PLOS ONE

RESEARCH ARTICLE

An improved deep learning method for predicting DNA-binding proteins based on contextual features in amino acid sequences

Siquan Hu[1,2]*, **Ruixiong Ma**[1], **Haiou Wang**[3]

1 School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China, **2** Sichuan Jiuzhou Video Technology Co., Ltd, Mianyang, China, **3** School of Chemistry and Biological Engineering, University of Science and Technology Beijing, Beijing, China

* husiquan@ustb.edu.cn

Citation: Hu S, Ma R, Wang H (2019) An improved deep learning method for predicting DNA-binding proteins based on contextual features in amino acid sequences. PLoS ONE 14(11): e0225317. https://doi.org/10.1371/journal.pone.0225317

Editor: Seyed Reza Shahamiri, Manukau Institute of Technology, NEW ZEALAND

Received: February 24, 2019

Accepted: November 2, 2019

Published: November 14, 2019

Data Availability Statement: All the source codes and data used in this study are available from the figshare server https://doi.org/10.6084/m9.figshare.8131244.

Funding: This work was supported by: (1) Sichuan Science and Technology Program (No. 2018GZDZX0034), P.R. China, author: Siquan Hu; (2) Sichuan Jiuzhou Video Technology Co., Ltd, Mianyang, China, author: Siquan Hu. The funders provided support in the form of salaries for author (SH), but did not have any additional role in the study design, data collection and analysis, decision

Abstract

As the number of known proteins has expanded, how to accurately identify DNA binding proteins has become a significant biological challenge. At present, various computational methods have been proposed to recognize DNA-binding proteins from only amino acid sequences, such as SVM, DNABP and CNN-RNN. However, these methods do not consider the context in amino acid sequences, which makes it difficult for them to adequately capture sequence features. In this study, a new method that coordinates a bidirectional long-term memory recurrent neural network and a convolutional neural network, called CNN-BiLSTM, is proposed to identify DNA binding proteins. The CNN-BiLSTM model can explore the potential contextual relationships of amino acid sequences and obtain more features than can traditional models. The experimental results show that the CNN-BiLSTM achieves a validation set prediction accuracy of 96.5%—7.8% higher than that of SVM, 9.6% higher than that of DNABP and 3.7% higher than that of CNN-RNN. After testing on 20,000 independent samples provided by UniProt that were not involved in model training, the accuracy of CNN-BiLSTM reached 94.5%—12% higher than that of SVM, 4.9% higher than that of DNABP and 4% higher than that of CNN-RNN. We visualized and compared the model training process of CNN-BiLSTM with that of CNN-RNN and found that the former is capable of better generalization from the training dataset, showing that CNN-BiLSTM has a wider range of adaptations to protein sequences. On the test set, CNN-BiLSTM has better credibility because its predicted scores are closer to the sample labels than are those of CNN-RNN. Therefore, the proposed CNN-BiLSTM is a more powerful method for identifying DNA-binding proteins.

Introduction

As a part of the protein family, DNA binding proteins play an important role in RNA editing, methylation and other biological processes. [1]. According to current research, DNA can bind to more than 3% of eukaryotic and prokaryotic proteins [2,3]. In cellular function research, the ability to recognize DNA-binding proteins is a highly meaningful task [4].

to publish, or preparation of the manuscript. The specific roles of these authors are articulated in the 'author contributions' section.

Competing interests: The author (SH) is employed by Sichuan Jiuzhou Video Technology Co., Ltd, Mianyang, China. This does not alter our adherence to PLOS ONE policies on sharing data and materials.

In recent decades, some biological experimental approaches have been proposed to discriminate among DNA-binding proteins. For example, "protein blotting" uses SDS-polyacrylamide gel to detect DNA-binding proteins [5]. Hugh et al. identified DNA binding proteins by electrostatic potential and structural units [6]. However, these biological experimental approaches often require considerable time and involve expensive materials; thus, computational methods have advantages compared to experimental methods for identifying DNA binding proteins from massive amounts of data [7].

Computational approaches have advantages in processing sequential data, and there is growing evidence that predicting DNA-binding proteins solely from primary sequences is effective compared with the experimental methods [8–10]. Many computational approaches for predicting DNA binding proteins by primary sequences have been introduced, and machine learning and deep learning methods are among the best.

Models such as the support vector machine (SVM), random forest and other algorithms that belong to the machine learning category have been used to predict DNA binding proteins, [11]. For example, Cai et al. used nonlinear characteristic sets in amino acid sequences to predict DNA-binding proteins using an SVM [12]. Some methods that combine machine learning with mathematical techniques have also been created. For example, by combining the random forest algorithm with the "gray model", Lin et al. created a DNA-binding protein recognizer called iDNA-Prot [13], which reduced the computational time and is thus suitable for large-scale analysis. Wang et al. used normalization, discrete wavelet alteration and cosine alteration to address sequence features, and they used the processed feature set to train an SVM to obtain the predictors of DNA binding proteins [14]. To better capture sequence features, Zou et al. used features from four types of proteins to train an SVM classifier that used three diverse approaches in the eigen transformation process [15]. Rahman et al. proposed a new computational method to identify DNA binding proteins that used a random forest to recognize sequence characteristics and an SVM as the classifier [16]. Chowdhury et al. proposed a new method called iDNAProt-ES, which trains an SVM to obtain a classification model based on the evolutionary messages and structural characteristics of proteins [17]. Liu et al. constructed a modular model framework by combining multi view features and classifiers; the features come from the sequence structure, physical and chemical properties, evolutionary messages and predictive structure messages [18]. This framework can flexibly coordinate different prediction models and perform well on datasets. Adilinaet al. improved the method for extracting sequence features by adopting grouping and recurrent selection to process the obtained feature sets. Their approach reduced the overfitting degree of the model [19]. With the development of network technology, some web implementations for discriminating DNA-binding proteins have been created that can provide online predictive services. One such web site is iDNA-Prot|dis, which incorporated an SVM and linkage information of the primary protein sequence [20] to improve the predictive accuracy of DNA-binding proteins compared to previous methods. Ma et al. designed a more accurate DNA-binding proteins predictor, DNABP, by adopting the random forest algorithm and considering the physical and chemical characteristics of amino acids [21].

The traditional machine learning methods have shown unmatched superiority for solving small-scale data identification problems [22,23]. However, the traditional machine learning methods are difficult to apply to massive samples [24]. Fortunately, the emergence of deep learning has solved the dilemma of traditional machine learning. Deep learning is a new technology based on neural network architecture that has been highly successful at image recognition, voice recognition and many other tasks. [25–27]. Importantly, deep learning can be applied to large amounts of sample data.

Some scholars have perceived the advantages of deep learning methods and applied them to predict DNA-binding proteins [7,28,29]. Delong et al. were the first to show that protein

features can be identified by deep learning; this work provided the original idea for the prediction of DNA-binding proteins [28]. Zeng et al. went further, using a convolutional neural network (CNN) to predict DNA-binding proteins [29]. Qinhu et al. proposed a new method based on a CNN combined with instance learning to identify DNA binding proteins; this method fully considers the inherent weak supervision information of sequences to improve the effect [30]. Recently, Qu et al. combined a CNN with a recurrent neural network (CNN-RNN) to predict DNA-binding proteins [7]. Previous work has improved the flexibility with which features of protein sequences can be extracted. Compared with the machine learning methods, the application of deep learning not only made it possible to use millions of protein sequences for model training but also improved the prediction accuracy by approximately 5.9 percentage points.

However, it is worth noting that neither the CNN used in [29] nor the recurrent neural network used in [7] take context into account when processing sequence information. From previous works other than protein sequences, contextual relationship has been found to be important features of sequence information [31]. These considerations inspired us to wonder whether amino acid sequence also contain contextual features. If contextual relationships exist in amino acid sequences, capturing those features might improve the ability to predict DNA-binding proteins.

Regarding the above question, some other studies on the context of amino acid sequences have also inspired us. Ashraf et al. found that the contextual relationships among amino acid sequences are important sequence features and have a positive effect when predicting protein structures [32,33]. The early GOR method has achieved preliminary success in second level architecture prognosis by using alternation statistics based on context and information theory. [34]. Starosta et al. found that a change in the context of consecutive proline triggers (PPP) in both the NlpD and LepA sequences had an important impact on its function, as shown in Fig 1 [35]. These studies imply that contextual relationships do exist in amino acid sequences; thus, exploiting this information will contribute to improving the prediction accuracy of DNA-binding proteins.

Assuming that there is a primary sequence of a protein S = (ALQPGGS. . .), the contextual features of S can be expressed as follows:

$$F(S) = \sum_{i=1}^{n} \sum_{j=1}^{n} com(S_i, S_j)(i \neq j).$$

In the above formula, n represents the length of the sequence, and S_i and S_j represent the i and j elements in the sequence. In addition, $com(S_i,S_j)$ represents the functional affinity scores

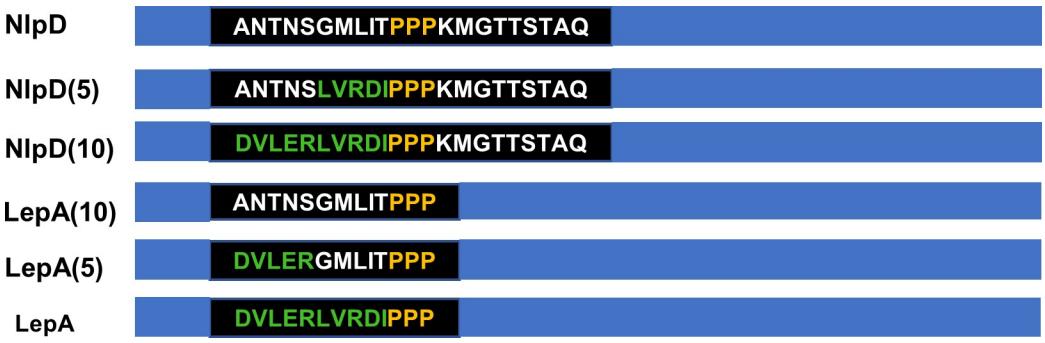

Fig 1. Changes in the context of PPP in both the NlpD and LepA sequences had an important impact on its function.

https://doi.org/10.1371/journal.pone.0225317.g001

of the two elements. The affinity scores represent the functional information expressed by amino acids in a specific context. The sum of the affinity scores between different amino acids in the whole sequence constitutes the contextual features.

