


RESEARCH

Open Access



# iProDNA-CapsNet: identifying protein-DNA binding residues using capsule neural networks

Binh P. Nguyen<sup>1\*</sup> , Quang H. Nguyen<sup>2</sup>, Giang-Nam Doan-Ngoc<sup>2</sup>, Thanh-Hoang Nguyen-Vo<sup>1</sup> and Susanto Rahardja<sup>3\*</sup>

From Joint 30th International Conference on Genome Informatics (GIW) & Australian Bioinformatics and Computational Biology Society (ABACBS) Annual Conference  
Sydney, Australia. 9–11 December 2019

## Abstract

**Background:** Since protein-DNA interactions are highly essential to diverse biological events, accurately positioning the location of the DNA-binding residues is necessary. This biological issue, however, is currently a challenging task in the age of post-genomic where data on protein sequences have expanded very fast. In this study, we propose iProDNA-CapsNet – a new prediction model identifying protein-DNA binding residues using an ensemble of capsule neural networks (CapsNets) on position specific scoring matrix (PSSM) profiles. The use of CapsNets promises an innovative approach to determine the location of DNA-binding residues. In this study, the benchmark datasets introduced by Hu et al. (2017), i.e., PDNA-543 and PDNA-TEST, were used to train and evaluate the model, respectively. To fairly assess the model performance, comparative analysis between iProDNA-CapsNet and existing state-of-the-art methods was done.

**Results:** Under the decision threshold corresponding to false positive rate (FPR)  $\approx 5\%$ , the accuracy, sensitivity, precision, and Matthews's correlation coefficient (MCC) of our model is increased by about 2.0%, 2.0%, 14.0%, and 5.0% with respect to TargetDNA (Hu et al., 2017) and 1.0%, 75.0%, 45.0%, and 77.0% with respect to BindN+ (Wang et al., 2010), respectively. With regards to other methods not reporting their threshold settings, iProDNA-CapsNet also shows a significant improvement in performance based on most of the evaluation metrics. Even with different patterns of change among the models, iProDNA-CapsNets remains to be the best model having top performance in most of the metrics, especially MCC which is boosted from about 8.0% to 220.0%.

**Conclusions:** According to all evaluation metrics under various decision thresholds, iProDNA-CapsNet shows better performance compared to the two current best models (BindN and TargetDNA). Our proposed approach also shows that CapsNet can potentially be used and adopted in other biological applications.

**Keywords:** Protein-DNA interaction, Residue, Prediction, PSSM, Capsule neural network, Deep learning

\*Correspondence: [binh.p.nguyen@vuw.ac.nz](mailto:binh.p.nguyen@vuw.ac.nz); [susantorahardja@ieee.org](mailto:susantorahardja@ieee.org)

<sup>1</sup>School of Mathematics and Statistics, Victoria University of Wellington, Gate 7, Kelburn Parade, 6140 Wellington, New Zealand

<sup>3</sup>School of Marine Science and Technology, Northwestern Polytechnical University, 127 West Youyi Road, 710072, Xi'an, China

Full list of author information is available at the end of the article



## Introduction

In biochemistry, the protein-DNA interaction is considered to be one of the vital activities that has strong impacts on diverse biological events including DNA synthesis, transcription, splicing, and restoration [1–3]. Therefore, high precision in determining the protein-DNA binding residues is crucial not only for protein function analysis but also for novel drug discovery [4]. For years, a lot of studies have been conducted to gain more understandings about the natural mechanism of protein-DNA interactions [5, 6]. Besides, to make an empirical confirmation on protein-DNA interaction, many high-throughput experimental advances have been designed such as protein microarray assays [7], ChIP-Seq [8], and protein binding microarray (PBM) [9]. Nevertheless, the identification of protein-DNA binding residues using experimental advances usually brings a great burden of cost and time. Since handling on experimental assays is complicated, using computation advances to identify DNA-binding residues is now preferable. On the other hand, the ceaselessly increasing number of unannotated protein sequences has motivated researchers to find better tools for this biological problem. Thus, employing computational models to predict the locations of the protein-DNA binding residues has become one of the most concerned topics in bioinformatics [1, 5, 10]. In the last decade, a number of computational models have been developed to identify DNA-binding residues [1, 11]. Based on the used features, these methods can be classified into three major groups: (i) structure-based models [12, 13], (ii) sequence-based models [10, 14], and (iii) hybrid models (using both sequence and structural features) [15].

In general, the structure-based and hybrid models frequently come up with better prediction accuracies compared to sequence-based models due to efficient exposure of specific distinctions between DNA-binding and non-binding residues [15]. The B-factor, surface curvature, and depth index (DPX), for instances, are three of numerous structure-based features that have been vastly employed to identify DNA-binding residues [15] with fairly good performances. Structure-based and hybrid models, however, also find more difficulties in circumstances in which no defined 3D-structure proteins are available. This situation is common for newly investigated proteins with only peptide sequences being determined because performing 3D-structural reconstruction for a particular protein usually time-consuming. Despite being supported by some popular homology modeling tools (e.g., MODELLER [16] and I-TASSER [17]), 3D-structural reconstruction for a new protein is still not a simple work because of large structural inconsistencies between the computer-aided rebuilt structure and the actual one, especially when appropriate structural templates are unavailable [18]. Moreover, the fast ever-growing

genome sequencing technologies also add more distance between a number of protein sequences and their rebuilt structures. Hence, using sequence-based computational models to identify DNA-binding residues seems to be more realistic and reasonable to meet the needs.

In comparison with structure-based methods, sequence-based models do not require protein structural information to predict DNA-binding residues. The last decade has seen significant growth in machine learning-based models (e.g., DNABR [19], DP-Bind [12], BindN [10], and MetaDBsite [6]) used for prediction of DNA-binding residues based on given sequences. These sequence-based methods use only protein sequence information to identify DNA-binding residues under the support of several common learning algorithms including Random Forest (RF) [20], Support Vector Machine (SVM) [21], k-Nearest Neighbors (k-NN) [22], and Extreme Gradient Boosting (XGBoost) [23]. In 2006, Wang et al. proposed BindN [10], a prediction model using the SVM algorithm receiving sequence features comprising of the hydrophobicity index, the molecular mass of a residue, and the pKa value of the side chain as model inputs. A year later, DP-Bind [14] developed by Hwang et al. was introduced as a web-based prediction tool that combined three learning algorithms encompassing kernel logistic regression, penalized logistic regression and SVM to improve the performance. This tool utilizes the position-specific scoring matrix (PSSM) profiles generated from protein sequences. In 2015, Wong et al. published a new computational model that utilized not only protein sequences but also DNA sequences to enhance feature specificity to predict possible interactions between a nucleotide and protein residues from distinctive defined DNA-binding domain families [24]. Additionally, Wong et al. also proposed the kmerHMM [25] - a hidden Markov model (HMM) utilizing belief propagations. This model is capable of adapting and converting protein binding microarray raw data into another form so-called median-binding intensities of single k-mers to recognize DNA motifs. Although these existing models have come up with certain achievements, further studies for model improvement is still needed.

Using machine learning algorithms to construct prediction models for protein-DNA binding residues is not straightforward due to the inherent data imbalance in the residues. In fact, the number of non-binding residues is predominant over that of DNA-binding residues. Therefore, using resampling techniques is currently the most frequent solution for class imbalance [26, 27]. In this scenario, over-sampling and under-sampling are the two most commonly applied techniques as described in previous studies [26–28]. Over-sampling expands the training dataset and hence training time and predicting time usually elongate. Additionally, this technique is often claimed

to cause the over-fitting problem. On the contrary, under-sampling reduces the training dataset and therefore leads to implicit risk of feature losses or weak feature characterization. In 2016, Hu et al. suggested a prediction model with a solution for the class imbalance. They proposed combining an under-sampling technique and a suitable boosting ensemble algorithm. Then, their proposed ensemble learning model was constructed using various distinctive classifiers on the modified balanced dataset [29].

