Interdisciplinary Sciences: Computational Life Sciences (2023) 15:32–43
https://doi.org/10.1007/s12539-022-00537-9

ORIGINAL RESEARCH ARTICLE

Improving language model of human genome for DNA–protein binding prediction based on task-specific pre-training

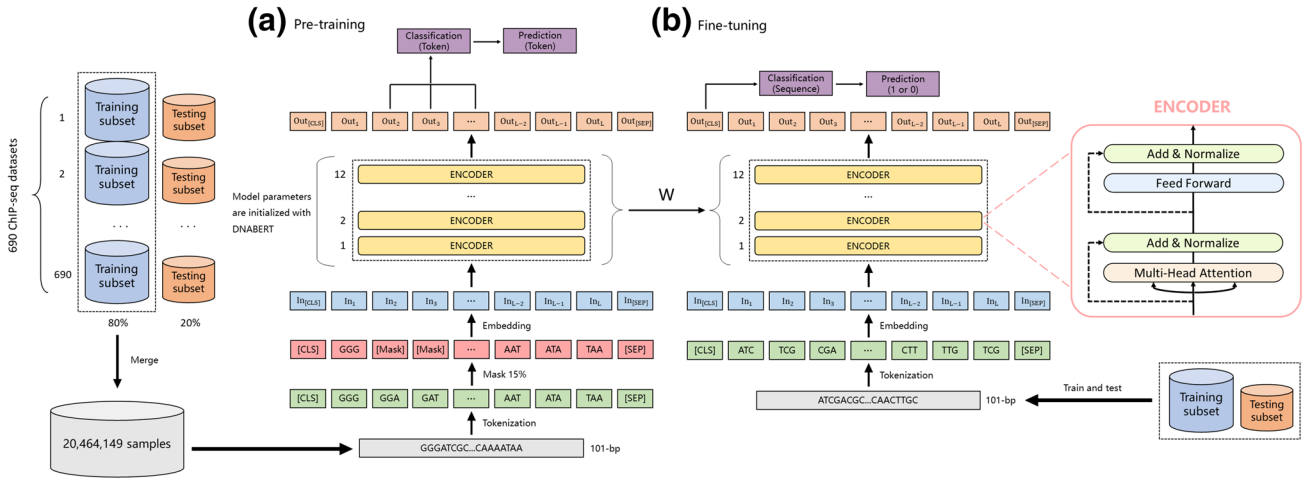
Hanyu Luo[1] · Wenyu Shan[1] · Cheng Chen[1] · Pingjian Ding[1] · Lingyun Luo[1,2]

Received: 31 March 2022 / Revised: 30 August 2022 / Accepted: 7 September 2022 / Published online: 22 September 2022

Abstract

The DNA–protein binding plays a pivotal role in regulating gene expression and evolution, and computational identification of DNA–protein has drawn more and more attention in bioinformatics. Recently, variants of BERT are also used to capture the semantic information of DNA sequences for predicting DNA–protein bindings. In this study, we leverage a task-specific pre-training strategy on BERT using large-scale multi-source DNA–protein binding data and present TFBert. TFBert treats DNA sequences as natural sentences and k-mer nucleotides as words. It can effectively extract upstream and downstream nucleotide context information by pre-training the 690 unlabeled ChIP-seq datasets. Experiments show that the pre-trained model can achieve promising performance on every single dataset in the 690 ChIP-seq datasets after simple fine tuning, especially on small datasets. The average AUC is 94.7%, outperforming existing popular methods. In conclusion, this study provides a variant of BERT based on pre-training and achieved state-of-the-art results in predicting DNA–protein bindings. We believe that TFBert can provide insights into other biological sequence classification problems.

Graphical abstract



Keywords Biological sequence · DNA–protein binding · BERT · Pre-training

✉ Lingyun Luo
luoly@usc.edu.cn

1 School of Computer Science, University of South China, Hengyang, Hunan 421001, People's Republic of China

2 Hunan Medical Big Data International Science and Technology Innovation Cooperation Base, Hengyang, Hunan 421001, People's Republic of China

1 Introduction

Transcription factor (TF) is a kind of protein that regulates the expression of downstream genes by binding to DNA sequences thus plays an integral role in the regulation of genome function [1, 2]. The DNA fragment that

binds to TF is called transcription factor-binding site (TFBS), which is small but highly variable, usually in the 4–30 bp range [3, 4]. Studies have demonstrated that TFBS and its adjacent mutations increase the risk of complex diseases in humans. Therefore, accurate predictions of DNA–protein binding are important for deciphering gene expression mechanisms and drug design. [5–7]. With the rapidly growing popularity of high-throughput sequencing technology, more and more high-quality datasets of TFBS were proposed, such as TRANSFAC [8], JASPAR [9], etc. However, these methods are very time-consuming and laborious. Therefore, it is urgent to develop rapid and highly accurate computational methods for the prediction of DNA–protein binding sites.

In recent years, deep learning methods have made great progress in various fields, especially Natural Language Processing (NLP) and Computer Vision (CV) [10–12]. Moreover, deep learning is becoming increasingly popular in bioinformatics and biocomputing including DNA–protein binding site prediction [13]. DeepBind [14] is the first method to predict DNA–protein bindings based on deep learning, using a single-layer convolutional neural network. Zeng et al. [15] investigate the effects of using various CNN architectures on this problem. However, as the convolution operation can only extract local features, it is not suitable for processing long DNA sequences. Shen et al. used the Bidirectional Gated Recurrent Unit (GRU) network for DNA–protein binding prediction to capture long-distance features [16]. However, the above methods still have two shortcomings: Firstly, the training of these models is supervised that relies on a large amount of labeled data. Secondly, when trained on small DNA–protein datasets, the performances of the models will decrease sharply. Therefore, advanced models that overcome these limitations are awaited.

Since pre-training on the large volume of unlabeled data can avoid the disadvantage introduced by insufficient data, it is adopted by many embedding models (such as word2vec [17]) as well as natural language processing (NLP) models (such as BERT [18]). As BERT has achieved strong performances in various tasks after large-scale corpus pre-training, variants of BERT that pre-train on a large corpus of protein sequences have been proposed recently in the field of bioinformatics [19–22]. Besides, another variant of BERT, named DNABERT [23], that pre-train on the whole human reference genome, is also proposed. DNABERT has proven capable of achieving state-of-the-art performance on many DNA sequence prediction tasks simultaneously. However, as DNABERT is trained and tested on general domain sequences, i.e., human reference genome, it cannot learn sufficient task-specific features in downstream tasks. For the task of identifying DNA–protein bindings, we need more specific DNA–protein binding data to improve the prediction results.