To capture the contextual features of amino acid sequences, it is helpful to use the bidirectional long short-term memory recurrent neural network (Bi-LSTM), which is a recent deep learning network development [36–38]. A Bi-LSTM effectively captures the contextual information of statements or sequences. which can potentially improve contextual feature extraction and thus achieve a better recognition effect for DNA-binding proteins.

In this paper, we apply a new deep learning model, named CNN-BiLSTM, to identify DNA-binding proteins. The CNN improves model robustness and reduces the error by using the backpropagation algorithm and loss function, while the Bi-LSTM mines the relationships between the contexts in both the forward and backward directions. CNN-BiLSTM includes two convolutional layers, two pooling layers and a Bi-LSTM recurrent neural network layer. In this model, an amino acid can be considered a "word" in a sequence, and multiple amino acids are considered to be a "phrase". The roles of these "words" or "phrases" are influenced by the context in which they are located, just as a word has different meanings in different contexts, as shown in Fig 2.

Compared with the preceding models for identifying DNA binding proteins, our method captures the contextual features of amino acid sequences and exhibits better performance. The experiments show that our method is more robust than are previous methods in terms of its ability to generalize from the training dataset; consequently, it is more accurate at predicting DNA-binding proteins.

Materials

The Universal Protein Resource (UniProt) is a repository that contains a large number of protein sequences, and the raw dataset was manually annotated and reviewed [39]. We obtained the sequences of DNA binding proteins from UniProt. Our use of the UniProt website conformed with its terms of use.

In the process of extracting protein data from UniProt, we excluded sequences shorter than 50 and longer than 1,280; we need to limit the input data length to the deep learning model because a longer length requires more computing resources needed. Most of the proteins in the database are in the 50–1280 range. Therefore, to collect data of different lengths effectively, we chose this standard. Finally, we obtained 17,151 positive samples from UniProt, all of which were marked as DNA binding protein sequences. Simultaneously, we obtained 50,000 negative samples from UniProt, none of which were marked as DNA binding protein sequences. For these positive and negative samples, we randomly selected 85% for training and used the remaining 15% as validation sets for model training. (see Table 1 for specific circumstances).

The number of positive and negative samples in the table above are not equivalent. Therefore, we established a balanced dataset to carry out comparative experiments. The training

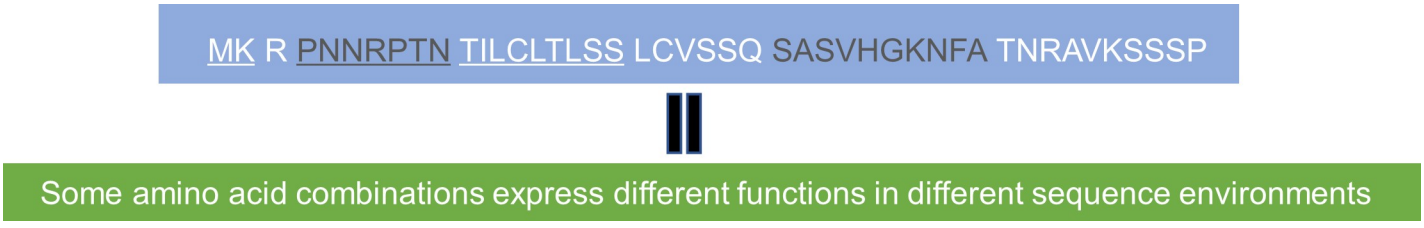

Fig 2. An amino acid sequence can be considered a sentence.

https://doi.org/10.1371/journal.pone.0225317.g002

Table 1. Unbalanced dataset for model training.

Dataset	Positive	Negative	Total
Original set	17,151	50,000	67,151
Training set (85%)	14,578	42,500	57,078
Validation set (15%)	2,573	7,500	10,073

https://doi.org/10.1371/journal.pone.0225317.t001

data include both positive and negative samples of 14,578 protein sequences, and the validation set data included 2,573 protein sequences. The comparative experiment is helpful for investigating the effect of having balanced positive and negative samples on the prediction accuracy. Additionally, to confirm the universality of CNN-BiLSTM, we adopted the datasets used in [40] for testing. In addition, we used 32,000 samples from [7] as training sets to compare the differences in the predictive scores of different models on independent sample sets. (see Table 2 for the specific circumstances).

To test the model generalizability, we have also prepared a tagged test set whose data are not used during model training. This dataset included 500 units with 500 positive and 500 negative samples. At the same time, we also prepared a large test set with 10,000 units to test the model effect on a large dataset. The test results of the model on these sets can reflect the real prediction level of the model. (see Table 3 for the specific circumstances).

Methods

Deep learning is both a set of algorithms and a branch of machine learning. Deep learning can approximate complex functions through multiple neural network layers and represent abstract data. Deep learning uses the backpropagation algorithm to update the internal model weights, and deep learning can discover the characteristics of complex data and pass them on to the next layer of the network [41].

A CNN can process data in matrix form well and extract its features effectively through feature mapping [42]. Unlike a CNN, RNN is designed to address sequence information [43]. However, RNNs suffer from time lag problems during training because they often encounter disappearing gradient or gradient explosion problems during training.

Bidirectional long-term memory recurrent neural network

The long short-term memory recurrent neural network, first proposed by Hochreiter & Schmidhuber in 1997, was originally designed to address long time lag problems in RNNs [44]. Sometimes, however, predictions need to be determined by considering both previous and subsequent inputs. Therefore, Zhang et al. proposed the bidirectional long short-term memory network to process sequence information [45]. The network is first calculated forward from time 1 to time t in the forward layer. The output of the hidden layer at each timepoint is obtained and saved, as shown in Fig 3. Then, in the backward layer, the outcome of

Table 2. Balanced experiment dataset for model training.

Dataset	Positive samples	Negative samples	Total
Original set	17,151	17,151	34,302
Training set	14,578	14,578	29,156
Validationt set	2,573	2,573	5,146
Arabidopsis	100	100	200
Dataset from the literature	16,000	16,000	32,000

https://doi.org/10.1371/journal.pone.0225317.t002

Table 3. Test set.

Dataset	Positive	Negative	Total
Test set (500)	500	500	1,000
Test set (10,000)	10,000	10,000	20,000

https://doi.org/10.1371/journal.pone.0225317.t003

the hidden layer at each time is obtained and saved by the reverse calculation, from time t to time 1. Finally, the output takes into account the results of both the forward layer and the backward layer.

Deep learning model

There are 20 amino acids that make up proteins. Each amino acid is represented by a capital letter [46]. We use different numbers to represent different types of amino acids. (see Fig 4 for details).

The deep learning model is composed of the following four parts: a coding layer, an embedding layer, a convolutional layer and a Bi-LSTM layer. The coding layer represents each amino acid as a particular number. The embedding layer translates the amino acid sequences into continuous vectors. The convolution layer consists of two convolutions and two max pooling operations. The goal of the Bi-LSTM is to grasp the contextual features of amino acid sequences. (see Fig 5 for the specific circumstances).

Similar to the use of the deep learning method in the field of image recognition, we use filters in the convolutional layer to obtain the features of protein sequences and further extract their main features in the pooling layer. Then, we utilize the Bi-LSTM to acquire the contextual features of amino acid sequences. In this work, an amino acid sequence is treated as a complete sentence, and each amino acid is treated as a word. We not only capture the characteristic information of the entire sequence but also capture the effects of the contextual features of sequences on amino acid combinations. All the information obtained by our model serves as a predictor of DNA-binding proteins.

The model execution process

To clarify the above process, we take the sequence ***Seq* = *MAAITIAN*** as an example input and illustrate its flow through the layers. (see Fig 6 for details).

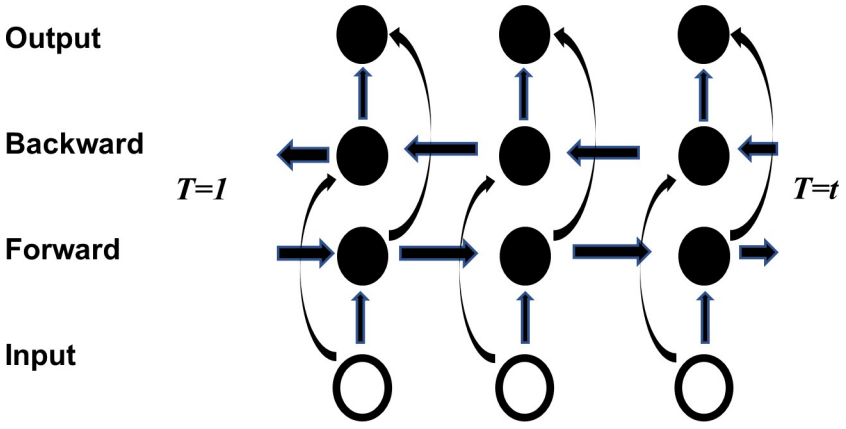

Fig 3. Bi-LSTM structure.