In this study, we applied the capsule neural network (CapsNet) architecture [30], one of the latest deep learning approaches, on the PSSM features generated from the training and test datasets introduced by Hu et al. [29]. PSSM has been shown to be a suitable data representation for applying deep learning architectures, especially convolutional neural networks (CNNs), on various bioinformatics problems in general and on binding prediction problems in particular [31]. CapsNet is an advanced deep learning architecture; however, it has not been widely applied in bioinformatics except for a recent work on prediction of protein post-translational site modification [32]. Compared to CNN architectures having similar computation costs, CapsNet often archives better performance [30]. On small training datasets, CapsNet also outperforms CNNs as the result of having ability to characterize hierarchical relationships between simple and complex features [30, 32]. We anticipated using CapsNet with PSSM features would outperform other algorithms which have been successfully used in prediction of protein-DNA binding residues. Our method used 10-fold cross-validation and trained 10 CapsNet models from 10 sub-training datasets. To deal with the class imbalance issue, random under-sampling (RUS) [33] was applied on each sub-training dataset. Eventually, these 10 CapsNet models were ensemble and fed with the test dataset to obtain the final testing results. For a fair assessment, we compared our proposed approach with other state-of-the-art methods using the same test dataset.

## Materials and methods

### Benchmark datasets

For model construction and evaluation, we used PDNA-543 and PDNA-TEST as the training dataset and the independent test dataset, respectively. These two datasets resemble those used in the TargetDNA method [29]. Totally there are 584 non-redundant protein sequences obtained after removing redundant sequences using the CD-hit software [34] with the identity threshold of 30%. The training dataset has 543 protein sequences while the independent test dataset has 41 protein sequences. The training dataset consists of DNA-binding residues (positive samples) and 134,995 non-binding residues (negative samples). In the test dataset, there are 734 positive

samples and 14,021 negative samples. The detailed information of PDNA-543 and PDNA-TEST is summarized in Table 1.

### Feature representation

The position-specific scoring matrix (PSSM) has been widely used to extract features from protein sequences. We used the PSI-BLAST (Position-Specific Iterative Basic Local Alignment Search Tool) [35] against the Swiss-Prot database [36] with three iterations and a cut-off E-value of 0.01 to generate a PSSM profile from an input protein sequence. In the generated matrix, each of  $L$  rows represents the corresponding amino acid in the input protein sequence with length  $L$ , and each of 20 columns represents a particular amino acid among the total of 20 standard amino acids building up the protein structure. Equation (1) shows how the PSSM profile with respect to a protein  $P$  with  $L$  amino acids being calculated.

$$P_{PSSM} = \begin{bmatrix} E_{1 \rightarrow 1} & E_{1 \rightarrow 2} & \cdots & E_{1 \rightarrow j} & \cdots & E_{1 \rightarrow 20} \\ E_{2 \rightarrow 1} & E_{2 \rightarrow 2} & \cdots & E_{2 \rightarrow j} & \cdots & E_{2 \rightarrow 20} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ E_{i \rightarrow 1} & E_{i \rightarrow 2} & \cdots & E_{i \rightarrow j} & \cdots & E_{i \rightarrow 20} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ E_{L \rightarrow 1} & E_{L \rightarrow 2} & \cdots & E_{L \rightarrow j} & \cdots & E_{L \rightarrow 20} \end{bmatrix}, \quad (1)$$

where  $E_{i \rightarrow j}$  is the score of the mutation from an amino acid in the  $i^{th}$  position of the protein sequence to the standard amino acid  $j$  ( $j = \overline{1, 20}$ ) during evolution. Positive scores suggest the mutation  $E_{i \rightarrow j}$  happens more often than expected by chance while negative scores indicate the opposite. Then each score  $x$  in the PSSM profile is rescaled to the interval (0, 1) using the standard logistic function:

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (2)$$

Then, the window sliding technique was applied to the rescaled PSSM to obtain the PSSM feature vector for each amino acid since the PSSM score of a particular amino acid and its neighbors may affect the DNA-binding ability. We set the window size to 21, leading to the size of  $21 \times 20$  for each PSSM feature vector.

### Model architecture

The architecture of our proposed CapsNet model, as shown in Fig. 1, consists of two 2-dimensional convolutional layers (CNN and PrimaryCaps) and one fully connected layer (BindCaps). The first layer, CNN, detects basic features of the input PSSM corresponding to a protein sequence. The  $21 \times 20$  PSSM is convolved with 256 filters of size  $7 \times 7$  at stride 1 with ReLU activation function to produce a  $15 \times 14 \times 256$  tensor. For improving the training speed, performance, and stability of the CapsNet model and preventing overfitting, we added a batch

**Table 1** Data distribution in the training set (PDNA-543) and the independent testing set (PDNA-TEST)

Dataset	No. of Sequences	No. of Positive Samples ( <i>a</i> )	No. of Negative Samples ( <i>b</i> )	Ratio ( <i>a/b</i> )
PDNA-543	543	9,549	134,995	14.137
PDNA-TEST	41	734	14,021	19.102

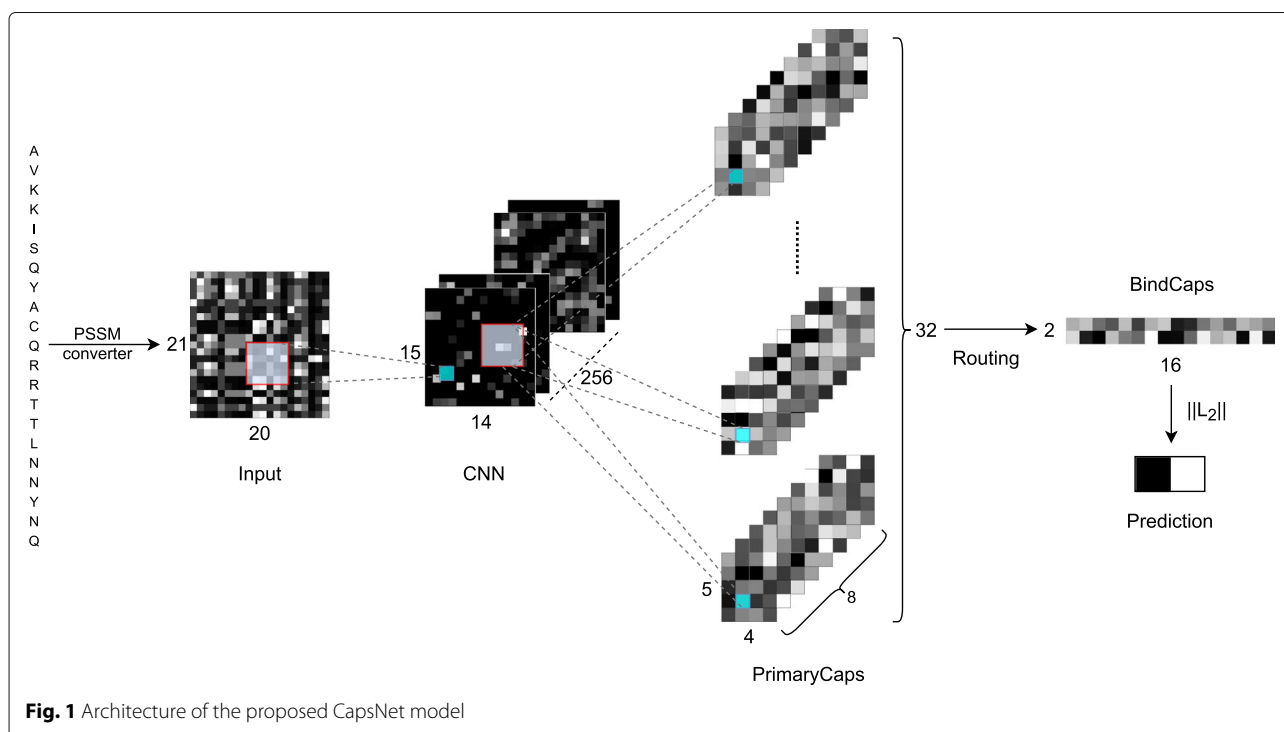
normalization sub-layer [37] and a dropout sub-layer [38] with the neuron dropping rate of 0.7 at the end of the first layer.

