amino acid composition. Proteins: Struct Func Genet. 2001; 43: 246-55.
6. Chou KC. Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology. Curr Proteonomics. 2009; 6: 262-74.
7. Cai YD, Lin SL. Support vector machines for predicting rRNA-, RNA-, and DNA-binding proteins from amino acid sequence. Biochim Biophys Acta Protein Proteonomics. 2003; 1648: 127-33.
8. Liu B, *et al.* 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. 2014; 9(9): e106691. doi: 10.1371/journal.pone.0106691.
9. Liu B, Xu J, Fan S, Xu R, Zhou J, Wang X. PseDNA-pro: DNA-binding protein identification by combining Chou's PseAAC and physicochemical distance transformation. Molecular Inform. 2015; 34: 8-17.
10. Nanni L, Lumini A. An ensemble of reduced alphabets with protein encoding based on grouped weight for predicting DNA-binding proteins. Amino Acids. 2009; 36: 167-75.
11. Zhang ZH, Wang ZH, Zhang ZR, Wang Y. A novel method for apoptosis protein subcellular localization prediction combining encoding based on grouped weight and support vector machine. FEBS (Fed Eur Biochem Soc) Lett. 2006; 580(26): 6169-74.
12. Lin H. Predicting ion channels and their types by the dipeptide mode of pseudo amino acid composition. J Theor Biol. 2011; 269(1): 64-9.
13. Fang Y, Guo Y, Feng Y, Li M. Predicting DNA-binding proteins: approached from Chou's pseudo amino acid composition and other specific sequence features. Amino Acids. 2008; 34(1): 103-9.

14. Nanni L, Lumini A. Combing ontologies and dipeptide composition for predicting DNA-binding proteins. Amino Acids. 2008; 34: 635-41.
15. Waris M, Ahmad K, Kabir M, Hayat M. Identification of DNA binding proteins using evolutionary profiles position specific scoring matrix. Neurocomputing. 2016; 199: 154-62.
16. Ding S, Zhang S, Li Y, Wang T. A novel protein structural classes prediction method based on predicted secondary structure. Biochimie. 2012; 94: 1166-71.
17. Lin H, Chen W, Yuan LF, Li ZQ, Ding H. Using over-represented tetrapeptides to predict protein locations. Acta Biotheor. 2013; 61(2): 259-68.
18. Adilina S, Farid DM, Shatabda S. Effective DNA binding protein prediction by using key features via Chou's general PseAAC. J Theor Biol. 2019; 460: 64-78.
19. Rahman MS, Shatabda S, Saha S, Kaykobad M, Rahman M. DPP-PseAAC: a DNA-binding protein prediction model using Chou's general PseAAC. J Theor Biol. 2018; 452: 22-34.
20. Hu S, Ma R, Wang H. An improved deep learning method for predicting DNA-binding proteins based on contextual features in amino acid sequences. PloS One. 2019; 14. doi: 10.1371/journal.pone.0225317.
21. Zhang Q, Zhu L, Bao W, Huang D. Weakly-supervised convolutional neural network architecture for predicting protein-DNA binding. IEEE ACM Trans Comput Biol Bioinf. 2020; 17: 679-89.
22. Qu Y, Yu HE, Gong X, Xu J, Lee HS. On the prediction of DNA-binding proteins only from primary sequences: a deep learning approach. PloS One. 2017; 12.
23. Shadab S, Khan T, Neezi NA, Adilina S, Shatabda S. DeepDBP: deep neural networks for identification of DNA-binding proteins. Informat Med Unlocked. 2020; 19: 100318.
24. Shen Z, Bao W, Huang D. Recurrent neural network for predicting transcription factor binding sites. Sci Rep. 2018; 8. doi: 10.1038/s41598-018-33321-1.
25. Yang B, *et al.* BiRen: predicting enhancers with a deep-learning-based model using the DNA sequence alone. Bioinformatics. 2017; 33(13): 1930-36. doi: 10.1093/bioinformatics/btx105.
26. Trabelsi A, Chaabane M, Ben-Hur A. Comprehensive evaluation of deep learning architectures for prediction of DNA/RNA sequence binding specificities. Bioinformatics. 2019; 35(14): i269-77. doi: 10.1093/bioinformatics/btz339.
27. Altschul SF, *et al.* Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 1997; 25: 3389-402.
28. Kumar KK, Pugalenthi G, Suganthan PN. DNA-prot: identification of DNA binding proteins from protein sequence information using random forest. J Biomol. Struct Dyn. 2009; 26(6): 679-686. doi: 10.1080/07391102.2009.10507281.
29. Liu B, Wang S, Wang X. DNA binding protein identification by combining pseudo amino acid composition and profile-based protein representation. Sci Rep. 2015; 5: 15479.
30. Xu R, *et al.* Identification of DNA-binding proteins by incorporating evolutionary information into pseudo amino acid composition via the top-n-gram approach. J Biomol Struct Dyn. 2015: 1720-30.
31. Chowdhury SY, Shatabda S, Dehzang A. iDNAProt-ES: identifcation of dna-binding proteins using evolutionary and structural features. Sci Rep. 2017; 7(1493): 1-14. [Online]. Available From: https://www.nature.com/articles/s41598-017-14945-1.pdf.
32. Chowdhury S, Shatabda S, Dehzangi A. iDNAProt-ES: identification of DNA-binding proteins using evolutionary and structural features. Sci Rep. 2017; 7. doi: 10.1038/s41598-017-14945-1.
33. Liu B, Wu H, Chou K. 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. 2017; 09: 67-91.
34. Gribskov M, McLachlan AD, Eisenberg D. Profile analysis: detection of distantly related proteins. In: Presented at the Proceedings of the National Academy of Sciences (PNAS); 1987.