In this study, we propose a large-scale pre-training model based on BERT, named TFBert, for predicting DNA–protein binding sites in DNA sequences. Firstly, we preprocess all the sequences on ChIP-seq datasets using the k-mer method. Secondly, TFBert is initialized with DNABERT and further pre-trained on the fusion of 690 unlabeled ChIP-seq datasets to fully extract the semantic of context information in DNA sequences. Lastly, to test the effect of TFBert on each ChIP-seq dataset, we investigate three fine-tuning strategies: (1) Individual learning: The model is trained and tested on each individual dataset. (2) Global learning: The model is trained on a global dataset [23] that is collected from the 690 datasets, and further tested on each individual dataset. (3) Transfer learning: Instead of directly applying the global model to the individual dataset, we transfer it to the individual dataset using transfer learning [24]. Experiments show that TFBert can achieve state-of-the-art performance on the 690 ChIP-seq datasets, especially on small datasets. We believe that it can provide insights into other biological sequence classification problems.

2 Background

2.1 K-mers

For a particular DNA sequence, k-mers of the sequence is to extract several subsequences of length k by iterative segmentation. For example, given a sequence "ATCAGGCC", the 3-mers of this sequence: {ATC, TCA, CAG, AGG, GGC, GCC}, and the 6-mers: {ATCAGG, TCAGGC, CAGGCC }. K-mers captures contextual information better than single bases as words. In this paper, we preprocess all the sequences in the 690 ChIP-seq datasets using the k-mer method.

2.2 DNA-language model

DNABERT [25], a bidirectional Transformer-based architecture, had 12 layers Transformer encoder. As shown in the ENCODER in Fig. 1, the attention layer, the feed-forward layer and the regularization layer together form the encoding layer.

The multi-headed self-attentive layer is a variant of the self-attentive layer, extending the model's capability to concern itself with different positions and to better extract information from the context.

$$\text{MultiHead}\,(Q,K,V) = \text{Concat}\left(\text{head}_1,\ldots,\text{head}_h\right)W^o$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right)$$

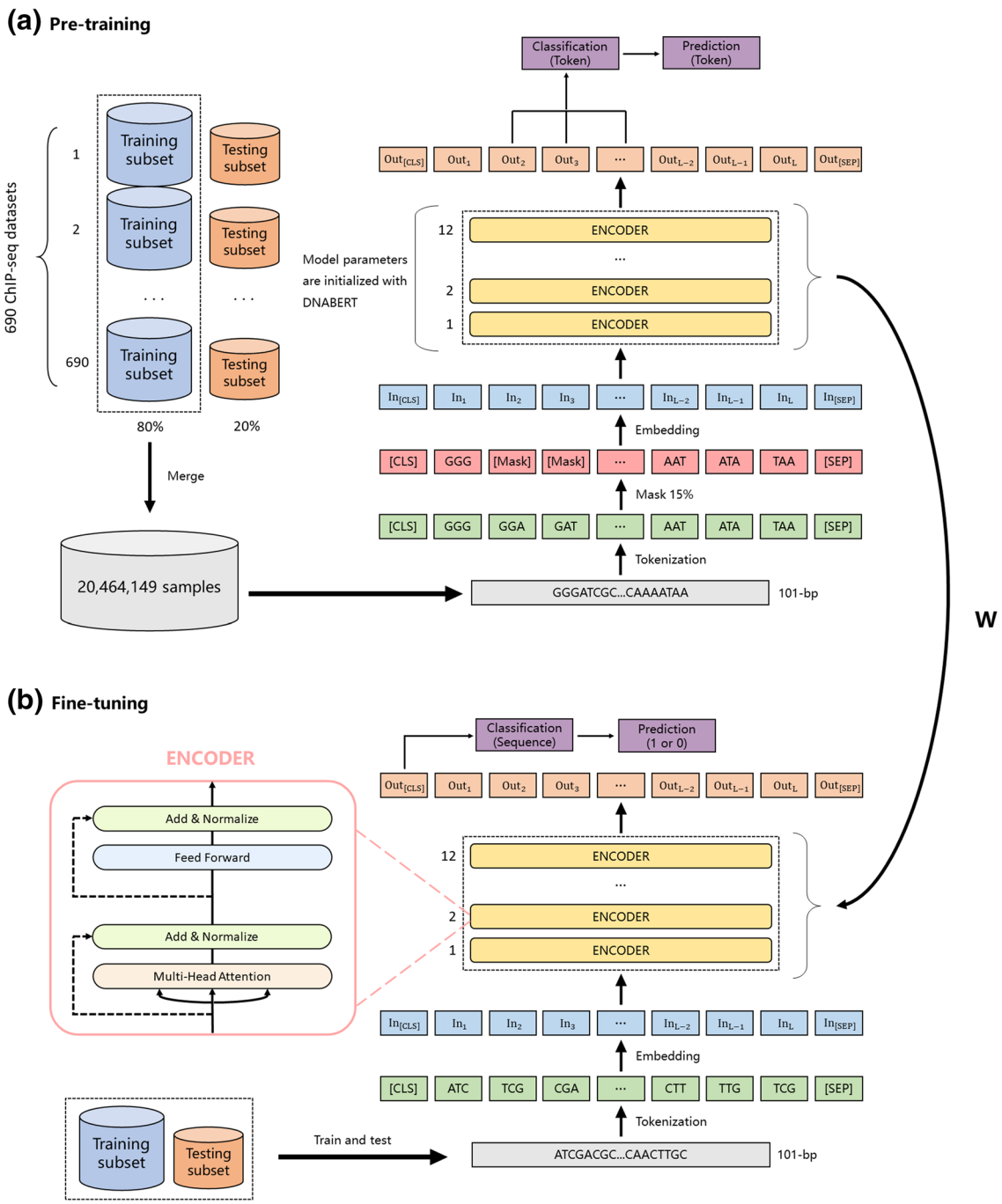


Fig. 1 The architecture of TFBert mainly composed of two phases: **a** pre-training phase; **b** fine-tuning phase

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \bullet V$$

where W_i^Q, W_i^K, W_i^V are the query, key and value linear transformation layers of the i-th head while h is the number of head. W^o is a linear conversion layer that can map the output dimension of multi-headed attention to the initial input dimension of ENCODER. They are all learnable parameters. The reader is referred to Vaswani et al. [11] for a more detailed description of multi-head self-attention.

The feed-forward layer consists of two simple fully connected layers and ReLU activation function.

$$F(x) = max\left(0, xW_1 + b_1\right)W_2 + b_2$$

The regularization layer consists of two operations, a residual connection (or shortcut connection) [26] to resolve the gradient update instability and a LayerNormalization [18] to accelerate convergence.

$$x_{l+1} = x_l + \text{MultiHead}\,(Q, K, V)$$

$$x_{l+2} = x_{l+1} + F(x_{l+1})$$

BERT-style pre-train and fine-tune scheme is ideal for DNA sequence understanding. In particular, with the development of deep learning models, most bioinformatics researchers are increasingly demanding data. Thus, the model is prone to overfitting and poor performance when the size of the labeled data volume is small. In contrast, BERT-style pre-training of large-scale unlabeled data solves this problem well. The contextual information captured by pre-training is well transferred to downstream tasks.