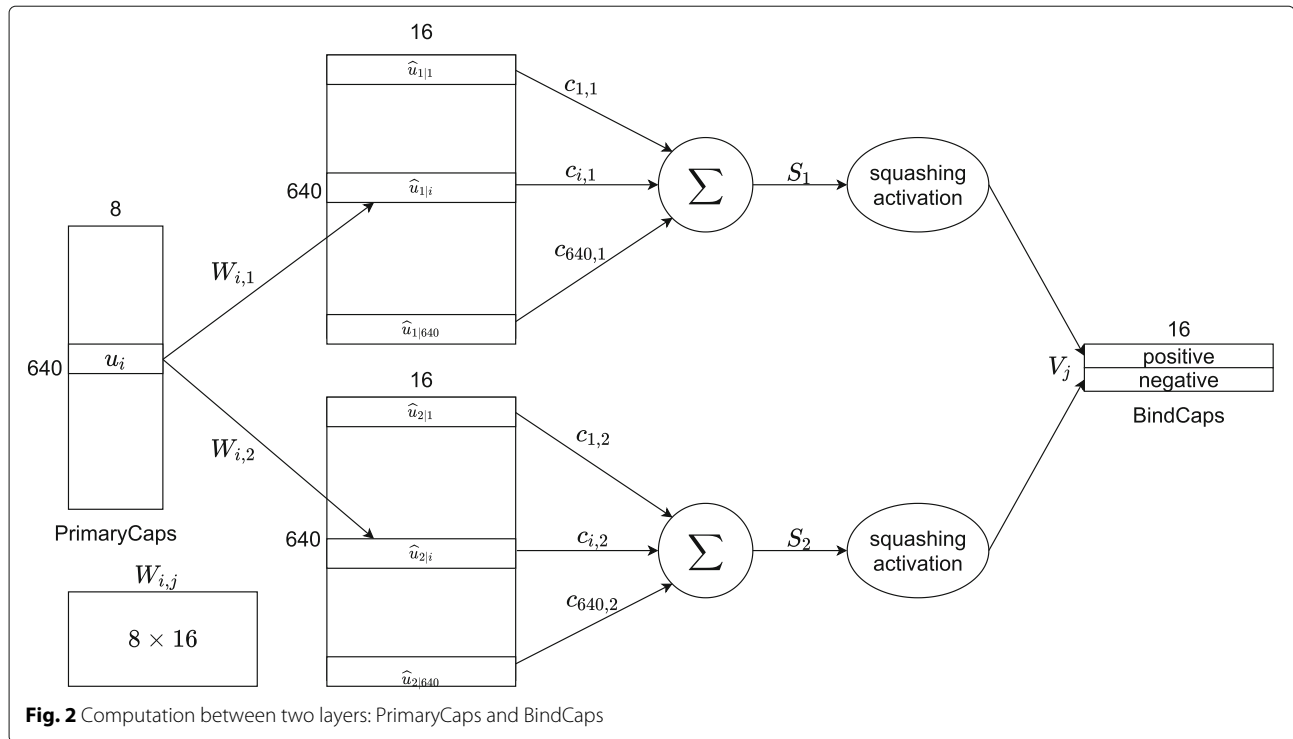
The second layer, PrimaryCaps, consists of 32 primary capsules which combines the basic features detected in the first layer. This is done by the use of 8 filters with the size of  $7 \times 7 \times 256$  with stride 2 in each capsule that takes the  $15 \times 14 \times 256$  tensor from the CNN layer as input and produces a  $5 \times 4 \times 8$  output tensor. Here, 8 is the dimension of the capsule vectors in PrimaryCaps which is similar to that in the original CapsNet architecture [30]. Since there are 32 capsules, the shape of the output of this layer is  $5 \times 4 \times 8 \times 32$ . A batch normalization sub-layer is included with a dropout sub-layer with the neuron dropping rate of 0.2 at the end of PrimaryCaps. A non-linear “squashing” function is used to scale the length of the output vector of each capsule to  $[0, 1]$  since it is the probability that the current input represents the encoded entity:

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|}, \quad (3)$$

where  $v_j$  is the vector output of capsule  $j$  and  $s_j$  is its input.

The next layer, BindCaps, has two 16-dimensional “binding” capsules corresponding to 2 possible labels of the input protein sequence: positive and negative (indicating whether protein-DNA binding exists at the centered amino acid or not). The input of each capsule in this layer is a  $5 \times 4 \times 8 \times 32$  tensor. In other words, they are  $5 \times 4 \times 32$  8-dimensional vectors, each is assigned with an  $8 \times 16$  weight matrix which then multiplies with an 8-dimensional input vector to produce a 16-dimensional vector. These 16-dimensional vectors are weighted (the weights are determined by the dynamic routing algorithm) and summed over, and the results are then passed through the squashing function to produce two 16-dimensional vectors as the output. The computation between the PrimaryCaps and BindCaps layers is illustrated in Fig. 2 and the complete dynamic routing algorithm is described in [30]. There are 640 8-dimensional capsules (each  $u_i$  is an 8-D vector) in PrimaryCaps. Each is produced by multiplying  $u_i$  by a weight matrix  $W_{ij}$  (size of  $8 \times 16$ ). Capsule vector  $V_j$  ( $j = 1$  or  $2$ ) in BindCaps is a 16-dimensional vector which is computed





by passing the weighted sum over all output  $\hat{u}_{ji}$  from PrimaryCaps through the squashing function. Parameter  $c_{ij}$  is determined by the iterative dynamic routing process. Similar to the previous two layers, a batch normalization sub-layer and a dropout sub-layer with the neuron dropping rate of 0.1 are included at the end of BindCaps. The L2-norms of the two 16-dimensional vectors are then computed to obtain the final output of the CapsNet model as a 2-dimensional vector.