https://doi.org/10.1371/journal.pone.0225317.g003

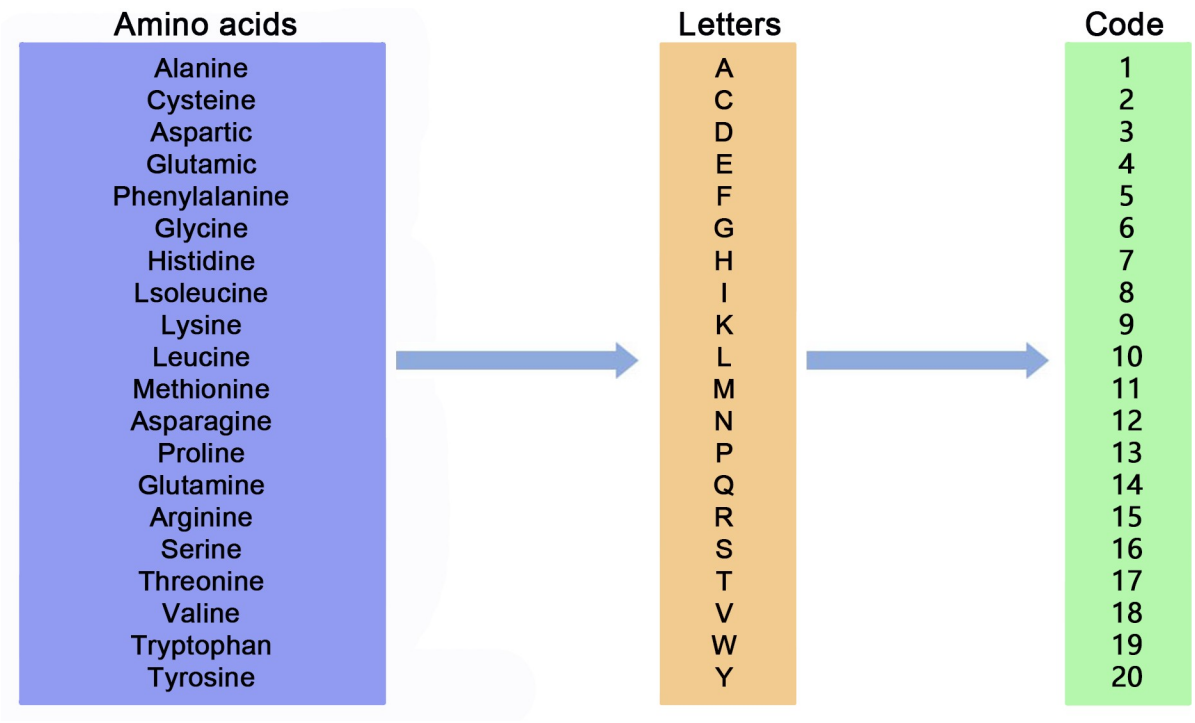

Fig 4. Amino acid encoder.

https://doi.org/10.1371/journal.pone.0225317.g004

Note that the maximum length of protein sequences in the dataset used in this paper is 1,280 because the length range of most of the protein sequences in our dataset is 50–1280. During the coding process, sequences shorter than 1,280 are zero-filled at the end to keep all the coded sequences aligned. To simplify this concept, in Fig 6, we assume that the maximum sequence length is 9.

Thus, during coding, the sequence ***Seq*** becomes a list of numbers after passing through the coding layer, as shown in (1).

$$Seq_1 = Encoding(Seq) = (11, 1, 1, 8, 17, 8, 1, 12) \quad (1)$$

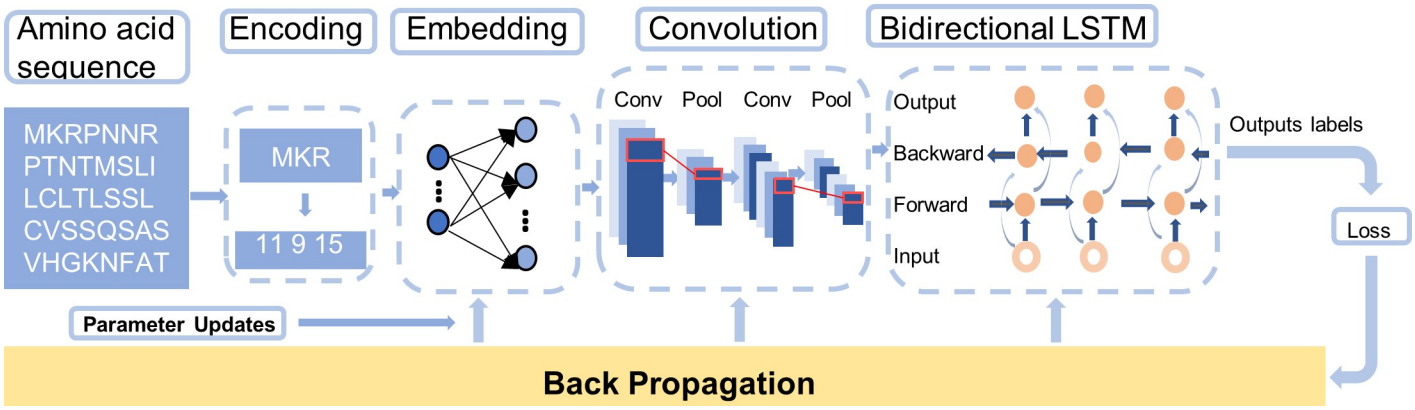

Fig 5. Model.

https://doi.org/10.1371/journal.pone.0225317.g005

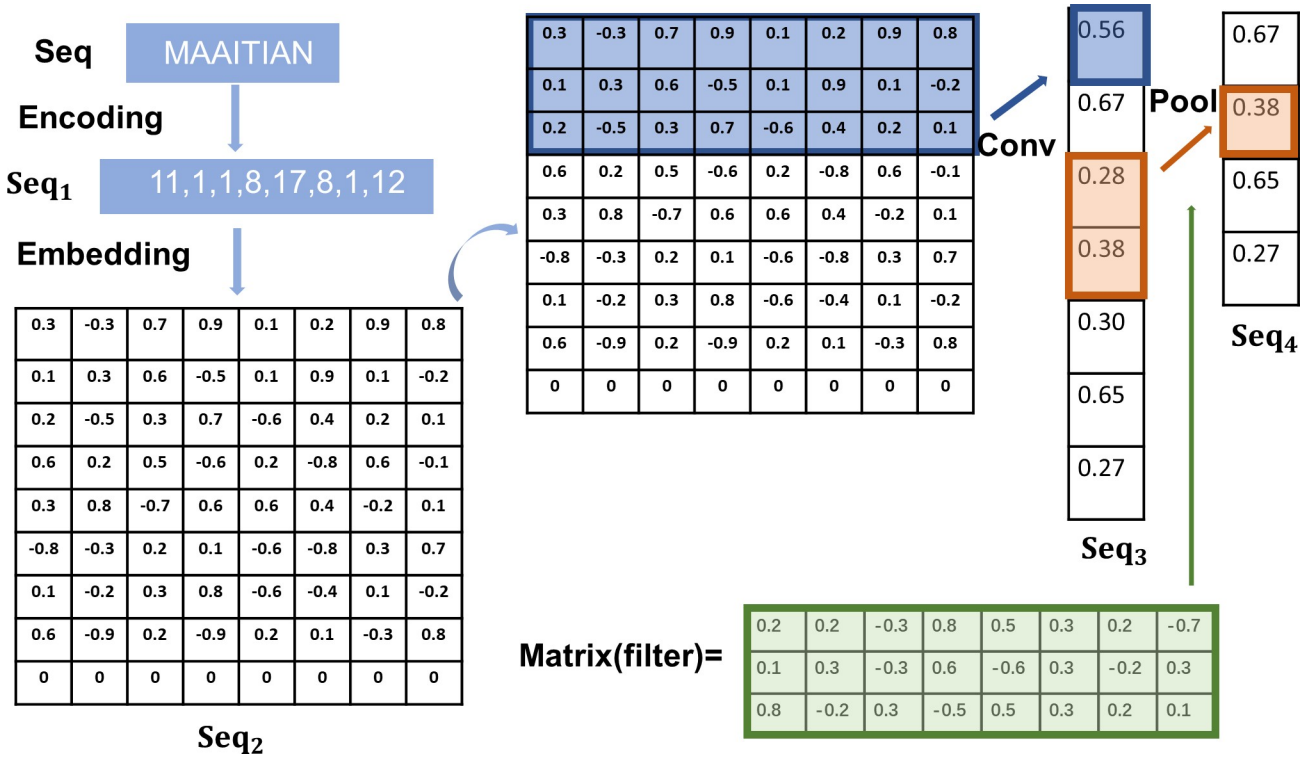

Fig 6. Embedding and convolution.

https://doi.org/10.1371/journal.pone.0225317.g006

Next, the sequence is transformed into a multidimensional matrix, as shown in (2).

$$Seq_2 = Embedding(Seq_1) \begin{bmatrix} 0.3 & -0.3 & 0.7 & 0.9 & 0.1 & 0.2 & 0.9 & 0.8 \\ 0.1 & 0.3 & 0.6 & -0.5 & 0.1 & 0.9 & 0.1 & -0.2 \\ 0.2 & -0.5 & 0.3 & 0.7 & -0.6 & 0.4 & 0.2 & 0.1 \\ 0.6 & 0.2 & 0.5 & -0.6 & 0.2 & -0.8 & 0.6 & -0.1 \\ 0.3 & 0.8 & -0.7 & 0.6 & 0.6 & 0.4 & -0.2 & 0.1 \\ -0.8 & -0.3 & 0.2 & 0.1 & -0.6 & -0.8 & 0.3 & 0.7 \\ 0.1 & -0.2 & 0.3 & 0.8 & -0.6 & -0.4 & 0.1 & -0.2 \\ 0.6 & -0.9 & 0.2 & -0.9 & 0.2 & 0.1 & -0.3 & 0.8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2)$$

In the convolutional layer, we use the filter matrix to scan $\boldsymbol{Seq_2}$ and obtain $\boldsymbol{Seq_3}$, which is also a matrix, as shown in (3) and (4).

$$\text{Matrix(filter)} = \begin{bmatrix} 0.2 & 0.2 & -0.3 & 0.8 & 0.5 & 0.3 & 0.2 & -0.2 \\ 0.1 & 0.3 & -0.3 & 0.6 & -0.6 & 0.3 & -0.2 & 0.3 \\ 0.8 & -0.2 & 0.3 & -0.5 & 0.5 & 0.3 & 0.2 & 0.1 \end{bmatrix} \quad (3)$$

$$Seq_3 = Conv(Seq_2) = \begin{bmatrix} 0.56 \\ 0.67 \\ 0.28 \\ 0.38 \\ 0.30 \\ 0.65 \\ 0.27 \end{bmatrix} \quad (4)$$

In the pooling layer, we adopt the max pooling method. This method adopts the maximum value of two numbers as their representative, as shown in (5).

$$Seq_4 = Pool(Seq_3) = \begin{bmatrix} 0.67 \\ 0.38 \\ 0.65 \\ 0.27 \end{bmatrix} \quad (5)$$

The Bi-LSTM layer computes $\boldsymbol{Hid} = (\boldsymbol{h_1},\cdots,\boldsymbol{h_t})$ and $\boldsymbol{Out} = (\boldsymbol{o_1},\cdots,\boldsymbol{o_t})$ for the input $\boldsymbol{Seq_4} = (\boldsymbol{s_1},\cdots,\boldsymbol{s_t})$, iterating the formulas below from $\boldsymbol{t} = 1$ to $\boldsymbol{T}$, as shown in (6) and (7). (see Fig 7 for details).

$$h_t = Act(W_{sh}h_t + W_{hh}h_{t-1} + Bia_h) \quad (6)$$

$$o_t = W_{ho}h_t + Bia_o \quad (7)$$

$\boldsymbol{W_{sh}}$ is the weight matrix between the input and intermediate layer, $\boldsymbol{Bia_h}$ is the bias vector for intermediate layer vectors, and ***Act*** is a nonlinear activation function. Finally, for a given sequence ***Seq***, we use the function ***F(Seq)*** to calculate its score to determine whether it is a DNA-binding protein, as shown in (8).

$$F(Seq) = Bi - LSTM(CNN(Embedding(Encoding(Seq)))) \quad (8)$$

We implemented the method on the Keras platform [47]. The laboratory protocols for this study are available at (http://dx.doi.org/10.17504/protocols.io.2rdgd26) and that site contains both the tools and the steps required for the experiment. All the source code and data used in this study are available from the Figshare server at (https://doi.org/10.6084/m9.figshare.8131244).