3 Materials and methods

3.1 Benchmark datasets

We retrieved the 690 ChIP-seq datasets downloaded from http://cnn.csail.mit.edu/ and provided by the ZengCNN study [15]. The 690 ChIP-seq datasets consisted of 91 human cell types and 161 specific DNA-binding proteins collected under different conditions. Each ChIP-seq dataset contains a training set (80%) and a corresponding test set (20%). These datasets were traditionally used as the benchmark to assess the deep learning models in DeepBind [14], ZengCNN, DeepARC [27] and SAResNet [23]. ZengCNN intercepts the 101 bp DNA sequences in the central region of each ChIP-seq peak as the positive subset and the positive subset is shuffled as the negative subset using the MEME tools [28]. Each positive subset has at least one ChIP-seq peak, and one DNA protein binding event, while the negative subset has none. To ensure the completeness of the experiment, we used all 690 ChIP-seq datasets. In addition, we experimented on the Global dataset from SAResNet [23] that was retrieved from the above 690 ChIP-seq datasets. This dataset contains 4,153,122 training samples (denoted as g-TR), 461,458 validation samples (g-VL) and 800,000 testing samples (g-TS). These global datasets could provide an overall measure for different learning models. Moreover, models trained from the global datasets can also be used on the 690 individual datasets using transfer learning to improve performances on them.

3.2 Methods

We designed and implemented our TFBert pipeline based on pre-training and fine tuning. First, we initialized TFBert with DNABERT pre-training model parameters. Second, TFBert was further pre-trained with a masked language model (MLM) on a fused dataset. Lastly, we leveraged three fine-tuning strategies on TFBert to perform task-specific DNA–protein predictions. The whole architecture is shown in Fig. 1. In the process, a k-mer DNA sequence was embedded as the composition of the token embedding and the positional embedding. Subsequently, the bidirectional feature information of the sequence was extracted through a 12-layer encoder. The header vector [CLS] in the output was ultimately placed into a classification layer to predict DNA–protein binding sites. In the following, we introduce the proposed framework in detail.

3.2.1 Task-specific pre-training

In BERT-like models, task-specific further pre-training is a very common technique [29]. As the DNABERT model is pre-trained on the gene sequences of all human chromosomes, whereas our DNA–protein binding prediction task is performed on specific DNA sequence data. Therefore, we perform task-specific pre-training on the 690 datasets. We fused all the data from the 690 datasets (only training datasets) as the pre-training dataset, which consists of 20,464,149 samples with labels removed. Subsequently, the unlabeled DNA sequences were tokenized as k-mers and expanded with a head token [CLS] and a tail token [SEP]. Specifically, 15% of the k-mers were randomly masked (replaced with [MASK] token), and we trained the model to predict the masked k-mers, as shown in Fig. 1a. This methodology is usually called the masked language model (MLM), which is an unsupervised learning model that uses data without manually labeled tags. Pre-training using MLM allows the model to learn the contextual information and obtain a better representation of the sequence vectors. However, this method has a drawback. There is a mismatch between pre-training phases and fine-tuning phases because the [MASK] token is never seen during fine-tuning phases. To address this drawback we use the following rule:

- The k-mers will be replaced by a special token [MASK] 80% of the time.
- The k-mers will be replaced by a random k-mers 10% of the time.
- The k-mers will keep unchanged the other 10% of the time.

Due to the k-mer DNA sequence generating method, using each k-mer as a word in token masking will introduce

an issue: the masked k-mer can be determined by its previous and following k-mers. For example, in a 3-mer sequence {ATC, TCG, [MASK], GAA}, [MASK] is the concatenation of "CG" in "TCG" and "A" in "GAA", so the [MASK] token is "CGA". Therefore, instead of randomly masking each k-mer, we mask continuous blocks of k-mers. In TFBert, for each masked 3-mer, we took its corresponding output vector and put it into a classification layer to determine its token in the overall vocabulary. For 3-mer, the number of classes in the vocabulary is 64 (4^3).

In the MLM training process, we optimize TFBert with AdamW [30] using the following parameters: $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1e-6, lr = 4e-4$ and weight decay as 0.01. The process took about 14 days on a single Tesla V100 GPU (32 GB memory).

3.2.2 Fine-tuning

To apply the pre-trained TFBert to each ChIP-seq dataset for DNA–protein prediction, we need to perform further fine tuning on it. The whole procedure is shown in Fig. 1(b). By feeding the embeddings of DNA sequences into TFBert, we can obtain the head token [CLS] of the last encoder layer. We then input [CLS] to the classifier layer that is comprised of one dropout layer, one fully connected neural network, and one softmax function. The dropout rate is set to 0.1 to prevent overfitting. We chose the binary cross-entropy loss function, shown as follows:

$$\text{loss} = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n * \log x_n + \left(1-y_n\right) * \log\left(1-x_n\right)\right]$$

where x_n is the prediction value and y_n is the label.

To investigate the performance of TFBert, we propose three fine-tuning strategies on the 690 datasets. The most straightforward strategy is to train a separate training subset of each dataset and subsequently test it on the corresponding test subset. We call it individual learning. The second strategy is called global learning: We train the classification model on a global dataset and apply it to the testing subset in each individual dataset. This means that one model is used to predict all the test subsets. The global training dataset we used is retrieved from the training subsets in the 690 datasets by under-sampling[23]. The last strategy uses transfer learning [24]: before applying the above global classification model directly to each individual testing subset, we conduct further training of it using the corresponding training subset.

3.3 Competing methods

Our study is to predict DNA–protein binding sites at the sequence level. Therefore, by comparing it with solutions related to this problem, DeepBind [14], ZengCNN [15], DeepARC [27], DNABERT-TF [25] and SAResNet [23], we prove the usability and robustness of our proposed deep learning approach.

3.3.1 DeepBind

This method is the first to use deep convolutional neural networks to predict DNA–protein binding sites, and it treats a window of genomic sequences as a single picture to be classified.

3.3.2 ZengCNN

This method illustrates the ability of the model to extract features by further analyzing the number of convolutional kernels and convolutional layers in DeepBind.

3.3.3 DeepARC

This model predicts transcription factors (TFs) by combining one-hot and dna2vec embedding, which consists of a CNN layer, a BiLSTM layer and an attention layer.

3.3.4 DNABERT-TF

This method is the first model to use the BERT architecture to pre-train the whole human gene sequence, which can be applied to many downstream tasks of gene sequences.

3.3.5 SAResNet

This method combines attention mechanisms and residual structures and further enhances the performance of DNA–protein binding using transfer learning.

