OXFORD

SAResNet: self-attention residual network for predicting DNA-protein binding

Long-Chen Shen, Yan Liu, Jiangning Song and Dong-Jun Yu

Corresponding authors: Jiangning Song, Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, Monash University, Melbourne, VIC 3800, Australia. Email: jiangning.song@monash.edu; Dong-Jun Yu, School of Computer Science and Engineering, Nanjing University of Science and Technology, China. Email: njyudj@njust.edu.cn

Abstract

Knowledge of the specificity of DNA-protein binding is crucial for understanding the mechanisms of gene expression, regulation and gene therapy. In recent years, deep-learning-based methods for predicting DNA-protein binding from sequence data have achieved significant success. Nevertheless, the current state-of-the-art computational methods have some drawbacks associated with the use of limited datasets with insufficient experimental data. To address this, we propose a novel transfer learning-based method, termed SAResNet, which combines the self-attention mechanism and residual network structure. More specifically, the attention-driven module captures the position information of the sequence, while the residual network structure guarantees that the high-level features of the binding site can be extracted. Meanwhile, the pre-training strategy used by SAResNet improves the learning ability of the network and accelerates the convergence speed of the network during transfer learning. The performance of SAResNet is extensively tested on 690 datasets from the ChIP-seq experiments with an average AUC of 92.0%, which is 4.4% higher than that of the best state-of-the-art method currently available. When tested on smaller datasets, the predictive performance is more clearly improved. Overall, we demonstrate that the superior performance of DNA-protein binding prediction on DNA sequences can be achieved by combining the attention mechanism and residual structure, and a novel pipeline is accordingly developed. The proposed methodology is generally applicable and can be used to address any other sequence classification problems.

Key words: DNA-protein binding; self-attention mechanism; deep residual network; transfer learning; sequence analysis; bioinformatics

Introduction

Transcription factors (TFs) are proteins that bind to DNA sequences and regulate gene expression and play a major role in the regulation of genome function and also have important implications for personalized medicine [1–3]. The transcription factor-binding site (TFBS) is a DNA fragment that a TF binds to, which is typically within the range of 4–30 bp [4, 5]. The transcription factor usually regulates several genes at the same time, and to some extent its binding sites on different genes are conservative, but not identical [6, 7]. Accordingly, accurate prediction of DNA-protein binding is important for understanding the physiological role of transcription factors, characterizing specific functional characteristics of the genome, and elucidating how highly specific sequence expression program is orchestrated in complex organisms [8, 9]. With the development of high-throughput sequencing technology, a variety of experimental methods can identify these binding sites *in vivo*, such as ChIP-seq [10] and SMiLE-seq [11]. However, these methods are expensive and time consuming. In this context, development of fast and accurate computational methods for the identification of DNA-protein binding sites is needed and as such, many machine-learning-based methods have emerged [12, 13].

Early-stage methods such as those developed based on traditional machine learning algorithms mainly focused on identifying DNA-protein binding sites [14]. For example, Wong et al. proposed kmerHMM [15] based on Hidden Markov models (HMMs) and belief propagation to predict DNA-binding sites. Ghandi et al. designed gkm-SVM [16] based on the gap k-mers support vector machine method to identify DNA-binding sites. However, with the increasing and rapid accumulation of DNA sequence data, the performance of traditional machine learning algorithms is not satisfactory.

In recent years, deep learning techniques have achieved remarkable success in computer vision [17–19]. Moreover, they have been successfully applied to solve many bioinformatics and computational biology problems including DNA-protein binding site identification [20, 21]. Alipanahi et al. proposed the first deep-learning-based method DeepBind [22] to identify the sequence specificity of DNA- and RNA-binding protein through a single-layer convolutional neural network. Zeng's work [23] further systematically explored the effects of various structural parameters such as the number of the convolution layers and pooling methods and proved the value of CNN. Luo et al. proposed a novel global pooling method [24] based on the EM algorithm [25] and redesigned the pooling layer based on the network architecture of DeepBind. These CNN-based algorithms have achieved promising performance; however, due to the limits of the convolution, they can only focus on extracting the local information and cannot handle long sequence features well. To address this problem, KEGRU [26] successfully combined the Bidirectional Gated Recurrent Unit (GRU) network framework with the k-mer embedding to identify DNA-protein binding. Researchers have also constructed hybrid models by combining CNNs and RNNs [27] to predict the DNA-binding sites, such as DeepSite [28] and DeepTF [29]. The increased LSTM layers can improve the prediction performance of binding sites by learning the long-distance dependence in sequences. Nevertheless, despite the efficiency and accuracy achieved, the existing methods still have the two following critical deficiencies: First, since 690 ChIP-seq experimental datasets produced by the Encyclopedia of DNA Elements (ENCODE) project [30], each dataset corresponded to a combination of human cells and

specific regulatory factors, and the amount of data included in different datasets varied considerably. Most methods used shallow networks to fit the data because some datasets are not sufficiently large to support the training of deep networks. Second, although these algorithms used shallow networks, their performance on small datasets is still unsatisfactory.

In this study, we propose a novel deep-learning algorithm, termed Self-Attention Residual Network (SAResNet), for improving the prediction of DNA-protein binding sites in DNA sequences. Importantly, SAResNet contains two specific modules, the self-attention mechanism and residual structures. The self-attention module focuses on capturing the position information of the sequence, whereas the residual structure guarantees the extraction of the high-level features of the binding site. At the data preprocessing stage, we combine the training subsets containing 690 ChIP-seq datasets; however, the merged dataset may not be reliable. Because the combined data come from different types of human cells and different transcription factors, the sequence that needs to be predicted is the combination of a certain human cell and a specific transcription factor. To overcome this limitation, we first utilize the dataset to train a global model, and then we employ each small dataset to fine-tune the model. Compared with the suboptimal method, our method achieves a 4.4% performance improvement in terms of the average AUC. The AUC improves on 618 out of the 690 datasets. The online web server of SAResNet is implemented and publicly freely available at http://csbio.njust.edu.cn/bioinf/saresnet. In addition, the source code of SAResNet is available at https://github.com/shenlongchen/saresnet.

Materials and methods

Benchmark datasets

To evaluate the performance of the proposed method, we chose 690 ChIP-seq datasets, which were previously used to evaluate the deep learning architectures in DeepBind [22], DeepSEA [31] and CNN-Zeng [23] as the benchmark datasets. These 690 ChIP-seq datasets included 91 human cell types and 161 specific DNA-binding proteins, some of which were collected under different treatment conditions. For each of the 690 ChIP-seq datasets, Zeng et al. divided it into the training subset (80%) and the corresponding testing subset (20%). All the datasets were downloaded from http://cnn.csail.mit.edu/. Each training subset (testing subset) consists of a positive subset and a corresponding negative subset. The positive subset consists of 101-bp DNA sequences, each of which has at least one transcription factor binding event, while the negative subset was generated by shuffling the positive subset by matching the dinucleotide composition [22]. The 'fasta-dinucleotide-shuffle' package in MEME [32] is used for shuffling. DeepBind [22], CNN-Zeng [23] and Expectation-Luo [24] all use the same method for negative sample generation.