Results and discussion

Program architecture

Based on the method mentioned above, we developed our program on the Keras platform and used its functions to construct our program architecture. (see Fig 8 for details).

Experimental setup and results

We used both a balanced dataset and unbalanced dataset in our experiments. We also used cross-validation methods to train the model. Here, K represents the proportion of training set data to total data in the training model, while 1-K represents the proportion of the validation set data to

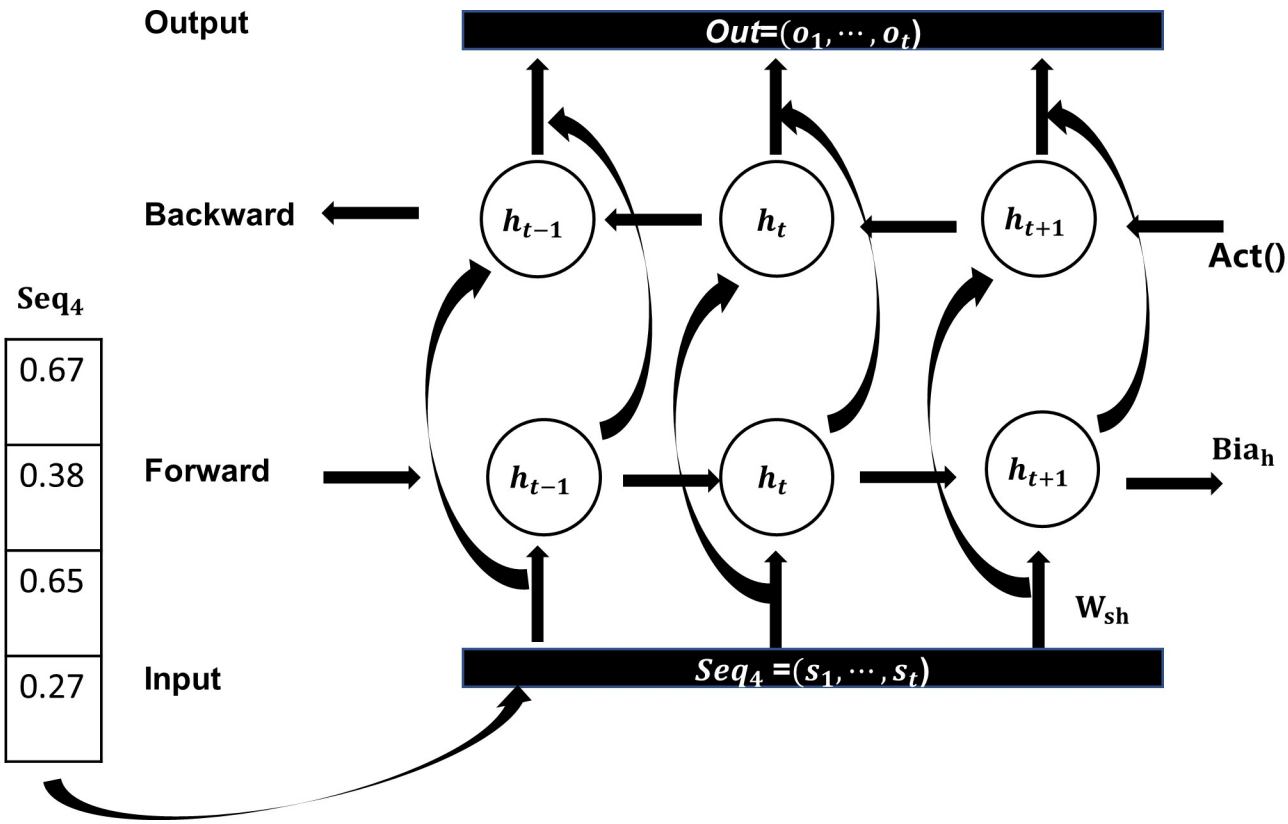

Fig 7. Bi-LSTM layer.

https://doi.org/10.1371/journal.pone.0225317.g007

the total data. For each experiment, we checked the performance of the model on the validation set under three different conditions: k = 0.85, k = 0.8 and k = 0.9 and found that the model fits the samples better and predicts the sequences more accurately when k = 0.85. (see Table 4 for details).

The validation accuracy (Validation-Acc) by the best model of the balanced experiment is 94.6%. The test accuracy (Test-Acc) of the model was 94.1% for the 500-unit test set (500

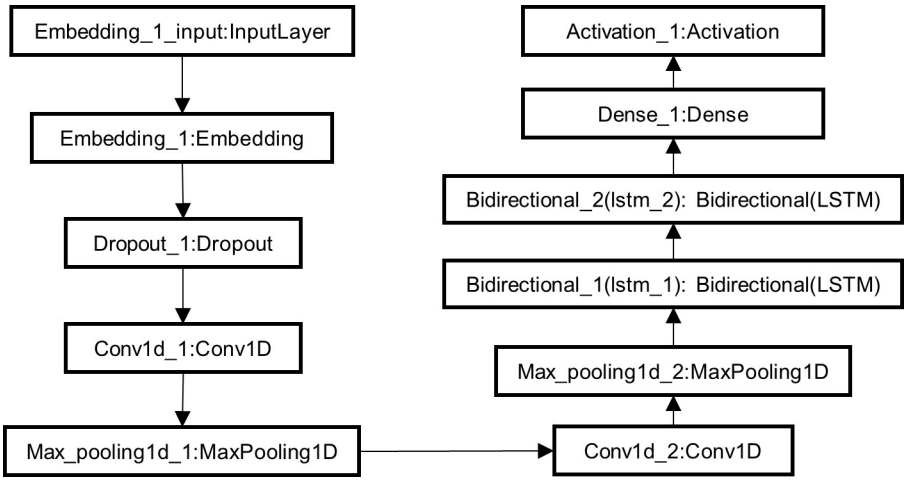

Fig 8. Program architecture.

https://doi.org/10.1371/journal.pone.0225317.g008

positive samples and 500 negative samples) and 94.5% for the 10,000-unit test set (10,000 positive samples and 10,000 negative samples). For the unbalanced experimental group, the Validation-Acc of the best model is 96.5%. This model achieved a Test-Acc of 90.4% on the 500-unit test set and 90.7% on the 10,000-unit test set. (see Table 5 for details).

In the balanced experiment, the model Test-Acc on the test samples is highly similar to its Validation-Acc. This result shows that the model trained by the balanced dataset exhibits almost no overfitting, and its predictions are both sensitive and accurate. On the unbalanced experiments, the Validation-Acc of the model was higher than that on the balanced experiments, but it performed worse on the test samples. This result indicates that the model trained on the unbalanced data has a slight overfitting phenomenon; thus, its prediction ability is weaker than the model trained on the balanced data.

Comparison of the results of different models

We also compared other models for predicting DNA binding proteins with the model proposed in this paper [7,21, 48,49]. Judging from the results of different experiments, the Validation-Acc of CNN-BiLSTM is 96. 5%—7.8% higher than that of SVM, 9.6% higher than that of DNABP and 3.7% higher than that of CNN-RNN, respectively. (see Table 6 for details).

The results in Table 6 show that the proposed CNN-BiLSTM is more capable of capturing protein sequence features and fitting data than are other models.

When tested on the test samples (*Arabidopsis*) in the literature [40], the Test-Acc of CNN-BiLSTM reached 93%—12% higher than that of SVM, 19% higher than that of DNA Binder and 4% higher than that of CNN-RNN. (see Table 7 for details).

Table 4. Experiment with different training set proportions.

Experimental category	K
Balanced experiment	0.85
	0.8
	0.9
Unbalanced experiment	0.85
	0.8
	0.9

https://doi.org/10.1371/journal.pone.0225317.t004

Table 5. Validation and test results.

Experimental category	Validation-Acc	Test samples	Test-Acc
Balanced experiment	94.6%	1,000	94.1%
		20,000	94.5%
Unbalanced experiment	96.5%	1,000	90.4%
		20,000	90.7%

https://doi.org/10.1371/journal.pone.0225317.t005

Table 6. Validation-Acc of different models.

Model category	Validation-Acc
SVM	88.7%
DNABP	86.9%
CNN-RNN	92.8%
CNN-BiLSTM	96.5%

https://doi.org/10.1371/journal.pone.0225317.t006

Table 7. Test-Acc of different models (*Arabidopsis*).

Model categories	Test-Acc
SVM	81.0%
DNABP	89.6%
CNN-RNN	89.0%
CNN-BiLSTM	93.0%

https://doi.org/10.1371/journal.pone.0225317.t007

In addition, in order to further verify our model, we used independent test samples in literatures [14,16,17,19] to test our model. Four methods are used to estimate the performance of our method, including Accuracy, Recall, Specificity, and MCC (Mathew's correlation coefficient). Their expressions are listed below:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{9}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{10}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{11}$$

$$\text{MCC} = \frac{TP * TN - FP * FN}{\sqrt{(TP + FN) * (TN + FP) * (TP + FN) * (TN) * FN}} \tag{12}$$

Where TP, TN, FP, FN indicates the number of true positive, true negative, false positive and false negative respectively.

In Table 8, our model is compared with the models in the above literatures including: iDNA-Prot [13] Compression technology on PSSM [14], DPP-PseAAC [16], iDNAProt-ES [17].