3.4 Performance evaluation metrics

In this article, the problem we study is the prediction of DNA–protein binding sites, which can be regarded as a binary classification machine learning problem. We used different measurement metrics to evaluate the performance of TFBert predictions, such as accuracy, precision, recall, F1 score, MCC and AUC. These metrics are common in bioinformatics [31; 32].

$$Accuracy = \frac{T_{*+} + T_{*-}}{T_{*+} + T_{*-} + F_{*+} + F_{*-}}$$

$$\text{Precision} = \frac{T_{*+}}{T_{*+} + F_{*+}}$$

$$\text{Recall} = \frac{T_{*+}}{T_{*+} + F_{*-}}$$

$$F1score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$MCC = \frac{(T_{*+} \times T_{*-}) - (F_{*+} \times F_{*-})}{\sqrt{(T_{*+} + F_{*+})(T_{*+} + F_{*-})(T_{*-} + F_{*+})(T_{*-} + F_{*-})}}$$

where T_{*+}, T_{*-}, F_{*+} and F_{*-} denote the numbers of true positives, true negatives, false positives and false negative, respectively.

4 Experimental results

To verify the efficiency of our presented model and method, we perform a comprehensive experiment on all 690 datasets. Firstly, we searched for the optimal length of the k-mer pre-training model (cf. Section 4.1) and then focused on exploring the effect of task-specific pre-training on the model (cf. Section 4.2). In addition, we compared and analyzed three fine-tuning strategies (cf. Section 4.3). Finally, we compared the performance of our method with other competing methods in predicting DNA–protein binding sites (cf. Section 4.4).

4.1 K-mer selection

In this section, we investigate the effect of different k-mer pre-training models on the predictions and obtain the optimal results for the global dataset. In the experiment, we split the sequence of the global dataset into 3-mer to 6-mer, and input them into the corresponding pre-training model for fine tuning. We train our models with different k-mer on g-TR and choose the best parameters by validating on g-VL. During the testing phase, we tested the performance of the TFBert-k-mer model on the g-TS. Figure 2 provides the AUC and AUPR of the global learning for different k-mer. From Fig. 2, the four k-mer models had reached AUC and AUPR values above 0.9, and we can observe that TFBert-3mer has the best classification performance for both AUC and AUPR. This experiment demonstrated that the 3-mer pre-training model had a better ability to capture DNA–protein binding sites. We all use the 3-mer model in the following experiment.

4.2 Effectiveness of task-specific pre-training

In this section, it is analyzed how task-specific pre-training of the 690 datasets affects the final prediction performance. First, to investigate the effectiveness of task-specific pre-training, we performed pre-training of the model with the different steps and then found an optimal global model by fine tuning the global dataset.

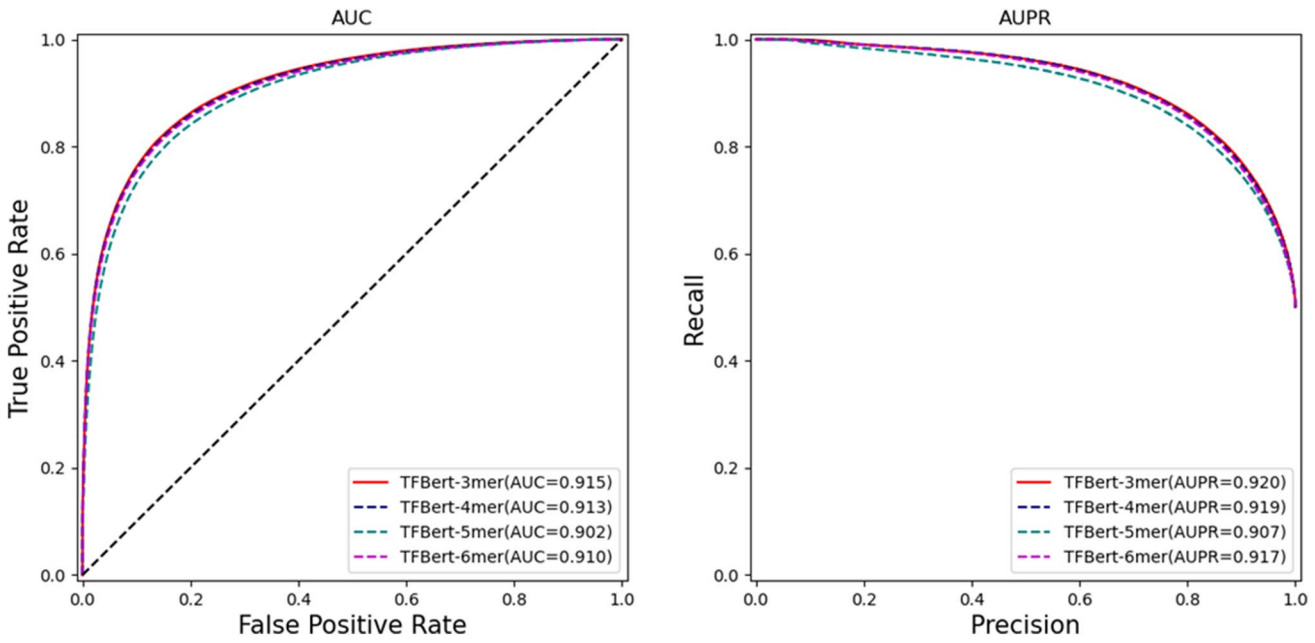


Fig. 2 ROC curves and PR curves of different k-mer pre-trained models. The models are trained on g-TR, validated on g-VL and finally, the performance is evaluated on g-TS. The result with TFBert-3mer (AUC = 0.915, AUPR = 0.920) outperforms other k-mer pre-training models

As illustrated in Fig. 3, the AUC value gradually increased at first with the increase of the number of steps, and then gradually decreased after reaching the maximum value when the number of steps was 20 000. The experimental results indicate that the pre-training for a specific task is very effective for improving the capability of TFBert to predict DNA–protein binding sites. Therefore, we selected the 20 000-step model as the final pre-trained model for subsequent experiments. Catastrophic forgetting was the main reason for the decrease in AUC after 20,000 steps, suggesting that the pre-trained knowledge is wiped out in the process of studying new knowledge [29].

To better represent the effect of the proposed task-specific pre-training, we make a comparison between the effect of pre-training and no pre-training for a specific task. We trained the two models on 690 ChIP-seq datasets with individual learning, global learning, and migration learning, respectively. As indicated in Table 1, the performance indicators of almost all models improved after the inclusion of task-specific pre-training, which demonstrates the effectiveness and robustness of performing task-specific pre-training in the models. Regardless of the learning method employed, including individual, global, or transfer, the overall model with task-specific pretraining consistently outperformed models without further pretraining. For example, adding task-specific pre-training in the model with transfer learning improved the average AUC of the model from 0.935 to 0.947. Figure 4 shows that models with task-specific pre-training can achieve significant performance improvements on models without further pre-training on 98.4%, 89.2% and 93.8% of the 690 datasets, respectively. As results show models with task-specific pre-training can achieve significant performance improvements.