In this study, we combined these training subsets as the global training dataset and testing subsets as the global testing dataset, respectively. To avoid overfitting and improve the generalization ability of the model trained by unbalanced samples, we partitioned the positive and negative samples and used the under-sampling strategy (For the global training set, random sampling was used to keep the balance between the positive and negative samples from the 690 training subsets. For the global testing set, 400 000 positive samples and 400 000 negative samples were obtained by random sampling from the 690 testing subsets) to rebuild the global dataset. Finally, we obtained 4 614 580 training sequences and 800 000 testing
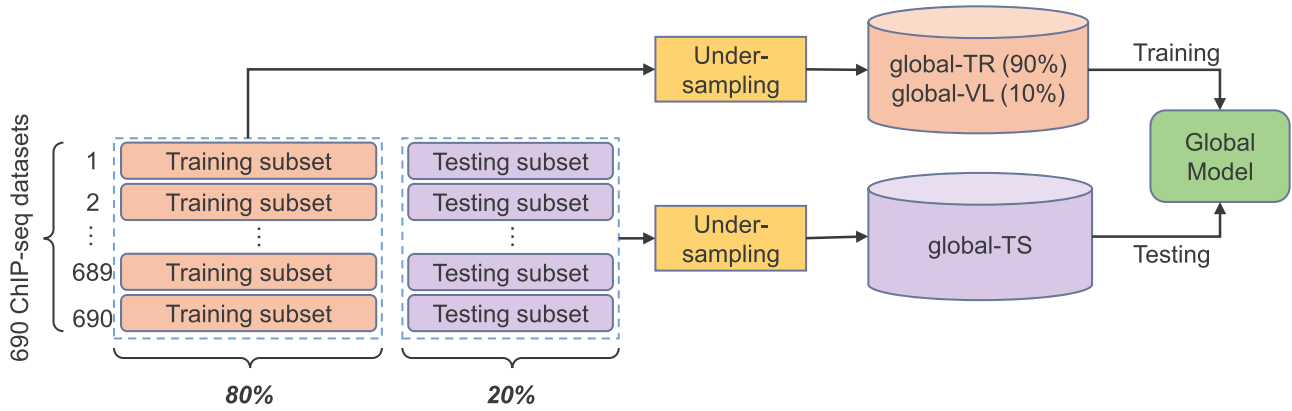
Figure 1. The procedures for generating the global training dataset (global-TR), global validation dataset (global-VL) and global testing dataset (global-TS).

sequences for the global training. These training sequences were randomly split into 90% for training (denoted as global-TR), 10% for validation (denoted as global-VL) and 800 000 testing sequences (denoted as global-TS) for independent test, respectively. In order to examine the impact of training sequence division on our proposed model, we compared the performance of the model based on different proportions of data division in Supplementary Table S1. Finally, considering the scale of the global dataset and the performance of the model, we adopted the 90/10% data splitting for model generation. Figure 1 illustrates the procedures for generating global training dataset (global-TR), global validation dataset (global-VL) and global testing dataset (global-TS). As shown in Figure 1, we divided the global dataset on the basis of the training and testing subsets divided by Zeng et al. Specifically, our global-TR/global-VL (i.e., training) are constructed based on 690 divided training subsets, while global-TS (i.e., testing) is constructed based on 690 divided testing subsets. As a result, this partition could effectively ensure the independence of the training and testing subsets. Moreover, the model trained on such global training subset can learn the common knowledge on the 690 ChIP-seq datasets. Therefore, in order to improve the performance on each of the 690 datasets, it is necessary to transfer this global model to each of the 690 datasets by further fine-tuning.

Performance evaluation metrics

DNA-protein binding site prediction is formulated and solved as a binary classification problem. We used the accuracy, precision, recall and F1 score as the primary performance measures to evaluate the performance of the developed method. These are calculated using the following equations:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

where TP, FN, TN and FP denote the numbers of true positives, false negatives, true negatives and false positives, respectively.

However, all these four performance measurements depend on the prediction cutoff threshold. Therefore, it is critical to find a rational method to compare different predictors. In this study, the area under the receiver operating characteristic (ROC) curve (AUC) [33], which is classification-threshold-invariant and reflects the most comprehensive prediction performance, serves as another important evaluation metric [34].

Feature representation

The input to SAResNet is a DNA sequence represented by a binary one-hot vector of size $L \times 4$. L is the length of the DNA sequence (101 bp in our tests) and '4' corresponds to the number of base pairs (A, C, G, T). In one-hot encoding, each base pair in a sequence is denoted as one of four one-hot vectors [1,0,0,0], [0,1,0,0], [0,0,1,0] and [0,0,0,1], the value of 1 corresponding to the nucleotide at that position and 0 elsewhere [35].

Model architecture and training procedures

If we considered a DNA sequence as a 1-D sequence with four channels (A, C, G, T), then the task of sequence motif discovery can be interpreted as the computer vision task of two-class image classification [36]. Recent works show that deep learning architectures are effective in solving this problem [23, 37]. It has also been shown that the attention mechanism plays an important role in the field of computer vision [38] and NLP [39] and has been widely used in imaging analysis tasks.

Self-attention module

Since the information obtained by the convolution process is often confined to a local neighborhood, it is inefficient to employ convolutional layers solely to model long-range dependencies in the sequences. In this study, we referred to the idea of non-local model [40] and proposed a new self-attention module, which allowed the model to efficiently identify connections between long-distance separated regions. The self-attention module and the residual blocks are illustrated in Figure 2B and C, respectively.

We defined the self-attention module in deep neural networks as follows:

$$y_i = \frac{1}{N} \sum_{\forall j} F\left(x_i, x_j\right) h\left(x_j\right) \tag{5}$$

where x is the representation of the previous hidden layer and y is the output feature of the attention module. i and j are the
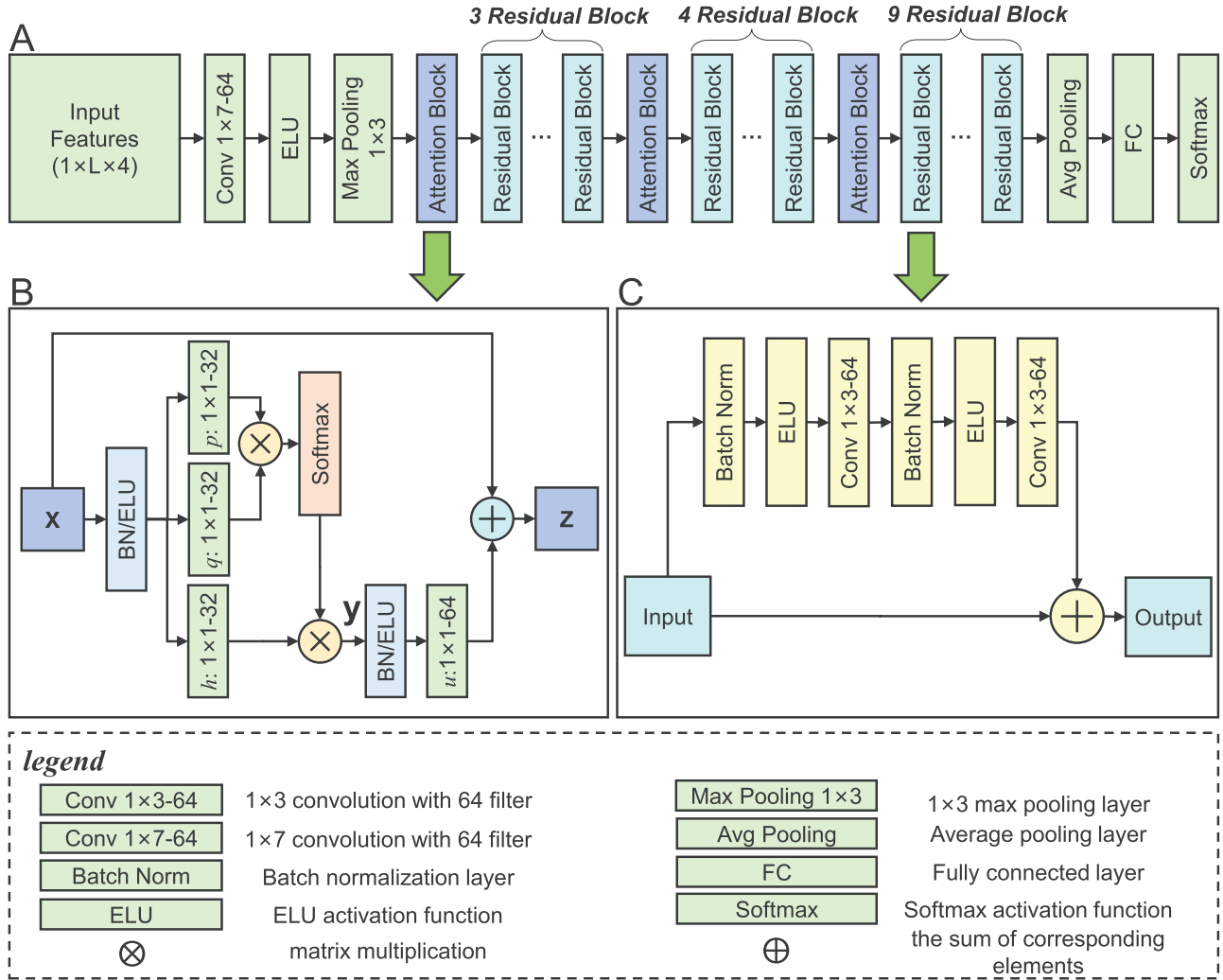
Figure 2. Illustration of SAResNet: (A) Block diagram of the network architecture of SAResNet. (B) Its self-attention module. (C) Its residual block.