As shown in Table 7 and Table 8, the performance of CNN-BiLSTM on the independent test samples is better than the performances of other models, which indicates that CNN-BiLSTM is more stable and more trustworthy in practical application.

Comparison of characteristics of different models

To reveal the differences between the different models in amino acid sequence processing, we compared the CNN-BiLSTM model with the SVM and CNN-RNN models. According to its length, a sequence is divided into the three parts, N, Middle and C, in the method (SVM) proposed in: [15].Then, the sequence features of the three parts are abstracted, and the DNA-binding protein prediction is achieved by the SVM package. In [7], an LSTM was used to process the sequences. During this process, the sequences are scanned unidirectionally. In contrast,

Table 8. The performance of our methods and other existing methods on independent datasets.

Model category	Accuracy	Recall	Specificity	MCC
iDNA-Prot	67.20%	67.7%	66.7%	0.344
Compression technology on PSSM	76.3%	92.5%	60.2%	0.557
DPP-PseAAC	77.4%	83.9%	71.0%	0.553
iDNAProt-ES	80.6%	81.3%	80.0%	0.613
CNN-BiLSTM	81.2%	89.2%	73.1%	0.632

https://doi.org/10.1371/journal.pone.0225317.t008

the process of scanning sequences in the CNN-BiLSTM model is bidirectional, and the output is constructed by synthesizing the contextual features (see Fig 9 for details).

After the comparison, we find that the CNN-BiLSTM collects more sequence information than do the SVM and CNN-RNN.

Comparison of score predicted by the different models

A deep learning model yields a prediction score for each test sample; when the prediction score is between 0 and 0.5, we consider it as a negative sample. In contrast, when the predicted sample score is between 0.5 and 1, we consider it as a positive sample. The score actually represents the probability that a sample is a positive sample, that is, the closer the score is to 1, the greater the probability that it is a positive sample. CNN-BiLSTM relies on the predictive score to determine whether a sequence is a positive sample (see Fig 10 for details).

We trained the CNN-RNN model on the 32,000 samples provided in the literature and obtained a final CNN-RNN model, which does not consider the contextual features of the amino acid sequence. Then, we applied the final CNN-RNN model to the 500-unit test set and the 10,000-unit test set, respectively. It is worth mentioning that these test samples are labeled. We drew the plots of the scores shown in Fig 11 and Fig 12.

In the above diagrams, the horizontal axis expresses the ordinal number of the sequences, and the ordinate expresses the predicted scores of the sequences. A green flag represents a correct sample judgment, and a red sign represents an incorrect prediction.

We then used the same data to train a CNN-BiLSTM model. As with the CNN-RNN results, we also obtained the distribution of the predicted scores of CNN-BiLSTM on the test set as shown in Fig 13 and Fig 14.

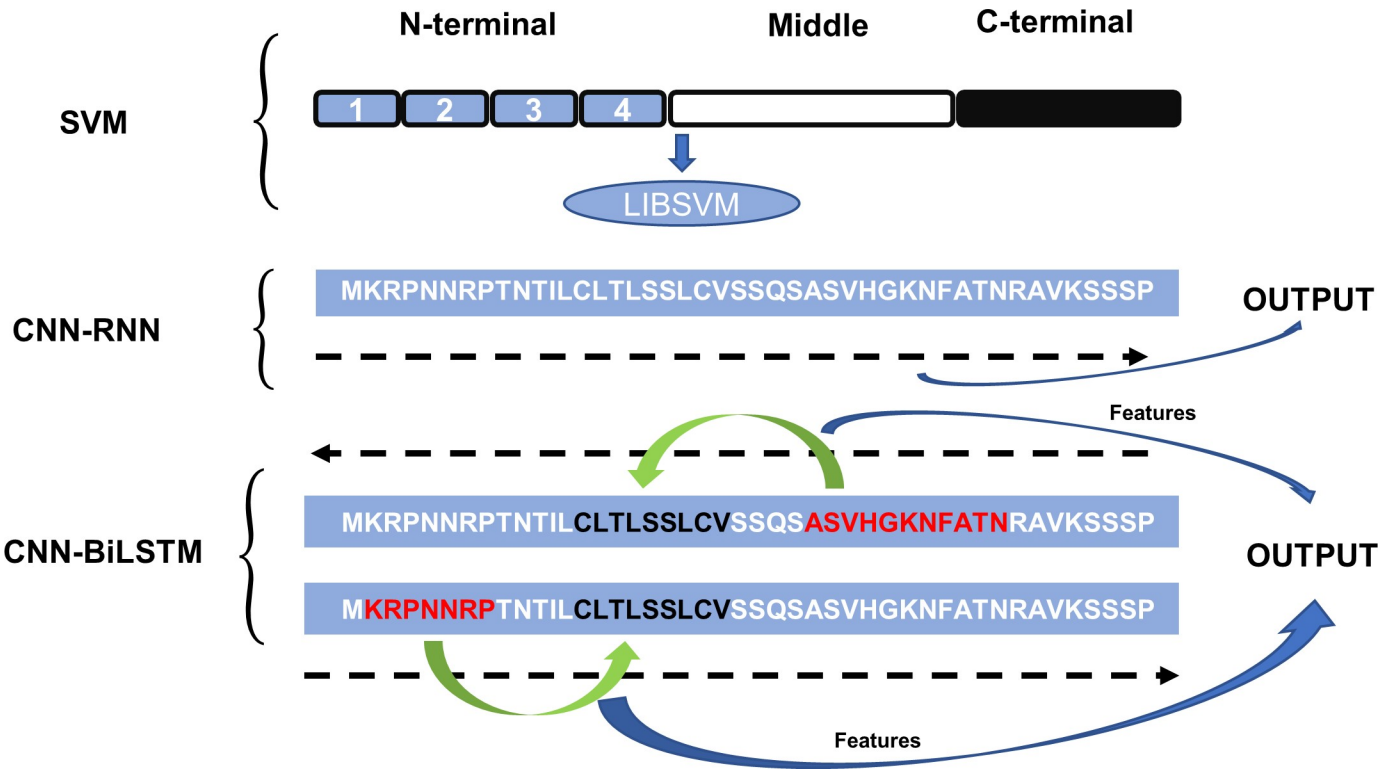

Fig 9. Methods of processing the sequences of different models.

https://doi.org/10.1371/journal.pone.0225317.g009

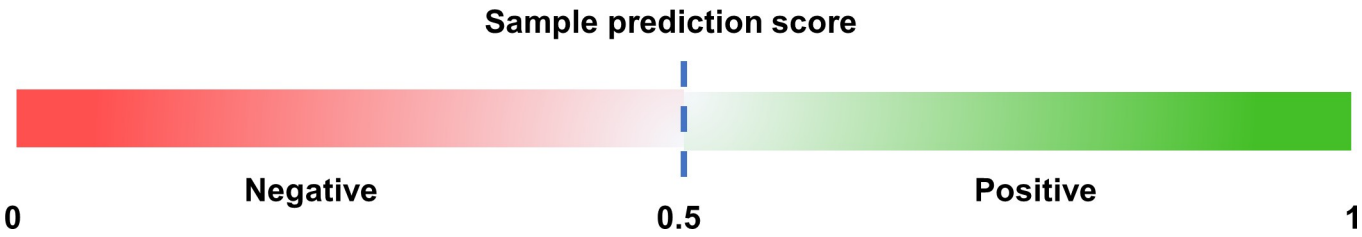

Fig 10. Evaluation criteria for predicted scores.

https://doi.org/10.1371/journal.pone.0225317.g010

We can see that the scores obtained by CNN-BiLSTM are more concentrated in the vicinities of 0 and 1 than are those of the CNN-RNN model, which indicates that the CNN-BiLSTM model has a better predictive score tendency. The prediction score is a probability value. The closer it is to 0 or 1, the more reliable the prediction results are. However, when the prediction score is close to 0.5, the prediction reliability is low. Note that the CNN-BiLSTM model graphs have fewer red marks in the predicted score distribution map than do the CNN-RNN model graphs. Therefore, the CNN-BiLSTM is more trustworthy and robust regarding data fitting and prediction accuracy.

Model training process visualization

After reconstructing the experiments in [7], we obtained the data for the running process of the model (CNN-RNN). We show the variations of the training accuracy (Train-Acc) and Validation-Acc during the model training process in a single chart. Importantly, Train-Acc refers

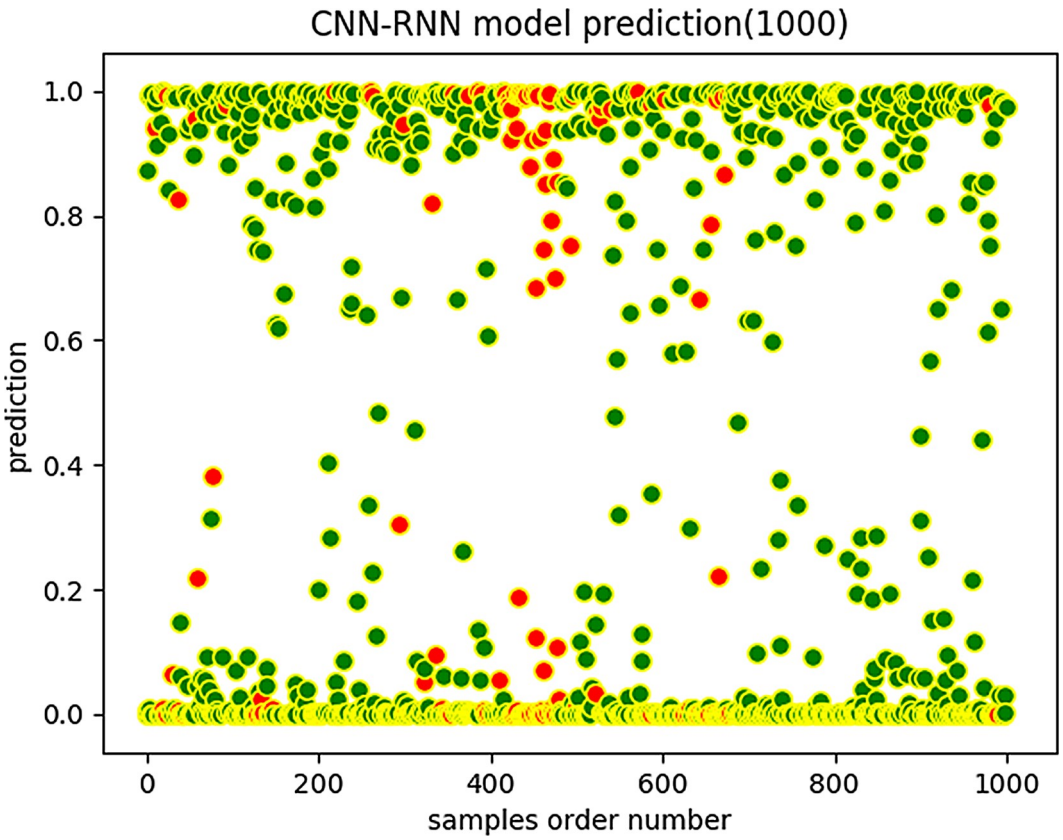

Fig 11. CNN-RNN prediction (1,000).