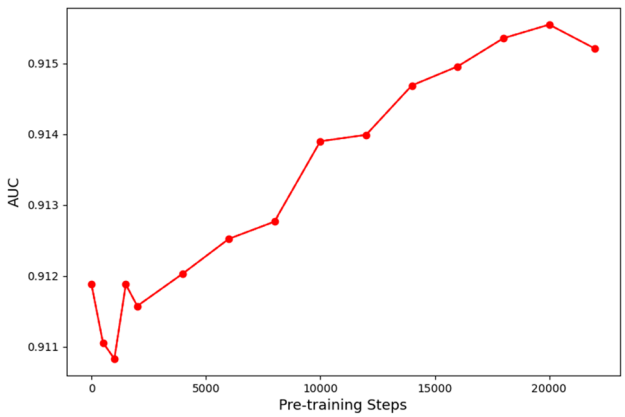


Fig. 3 AUC of the TFBert model for different further pre-training steps on the global dataset. After different steps of unsupervised MLM pre-training, the model is trained on g-TR, validated on g-VL, and finally tested on g-TS

4.3 Three fine-tuning strategies

In this section, we discuss the influence of three fine-tuning strategies on the final results. We initialized the parameters of TFBert using an optimal pre-training model with 20,000 steps, followed by training and testing on the 690 datasets using individual learning, global learning and transfer learning, respectively. The results of the three fine-tuning strategies are shown in Table 2. The average AUC of transfer learning is 0.2% higher than that of individual learning, the average accuracy is 0.1% higher, the average precision is 0.2% higher, the average recall is 0.1% higher, the average F1 score is 0.1% higher, and the average MCC is 0.3% higher. There is not much advantage over individual learning. The global model learns common features of all datasets and is suitable for prediction tasks across cell lines or unknown cell lines. Figure 5A-C scatter plots show head-to-head AUC score comparisons for the individual, global, and transfer models on 690 testing subsets, where the X and Y coordinates of each point represent the AUC score of the corresponding method, with blue points indicating above the diagonal and red points indicating below the diagonal. For example, as shown in Fig. 5B, there are 631 blue points located above the diagonal and 59 red points located below the diagonal. This indicates that transfer learning has a higher AUC value than individual learning on most of the datasets. This is because the initial performance of the transfer learning model is higher than that of the individual learning model, with more desirable initialization parameters.

Table 1 Performance of the different learning models on 690 ChIP-seq testing sets with or without the task-specific pre-training mechanism

Method	AUC	Accuracy	Precision	Recall	F1 score	MCC
Individual	0.901 ± 0.065	0.789 ± 0.087	0.814 ± 0.073	0.790 ± 0.087	0.783 ± 0.094	0.603 ± 0.160
Pre-individual	0.945 ± 0.043	0.879 ± 0.063	0.880 ± 0.062	0.879 ± 0.063	0.879 ± 0.063	0.759 ± 0.125
Global	0.896 ± 0.036	0.767 ± 0.025	0.710 ± 0.033	0.907 ± 0.054	0.795 ± 0.021	0.560 ± 0.047
Pre-global	0.917 ± 0.050	0.824 ± 0.044	0.830 ± 0.048	0.824 ± 0.044	0.822 ± 0.044	0.654 ± 0.092
Transfer	0.935 ± 0.042	0.817 ± 0.066	0.842 ± 0.054	0.818 ± 0.066	0.813 ± 0.071	0.659 ± 0.120
Pre-transfer	0.947 ± 0.041	0.880 ± 0.062	0.882 ± 0.061	0.880 ± 0.062	0.880 ± 0.062	0.762 ± 0.122

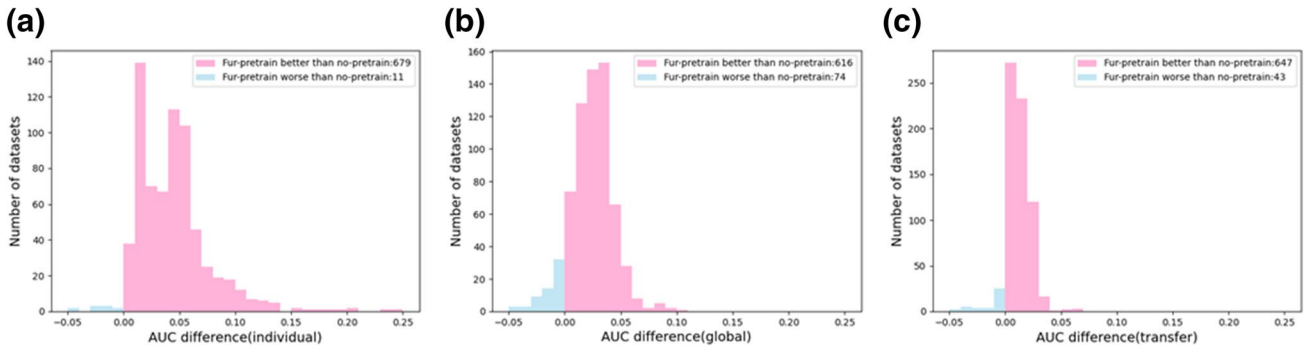


Fig. 4 There is some improved performance on the pre-training mechanism model compared to the no-pre-training model on the 690 datasets. The x-axis shows the difference in AUC on the models whether having task-specific pre-training. In panel A, the models with the task-specific pre-training mechanism outperformed those no-task-specific pre-training models on 679 datasets but showed worse than them on the other 11 datasets. In panel B, the models with the task-specific pre-training mechanism outperformed those no-task-specific pre-training models on 616 datasets but showed worse than them on the other 74 datasets. In panel C, the models with the task-specific pre-training mechanism outperformed those no-task-specific pre-training models on 647 datasets but showed worse than them on the other 43 datasets

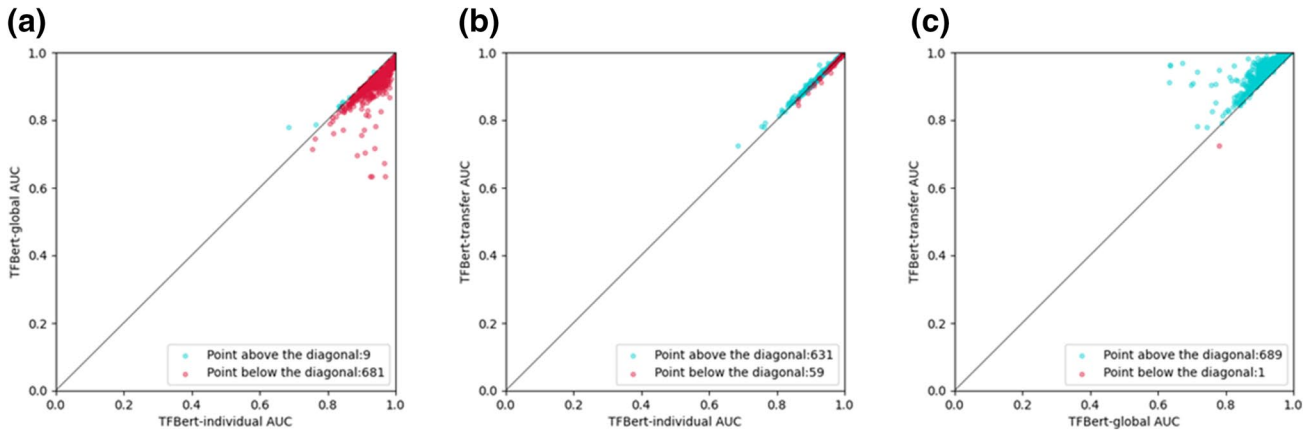


Fig. 5 The comparison of head-to-head AUC score on 690 ChIP-seq testing subset is about three fine-tuning strategies. Fig. A represents the comparison in performance between individual learning and global learning. Fig B represents the comparison in performance between individual learning and transfer learning, and Fig. C represents the comparison in performance between global learning and transfer learning