indices of the output position of the input signal and the index of enumerating all possible positions, respectively. A pairwise function F computes the attention between i and all j. The function h calculates a representation of the input feature map at the position j, while $h(x_j) = W_p\hat{f}(x_j)$. N is the number of positions in x.

$$F(x_i, x_j) = softmax\left(p(x_i)^T q(x_j)\right) \quad (6)$$

In this module, sequence features transform the feature space through the functions p and q, while $p(x_i) = W_p\hat{f}(x_i)$, $q(x_j) = W_q\hat{f}(x_j)$. The function $\hat{f}$ means batch normalization (BN) [41] and the activation function. Here, the activation function is softmax [42]. In the above formulation, $W_p \in \mathbb{R}^{\overline{C} \times C}$, $W_q \in \mathbb{R}^{\overline{C} \times C}$ and $W_h \in \mathbb{R}^{\overline{C} \times C}$ are the learnable weight matrices. C is the channel number of x, while $\overline{C}$ is the number of channels reduced by the 1×1 convolution kernel. For memory efficiency and model accuracy, 32 filters were selected in all our experiments. In other words, $\overline{C} = 32$. In addition, we further enhanced the dimension of the output of the attention layer through the 1×1 convolution and added it back to the input feature map. Therefore, the final output is

$$z = \theta W_u\hat{f}(y) + x \quad (7)$$

where $W_u \in \mathbb{R}^{C \times \overline{C}}$, θ is a learnable weight, initialized to 0. θ was introduced to allow the network to focus on the local neighborhood, and then gradually learn non-local information. Let the network learn simple content first, and then gradually learn complex information. We verified our hypothesis in the experiment, which showed that this parameter is conducive to improving the generalization of the model.

Through adding layers of the neural network, the expression ability of the model can be enhanced [23]. However, traditional feedforward networks with deep layers are difficult to train because of the instability of gradient renewal. In this regard, the residual network (ResNet) [43] provides a new solution to the problem through 'shortcut connections'. The pre-activated residual unit proposed by He et al. [44] was used in SAResNet. It is shown in Figure 2C that the implementation of the l-th residual block can be expressed as follows:

$$x_{l+1} = x_l + \mathcal{F}\left(\hat{f}(x_l), \mathcal{W}_l\right) \quad (8)$$

where x_l and x_{l+1} denote the input and output of the l-th residual basic block, respectively; $\mathcal{W}_l$ is a set of weights associated with the l-th residual block, while $\mathcal{F}$ stands for the residual function.

Table 1. The hyper-parameters of SAResNet and the corresponding search space

Calibration parameters	Search space	Sampling
Learning rate (global model)	[0.001, 0.0005][a]	Evaluate all
Learning rate (transfer learning)	[0.001, 0.0003][a]	Evaluate all
Kernel numbers	{32, 64, 128}	Uniform
Attention block	{1, 2, 3}	Uniform
Optimizer	Adam	Fixed
Weight initialization	Truncated normal	Fixed
Dropout ratio	{0.5, 0.7}	Uniform

[a]step = 1e-4.

SAResNet pipeline and network architecture

Leveraging the power of the self-attention module and the residual structure, we designed and implemented our SAResNet pipeline. As shown in Figure 2A, the one-hot encoded DNA sequence was used as the input feature ($1 \times L \times 4$). Since the length of the sequence is long, the computational cost would be intensive if the original feature was directly input into the residual block. Hence, we first employed a convolutional layer of 1×7 to transform the input into a signal with a big channel size; i.e., 64. A larger convolution kernel saves as much information as possible from the original input. The ELU activation function follows, and the experiments show that training models using ELU converges faster than RELU [45]. Then, we used the max-pooling layer for further down-sampling. The learned features were then fed into a group of residual basic blocks. Each block contains two convolutional layers, and the size of both kernels was 1×3. The ELU activation function and the Batch Normalization technology [41] were used in each of these blocks. Next, the average pooling layer was used to reduce the amounts of parameters and save computing power. To a certain extent, the application of the average pooling layer can control overfitting and improves the model performance. All convolution padding parameters were set to 'SAME', which means that the size of the feature map remained unchanged after convolution. Finally, two fully connected layers (FC) with dropout [46] and a softmax activation function were used in the final stage of the SAResNet architecture. A dropout rate of 0.7 was utilized in the hidden layers to suppress overfitting. The softmax function converts the output into the probability distribution over two classes.

Model implementation and hyperparameter settings

The model, which was implemented using the Tensorflow framework (v1.12) [47], was trained on a single NVIDIA TITAN X Graphics Card. In the model training process, we utilized the softmax cross-entropy function to calculate the loss and optimized the model by the Adam method [48]. We adjusted the network's hyperparameters by observing the model performance on the validation set (global-VL). The detailed hyperparameter settings are summarized in Table 1. In this study, we tuned the hyper-parameters by optimizing on global-TR and testing on global-TS through grid search (i.e., by enumerating the possible value of each hyper-parameter, a set of which can lead to a relatively high accuracy and ensure the execution efficiency of the model). Then, transfer learning was performed to further train the model on the respective training subsets. The trained models were tested independently with the corresponding testing subsets.

Transfer learning

Transfer learning [49]: Based on a large model, which is trained on a large dataset for a specific task, we utilized limited data to further train the models for other related tasks. In the present study, we used the large dataset global-TR for initial training, and then transfer learning was employed by using 690 ChIP-seq datasets. Irrespective of global training or transfer learning, search for hyper-parameters was conducted on the validation set and the model performance was evaluated on each testing subset. During transfer learning, we trained all the weights of the initial model without freezing any layers, because we found that this worked better than the models that froze part of the layers. We reached the above conclusions by comparing the performance of the transfer learning models under different freezing conditions on 690 ChIP-seq testing sets. The corresponding experimental results are shown in Supplementary Table S2. We can see that the SAResNet-no-freeze method scored higher on various performance indicators than the other models that froze part of the layers. At this stage, we used the same hyper-parameters as those used for training on global-TR, except for the learning rate, which was set at smaller values. The smaller learning rate ensures that the model converges more smoothly.

Results and discussion

Global learning is better than direct learning

We trained our SAResNet model using the curated non-redundant set of DNA sequences (i.e., global-TR, global-VL and global-TS). We used the optimal hyper-parameters obtained by grid search to train the global model on the training set containing the global-TR and global-VL. At the test stage, the performance of SAResNet model was tested on the global-TS through the global model. The architecture of our proposed SAResNet model is shown in Figure 2. The performance of the global model on 690 ChIP-seq testing subsets is given in Table 2. The individual models were generated by training on 690 ChIP-seq datasets, respectively. In detail, the global model achieved a 11.1% improvement in terms of the average AUC for predicting DNA-protein binding over the individual model. As shown in Table 2, the global model has significantly improved the accuracy, precision, recall and F1 score. 'Direct learning' means the process of training individual models. 'Global learning' means the process of training the global model. Figure 3A–C provides the head-to-head AUC score comparison among direct learning, global learning and transfer learning on 690 ChIP-seq testing subsets through the scatter diagrams, where each of 690 testing subsets corresponds to a point whose X coordinate and Y coordinate indicate the AUC scores of the corresponding methods. Figure 3D–F correspond to Figure 3A–C respectively,

Table 2. Performance of the individual model and global model on the 690 ChIP-seq testing sets

Method	Accuracy[a]	Precision[a]	Recall[a]	F1 score[a]	AUC[a]
Individual model	0.728	0.735	0.723	0.723	0.790
Global model	0.794	0.808	0.771	0.786	0.878