The loss function for training our CapsNet model is the sum of two separate margin losses,  $L_k$  for each “binding” capsule,  $k$ :

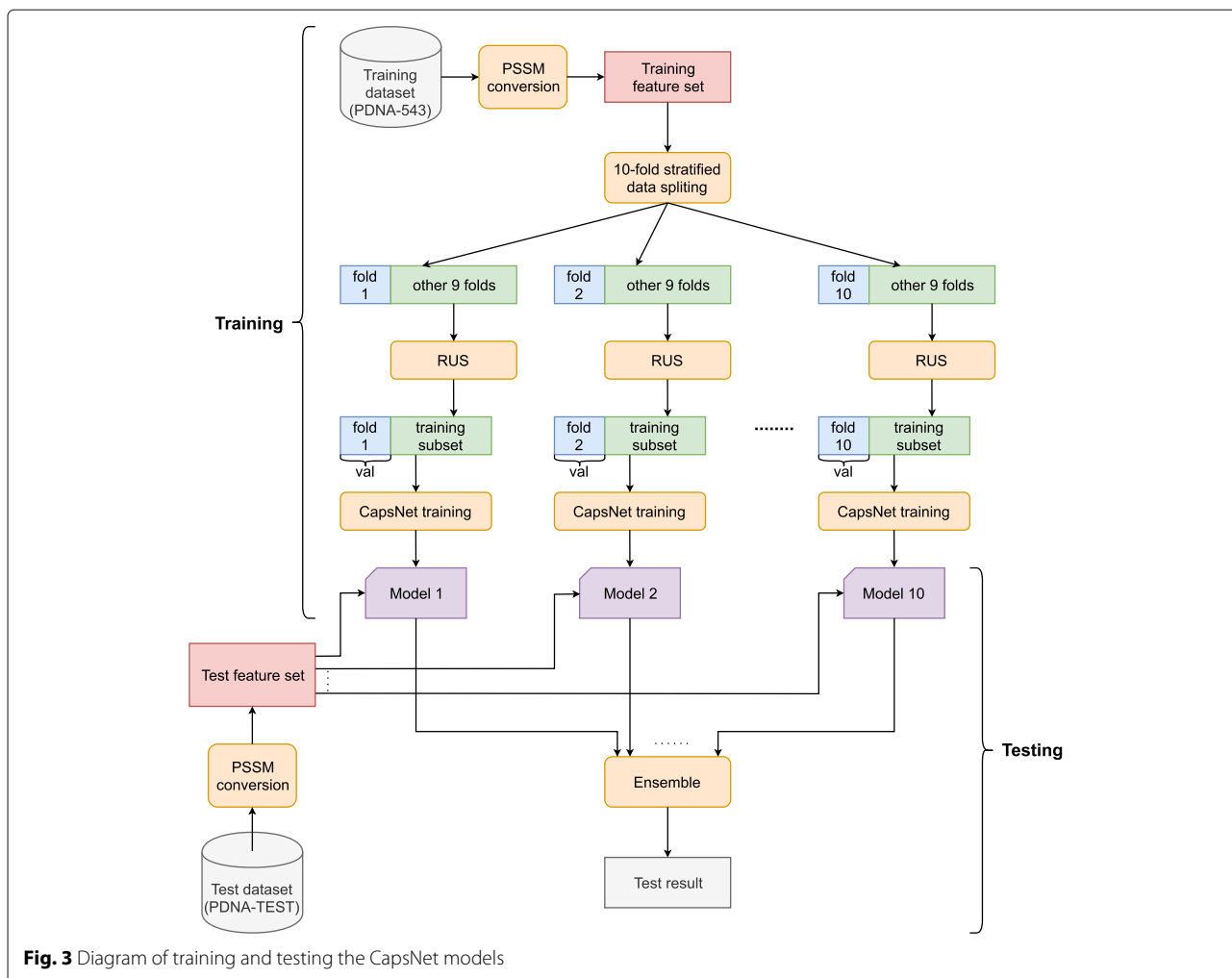
$$L_k = T_k \max(0, 0.9 - \|v_k\|)^2 + 0.5(1 - T_k) \max(0, \|v_k\| - 0.1)^2, \quad (4)$$

where  $v_k$  is the output of “binding” capsule  $k$  and  $T_k = 1$  if protein-DNA binding exists.

The architecture of our CapsNet model is relatively similar to the encoder in the original capsule neural network [30]. In addition to the differences in the number filters in each layer, we decreased the filter size in the first two layers from  $9 \times 9$  to  $7 \times 7$  and included batch normalization and dropout in all the layers. Our preliminary experimental results confirmed that reducing filter size and including dropout helped the model to be less prone to overfitting and integrating batch normalization improved validation performance.

### Model training and testing

The diagram of training and testing our model is described in Fig. 3. First, PSSM features are extracted from both the training set (PDNA-543) and the test set (PDNA-TEST) to form the training feature set and the test feature set, respectively. The training feature set is then divided into 10 mutually exclusive stratified folds, and they are combined to form 10 combinations of 9 training folds and 1 validation fold. For each of the combinations, a CapsNet model was trained using a balanced training subset created from the 9 training folds using random under-sampling (RUS) [33], and the model was optimized and validated on the validation fold. Adam optimization algorithm [39] was used along with each minibatch of 256 samples. Under the learning rate of 0.0001, the model was trained with a maximum of 300 epochs. During the training iteration, the early stopping strategy was used in such the way that if no improvement in validation loss after 20 consecutive epochs, the training process would be automatically terminated. The learning rate would be halved whenever the validation loss did not improve after 10 consecutive epochs. The number of iterations used in the routing algorithm was set to 3 (by default) and the margin loss function was employed. The best model was saved at the end of the training process. During testing, these 10 trained CapsNet models were fed with the test feature set and then the final testing results were calculated by averaging the predictions from all the models and compared with the truth labels.



**Fig. 3** Diagram of training and testing the CapsNet models

In our experiments, all the deep-learning models were implemented using Keras 2.2.4 and TensorFlow 1.13.1. Model training and testing were performed on an i5 9600k workstation with the Ubuntu 18.04.1 LTS operating system and equipped with 16GB RAM and one GPU NVIDIA GTX 1080Ti. It took about 6 seconds to train 1 epoch and 43 seconds to complete testing.

### Cross-Validation

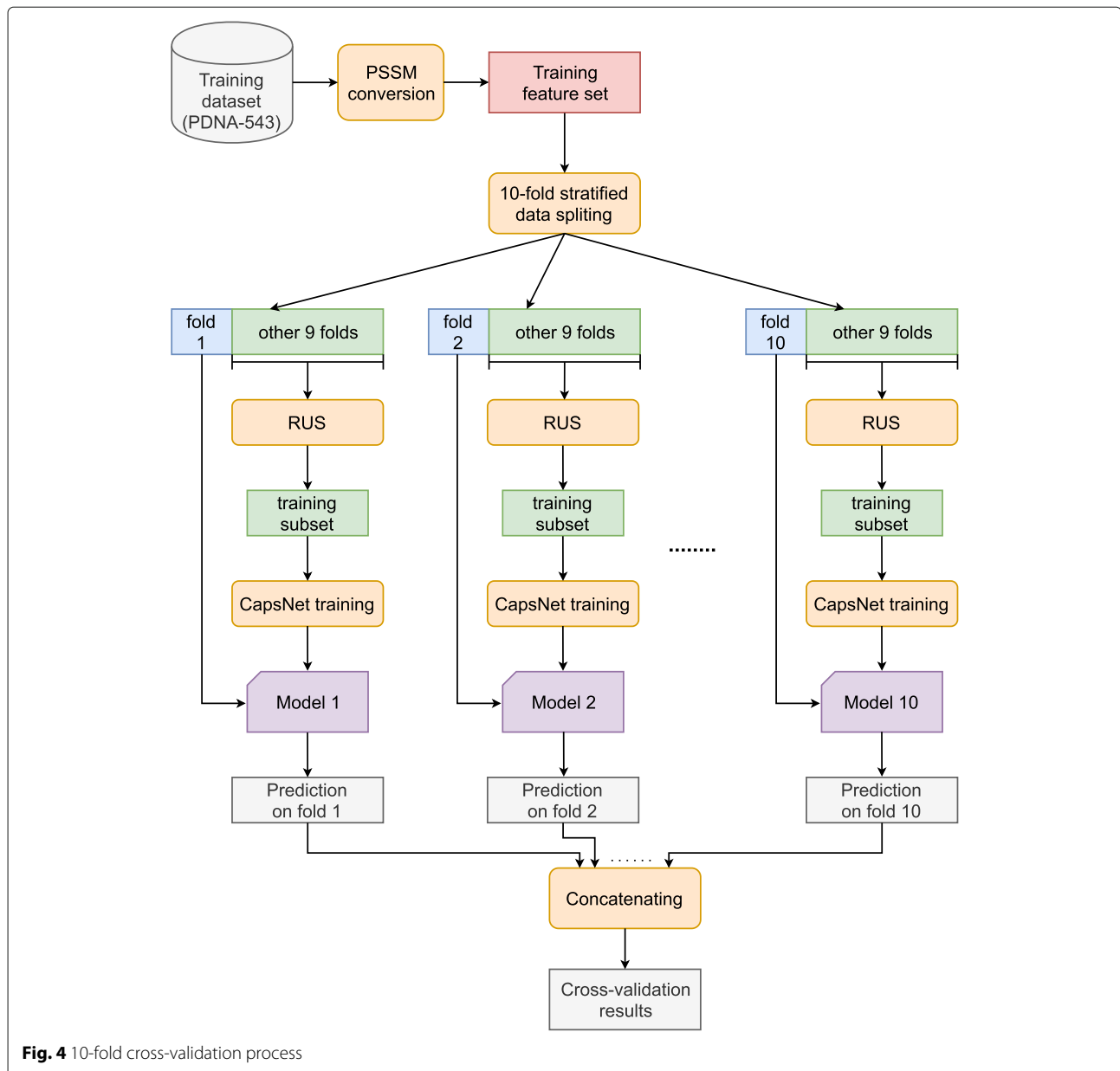
In order to compare our framework with other related methods, we also performed 10-fold cross-validation on the training dataset (PDNA-543). As shown in Fig. 4, the model training process in this case is somewhat different from that in “**Model training and testing**” section. First, we extracted the PSSM feature set from the PDNA-543 dataset. Then, the feature set was randomly split into 10 mutually exclusive folds using stratified sampling. Each fold was in turn used as the validation set while the remaining 9 folds were use as the training set for