35. Nanni L, Lumini A, Brahnam S. An empirical study on the matrix-based protein representations and their combination with sequence-based approaches. Amino Acids. 2013; 44(3): 887-901.
36. Wang Y, Ding Y, Guo F, Wei L, Tang J. Improved detection of DNA-binding proteins via compression technology on PSSM information. PloS One. 2017; 12(9): e0185587.
37. Feng ZP, Zhang CT. Prediction of membrane protein types based on the hydrophobic index of amino acids. J Protein Chem. 2000; 19(4): 269-75.
38. Wei L, Tang J, Zou Q. Local-DPP: an improved dna-binding protein prediction method by exploring local evolutionary information. Inf Sci. 2017; 384(April): 135-44.
39. Chou KC, Shen HB. MemType-2L: a Web server for predicting membrane proteins and their types by incorporating evolution information through -PSSM. Biochem Biophys Res Comm. 2007; 360: 339-45.
40. Breiman L. Random forest. Machine Learn. 2001; 45(1): 5-32. [Online]. Available From: http://oz.berkeley.edu/users/breiman/randomforest2001.pdf.
41. Nanni L, Shi JY, Brahnam S, Lumini A. Protein classification using texture descriptors extracted from the protein backbone image. J Theor Biol. 2010; 3(7): 1024-32.
42. Nanni L, Brahnam S, Lumini A. Wavelet images and Chou's pseudo amino acid composition for protein classification. Amino Acids. 2012; 43(2): 657-65.
43. Kavianpour H, Vasighi M. Structural classification of proteins using texture descriptors extracted from the cellular automata image. Amino Acids. 2017; 49(2): 261-71. doi: 10.1007/s00726-016-2354-5.
44. Nanni L, Lumini A, Pasquali F, Brahnam S. iProStruct2D: identifying protein structural classes by deep learning via 2D representations. Expert Syst Appl. 2020; 142(March): 113019. doi: 10.1016/j.eswa.2019.113019.
45. Kawashima S, Kanehisa M. AAindex: amino acid index database. 374. Nucleic Acids Res. 1999; 27(1): 368-69. [Online]. Available From: https://pdfs.semanticscholar.org/0e92/23abb1f973eff54d20486f0dab90c7dde9e0.pdf.
46. Vapnik V. The support vector method. presented at the Artificial Neural Networks ICANN. 1997; 97.
47. Cristianini N, Shawe-Taylor J. An introduction to support vector machines and other kernel-based learning methods. Cambridge: Cambridge University Press; 2000.
48. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ (Eds). Advances in neural information processing systems. Red Hook, New York, NY: Curran Associates; 2012; 1097-105.
49. Szegedy C, *et al.* Going deeper with convolutions. In: Presented at the IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2015.
50. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Presented at the IEEE Conference on Computer Vision and Pattern Recognition; 2016.
51. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. Cornell University. arXiv:1409.1556v6 2014.
52. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV; 2016.
53. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. CVPR. 2017; 1(2): 3.
54. Yu X, Zheng X, Liu T, Dou Y, Wang J. Predicting subcellular location of apoptosis proteins with pseudo amino acid composition: approach from amino acid substitution matrix and auto covariance transformation. Amino acids. 2011; 1619-25. doi: 10.1007/s00726-011-0848-8.
55. Li FM, Li QZ. Predicting protein subcellular location using Chou's pseudo amino acid composition and improved hybrid approach. Protein Peptide Lett. 2008; 15(6); 612-6.