[a]All performance evaluation metrics are obtained by averaging the performance on the 690 testing subsets.
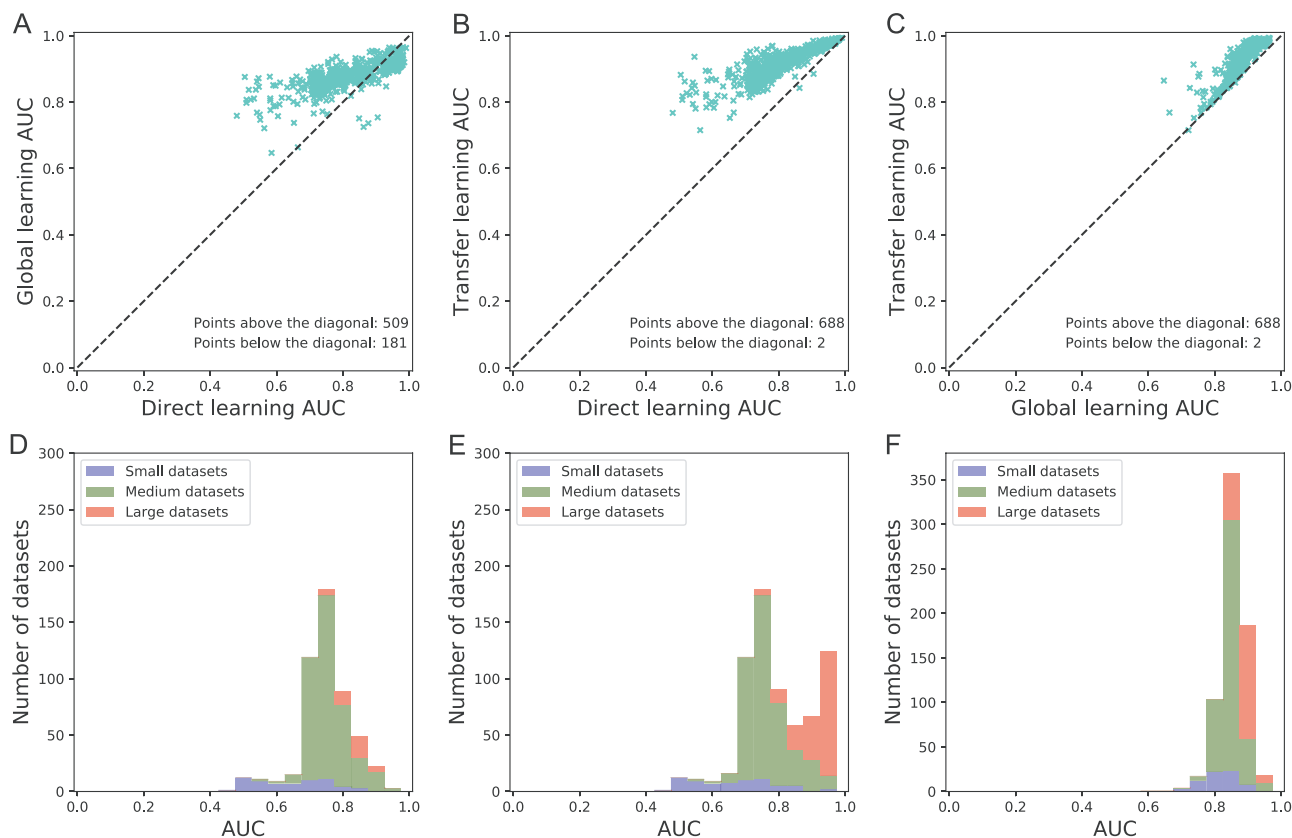
Figure 3. Head-to-head AUC score comparison of direct learning, global learning and transfer learning on 690 ChIP-seq testing subsets. The scatter plot A shows the performance comparison between direct learning and global learning. The scatter plot B shows the performance comparison between direct learning and transfer learning, while the scatter plot C shows the performance comparison between global learning and transfer learning. The stacked histograms D, E and F correspond to the scatter plots A, B and C, respectively. Each of them describes the performance improvement in terms of the AUC score of the corresponding scatter plot's y-axis method compared to the x-axis method with respect to the scale of the datasets. For example, the stacked histogram D corresponds to scatter plot A, where the x-axis in panel D corresponds to the x-axis of panel A, with the x-axis representing the AUC score and the y-axis representing the number of datasets of different scales, respectively. Panel D shows the performance improvement on different scaled datasets represented by the points above the diagonal in panel A. Explanations of Panels E and F are similar to that of panel D.

reflecting the performance differences of direct learning, global learning and transfer learning from the perspective of different scaled datasets. Taking Figure 3A as an example, we can see that 509 points are located above the diagonal line, while the other 181 points are below the diagonal line. This illustrates that global learning performed better than direct learning on the majority of the testing subsets. Figure 3D clearly shows that the performance improvement of global learning compared with direct learning is mainly on small and medium datasets, while the performance improvement on large datasets is small (The definition of small datasets, medium datasets and large datasets refers to the Section 'Comparison of SAResNet with other predictors'.) Figure 4 provides the ROC curve and PR curve of the global learning on the global-VL and global-TS subsets. From Figure 4, we can observe that the classification performance of the model is approximately the same for both the global-VL and global-TS irrespective of the ROC curve and

the PR curve. Such consistent performance in terms of the AUC highlights the robustness of the global model, which achieved an AUC of 0.889 and 0.892 for global-VL and global-TS, respectively.

Transfer learning further improves the prediction performance

The global model obtained from pre-training was transferred to each of the 690 ChIP-seq datasets for further training. The comprehensive performance of the transfer model is given in Table 3. The results indicate the performance of directly applying the global model to predict 690 ChIP-seq datasets, which provides inferior but reasonable performance compared with the model after transfer learning. In 690 testing subsets, the average AUC increased by 4.8% from 0.878 to 0.920. Its F1 score improved over the global model by 7.5% from 0.786 to 0.845.
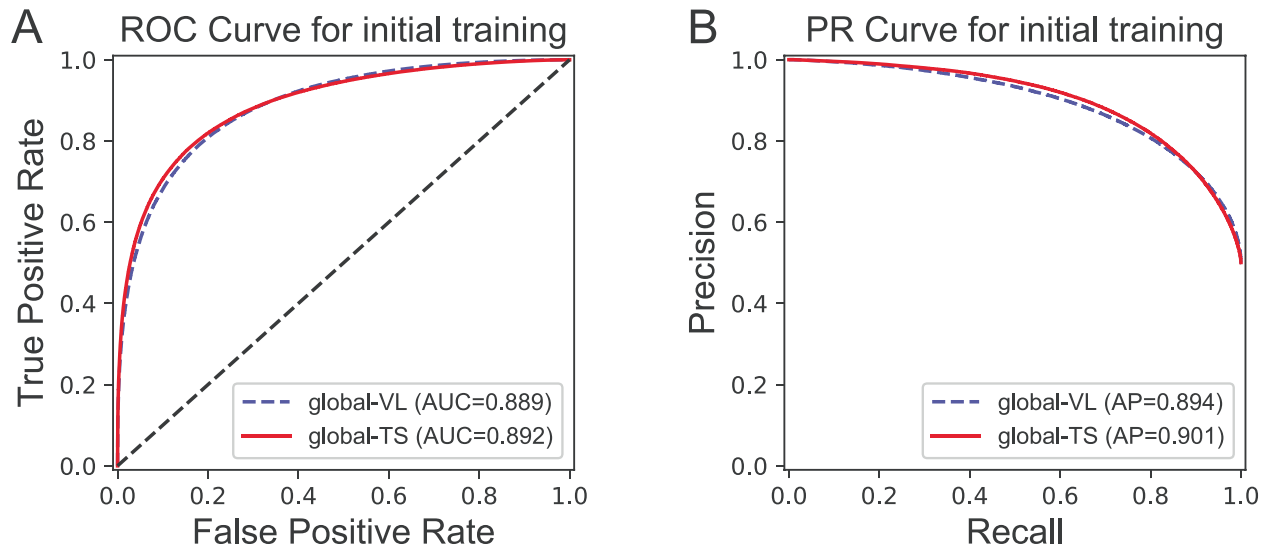
Figure 4. ROC Curve and PR Curve of the global training assessing the performance on global-VL and global-TS.

Table 3. Performance of the global model and transfer model on the 690 ChIP-seq testing sets

Method	Accuracy[a]	Precision[a]	Recall[a]	F1 score[a]	AUC[a]
Global model	0.794	0.808	0.771	0.786	0.878
Transfer model	0.849	0.861	0.831	0.845	0.920

[a] All performance evaluation metrics are obtained by averaging the performance on the 690 testing subsets.