training a CapsNet model. RUS was then applied to the training set for rebalancing whereas the validation set was left intact. The model was trained using the balanced 9-fold training dataset for 100 epochs and then tested once on the validation fold to obtain the predictions on that fold. This process was repeated for 10 times with all 10 different validation folds to produce 10 different prediction arrays. These prediction arrays were concatenated for the predictions on the whole PDNA-543 dataset and used to compare with the truth labels to produce the cross-validation results.

### Evaluation metrics

In this study, five evaluation metrics, including Sensitivity (SN), Specificity (SP), Accuracy (ACC), Precision (PR), and Matthews’s correlation coefficient (MCC) were used to evaluate the model performance. These mathematical expressions of these evaluation metrics are specified below:





$$Accuracy (ACC) = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Sensitivity (SN) = \frac{TP}{TP + FN}$$

$$Specificity (SP) = \frac{TN}{TN + FP}$$

$$Precision (PR) = \frac{TP}{TP + FP}$$

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

- where TP, FP, TN, and FN are abbreviated terms of True Positive, False Positive, True Negative, and False Negative values, respectively. These evaluation metrics, however, changes under the adjustment of the decision threshold when making prediction. To fairly compare our proposed approach with state-of-the-art methods, we set the threshold in the same way as in [29], i.e., select the threshold so that we have the following cases: (i) FPR (False Positive Rate, which is equal to 1 - Specificity)  $\approx 15\%$ , (ii) FPR  $\approx 5\%$ , and (iii) SN  $\approx$  SP (Sensitivity is approximately equal to Specificity) during cross-validation and testing. We also set the threshold so that FPR  $\approx 8\%$  when testing. In addition, since the Area Under the Curve (AUC)

of the Receiver Operating Characteristic (ROC) curve is independent on the threshold, we also used AUC as an important metric to evaluate the performance of our models. Higher AUC values mean better performances.

## Results and discussion

### Cross-Validation and model evaluation

Tables 2 and 3 show the performances of our models on the training dataset (PDNA-543) using 10-fold cross-validation and on the test dataset (PDNA-TEST), respectively, under four different settings of the decision threshold including (i) default threshold = 0.5, (ii) FPR  $\approx$  5% (SP  $\approx$  0.95), (iii) FPR  $\approx$  15% (SP  $\approx$  0.85), and (iv) SN  $\approx$  SP. The agreement between cross-validation outcomes and testing outcomes under the four different settings shows that our models are good at generalization. This is also confirmed by the two almost-identical ROC curves in Fig. 5.

For the PDNA-543 dataset, among the four settings, the accuracy when FPR  $\approx$  5% holds the first place, followed by those when FPR  $\approx$  15%, SP  $\approx$  SN, and the default threshold. The specificity and precision when FPR  $\approx$  5% are also significantly higher than those of other thresholds. When FPR  $\approx$  5%, the accuracy, specificity, and precision are increased by about 9.0–22.0%, 12.0–27.0%, and 46.0–92.0%, respectively, compared to the other setups. The MCC of the model increases following the decrease in FPR and varies between 0.282 and 0.313. Under the threshold corresponding to FPR  $\approx$  15%, the MCC is higher than the other setups. In contrast, using the default threshold leads to the highest value of sensitivity.

For the PDNA-TEST dataset, under the threshold corresponding to FPR  $\approx$  5%, the accuracy, specificity, precision, and MCC come up with significantly higher values compared to other setups. The accuracy drops by about 10.0%, 22.0%, and 22.0% when changing the setting from FPR  $\approx$  5% to FPR  $\approx$  15%, SP  $\approx$  SN, and the default threshold, respectively. Using the threshold corresponding to SP  $\approx$  SN returns higher sensitivity compared to other setups but not significantly different from using the default threshold due to only small adjustment between sensitivity and specificity.

Besides, we also set another threshold so that FPR  $\approx$  8% in order to observe possibly new trend of change. This set-

**Table 2** 10-fold cross-validation performances of iProDNA-CapsNet on the training dataset (PDNA-543) under various decision thresholds

Setting	ACC (%)	SN (%)	SP (%)	PR (%)	MCC	AUC
Threshold = 0.5	74.73	<b>77.38</b>	74.55	17.32	0.282	0.832
FPR $\approx$ 5%	<b>91.21</b>	36.31	<b>95.00</b>	<b>33.34</b>	0.301	0.832
FPR $\approx$ 15%	83.66	64.21	85.00	22.78	<b>0.313</b>	0.832
SP $\approx$ SN	76.02	76.02	76.02	17.93	0.287	0.832

Values which are significantly higher than the others are in bold

**Table 3** Performances of iProDNA-CapsNet on the test dataset (PDNA-TEST) under various decision thresholds

Setting	ACC (%)	SN (%)	SP (%)	PR (%)	MCC	AUC
Threshold = 0.5	75.72	74.79	75.77	13.59	0.245	0.833
FPR $\approx$ 5%	<b>92.38</b>	42.17	<b>94.93</b>	<b>29.78</b>	<b>0.315</b>	0.833
FPR $\approx$ 8%	91.13	45.73	93.45	26.23	0.302	0.833
FPR $\approx$ 15%	84.05	65.38	85.00	18.17	0.285	0.833
SP $\approx$ SN	75.34	<b>75.36</b>	75.34	13.47	0.245	0.833