https://doi.org/10.1371/journal.pone.0225317.g011

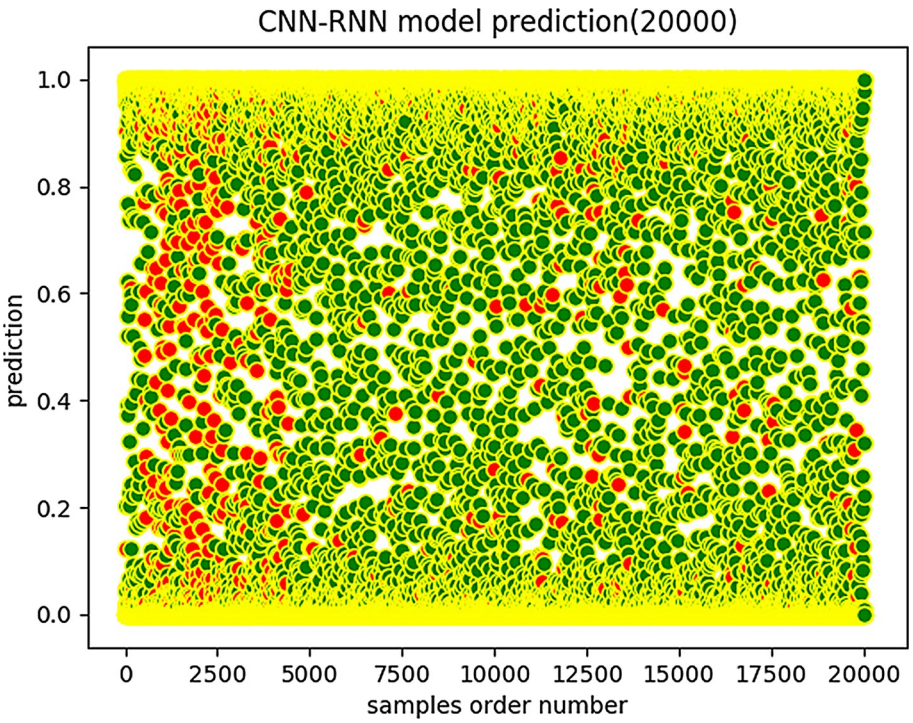

Fig 12. CNN-RNN prediction (20,000).

https://doi.org/10.1371/journal.pone.0225317.g012

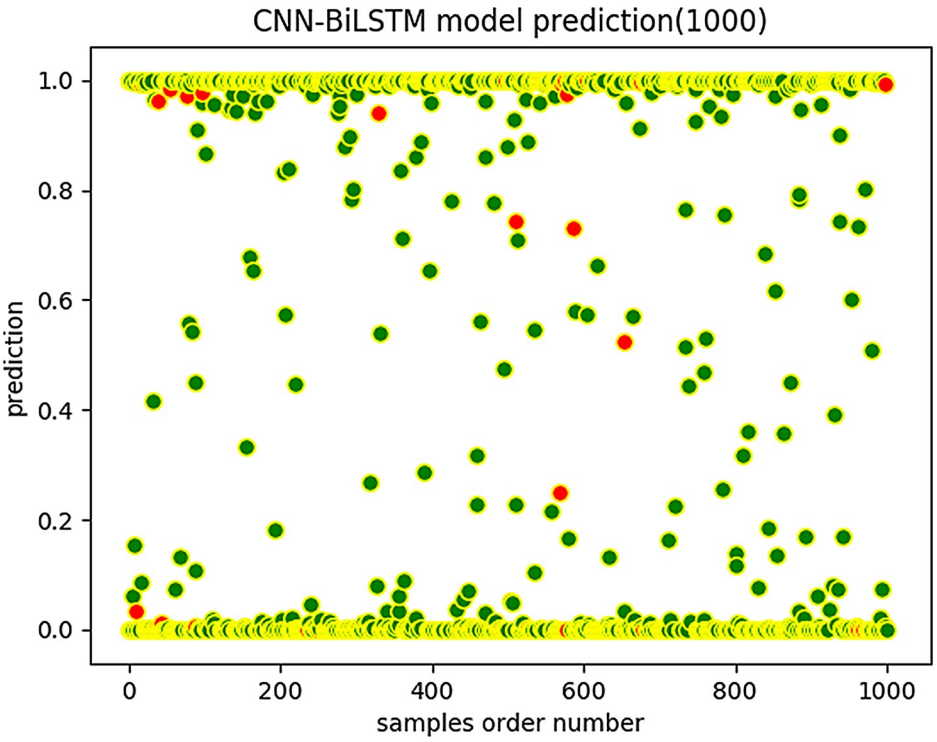

Fig 13. CNN-BiLSTM prediction (1,000).

https://doi.org/10.1371/journal.pone.0225317.g013

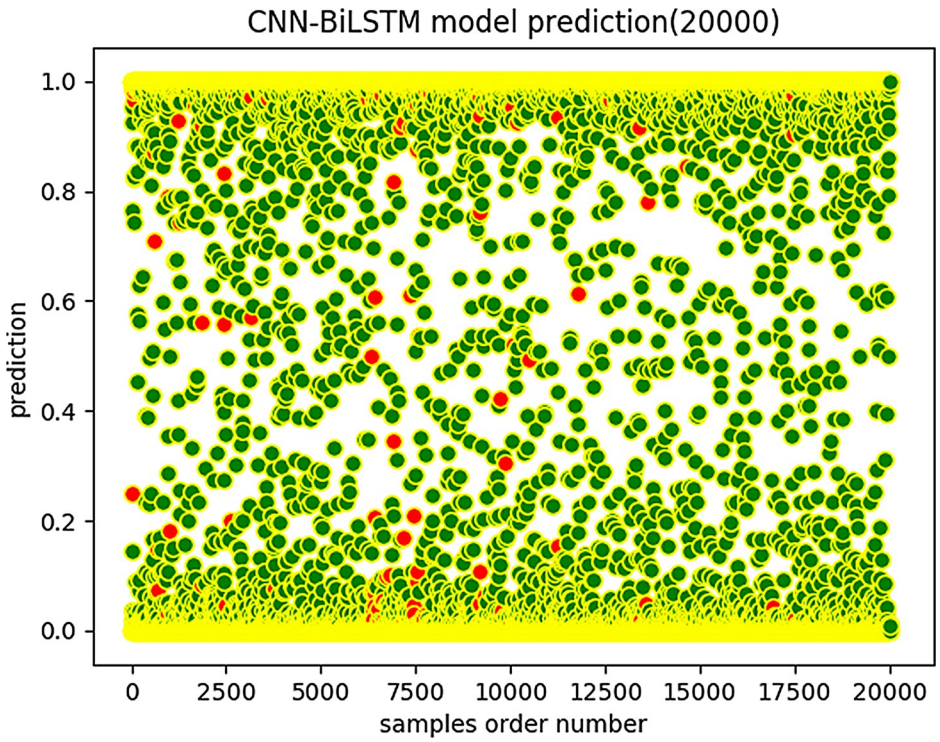

Fig 14. CNN-BiLSTM prediction (20,000).

https://doi.org/10.1371/journal.pone.0225317.g014

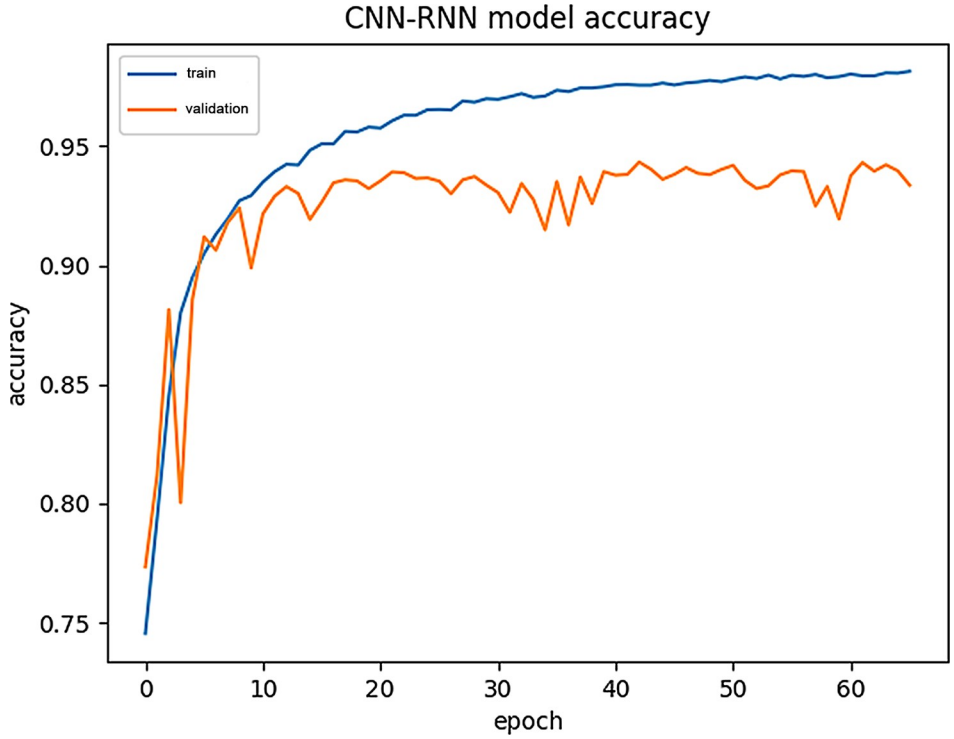

Fig 15. Accuracy variations during CNN-RNN training.