4.4 Performance comparison

In this section, we compare TFBert with other methods. The existing models basically use ChIP-seq as the benchmark dataset. To evaluate the performance of the models fairly and completely, we compared our model with DeepBind [14], ZengCNN [15], DeepARC [27], DNABERT-TF [25] and SAResNet [23] models in terms of AUC, Acc, Pre, Rec, F1 score and MCC on all 690 datasets.

According to the DeepBind paper, we first trained the DeepBind model with hyperparameter search and threefold cross-validation, and then tested the best model on the 690 test subsets. ZengCNN had modified the original DeepBind model architecture (original 1layer_16motifs). We took the best performance architecture (1layer_128motifs) to train the model using the same way. In addition, we reproduced the experiments of DeepARC and DNABERT-TF, and trained and tested the models in an equal experimental environment. All of the above models were trained and tested on a single Tesla V100 GPU (32 GB). The experimental data of SAResNet were obtained from http://csbio.njust.edu.cn/bioinf/sarsenet.

The violin plot in Fig. 6 shows the AUC distributions of DeepBind, ZengCNN, DeepARC, DNABERT-TF, SAResNet and TFBert on the 690 datasets. We applied the ANOVA test to show that TFBert achieved significant improvement in terms of AUC compared with other models (p-value is equal to 5.02e-28 when comparing TFBert with SAResNet, the other p-values are 1.78e-52, 1.90e-91, 2.14e-98 and 8.38e-121 when comparing with

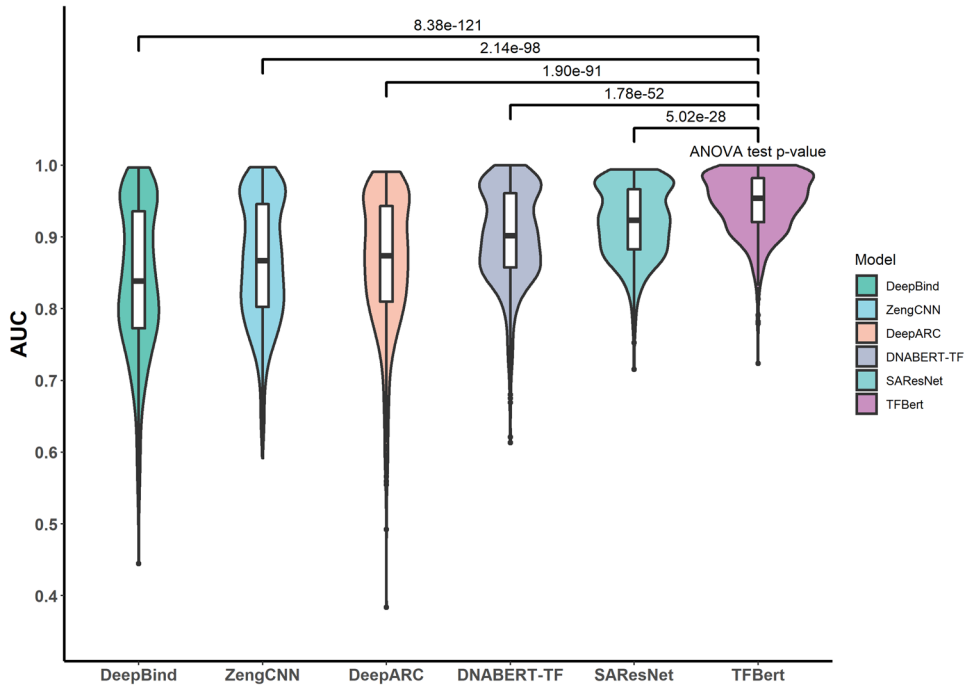


Fig. 6 Performance comparison and ANOVA test between TFBert and other prediction methods on 690 experimental datasets for DNA–protein binding prediction. For each of the violin plot with accompanying box plots, the top and bottom edges indicate the maximum and minimum values, and the upper and lower edges of the box indicate the upper and lower quartiles, respectively. The middle line indicates the median

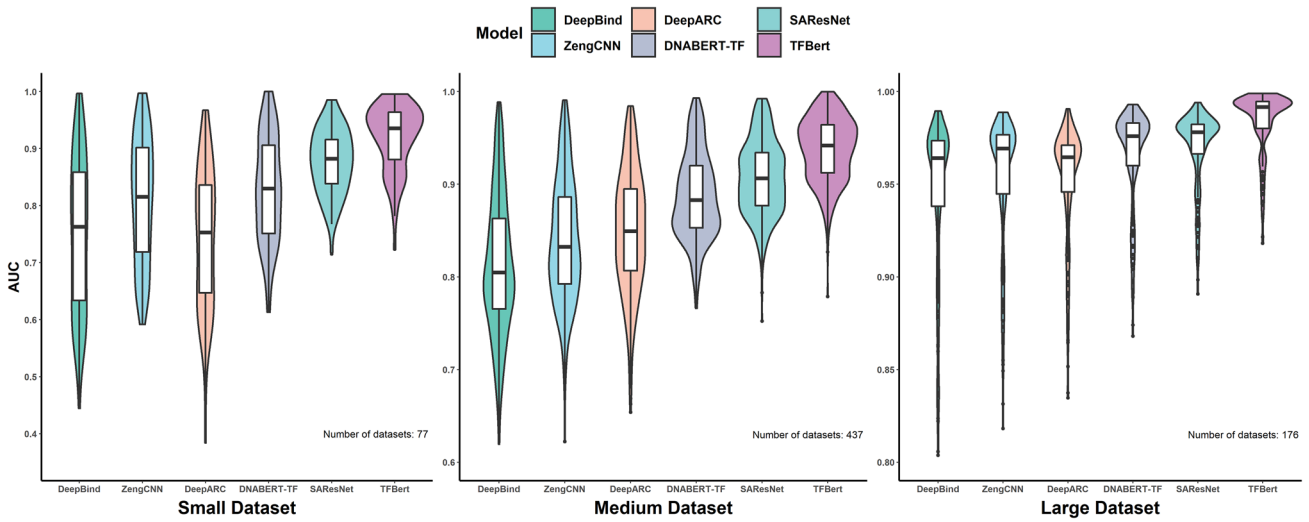


Fig. 7 The distribution of AUCs between TFBert and the other prediction methods on the datasets with different scales