Values which are significantly higher than the others are in bold

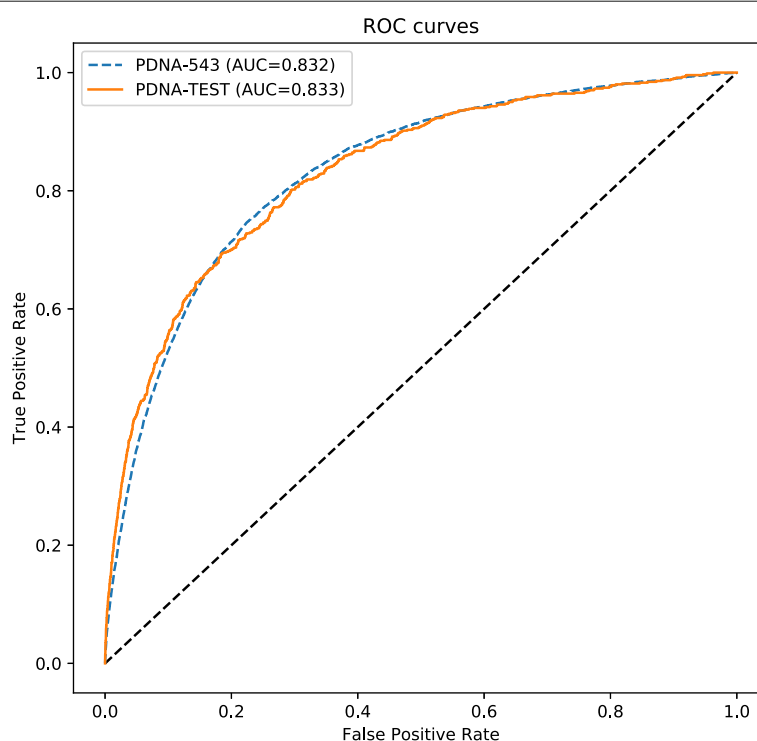
ting gives similar performance compared to the case when FPR  $\approx$  5% with smaller values of all the metrics except sensitivity.

### Comparative analysis

Table 4 shows the performance of our models compared with that of other state-of-the-art methods (data excerpted from [29]) including BindN [10], ProteDNA [40], MetaDBSite [6], DP-Bind [14], DNABind [41], BindN+ [42], and TargetDNA [29]. All the methods were tested on the same test dataset (PDNA-TEST), and among those methods, only BindN+ and TargetDNA provided performance information with two settings corresponding to FPR  $\approx$  5% and FPR  $\approx$  15%.

Under the threshold corresponding to FPR  $\approx$  5%, the accuracy, sensitivity, precision, and MCC of our model increases by about 2.0%, 2.0%, 14.0%, and 5.0% with respect to TargetDNA and 1.0%, 75.0%, 45.0%, and 77.0% with respect to BindN+, respectively. In comparison with BindN+, our model come up with a significant improvement in precision (45.0%) and MCC (about 77.0%) and these surges are very meaningful to indicate the small variation among different-run values as well as high stability of our model. In comparison with TargetDNA, our model's precision remarkably rises by 14.0% and this outgrowth reflects the considerable decline in variation among the different-run values. Besides, the specificity among methods in comparison is not significantly different. Therefore, under the threshold corresponding to FPR  $\approx$  5%, our proposed method seems to cover the weaknesses of both TargetDNA and BindN+. On the other hand, under the threshold corresponding to FPR  $\approx$  15%, the accuracy in our model, TargetDNA, and BindN+ all decrease by about 10.0%, 7.5%, and 9.5% and these declines are not far different from each other. Additionally, the specificity of the three models also drops in the range from 9.0% to 12.0%. In terms of precision, the fall in this metric decreases from our model (64.0%), followed by BindN+ (33.0%) and TargetDNA (27.0%). With regards to MCC, our method and TargetDNA share a common pattern of downward change, while upward change is observed in BindN+. When changing the threshold corresponding to FPR  $\approx$  5%





**Fig. 5** ROC curves for iProDNA-CapsNet on PDNA-543 (blue dashed line) in model testing and on PDNA-TEST (orange solid line) in 10-fold cross-validation

to that corresponding to  $FPR \approx 15\%$ , the sensitivity of our model, TargetDNA, and BindN+ all demonstrates with a remarkable growth by about 65.0%, 80.0%, and 100.0%, respectively, as a result of a trade-off between sensitivity and specificity.

Among the methods with unknown threshold settings, ProteDNA [40] is the only one having higher accuracy and precision compared to our model. However, our model's

MCC is twice as high as ProteDNA's MCC. Given the crucial role of MCC over other metrics as confirmed [43], we can therefore take that our model remains its competitive role with high stability. For the rest of the other approaches including MetaDBSite [6], DP-Bind [14], and DNABind [41]), our proposed method also shows a significant improvement in accuracy, precision, and MCC. The sensitivity of our model under the threshold

**Table 4** Performance comparison between iProDNA-CapsNet and other state-of-the-art methods

Method	Setting	ACC (%)	SN (%)	SP (%)	PR (%)	MCC
BindN	Unknown	79.15	45.64	80.90	11.12	0.143
ProteDNA	Unknown	95.11	4.77	99.84	60.30	0.160
MetaDBSite	Unknown	90.41	34.20	93.35	21.22	0.221
DP-Bind	Unknown	81.40	61.72	82.43	15.53	0.241
DNABind	Unknown	79.78	70.16	80.28	15.70	0.264
BindN+	FPR $\approx 5\%$	91.58	24.11	<b>95.11<sup>a</sup></b>	20.51	0.178
	FPR $\approx 15\%$	83.69	50.81	85.41	15.42	0.213
TargetDNA	FPR $\approx 5\%$	90.89	<b>45.50<sup>a</sup></b>	93.27	26.13	0.300
	FPR $\approx 15\%$	<b>84.52<sup>b</sup></b>	60.22	<b>85.79<sup>b</sup></b>	18.16	0.269
iProDNA-CapsNet	FPR $\approx 5\%$	<b>92.38<sup>a</sup></b>	42.17	94.93	<b>29.78<sup>a</sup></b>	<b>0.315<sup>a</sup></b>
	FPR $\approx 15\%$	84.05	<b>65.38<sup>b</sup></b>	85.00	<b>18.17<sup>b</sup></b>	<b>0.285<sup>b</sup></b>

Values which are significantly higher than the others are in bold with <sup>a</sup>FPR  $\approx 5\%$  and <sup>b</sup>FPR  $\approx 15\%$

corresponding to  $FPR \approx 5\%$  is about 900.0% and 125.0% higher than ProteDNA and MetaDBSite respectively while this metric under the threshold corresponding to  $FPR \approx 15\%$  notably grown by about 150.0%, >1,300.0%, 200.0%, and 6.0% compared to BindN, ProteDNA, MetaDBSite, and DP-Bind. The precisions of our model under the threshold corresponding to  $FPR \approx 5\%$  and the threshold corresponding to  $FPR \approx 15\%$  are remarkably higher than those of other methods. The improvement in precision fluctuated from roughly 140.0% (compared to MetaDBSite) to 270.0% (compared to BindN). Although ProteDNA has been reported to have an accuracy of 95.11%, a specificity of 99.84%, and a precision of 60.30%. Among the three reported metrics of ProteDNA, only its precision is far higher than that of our model but its very low MCC has weakened the trust-ability of this model. Among all the methods, our model obtained a meaningfully higher MCC which are boosted from about 8.0% (compared to DNABind) to 220.0% (compared to ProteDNA).