56. Nanni L, Brahnam S. Set of approaches based on 3D structure and Position Specific Scoring Matrix for predicting DNA-binding proteins. Bioinformatics. 2019; 35(11): 1844-51.
57. Zacharaki EI. Prediction of protein function using a deep convolutional neural network ensemble. Peer J Comp Sci. 2017; 3: e123. [Online]. Available From: https://peerj.com/articles/cs-124/.
58. Murzin AG, Brenner SE, Hubbard T, Chothia C. SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol. 1995; 247(4): 536-40. doi: 10.1016/S0022-2836(05)80134-2.
59. Fan GL., Li QZ. Predicting protein submitochondrion locations by combining different descriptors into the general form of Chou's pseudo amino acid composition. Amino Acids. 2011; 20(Nov): 1-11.
60. Jeong JC, Lin X, Chen XW. On position-specific scoring matrix for protein function prediction. IEEE ACM Trans Comput Biol Bioinf. 2011; 8(2): 308-15.
61. Ojala T, Pietikainen M, Maeenpaa T. Multiresolution grayscale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell. 2002; 24(7): 971-87.
62. Chen J, *et al.* WLD: a robust local image descriptor. IEEE Trans Pattern Anal Mach Intell. 2010; 2(9): 1705-20.
63. Guo Z, Zhang L, Zhang D. A completed modeling of local binary pattern operator for texture classification. IEEE Trans Image Process. 2010; 19(6): 1657-63. doi: 10.1109/TIP.2010.2044957.
64. Nosaka R, Fukui K. HEp-2 cell classification using rotation invariant co-occurrence among local binary patterns. Pattern Recogn Bioinform. 2014; 47(7): 2428-36.
65. Strandmark P, Ulén J, Kahl F. HEp-2 staining pattern classification. In: Presented at the International Conference on Pattern Recognition (ICPR2012); 2012. [Online]. Available From: https://lup.lub.lu.se/search/ws/files/5709945/3437301.pdf.
66. San Biagio M, Crocco M, Cristani M, Martelli S, Murino V. Heterogeneous auto-similarities of characteristics (): exploiting relational information for classification. In: Presented at the IEEE Computer Vision (ICCV13), Sydney, Australia; 2013.
67. Lou W, Wang X, Chen F, Chen Y, Jiang B, Zhang H. Sequence based prediction of DNA-binding proteins based on hybrid feature selection using random forest and Gaussian naive Bayes. PloS One. 2014; 9(1): e86703.
68. Mishra A, Pokhrel P, Hoque MT. StackDPPred: a stacking based prediction of DNA-binding protein from sequence. Bioinformatics. 2018; 35(3): 433-41. doi: 10.1093/bioinformatics/bty653.
69. Hu J, Zhou X, Zhu Y, Yu D, Zhang G. TargetDBP: accurate DNA-binding protein prediction via sequence-based multi-view feature learning. IEEE ACM Trans Comput Biol Bioinf. 2019; 1. doi: 10.1109/TCBB.2019.2893634.
70. Chou KC. Some remarks on protein attribute prediction and pseudo amino acid composition (50th Anniversary Year Review). J Theor Biol. 2011; 273: 236-47.
71. Fawcett T. ROC graphs: notes and practical considerations for researchers. Palo Alto: HP Laboratories; 2004.
72. Landgrebe TCW, Duin RPW. Approximating the multiclass ROC by pairwise analysis. Pattern Recognit Lett. 2007; 28: 1747-58.
73. Qin ZC. ROC analysis for predictions made by probabilistic classifiers. In: Presented at the Fourth International Conference on Machine Learning and Cybernetics; 2006.
74. Dong Q, Wang S, Wang K, Liu X, Liu B. Identification of DNA-binding proteins by auto-cross covariance transformation. In: Presented at the IEEE International Conference on Bioinformatics and Biomedicine (BIBM). Washington, DC; 2015.
75. Zhang J, Liu B. PSFM-DBT: identifying dna-binding proteins by combing position specific frequency matrix and distance-bigram transformation. Int J Mol Sci. 2017; 25(18): E1856. doi: 10.3390/ijms18091856.

76. Shadab S, Alam Khan MT, Neezi NA, Adilina S, Shatabda S. DeepDBP: deep neural networks for identification of DNA-binding proteins. Inform Med Unlocked; 19: 100318. doi: 10.1016/j.imu.2020.100318.

77. Xu R, Zhou J, Wang H, He Y, Liu B, Wang X. Identifying DNA-binding proteins by combining support vector machine and PSSM distance transformation. BMC systems biology. 2015; 9(S10).

78. Kumar M, Gromiha MM, Gajendra PSR. Identification of DNA-binding proteins using support vector machines and evolutionary profiles. BMC Bioinf. 2007; 8: 463. doi: 10.1186/1471-2105-8-463.

79. Szilágyi A, Skolnick J. Efficient prediction of nucleic acid binding function from low-resolution protein structures. J Mol Biol. 2006; 358: 922-33.

80. Gao M, Skolnick J. A threading-based method for the prediction of DNA-binding proteins with application to the human genome. PLoS Comput. Biol. 2009; 5: e1000567.

81. Zaman R, Chowdhury SY, Rashid MA, Sharma A, Dehzangi A, Shatabda S. HMMBinder: DNA-binding protein prediction using HMM profile based features. BioMed Res Int.. 2017; 10. doi: 10.1155/2017/4590609.

Supplemental material

All the MATLAB code used in this paper is available at: https://github.com/LorisNanni.

Corresponding author

Sheryl Brahnam can be contacted at: sbrahnam@missouristate.edu