To further examine the effectiveness of transfer learning, we also directly trained the model with the same network architecture and hyper-parameters (i.e., the number of residual blocks, the number of self-attention blocks, the number of filter and the kernel size) on the 690 ChIP-seq datasets. The performance of the resulting model by individual learning is also shown in Table 2. However, this model achieved a much lower performance than that of transfer learning (i.e., 16.5% reduction of the average AUC). These results proved the difficulty of using small datasets to directly train the deep network and also showed that the large dataset could be used to effectively enhance the learning ability of the deep network. As shown in Figure 3B, we also found that 688 points were located above the diagonal line, while only two points were located below the diagonal line. The results indicate that the performance of transfer learning was significantly improved compared to that of direct learning. Figure 3C also exhibits the same phenomenon. Importantly, transfer learning outperformed direct learning and global learning across almost all the testing subsets. Both Figure 3E and F shows that the performance of transfer learning is better than direct learning and global learning on different scaled datasets.

Although the data volume of these datasets was very small, the performance of trained model has been significantly improved because of transfer learning. The pre-training phase of transfer learning is equivalent to giving a satisfactory set of initialization parameters, which enables the network to find a better local optimal solution even with less training data available. Individual learning initialized the model by truncating the normally distributed random numbers. In the case of sufficient data, more iterations are needed to approach the performance of the transfer learning model. If the amount of training data is insufficient, the performance of the model

after direct learning is hard to achieve the performance of the transfer learning model.

Of course, the prerequisite for transfer learning to work is that the pre-training data and the fine-tuning data have a certain similarity. In this study, the relationship between the pre-training data and fine-tuning data is commonality and individuality. More specifically, the pre-training dataset was used to predict whether the DNA sequence can be bound to any transcription factor, and the task of this work is to predict whether the DNA sequence can be bound to a specific transcription factor.

Is the self-attention mechanism effective?

To better understand the effect of the proposed self-attention mechanism, we added the self-attention block in front of the first three residual structures of the deep network and performed the side-by-side comparison with the network without the self-attention mechanism. As shown in Table 4, nearly all the performance metrics of the model after adding the self-attention mechanism have been improved, thus illustrating the effectiveness and reliability of the self-attention mechanism. In the case of different numbers of convolutions in the residual structure (i.e., 32 and 64), comparative experiments were performed respectively, and the distribution of AUC is shown in Figure 5A and B. Irrespective of which network architecture with convolution kernel of 32 or 64, the overall performance of the model with the self-attention mechanism was consistently better than that of the residual network with the same structure. For example, the average AUC of the model increased from 0.912 to 0.920 after adding the attention mechanism in the model with 64 convolution kernels. In Figure 5C and D, the models with the self-attention mechanism performed better

Table 4. Performance of the transfer learning models with different convolution quantities under the two conditions with or without the self-attention mechanism on 690 ChIP-seq testing sets

Method	Accuracy[a]	Precision[a]	Recall[a]	F1 score[a]	AUC[a]
Non-Attention-32	0.825	0.835	0.812	0.822	0.900
Self-Attention-32	0.838	0.850	0.822	0.835	0.911
Non-Attention-64	0.839	0.844	0.832	0.838	0.912
Self-Attention-64	0.849	0.861	0.831	0.845	0.920

[a]All performance evaluation metrics are obtained by averaging the performance on the 690 testing subsets.

than the models without the attention mechanism on 95.1 and 93.5% of the datasets, respectively. These results show that the models trained with the self-attention mechanism achieved a clear performance improvement.

The traditional convolution operation can only obtain local sequence information, whereas the global information can be obtained by superimposing the number of network layers [50]. The proposed residual structure makes the network deeper while ensuring effective learning [43] and achieves excellent results in computer vision tasks. However, due to continuous sampling and the layer-by-layer transfer of information, a large amount of information is lost, and the correlation within the long-distance sequence is also weakened. The self-attention mechanism enhances the attention to the protein binding fragments in the sequence by incorporating the correlation between different positions in the network model. It can well capture the long-distance dependence of the sequence, integrate the spatial information into the network, and then complement with the local information obtained by convolution, so that the network can effectively learn both the spatial location information and local information. The experimental results show that the spatial location information obtained by the self-attention mechanism indeed improved the performance of the model.

Comparison of SAResNet with other predictors

The existing DNA-protein binding prediction pipelines often use transcription factor ChIP-seq from the ENCODE project as the datasets. HOCNN [51], KEGRU [26] and DeepRAM [37] used 214, 125 and 83 ChIP-seq experiments, respectively, from the ENCODE project to evaluate their respective models. To ensure the integrity of the experiments and fairly evaluate the performance of the model, we utilized all 690 ChIP-seq experimental datasets to evaluate the model and compared it with gkm-SVM [52], DeepBind [22], CNN-Zeng [23], DeepTF [29] and Expectation-Luo models [24], all of which also used all the datasets.

We used the gkm-SVM R package (https://cran.r-project.org/web/packages/gkmSVM) [53] with the default parameters for performing the model training and prediction on 690 ChIP-seq datasets. As described by Zeng et al. [23] and Zhou et al. [31], because gkm-SVM cannot use all the training data to calculate the complete kernel matrix, we adopted a method consistent with these two studies: If the number of positive samples in the training subsets was greater than 5000, we randomly selected 5000 positive samples from it, and also selected the same number of negative samples. We obtained the experimental data of CNN-Zeng and DeepBind from http://cnn.csail.mit.edu/. In addition, we obtained the source code of Expectation-Luo from https://github.com/gao-lab/ePooling and trained and tested it on a single NVIDIA TITAN X Graphics Card. The experimental data of DeepTF were provided by its authors. For the convenience of other researchers, the above experimental results can be downloaded from our website.

Figure 6 shows the performance of SAResNet on these datasets in comparison with gkm-SVM, DeepBind, CNN-Zeng, DeepTF and Expectation-Luo. It can be seen that the performance of SAResNet was better than all other methods. Compared with these state-of-the-art methods, SAResNet has achieved a significant performance improvement. Specifically, the median AUC of SAResNet was 0.923, which was 4.4% higher than the suboptimal model (0.884). This indicates that SAResNet performed better than the other models. In terms of the maximum value of AUC, the maximum values of all the six models were higher than 0.990, but the minimum AUC of SAResNet was considerably improved compared with the other models, indicating that our model has a strong generalization ability.

We further analyzed the impact of the amount of training data on the model and then compared the performance of these models on the datasets of different sizes. Since the sizes of 690 ChIP-seq experimental datasets were not consistent, we divided these datasets into three major categories according to the size of the training subset. The specific classification is detailed as follows:

- Small datasets: there were 73 datasets with the size of the training subset less than 3000.
- Medium datasets: there were 429 datasets with the size of the training subset between 3000 and 30 000.
- Large datasets: there were 188 datasets with the size of the training subset greater than 30 000.