Our method, iProDNA-CapsNet, achieves very good values for MCC [43], which is an important metric whose crucial role has been far confirmed. Among the common evaluation metrics, MCC is the only one that uses up all information of the confusion matrix. Moreover, with respect to imbalanced datasets, MCC is the most important and informative metric that correctly assesses whether a prediction model is stable and robust while a single accuracy metric would not be sufficient to determine that status. With this dataset, MCC is therefore the most important metric. Eventually, under the threshold corresponding to  $FPR \approx 5\%$  and  $FPR \approx 15\%$ , our model shows its superior performance and high stability compared to other methods.

### Software availability

We deployed our model to an user-friendly and freely accessible web server at <https://github.com/ngphubinh/iProDNA-CapsNet>. Users can easily submit a protein sequence in FASTA format and receive the prediction result of protein-DNA binding residues in the sequence. We provided four different settings of the decision threshold as specified in this manuscript, including the (i) default threshold = 0.5, (ii)  $FPR \approx 5\%$  ( $SP \approx 0.95$ ), (iii)  $FPR \approx 15\%$  ( $SP \approx 0.85$ ), and (iv)  $SN \approx SP$ . The procedure to predict protein-DNA binding residues starts on our web server when there is a query protein sequence submitted in the FASTA format along with a decision threshold and an optional email address. The corresponding PSSM profile is then extracted by PSI-BLAST incorporated into our server, and subsequently the PSSM feature set is derived by placing a sliding window on each amino acid in the sequence (with zero padding in some of the first and last amino acids). Finally, the feature set is submitted to our iProDNA-Capsnet model for prediction, and the results

will be sent back to users. When viewing the result page, users can choose a desired decision threshold from a list ranging from 0.05 to 0.95 with the step of 0.05 to refine the prediction result on the submitted protein sequence.

### Conclusions

In this paper, a novel deep learning framework - CapsNet combining with PSSM features is proposed for the prediction of protein-DNA binding residues. iProDNA-CapsNet has significantly better performance than the state-of-the-art methods. In particular, the robustness and efficiency of iProDNA-CapsNet have been demonstrated by the remarkable improvement in most of the evaluation metrics, especially for MCC and accuracy. Additionally, the application of a new deep learning architecture so-called CapsNet to address this biological issue opens a new direction in similar topics, for example, RNA-protein binding [44] and ATP-protein bindings [45].

### Abbreviations

ATP: Adenosine triphosphate; AUC: Area under the ROC curve; CapsNets: Capsule neural networks; CNN: Convolutional neural network; DNA: Deoxyribonucleic acid; FPR: False positive rate; HMM: Hidden Markov model; k-NN: k-nearest neighbors; MCC: Matthew's correlation coefficient; PSI-BLAST: Position-specific iterative basic local alignment search tool; PSMM: Position specific scoring matrix; ReLU: Rectified linear unit; RF: Random forest; RNA: Ribonucleic acid; ROC: Receiver operating characteristic; RUS: Random under-sampling; SVM: Support vector machine; XGBoost: Extreme gradient boosting

### Acknowledgements

B.P. Nguyen and Q.H. Nguyen gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPUs used for this research.

### About this supplement

This article has been published as part of *BMC Bioinformatics Volume 20 Supplement 23, 2019: Proceedings of the Joint International GIW & ABACBS-2019 Conference: bioinformatics*. The full contents of the supplement are available online at <https://bmcbioinformatics.biomedcentral.com/articles/supplements/volume-20-supplement-23>.

### Authors' contributions

BPN designed the framework and experiments. QHN and G-ND-N developed the code and performed the experiments. BPN, G-ND-N, and T-HN-V interpreted the experimental results and drafted the manuscript. SR contributed to interpretation of data and significantly revised the manuscript. All authors have read and approved the final manuscript.

### Funding

The authors received no specific funding for this work. The publication costs were covered by the authors.

### Availability of data and materials

The benchmark dataset used in this study were collected from the previous work by Hu et al., 2017. The benchmark dataset were downloaded and processed under the instruction described in the paper entitled "Predicting Protein-DNA Binding Residues by Weightedly Combining Sequence-Based Features and Boosting Multiple SVMs" by Hu et al. (<https://doi.org/10.1109/TCBB.2016.2616469>). A web server implementing the proposed method is available at <https://github.com/ngphubinh/iProDNA-CapsNet>.

### Ethics approval and consent to participate

Not applicable.

### Consent for publication

Not applicable.

# Competing interests

The authors declare that they have no competing interests.

# Author details

<sup>1</sup>School of Mathematics and Statistics, Victoria University of Wellington, Gate 7, Kelburn Parade, 6140 Wellington, New Zealand. <sup>2</sup>School of Information and Communication Technology, Hanoi University of Science and Technology, 1 Dai Co Viet, 100000 Hanoi, Vietnam. <sup>3</sup>School of Marine Science and Technology, Northwestern Polytechnical University, 127 West Youyi Road, 710072, Xi'an, China.