https://doi.org/10.1371/journal.pone.0225317.g015

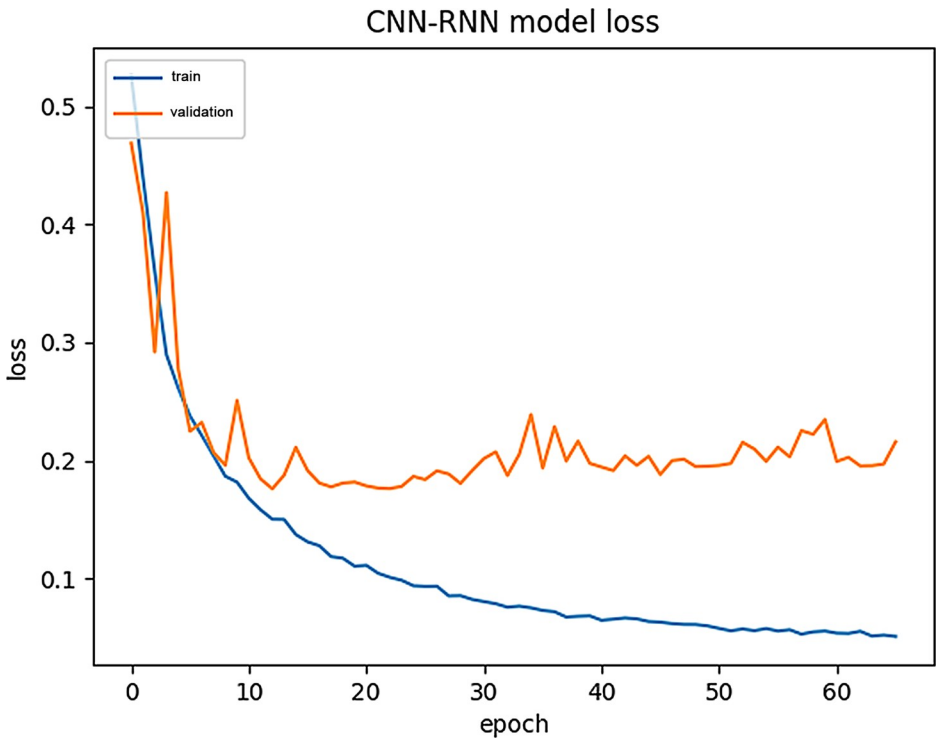

Fig 16. Loss variation during CNN-RNN training.

https://doi.org/10.1371/journal.pone.0225317.g016

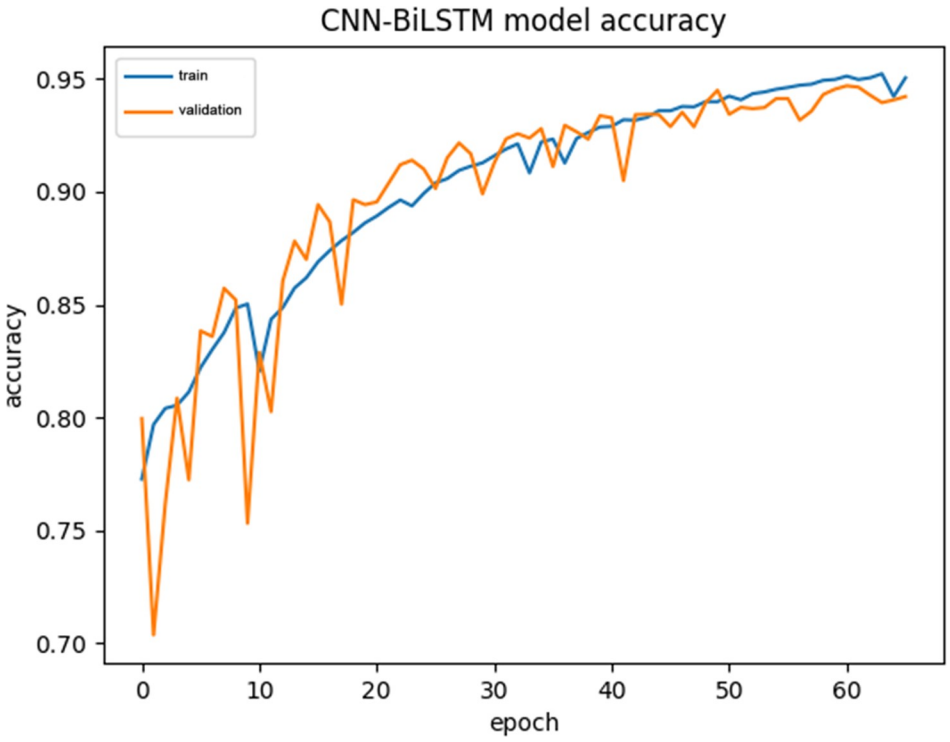

Fig 17. The variation of accuracy in CNN-BiLSTM training.

https://doi.org/10.1371/journal.pone.0225317.g017

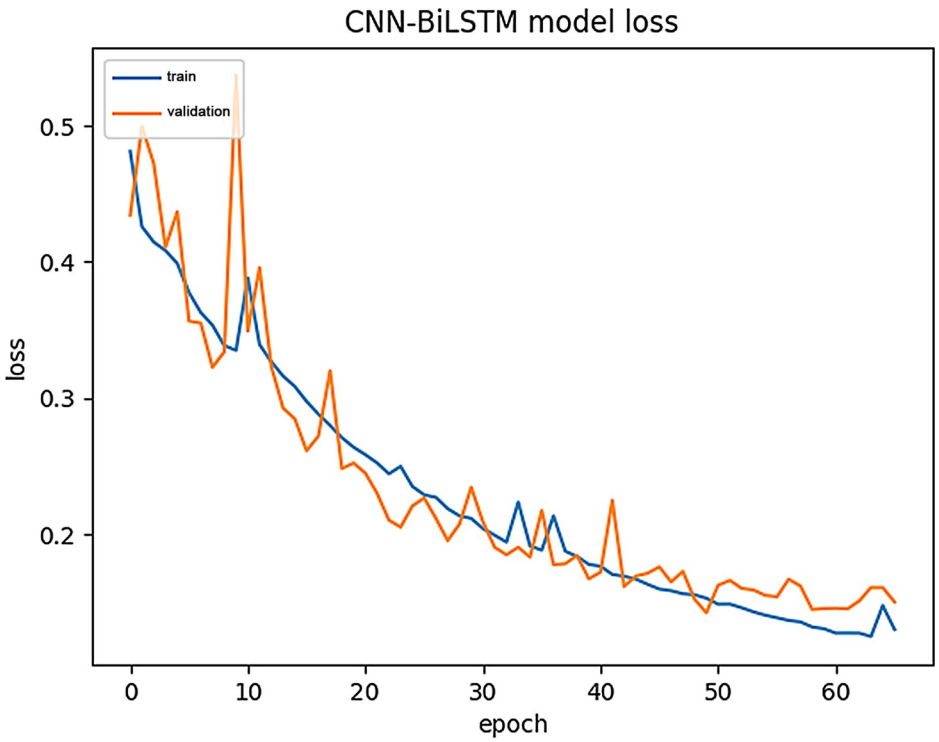

Fig 18. The variation of loss in CNN-BiLSTM training.

https://doi.org/10.1371/journal.pone.0225317.g018

to the prediction precision of the method on the training set, while Validation-Acc refers to the prediction precision of the method on the validation set used to calibrate the parameter weights of the model. (see Fig 15 for details).

We also show the variation in the training loss (Train-Loss) and validation loss (Validation-Loss) during the establishment of the above model. (see Fig 16 for details).

For comparison, we used the same data to visualize the CNN-BiLSTM training process. (see Fig 17 and Fig 18 for details).

From Figs 15–18, we can see that the training curve and the validation curve of the CNN-BiLSTM are closer than those of the CNN-RNN, both for Acc and loss. This indicates that CNN-BiLSTM experiences very little overfitting, but the opposite is true in CNN-RNN. Visibly, the training process of CNN-BiLSTM better reflects the real performance of the data.

Additionally, CNN-RNN converges extremely quickly at the beginning of training, but it reaches its upper limit quickly, and the Validation -Loss value (0.2) is still relatively high at this point. In contrast, CNN-BiLSTM converges more slowly, showing a slow climbing trend initially and ultimately reaching a Validation -Loss value close to 0.1. Because CNN-BiLSTM uses a more complex neural network than the other models, it cannot improve the Train-Acc quickly during the initial stage of training, but it does continuously reduce the los value during continuous operation. Although it requires a longer training period, CNN-BiLSTM can capture amino acid sequence features in more detail than can the CNN-RNN.

Conclusions

The prediction of DNA-binding proteins has been a focus of some computational biologists, and many methods have been proposed successfully. In this paper, we proposed a new deep

learning model to distinguish DNA-binding proteins. We combined a CNN and a Bi-LSTM to explore the potential relationships between amino acids that could be used to detect the functional domain of the protein sequence.

Compared with three previous models (SVM, DNABP and CNN-RNN), CNN-BiLSTM achieves a more advanced performance regarding both prediction accuracy and data fitting. In the test for independent samples, the tendency of the predicted scores obtained by CNN-BiLSTM is better than that of CNN-RNN. The use of deep learning methods to discriminate DNA-binding proteins will become more popular. In addition, the method proposed in this paper may have many potential applications elsewhere, such as in predicting plant hemoglobin.

Acknowledgments

We thank Zhizhou Liao for suggestions of the manuscript.

Author Contributions

Conceptualization: Siquan Hu.

Data curation: Ruixiong Ma.

Formal analysis: Ruixiong Ma.

Funding acquisition: Siquan Hu.

Investigation: Haiou Wang.

Methodology: Ruixiong Ma.

Project administration: Haiou Wang.

Resources: Haiou Wang.

Software: Ruixiong Ma.

Supervision: Siquan Hu.

Validation: Siquan Hu.

Visualization: Haiou Wang.

Writing – original draft: Ruixiong Ma.

Writing – review & editing: Siquan Hu.