From Table 5, it is clear that SAResNet achieved a statistically significant performance improvement in terms of the AUC (student's t-test, $p < 1.9 \times 10^{-11}$) on the three different-scaled datasets. Compared with Expectation-Luo, which is the second-best performer, SAResNet has improved by 4.9, 5.6 and 2.0% on the three types of datasets. In particular, we found that the performance improvement was the most significant on small- and medium-sized datasets. Moreover, to comprehensively understand the performance of the proposed SAResNet method, we further compared its performance with that of Expectation-Luo, CNN-Zeng, DeepBind and gkm-SVM on different-scaled datasets in terms of all five performance metrics (i.e., Accuracy, Precision, Recall, F1 score and AUC) based on bar charts, as shown in Figure 7. As can be seen, SAResNet outperformed the other four methods on different scaled datasets in terms of almost all the five performance metrics. As the precision of gkm-SVM was very high but the recall was low, for fair evaluation, we mainly compared its performance with SAResNet and other methods based on the F1 and AUC indicators. Taken together, the results demonstrate that transfer learning offers great advantages over traditional training methods especially on small-scale datasets.
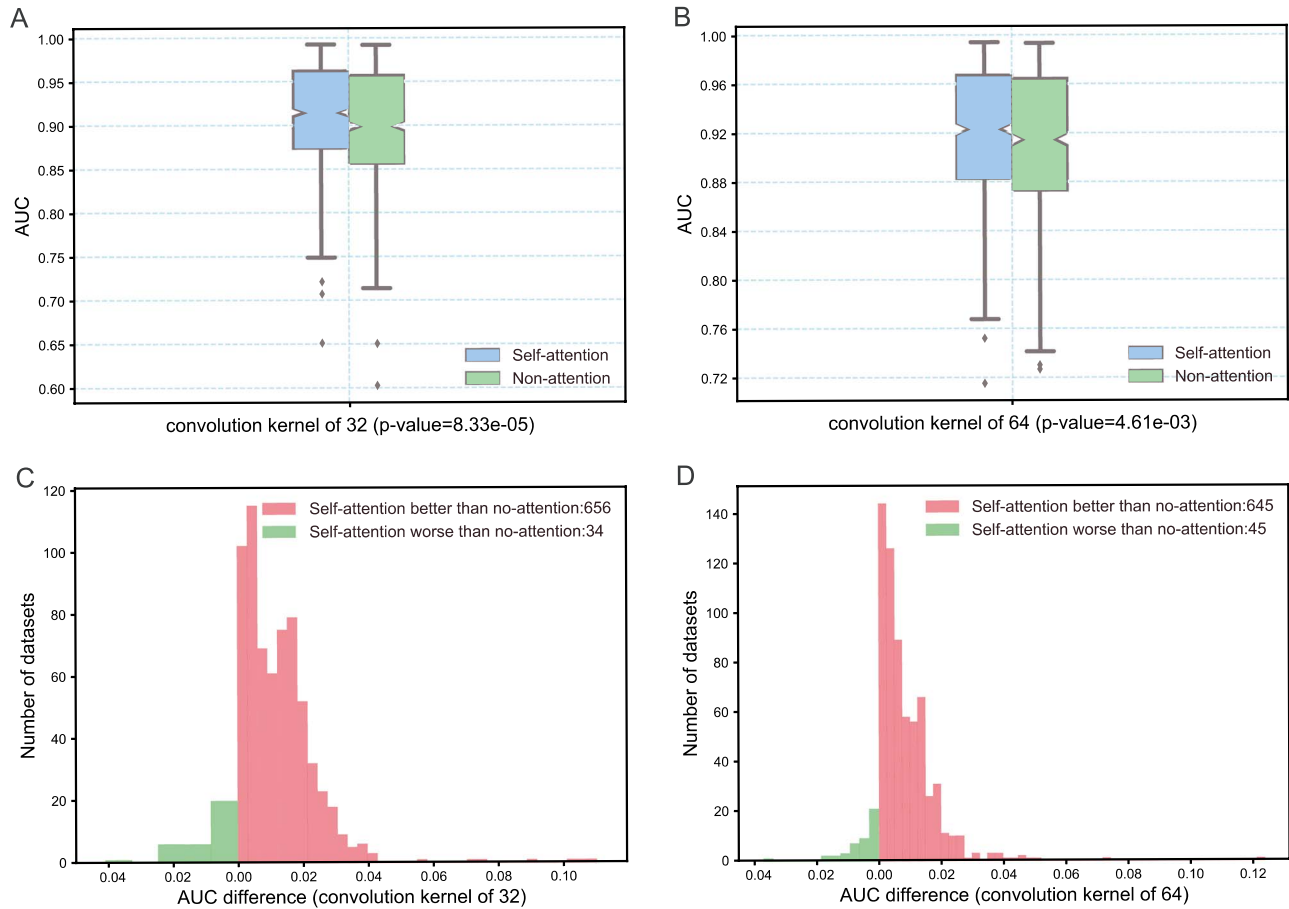
Figure 5. Performance comparison the models trained with and without the self-attention module. Panels A and B, respectively, compare the AUC distribution of the models with different convolution quantities under the two conditions with and without the self-attention mechanism. On each box, the intermediate mark indicates the median, and the top and bottom edges of the box indicate the upper and lower quartiles, respectively. The upper and lower sides indicate the upper and lower limits, while the diamond marks indicate outliers. Panels C and D, respectively, show the performance improvement effect of the model with self-attention mechanism compared with the no-attention model on 690 ChIP-seq datasets through histograms. The x-axis represents the AUC difference between the self-attention and no-attention model. In panel C, the models with the self-attention mechanism outperformed the no-attention models on 656 datasets, but performed worse than the no-attention models on 34 datasets. Similar results are shown in panel D.

Table 5. Performance comparison between SAResNet and the other prediction methods on the datasets with different scales

| Methods	All datasets	Small datasets	Medium datasets	Large datasets	P-value[a]
SAResNet	0.920	0.876	0.907	0.966	-[b]
Expectation-Luo	0.881	0.835	0.859	0.947	1.9×10^{-11}
CNN-Zeng	0.875	0.818	0.850	0.953	6.1×10^{-12}
DeepTF	0.845	0.809	0.818	0.919	9.8×10^{-14}
DeepBind	0.830	0.785	0.809	0.896	2.2×10^{-14}
gkm-SVM	0.818	0.798	0.805	0.856	5.7×10^{-168}

[a]The *P*-values of student's *t*-test for the difference in *AUC* values between SAResNet and the existing DNA-protein binding predictors.
[b]'-' indicates that the corresponding value does not exist.

Conclusions

In this study, we have designed and implemented a novel deep transfer learning approach, SAResNet, to predict DNA-protein binding in DNA sequences. In particular, we combined the self-attention mechanism and the residual structure to develop a deep learning architecture and trained the network model through transfer learning. Benchmarking experiments show that the performance of SAResNet is superior to other state-of-the-art methods on the 690 ChIP-seq datasets. The characteristics of this approach are summarized as follows:

First, we merged 690 ChIP-seq datasets and performed sequence homology removal to avoid any biased model training. The model was then initialized with the generated dataset, and fine-tuned on the respective datasets by transfer learning to predict specific binding sites. This approach has a faster convergence rate than other deep learning prediction methods. Second, we designed a self-attention mechanism to effectively learn the long-range dependencies from the DNA sequence, which can compensate for the global information loss caused by the stacking of residual structures. Furthermore, we demonstrate that
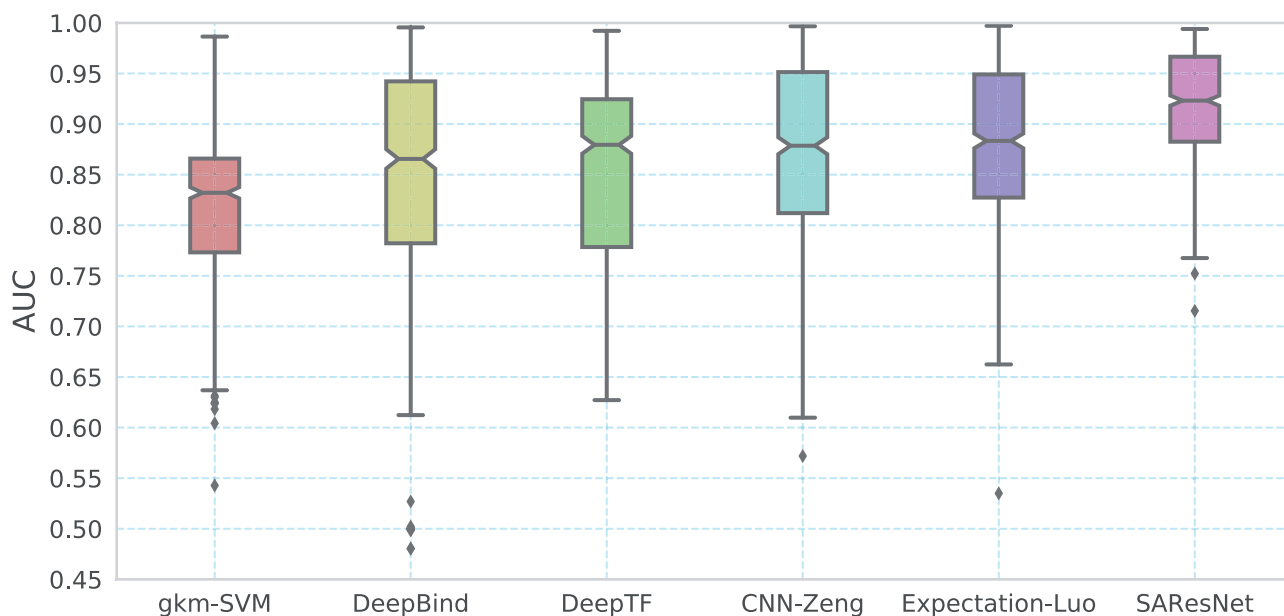
Figure 6. The distribution of AUCs across 690 experimental datasets for DNA-protein binding prediction.
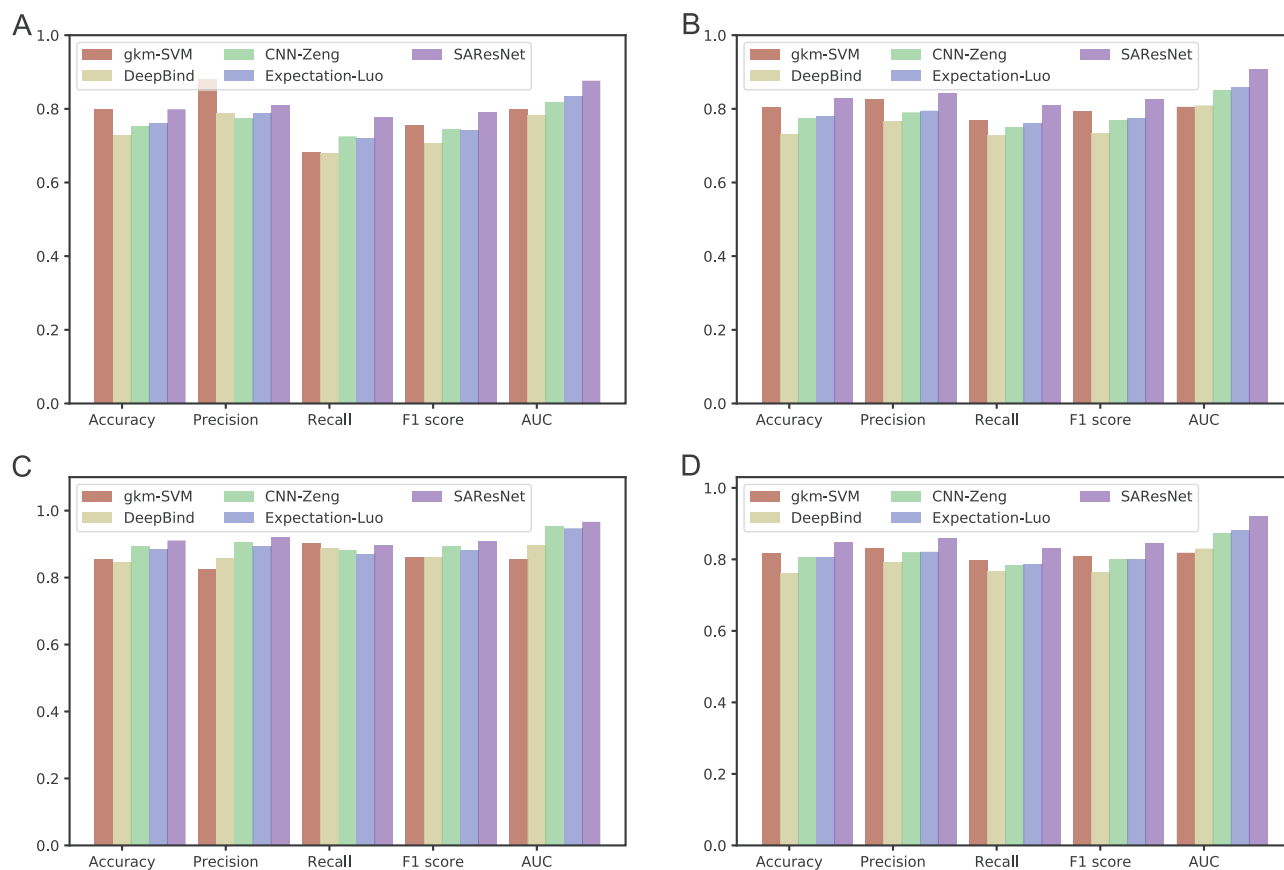
Figure 7. Performance evaluation on small, medium, large and all datasets. Panels A, B and C display the performance comparison of gkm-SVM, DeepBind, CNN-Zeng, Expectation-Luo and SAResNet on the small, medium and large datasets. Panel D shows the performance result of gkm-SVM, DeepBind, CNN-Zeng, Expectation-Luo and SAResNet on all 690 ChIP-seq datasets. Note that the values on the y-axis represent the performance metrics scores.

the attention mechanism and transfer learning can effectively improve the prediction performance of the trained models.

Although SAResNet has achieved a good performance on predicting DNA-protein binding, it has roughly determined the

hyper-parameter of SAResNet in limited experiments due to the limitation of time and computing resources. Therefore, there is still a further room for SAResNet to improve. In addition, there are some other important aspects that may be improved,

which include: First, SAResNet is currently only used to predict fixed-length DNA sequences. In this regard, we could consider adding bidirectional LSTM to process input sequences of different lengths. Second, SAResNet is a network architecture designed to predict DNA-protein binding. In the future, it can be generally applied to address other prediction problems in bioinformatics and computational biology such as predicting DNA-protein binding sites from protein sequences [53–55] and RNA-binding sites [56, 57]. Finally, we hope that SAResNet will be exploited as a useful tool to improve our further understanding of deep learning models and contribute to the elucidation of gene regulation mechanisms at the genomic level.

Key Points

- This study proposes a novel transfer learning-based deep learning pipeline, which combines the self-attention mechanism and transfer learning, to improve the prediction of DNA-protein binding from DNA sequences.
- The pre-training strategy employed in the pipeline improves the learning ability of the network and accelerates the convergence speed of the network during transfer learning.
- The self-attention mechanism can enable the effective learning of the long-range dependencies from the DNA sequence, thereby compensating for the global information loss caused by the stacking of residual structures.
- Based on the proposed pipeline, a novel DNA-protein binding predictor, termed SAResNet, is implemented. Benchmarking results demonstrates the superior performance of SAResNet compared to other existing state-of-the-art predictors on the 690 ChIP-seq datasets.
- A web server (http://csbio.njust.edu.cn/bioinf/saresnet/) has been made publicly available for the prediction of DNA-protein binding, providing a faster tool than other deep learning-based methods for DNA-binding prediction.

Supplementary Data

Supplementary data are available online at https://academic.oup.com/bib.

Funding

References

1. Jolma A, Yan J, Whitington T, *et al*. DNA-binding specificities of human transcription factors. *Cell* 2013;**152**:327–39.
2. Tuupanen S, Turunen M, Lehtonen R, *et al*. The common colorectal cancer predisposition SNP rs6983267 at chromosome 8q24 confers potential to enhanced Wnt signaling. *Nat Genet* 2009;**41**:885–90.
3. Wasserman WW, Sandelin A. Applied bioinformatics for the identification of regulatory elements. *Nat Rev Genet* 2004;**5**:276–87.
4. Tompa M, Li N, Bailey TL, *et al*. Assessing computational tools for the discovery of transcription factor binding sites. *Nat Biotechnol* 2005;**23**:137–44.
5. Tan G, Lenhard B. TFBSTools: an R/bioconductor package for transcription factor binding site analysis. *Bioinformatics* 2016;**32**:1555–6.
6. Kuntz SG, Williams BA, Sternberg PW, *et al*. Transcription factor redundancy and tissue-specific regulation: evidence from functional and physical network connectivity. *Genome Res* 2012;**22**:1907–19.
7. Kumar M, Gromiha MM, Raghava GP. Identification of DNA-binding proteins using support vector machines and evolutionary profiles. *BMC Bioinform* 2007;**8**:463.
8. Lambert SA, Jolma A, Campitelli LF, *et al*. The human transcription factors. *Cell* 2018;**172**:650–65.
9. Basith S, Manavalan B, Shin TH, *et al*. iGHBP: computational identification of growth hormone binding proteins from sequences using extremely randomised tree. *Comput Struct Biotechnol J* 2018;**16**:412–20.
10. Furey TS. ChIP–seq and beyond: new and improved methodologies to detect and characterize protein–DNA interactions. *Nat Rev Genet* 2012;**13**:840–52.
11. Isakova A, Groux R, Imbeault M, *et al*. SMiLE-seq identifies binding motifs of single and dimeric transcription factors. *Nat Methods* 2017;**14**:316.
12. Gromiha MM, Nagarajan R. Computational approaches for predicting the binding sites and understanding the recognition mechanism of protein–DNA complexes. Advances in protein chemistry and structural biology. *Elsevier* 2013;**91**:65–99.
13. Feng P, Yang H, Ding H, *et al*. iDNA6mA-PseKNC: identifying DNA N6-methyladenosine sites by incorporating nucleotide physicochemical properties into PseKNC. *Genomics* 2019;**111**:96–102.
14. Manavalan B, Shin TH, Lee G. DHSpred: support-vector-machine-based human DNase I hypersensitive sites prediction using the optimal features selected by random forest. *Oncotarget* 2018;**9**:1944.
15. Wong K-C, Chan T-M, Peng C, *et al*. DNA motif elucidation using belief propagation. *Nucleic Acids Res* 2013;**41**:e153–3.
16. Ghandi M, Lee D, Mohammad-Noori M, *et al*. Enhanced regulatory sequence prediction using gapped k-mer features. *PLoS Comput Biol* 2014;**10**:e1003711.
17. Ren S, He K, Girshick R, *et al*. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2017;**39**:1137–49.
18. Tian C, Xu Y, Zuo W, *et al*. Coarse-to-fine cnn for image super-resolution. *IEEE Transactions on Multimedia* 2020. doi: 10.1109/TMM.2020.2999182.
19. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, p. 3431–40. IEEE, New York.