Table 2 Performance of three fine-tuning strategies on the 690 ChIP-seq testing datasets

Method	AUC	Accuracy	Precision	Recall	F1 score	MCC
Individual	0.945 ± 0.043	0.879 ± 0.063	0.880 ± 0.062	0.879 ± 0.063	0.879 ± 0.063	0.759 ± 0.125
Global	0.917 ± 0.050	0.824 ± 0.044	0.830 ± 0.048	0.824 ± 0.044	0.822 ± 0.044	0.654 ± 0.092
Transfer	**0.947 ± 0.041**	**0.880 ± 0.062**	**0.882 ± 0.061**	**0.880 ± 0.062**	**0.880 ± 0.062**	**0.762 ± 0.122**

DNABERT-TF, DeepARC, ZengCNN and DeepBind, respectively). In particular, the average AUC of TFBert was 0.947, which was 2.7% higher than the second best model SAResNet (0.920). The minimum AUC was 0.724, which was 0.9% higher than SAResNet (0.715). Most of the AUC values of TFBert were higher than 0.9, which is sufficient to indicate the advantages of our model.

We further analyzed the effect of dataset size on the performance of the model. we divided the 690 datasets into three main categories: (1) Datasets with less than 4,000 training samples were defined as small datasets. (2) Datasets with 4,000 to 40,000 training samples were defined as medium datasets. (3) Datasets with more than 40,000 training samples were defined as large datasets. Based on the above division rules, we obtained 77 small datasets, 437 medium datasets and 176 large datasets. Figure 7 shows TFBert reached the best performances in all of the dataset categories. It improved the average AUC by 5.8%, 3.4% and 1.4% on small, medium and large datasets, respectively, compared to the second best performer SAResNet. A comparison of the average AUC of each model on datasets with different sizes as shown in Supplementary Table S1. It is notable that our model performs satisfactorily on small and medium-sized datasets, especially on small datasets. In addition, to evaluate the overall performance of TFBert, we further compared its performance with that of DeepBind, ZengCNN, DeepARC, DNABERT-TF, and SAResNet using six metrics (i.e., AUC, Acc, Pre, Rec, F1, and MCC) on the 690 datasets, as shown in Table 3. It can be seen that TFBert outperformed the other five methods on all of the six metrics. All of the experimental results were provided in the Supplementary material. Overall results demonstrate that after pre-training, satisfactory performance can be achieved even without transfer learning. Compared with traditional training methods, BERT-based pre-trained models have great advantages, especially on small and medium-sized datasets.

5 Discussion and Conclusion

In this study, we proposed TFBert, a task-specific pre-training model based on BERT for predicting DNA–protein binding sites, and investigated the effects of three fine-tuning strategies on its results. TFBert is featured by a stack of multiple Transformer Encoder and the multi-head attention mechanism in Encoder. In addition, the prior knowledge of 690 ChIP-seq datasets was integrated, and the context information of the DNA sequence was fully extracted by MLM pre-training. Benchmark experiments show that TFBert consistently outperforms several other existing methods on 690 datasets of different sizes. The advantages of our method can be demonstrated in the following three aspects. First, we provide a pre-training model of DNA–protein binding that can achieve satisfactory performance with simple fine tuning in a single dataset. Second, we investigate three fine-tuning strategies and explore the performance comparison among the three fine-tuning strategies. Third, we apply natural language processing techniques to biological sequence language, which could provide a new way of thinking about using sequence information for biological computation.

Although TFBert has achieved excellent performance for predicting DNA–protein binding sites, improvements are still needed. For instances, currently, TFBert only handles sequences with lengths of 101-bp, instead of sequences with long or various lengths; This study only shows the performance of TFBert on the ChIP-seq dataset, lacking external validation to better verify the generalization ability of the model. For future studies, to handle long sequences, we can cut the sequences into several 512-bp sequences and extract their features separately. For sequences with various lengths, we can consider adding a RNN structure to process the input sequences. For external validation, additional experiments will be performed on external validate data (i.e. ChIP-PCR assays).

TFBert can be seen not only as a biological language model to predict DNA–protein binding, but it can be used to solve various biological sequence prediction problems, such as predicting RNA–protein binding [33; 34] and DNA-binding protein [35–37] from sequences alone. Overall, our method provides a new perspective for studying biological sequences, as well as contributes to the development of pre-trained models of task-specific biological sequences.

Table 3 Performance evaluation of the proposed TFBert method and other existing methods for DNA–protein binding site prediction on 690 ChIP-seq datasets. TFBert-id is the model of individual learning. TFBert-gl is the model of global learning and TFBert-tf is the model of transfer learning

Method	AUC	Accuracy	Precision	Recall	F1 score	MCC
DeepBind	0.840 ± 0.101	0.773 ± 0.101	0.784 ± 0.106	0.755 ± 0.108	0.768 ± 0.105	0.548 ± 0.202
ZengCNN	0.866 ± 0.085	0.794 ± 0.091	0.806 ± 0.100	0.781 ± 0.095	0.791 ± 0.092	0.591 ± 0.182
DeepARC	0.864 ± 0.092	0.785 ± 0.097	0.866 ± 0.095	0.773 ± 0.121	0.776 ± 0.115	0.589 ± 0.175
DNABERT-TF	0.901 ± 0.065	0.789 ± 0.087	0.814 ± 0.073	0.790 ± 0.087	0.783 ± 0.094	0.603 ± 0.160
SAResNet	0.920 ± 0.049	0.849 ± 0.062	0.861 ± 0.063	0.831 ± 0.070	0.845 ± 0.064	0.698 ± 0.124
TFBert-id	0.945 ± 0.043	0.879 ± 0.063	0.880 ± 0.062	0.879 ± 0.063	0.879 ± 0.063	0.759 ± 0.125
TFBert-gl	0.917 ± 0.050	0.824 ± 0.044	0.830 ± 0.048	0.824 ± 0.044	0.822 ± 0.044	0.654 ± 0.092
TFBert-tf	**0.947 ± 0.041**	**0.880 ± 0.062**	**0.882 ± 0.061**	**0.880 ± 0.062**	**0.880 ± 0.062**	**0.762 ± 0.122**

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s12539-022-00537-9.

Acknowledgements This work has been supported by the Hunan Provincial Natural Science Foundation of China (No. 2019JJ50520, No. 2021JJ40467), the National Natural Science Foundation of China (No. 62002154), Research Foundation of Hunan Educational Committee (No. 20C1579), and Scientific Research Startup Foundation of University of South China (No. 190XQD096)

Funding Innovative Research Group Project of the National Natural Science Foundation of China, No. 62002154,Pingjian Ding, Natural Science Foundation of Hunan Province, No. 2019JJ50520,Lingyun Luo,No. 2021JJ40467, Pingjian Ding, Science and Technology Innovative Research Team in Higher Educational Institutions of Hunan Province, No. 20C1579, Pingjian Ding, Scientific Research Startup Foundation of University of South China, No. 190XQD096, Pingjian Ding

Availability and implementation Source code and the dataset are public in https://github.com/lhy0322/TFBert.