References

1. Kumar M, Gromiha MM, Raghava GP. Identification of DNA-binding proteins using support vector machines and evolutionary profiles. BMC bioinformatics. 2007 Dec; 8(1):463.
2. Luscombe NM, Austin SE, Berman HM, Thornton JM. An overview of the structures of protein-DNA complexes. Genome biology. 2000 Feb; 1(1): reviews 001–1.
3. Stawiski EW, Gregoret LM, Mandel-Gutfreund Y. Annotating nucleic acid-binding function based on protein structure. Journal of molecular biology. 2003 Feb 28; 326(4):1065–79. https://doi.org/10.1016/s0022-2836(03)00031-7 PMID: 12589754
4. 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 Jan 22; 20 (4):477–86. https://doi.org/10.1093/bioinformatics/btg432 PMID: 14990443
5. Bowen B, Steinberg J, Laemmli U, Weintraub H. The detection of DNA-binding proteins by protein blotting. Nucleic Acids Research. 1980 Jan 11; 8(1):1–20. https://doi.org/10.1093/nar/8.1.1 PMID: 6243775

6. Hugh P, Mario A, Susan Jones, Janet M Thornton. Identifying DNA-binding proteins using structural motifs and the electrostatic potential. Nucleic Acids Research, 2004, 32(16), 4732–4741. https://doi.org/10.1093/nar/gkh803 PMID: 15356290
7. Qu YH, Yu H, Gong XJ, Xu JH, Lee HS. On the prediction of DNA-binding proteins only from primary sequences: A deep learning approach. PloS one. 2017 Dec 29; 12(12): e0188129. https://doi.org/10.1371/journal.pone.0188129 PMID: 29287069
8. 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 Jan 24; 9(1): e86703. https://doi.org/10.1371/journal.pone.0086703 PMID: 24475169
9. Brown JB, Akutsu T. Identification of novel DNA repair proteins via primary sequence, secondary structure, and homology. BMC bioinformatics. 2009 Dec; 10(1):25.
10. Yan C, Terribilini M, Wu F, Jernigan RL, Dobbs D, Honavar V. Predicting DNA-binding sites of proteins from amino acid sequence. BMC bioinformatics. 2006 Dec; 7(1):262.
11. Caruana R, Niculescu-Mizil A. An empirical comparison of supervised learning algorithms. InProceedings of the 23rd international conference on Machine learning 2006 Jun 25 (pp. 161–168). ACM.
12. Cai YD, Lin SL. Support vector machines for predicting rRNA-, RNA-, and DNA-binding proteins from amino acid sequence. Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics. 2003 May 30; 1648(1–2):127–33.
13. Lin WZ, Fang JA, Xiao X, Chou KC. iDNA-Prot: identification of DNA binding proteins using random forest with grey model. PloS one. 2011 Sep 15; 6(9): e24756. https://doi.org/10.1371/journal.pone.0024756 PMID: 21935457
14. Wang Y, Ding Y, Guo F, Wei L, Tang J. Improved detection of DNA-binding proteins via compression technology on PSSM information[J]. PloS one, 2017, 12(9): e0185587. https://doi.org/10.1371/journal.pone.0185587 PMID: 28961273
15. Zou C, Gong J, Li H. An improved sequence-based prediction protocol for DNA-binding proteins using SVM and comprehensive feature analysis. BMC bioinformatics. 2013 Dec; 14(1):90.
16. Rahman M S, Shatabda S, Saha S, Kaykobad M, Rahman M S. DPP-PseAAC: a DNA-binding protein prediction model using Chou's general PseAAC[J]. Journal of theoretical biology, 2018, 452: 22–34. https://doi.org/10.1016/j.jtbi.2018.05.006 PMID: 29753757
17. Chowdhury S Y, Shatabda S, Dehzangi A. iDNAprot-es: Identification of DNA-binding proteins using evolutionary and structural features[J]. Scientific reports, 2017, 7(1): 14938. https://doi.org/10.1038/s41598-017-14945-1 PMID: 29097781
18. Liu X. J, Gong X. J, Yu H, Xu J. H. A Model Stacking Framework for Identifying DNA Binding Proteins by Orchestrating Multi-View Features and Classifiers[J]. Genes, 2018, 9(8): 394.
19. Adilina S, Farid D M, Shatabda S. Effective DNA binding protein prediction by using key features via Chou's general PseAAC[J]. Journal of theoretical biology, 2019, 460: 64–78. https://doi.org/10.1016/j.jtbi.2018.10.027 PMID: 30316822
20. Liu B, Xu J, Lan X, Xu R, Zhou J, Wang X, 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 Sep 3; 9(9): e106691. https://doi.org/10.1371/journal.pone.0106691 PMID: 25184541
21. Ma X, Guo J, Sun X. DNABP: Identification of DNA-binding proteins based on feature selection using a random forest and predicting binding residues. PloS one. 2016 Dec 1; 11(12): e0167345. https://doi.org/10.1371/journal.pone.0167345 PMID: 27907159
22. Bhardwaj N, Langlois RE, Zhao G, Lu H. Kernel-based machine learning protocol for predicting DNA-binding proteins. Nucleic Acids Research. 2005 Jan 1; 33(20):6486–93. https://doi.org/10.1093/nar/gki949 PMID: 16284202
23. Yu X, Cao J, Cai Y, Shi T, Li Y. Predicting rRNA-, RNA-, and DNA-binding proteins from primary structure with support vector machines. Journal of Theoretical Biology. 2006 May 21; 240(2):175–84. https://doi.org/10.1016/j.jtbi.2005.09.018 PMID: 16274699
24. Qiu J, Wu Q, Ding G, Xu Y, Feng S. A survey of machine learning for big data processing. EURASIP Journal on Advances in Signal Processing. 2016 Dec 1; 2016(1):67.
25. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. InAdvances in neural information processing systems 2012 (pp. 1097–1105).
26. Graves A, Mohamed AR, Hinton G. Speech recognition with deep recurrent neural networks. InAcoustics, speech and signal processing (icassp), 2013 ieee international conference on 2013 May 26 (pp. 6645–6649). IEEE.
27. Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. InAdvances in neural information processing systems 2014 (pp. 3104–3112).

28. Alipanahi B, Delong A, Weirauch MT, Frey BJ. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. Nature biotechnology. 2015 Aug; 33(8):831. https://doi.org/10.1038/nbt.3300 PMID: 26213851

29. Zeng H, Edwards MD, Liu G, Gifford DK. Convolutional neural network architectures for predicting DNA–protein binding. Bioinformatics. 2016 Jun 11; 32(12): i121–7. https://doi.org/10.1093/bioinformatics/btw255 PMID: 27307608

30. Zhang Qinhu, Zhu Lin, Bao Wenzheng, Huang De-shuang. Weakly-Supervised Convolutional Neural Network Architecture for Predicting Protein-DNA Binding[J]. IEEE/ACM transactions on computational biology and bioinformatics, 2018.

31. Melamud O, Goldberger J, Dagan I. context2vec: Learning generic context embedding with bidirectional lstm. InProceedings of the 20th SIGNLL Conference on Computational Natural Language Learning 2016 (pp. 51–61).

32. Yaseen A, Li Y. Context-based features enhance protein secondary structure prediction accuracy. Journal of chemical information and modeling. 2014 Mar 12; 54(3):992–1002. https://doi.org/10.1021/ci400647u PMID: 24571803

33. Figliuzzi M, Jacquier H, Schug A, Tenaillon O, Weigt M. Coevolutionary landscape inference and the context-dependence of mutations in beta-lactamase TEM-1. Molecular biology and evolution. 2015 Oct 6; 33(1):268–80. https://doi.org/10.1093/molbev/msv211 PMID: 26446903

34. Garnier J, Gibrat JF, Robson B. [32] GOR method for predicting protein secondary structure from amino acid sequence. InMethods in enzymology 1996 Jan 1 (Vol. 266, pp. 540–553). Academic Press.

35. Starosta AL, Lassak J, Peil L, Atkinson GC, Virumäe K, Tenson T, et al. Translational stalling at polyproline stretches is modulated by the sequence context upstream of the stall site. Nucleic acids research. 2014 Aug 20; 42(16):10711–9. https://doi.org/10.1093/nar/gku768 PMID: 25143529

36. Pennington J, Socher R, Manning C. Glove: Global vectors for word representation. InProceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) 2014 (pp. 1532–1543).

37. Wang P, Qian Y, Soong FK, He L, Zhao H. A unified tagging solution: Bidirectional LSTM recurrent neural network with word embedding. arXiv preprint arXiv:1511.00215. 2015 Nov 1.

38. Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991. 2015 Aug 9.

39. Pichler K, Warner K, Magrane M, UniProt Consortium. SPIN: Submitting Sequences Determined at Protein Level to UniProt Curr. Protoc. Bioinformatics 62(1):e52 (2018). https://doi.org/10.1002/cpbi.52 PMID: 29927080

40. Motion GB, Howden AJ, Huitema E, Jones S. DNA-binding protein prediction using plant specific support vector machines: validation and application of a new genome annotation tool. Nucleic acids research. 2015 Aug 24; 43(22):e158. https://doi.org/10.1093/nar/gkv805 PMID: 26304539

41. LeCun Y, Bengio Y, Hinton G. Deep learning. nature. 2015 May; 521(7553):436. https://doi.org/10.1038/nature14539 PMID: 26017442

42. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. InAdvances in neural information processing systems 2012 (pp. 1097–1105).

43. Medsker LR, Jain LC. Recurrent neural networks. Design and Applications. 2001; 5.

44. Hochreiter S, Schmidhuber J. LSTM can solve hard long-time lag problems. InAdvances in neural information processing systems 1997 (pp. 473–479).

45. Zhang S, Zheng D, Hu X, Yang M. Bidirectional long short-term memory networks for relation classification. InProceedings of the 29th Pacific Asia Conference on Language, Information and Computation 2015 (pp. 73–78).

46. Dobzhansky T. Nothing in biology makes sense except in the light of evolution. The american biology teacher. 2013 Feb; 75(2):87–91.

47. Chollet F. Keras: The python deep learning library[J]. Astrophysics Source Code Library, 2018.

48. Guo Y, Yu L, Wen Z, Li M. Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. Nucleic acids research. 2008 Apr 4; 36(9):3025–30. https://doi.org/10.1093/nar/gkn159 PMID: 18390576

49. Shen J, Zhang J, Luo X, Zhu W, Yu K, Chen K, et al. Predicting protein—protein interactions based only on sequences information. Proceedings of the National Academy of Sciences. 2007 Mar 13; 104(11):4337–41.