20. Hong J, Luo Y, Zhang Y, *et al*. Protein functional annotation of simultaneously improved stability, accuracy and false discovery rate achieved by a sequence-based deep learning. *Brief Bioinform* 2020;**21**:1437–47.

21. Hong J, Luo Y, Mou M, *et al*. Convolutional neural network-based annotation of bacterial type IV secretion system effectors with enhanced accuracy and reduced false discovery. *Brief Bioinform* 2020;**21**:1825–36.

22. Alipanahi B, Delong A, Weirauch MT, *et al*. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat Biotechnol* 2015;**33**:831.

23. Zeng H, Edwards MD, Liu G, *et al*. Convolutional neural network architectures for predicting DNA–protein binding. *Bioinformatics* 2016;**32**:i121–7.

24. Luo X, Tu X, Ding Y, *et al*. Expectation pooling: an effective and interpretable pooling method for predicting DNA–protein binding. *Bioinformatics* 2020;**36**:1405–12.

25. McLachlan GJ, Krishnan T. *The EM algorithm and extensions*. Hoboken: John Wiley & Sons, 2007.

26. Shen Z, Bao W, Huang D-S. Recurrent neural network for predicting transcription factor binding sites. *Sci Rep* 2018;**8**:15270.

27. Ma F, Chitta R, Zhou J *et al*. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017, p. 1903–11. ACM, New York.

28. Zhang Y, Qiao S, Ji S, *et al*. DeepSite: bidirectional LSTM and CNN models for predicting DNA–protein binding. *Int J Mach Learn Cybern* 2020;**11**:841–51.

29. Bao X-R, Zhu Y-H, Yu D-J. DeepTF: Accurate prediction of transcription factor binding sites by combining multi-scale convolution and long short-term memory neural network. In: *International Conference on Intelligent Science and Big Data Engineering*. 2019, p. 126–38. Springer, Cham, Switzerland.

30. Consortium EP. An integrated encyclopedia of DNA elements in the human genome. *Nature* 2012;**489**:57–74.

31. Zhou J, Troyanskaya OG. Predicting effects of noncoding variants with deep learning–based sequence model. *Nat Methods* 2015;**12**:931.

32. Bailey TL, Johnson J, Grant CE, *et al*. The MEME suite. *Nucleic Acids Res* 2015;**43**:W39–49.

33. Liu B, Wang S, Dong Q, *et al*. Identification of DNA-binding proteins by combining auto-cross covariance transformation and ensemble learning. *IEEE Trans Nanobioscience* 2016;**15**:328–34.

34. Manavalan B, Shin TH, Lee G. PVP-SVM: sequence-based prediction of phage virion proteins using a support vector machine. *Front Microbiol* 2018;**9**:476.

35. Manavalan B, Basith S, Shin TH, *et al*. Meta-4mCpred: a sequence-based meta-predictor for accurate DNA 4mC site prediction using effective feature representation. *Molecular Therapy-Nucleic Acids* 2019;**16**:733–44.

36. Nagarajan R, Ahmad S, Michael Gromiha M. Novel approach for selecting the best predictor for identifying the binding sites in DNA binding proteins. *Nucleic Acids Res* 2013;**41**:7606–14.

37. Trabelsi A, Chaabane M, Ben-Hur A. Comprehensive evaluation of deep learning architectures for prediction of DNA/RNA sequence binding specificities. *Bioinformatics* 2019;**35**:i269–77.

38. Wang F, Jiang M, Qian C *et al*. Residual attention network for image classification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, p. 3156–64. IEEE, New York.

39. Shen T, Zhou T, Long G *et al*. Disan: Directional self-attention network for rnn/cnn-free language understanding. In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018, p. 5446–55. AAAI, Palo Alto, CA.

40. Wang X, Girshick R, Gupta A *et al*. Non-local neural networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, p. 7794–803. IEEE, New York.

41. Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: *International conference on machine learning*. 2015, p. 448–56. PMLR, New York.

42. Liu W, Wen Y, Yu Z *et al*. Large-margin softmax loss for convolutional neural networks. In: Maria Florina B., Kilian Q. W. eds). *Proceedings of The 33rd International Conference on Machine Learning*. 2016, p. 507–16. PMLR, New York.

43. He K, Zhang X, Ren S *et al*. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 770–8. IEEE, New York.

44. He K, Zhang X, Ren S *et al*. Identity mappings in deep residual networks. In: *European conference on computer vision*. 2016, p. 630–45. Springer, Cham, Switzerland.

45. Xu B, Wang N, Chen T *et al*. Empirical evaluation of rectified activations in convolutional network, *arXiv* preprint arXiv:1505.00853, 30 November 2015, preprint: not peer reviewed.

46. Srivastava N, Hinton G, Krizhevsky A, *et al*. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 2014;**15**:1929–58.

47. Abadi M, Agarwal A, Barham P, *et al*. Tensorflow: large-scale machine learning on heterogeneous distributed systems. *arXiv* preprint arXiv:1603.04467, 2016. March 17 2016; preprint: not peer reviewed.

48. Kingma DP, Ba J. Adam: a method for stochastic optimization. *arXiv* preprint arXiv:1412.6980, 31 January 2014 2017; preprint: not peer reviewed.

49. Pan SJ, Yang Q. A survey on transfer learning. *IEEE Trans Knowl Data Eng* 2010;**10**:1345–59.

50. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv* preprint arXiv:1409.1556, 2014. April 13 2015; preprint: not peer reviewed.

51. Zhang Q, Zhu L, Huang D-S. High-order convolutional neural network architecture for predicting DNA-protein binding sites. *IEEE/ACM Trans Comput Biol Bioinform* 2018;**16**:1184–92.

52. Ghandi M, Mohammad-Noori M, Ghareghani N, *et al*. gkmSVM: an R package for gapped-kmer SVM. *Bioinformatics* 2016;**32**:2205–7.

53. Manavalan B, Basith S, Shin TH, *et al*. 4mCpred-EL: an ensemble learning framework for identification of DNA N4-methylcytosine sites in the mouse genome. *Cell* 2019;**8**:1332.

54. Chen W, Lv H, Nie F, *et al*. i6mA-Pred: identifying DNA N6-methyladenine sites in the rice genome. *Bioinformatics* 2019;**35**:2796–800.

55. Xu R, Zhou J, Wang H, *et al*. Identifying DNA-binding proteins by combining support vector machine and PSSM distance transformation. *BMC Syst Biol* 2015;1–12 BioMed Central.

56. Chen W, Tang H, Ye J, *et al*. iRNA-PseU: identifying RNA pseudouridine sites. *Molecular Therapy-Nucleic Acids* 2016;**5**:e332.

57. Kumar M, Gromiha MM, Raghava GPS. Prediction of RNA binding sites in a protein using SVM and PSSM profile, proteins: structure. *Function and Bioinformatics* 2008;**71**:189–94.