Declarations

Conflict of interest The authors declare that they have no competing interest.

References

1. Jolma A et al (2013) DNA-binding specificities of human transcription factors. Cell 152(1–2):327–339. https://doi.org/10.1016/j.cell.2012.12.009
2. Wasserman WW, Sandelin A (2004) Applied bioinformatics for the identification of regulatory elements. Nat Rev Genet 5(4):276–287. https://doi.org/10.1038/nrg1315
3. Tompa M et al (2005) Assessing computational tools for the discovery of transcription factor binding sites. Nat Biotechnol 23(1):137–144. https://doi.org/10.1038/nbt1053
4. Qu K, Wei L, Zou Q (2019) A review of DNA-binding proteins prediction methods. Curr Bioinform 14(3):246–254. https://doi.org/10.2174/1574893614666181212102030
5. Basith S et al (2018) iGHBP: computational identification of growth hormone binding proteins from sequences using extremely randomised tree. Comput Struct Biotechnol J 16:412–420. https://doi.org/10.1016/j.csbj.2018.10.007
6. Lambert SA et al (2018) The human transcription factors. Cell 172(4):650–665. https://doi.org/10.1016/j.cell.2018.01.029
7. Stormo GD (2000) DNA binding sites: representation and discovery. Bioinformatics 16(1):16–23. https://doi.org/10.1093/bioinformatics/16.1.16
8. Matys V et al (2006) TRANSFAC® and its module TRANSCompel®: transcriptional gene regulation in eukaryotes. Nucleic Acids Res 34(suppl_1):D108–D110. https://doi.org/10.1093/nar/gkj143
9. Fornes O et al (2020) JASPAR 2020: update of the open-access database of transcription factor binding profiles. Nucleic Acids Res 48(D1):D87–D92. https://doi.org/10.1093/nar/gkz1001
10. Chorowski J. et al. Attention-based models for speech recognition. arXiv preprint arXiv:1506.07503 2015.
11. Vaswani A et al (2017) Attention is all you need. Adv Neural Inform Proc Syst 2017:5998–6008
12. Xu K et al (2015) Show, attend and tell: neural image caption generation with visual attention. Int Conf Mach Learn 37:2048–2057
13. Hong J et al (2020) Protein functional annotation of simultaneously improved stability, accuracy and false discovery rate achieved by a sequence-based deep learning. Brief Bioinform 21(4):1437–1447. https://doi.org/10.1093/bib/bbz081
14. Alipanahi B et al (2015) Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. Nat Biotechnol 33(8):831–838. https://doi.org/10.1038/nbt.3300
15. Zeng H et al (2016) Convolutional neural network architectures for predicting DNA–protein binding. Bioinformatics 32(12):i121–i127. https://doi.org/10.1093/bioinformatics/btw255
16. Shen Z, Bao W, Huang D-S (2018) Recurrent neural network for predicting transcription factor binding sites. Sci Rep 8(1):1–10. https://doi.org/10.1038/s41598-018-33321-1
17. Mikolov, T., et al. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 2013. https://doi.org/10.48550/arXiv.1301.3781
18. Devlin, J, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 2018. https://doi.org/10.48550/arXiv.1810.04805
19. Elnaggar, A., *et al.* ProtTrans: towards cracking the language of Life's code through self-supervised deep learning and high performance computing. *arXiv preprint* arXiv:2007.06225 *2020. https://doi.org/10.48550/arXiv.2007.06225*
20. Iuchi H et al (2021) Representation learning applications in biological sequence analysis. Comput Struct Biotechnol J 19:3198. https://doi.org/10.1016/j.csbj.2021.05.039
21. Rao R et al (2019) Evaluating protein transfer learning with TAPE. Adv Neural Inf Process Syst 32:9689
22. Rives A et al (2021) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proc Natl Acad Sci. https://doi.org/10.1073/pnas.2016239118
23. Shen L-C et al (2021) SAResNet: self-attention residual network for predicting DNA-protein binding. Brief Bioinform. https://doi.org/10.1093/bib/bbab101
24. Pan SJ, Yang Q (2009) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359. https://doi.org/10.1109/TKDE.2009.191
25. Ji Y et al (2021) DNABERT: pre-trained Bidirectional encoder representations from transformers model for DNA-language in genome. Bioinformatics 37(15):2112–2120. https://doi.org/10.1093/bioinformatics/btab083
26. He, K., *et al.* Deep residual learning for image recognition. In, *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. p. 770–778.
27. Chen, J. and Deng, L. DeepARC: An Attention-based Hybrid Model for Predicting Transcription Factor Binding Sites from Positional Embedded DNA Sequence. In, *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE; 2020. p. 180–185. https://doi.org/10.1109/BIBM49941.2020.9313249
28. Bailey TL et al (2015) The MEME suite. Nucleic Acids Res 43(W1):W39–W49. https://doi.org/10.1093/nar/gkv416
29. Sun, C et al. (2019) How to fine-tune bert for text classification? China National Conference on Chinese Computational Linguistics. 194–206.
30. Loshchilov, I. and Hutter, F. Fixing weight decay regularization in adam. 2018.
31. Liu B et al (2016) Identification of DNA-binding proteins by combining auto-cross covariance transformation and ensemble learning. IEEE Trans Nanobiosci 15(4):328–334. https://doi.org/10.1109/TNB.2016.2555951
32. Do DT, Le TQT, Le NQK (2021) Using deep neural networks and biological subwords to detect protein S-sulfenylation sites. Brief Bioinform 22(3):bbaa128. https://doi.org/10.1093/bib/bbaa128

33. Kumar M, Gromiha MM, Raghava GPS (2008) Prediction of RNA binding sites in a protein using SVM and PSSM profile. Proteins 71(1):189–194. https://doi.org/10.1002/prot.21677
34. Chen W et al (2016) iRNA-PseU: Identifying RNA pseudouridine sites. Mol Ther Nucleic Acids 5:e332. https://doi.org/10.1038/mtna.2016.37
35. Xu R et al (2015) Identifying DNA-binding proteins by combining support vector machine and PSSM distance transformation. BMC Syst Biol. https://doi.org/10.1186/1752-0509-9-S1-S10
36. Chen W et al (2019) i6mA-Pred: identifying DNA N6-methyladenine sites in the rice genome. Bioinformatics 35(16):2796–2800. https://doi.org/10.1093/bioinformatics/btz015
37. Manavalan B et al (2019) 4mCpred-EL: an ensemble learning framework for identification of DNA N4-methylcytosine sites in the mouse genome. Cells 8(11):1332. https://doi.org/10.3390/cells8111332