Received: 22 November 2019 Accepted: 26 November 2019

Published: 27 December 2019

# References

1. Si J, Zhao R, Wu R. An overview of the prediction of protein DNA-binding sites. *Int J Mol Sci*. 2015;16(3):5194–215. <https://doi.org/10.3390/ijms16035194>.
2. Aeling KA, Steffen NR, Johnson M, Wesley Hatfield G, Lathrop RH, Senear DF. DNA deformation energy as an indirect recognition mechanism in protein-DNA interactions. *IEEE/ACM Trans Comput Biol Bioinforma*. 2007;4(1):117–25. <https://doi.org/10.1109/TCBB.2007.1000>.
3. Wong K-C, Li Y, Peng C, Wong H-S. A comparison study for DNA motif modeling on protein binding microarray. *IEEE/ACM Trans Comput Biol Bioinforma*. 2015;13(2):261–71. <https://doi.org/10.1109/TCBB.2015.2443782>.
4. Schmidtke P, Barril X. Understanding and predicting druggability. A high-throughput method for detection of drug binding sites. *J Med Chem*. 2010;53(15):5858–67. <https://doi.org/10.1021/jm100574m>.
5. Ahmad S, Gromiha MM, Sarai A. Analysis and prediction of DNA-binding proteins and their binding residues based on composition, sequence and structural information. *Bioinformatics*. 2004;20(4):477–86. <https://doi.org/10.1093/bioinformatics/btg432>.
6. Si J, Zhang Z, Lin B, Schroeder M, Huang B. MetaDBSite: a meta approach to improve protein DNA-binding sites prediction. *BMC Syst Biol*. 2011;5(1):7. <https://doi.org/10.1186/1752-0509-5-51-57>.
7. Ho S-W, Jona G, Chen CT, Johnston M, Snyder M. Linking DNA-binding proteins to their recognition sequences by using protein microarrays. *Proc Natl Acad Sci*. 2006;103(26):9940–5. <https://doi.org/10.1073/pnas.0509185103>.
8. Valouev A, Johnson DS, Sundquist A, Medina C, Anton E, Batzoglou S, Myers RM, Sidow A. Genome-wide analysis of transcription factor binding sites based on ChIP-Seq data. *Nat Methods*. 2008;5(9):829. <https://doi.org/10.1038/nmeth.1246>.
9. Berger MF, Philippakis AA, Qureshi AM, He FS, Estep III PW, Bulky ML. Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities. *Nat Biotechnol*. 2006;24(11):1429. <https://doi.org/10.1038/nbt1246>.
10. Wang L, Brown SJ. BindN: a web-based tool for efficient prediction of DNA and RNA binding sites in amino acid sequences. *Nucleic Acids Res*. 2006;34(Suppl 2):243–8. <https://doi.org/10.1093/nar/gkl298>.
11. Miao Z, Westhof E. A large-scale assessment of nucleic acids binding site prediction programs. *PLoS Comput Biol*. 2015;11(12):1004639. <https://doi.org/10.1371/journal.pcbi.1004639>.
12. Jones S, Barker JA, Nobeli I, Thornton JM. Using structural motif templates to identify proteins with DNA binding function. *Nucleic Acids Res*. 2003;31(11):2811–23. <https://doi.org/10.1093/nar/gkg386>.
13. Tjong H, Zhou H-X. DISPLAR: an accurate method for predicting DNA-binding sites on protein surfaces. *Nucleic Acids Res*. 2007;35(5):1465–77. <https://doi.org/10.1093/nar/gkm008>.
14. Hwang S, Gou Z, Kuznetsov IB. DP-Bind: a web server for sequence-based prediction of DNA-binding residues in DNA-binding proteins. *Bioinformatics*. 2007;23(5):634–6. <https://doi.org/10.1093/bioinformatics/btl672>.
15. Li B-Q, Feng K-Y, Ding J, Cai Y-D. Predicting DNA-binding sites of proteins based on sequential and 3D structural information. *Mol Genet Genom*. 2014;289(3):489–99. <https://doi.org/10.1007/s00438-014-0812-x>.
16. Webb B, Sali A. Comparative protein structure modeling using MODELLER. *Curr Protoc Bioinforma*. 2014;47(1):5–6. <https://doi.org/10.1002/0471250953.bi0506s15>.
17. Zhang Y. I-TASSER server for protein 3D structure prediction. *BMC Bioinformatics*. 2008;9(1):40. <https://doi.org/10.1186/1471-2105-9-40>.
18. Amirkhani A, Kolahdoozi M, Wang C, Kurgan L. Prediction of DNA-binding residues in local segments of protein sequences with fuzzy cognitive maps. *IEEE/ACM Trans Comput Biol Bioinforma*. 2018. <https://doi.org/10.1109/TCBB.2018.2890261>.
19. Ma X, Guo J, Liu H-D, Xie J-M, Sun X. Sequence-based prediction of DNA-binding residues in proteins with conservation and correlation information. *IEEE/ACM Trans Comput Biol Bioinforma*. 2012;9(6):1766–75. <https://doi.org/10.1109/TCBB.2012.106>.
20. Breiman L. Random forests. *Mach Learn*. 2001;45(1):5–32. <https://doi.org/10.1023/A:1010933404324>.
21. Cortes C, Vapnik V. Support-vector networks. *Mach Learn*. 1995;20(3):273–297. <https://doi.org/10.1023/A:1022627411411>.
22. Altman NS. An introduction to kernel and nearest-neighbor nonparametric regression. *Am Stat*. 1992;46(3):175–85. <https://doi.org/10.2307/2685209>.
23. Chen T, Guestrin C. XGBoost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM; 2016. p. 785–794. <https://doi.org/10.1145/2939672.2939785>.
24. Wong K-C, Li Y, Peng C, Moses AM, Zhang Z. Computational learning on specificity-determining residue-nucleotide interactions. *Nucleic Acids Res*. 2015;43(21):10180–9. <https://doi.org/10.1093/nar/gkv1134>.
25. Wong K-C, Chan T-M, Peng C, Li Y, Zhang Z. DNA motif elucidation using belief propagation. *Nucleic Acids Res*. 2013;41(16):153. <https://doi.org/10.1093/nar/gkt574>.
26. He H, Garcia EA. Learning from imbalanced data. *IEEE Trans Knowl Data Eng*. 2008;9:1263–84. <https://doi.org/10.1109/TKDE.2008.239>.
27. Hu J, Li Y, Yan W-X, Yang J-Y, Shen H-B, Yu D-J. KNN-based dynamic query-driven sample rescaling strategy for class imbalance learning. *Neurocomputing*. 2016;191:363–73. <https://doi.org/10.1016/j.neucom.2016.01.043>.
28. Yu D-J, Hu J, Tang Z-M, Shen H-B, Yang J, Yang J-Y. Improving protein-ATP binding residues prediction by boosting SVMs with random under-sampling. *Neurocomputing*. 2013;104:180–90. <https://doi.org/10.1016/j.neucom.2012.10.012>.
29. Hu J, Li Y, Zhang M, Yang X, Shen H-B, Yu D-J. Predicting protein-DNA binding residues by weightedly combining sequence-based features and boosting multiple SVMs. *IEEE/ACM Trans Comput Biol Bioinforma*. 2017;14(6):1389–98. <https://doi.org/10.1109/TCBB.2016.2616469>.
30. Sabour S, Frosst N, Hinton GE. Dynamic routing between capsules. In: *Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, editors. Advances in Neural Information Processing Systems 30*. New York: Curran Associates, Inc.; 2017. p. 3856–66.
31. Le N-Q-K, Nguyen BP. Prediction of FMN binding sites in electron transport chains based on 2-D CNN and PSSM profiles. *IEEE/ACM Trans Comput Biol Bioinforma*. 2019;1–9. <https://doi.org/10.1109/TCBB.2019.2932416>.
32. Wang D, Liang Y, Xu D. Capsule network for protein post-translational modification site prediction. *Bioinformatics*. 2019;35(14):2386–94. <https://doi.org/10.1093/bioinformatics/bty977>.
33. Lemaître G, Nogueira F, Aridas CK. Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. *J Mach Learn Res*. 2017;18(17):1–5.
34. Li W, Godzik A. CD-HIT: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*. 2006;22(13):1658–9. <https://doi.org/10.1093/bioinformatics/btl158>.
35. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*. 1997;25(17):3389–402. <https://doi.org/10.1093/nar/25.17.3389>.
36. Bairoch A, Apweiler R. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res*. 2000;28(1):45–8. <https://doi.org/10.1093/nar/28.1.45>.
37. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning, ICLR'15*; 2015. p. 448–56.
38. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. *J Mach Learn Res*. 2014;15:1929–58.

39. Kingma DP, Ba J. Adam: A method for stochastic optimization. CoRR. 2014;abs/1412.6980v1:1–9. [1412.6980v1](https://arxiv.org/abs/1412.6980v1).
40. Chu W-Y, Huang Y-F, Huang C-C, Cheng Y-S, Huang C-K, Oyang Y-J. ProteDNA: a sequence-based predictor of sequence-specific DNA-binding residues in transcription factors. *Nucleic Acids Res.* 2009;37(suppl\_2):396–401. <https://doi.org/10.1093/nar/gkp449>.
41. Szilágyi A, Skolnick J. Efficient prediction of nucleic acid binding function from low-resolution protein structures. *J Mol Biol.* 2006;358(3):922–33. <https://doi.org/10.1016/j.jmb.2006.02.053>.
42. Wang L, Huang C, Yang MQ, Yang JY. BindN+ for accurate prediction of DNA and RNA-binding residues from protein sequence features. *BMC Syst Biol.* 2010;4(1):3. <https://doi.org/10.1186/1752-0509-4-S1-S3>.
43. Chicco D. Ten quick tips for machine learning in computational biology. *BioData Min.* 2017;10(1):35. <https://doi.org/10.1186/s13040-017-0155-3>.
44. Ren H, Shen Y. RNA-binding residues prediction using structural features. *BMC Bioinformatics.* 2015;16(1):249. <https://doi.org/10.1186/s12859-015-0691-0>.
45. Chen K, Mizianty MJ, Kurgan L. ATPsite: sequence-based prediction of ATP-binding residues. In: *Proteome Sci. BioMed Central*; 2011. p. 4. <https://doi.org/10.1186/1477-5956-9-S1-S4>.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Ready to submit your research? Choose BMC and benefit from:**

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

**At BMC, research is always in progress.**

Learn more [biomedcentral.com/submissions](https://biomedcentral.com/submissions)

