mathematics

MDPI

Article

BTFBS: Binding Prediction of Bacterial Transcription Factors and Binding Sites Based on Deep Learning

Bingbing Jin [1], Song Liang [2], Xiaoqian Liu [1], Rui Zhang [1], Yun Zhu [1], Yuanyuan Chen [1], Guangjin Liu [2,*] and Tao Yang [1,*]

[1] College of Sciences, Nanjing Agricultural University, Nanjing 210095, China; 2022111009@stu.njau.edu.cn (B.J.); 2022111010@stu.njau.edu.cn (X.L.); 2024111007@stu.njau.edu.cn (R.Z.); chenyuanyuan@njau.edu.cn (Y.C.)

[2] College of Veterinary Medicine, Nanjing Agricultural University, Nanjing 210095, China; 2022207044@stu.njau.edu.cn

* Correspondence: liugj100@njau.edu.cn (G.L.); tao.yang@njau.edu.cn (T.Y.)

Abstract: The binding of transcription factors (TFs) to TF binding sites plays a vital role in the process of regulating gene expression and evolution. With the development of machine learning and deep learning, some successes have been achieved in predicting transcription factors and binding sites. In this paper, we develop a model, BTFBS, which predicts whether the bacterial transcription factors and binding sites combine or not. The model takes both the amino acid sequences of bacterial transcription factors and the nucleotide sequences of binding sites as inputs, and extracts features through convolutional neural network and MultiheadAttention. For the model inputs, we use two negative sample sampling methods: RS and EE. On the test dataset of RS, the accuracy, sensitivity, specificity, F1-score, and MCC of BTFBS are 0.91446, 0.89746, 0.93134, 0.91264, and 0.82946, respectively. Furthermore, on the test dataset of EE, the accuracy, sensitivity, specificity, F1-score and MCC of BTFBS are 0.87868, 0.89354, 0.86394, 0.87996, and 0.75796, respectively. Meanwhile, our findings indicate that the optimal approach for obtaining negative samples in the context of bacterial research is to utilize the whole genome sequences of the corresponding bacteria, as opposed to the shuffling method. The above results on the test dataset have shown that the proposed BTFBS model has a good performance and it can provide an experimental guide.

Keywords: transcription factors; binding sites; deep learning; bacteria

MSC: 68T07

check for updates

Academic Editor: Jonathan Blackledge

Received: 2 January 2025
Revised: 5 February 2025
Accepted: 7 February 2025
Published: 11 February 2025

Citation: Jin, B.; Liang, S.; Liu, X.; Zhang, R.; Zhu, Y.; Chen, Y.; Liu, G.; Yang, T. BTFBS: Binding Prediction of Bacterial Transcription Factors and Binding Sites Based on Deep Learning. *Mathematics* **2025**, *13*, 589. https://doi.org/10.3390/math13040589

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the network of prokaryotic transcriptional regulation, one of the most common forms is a transcription factor (TF), which has a single polypeptide containing both a sensory domain and a DNA-binding helix-turn-helix domain that directly binds DNA located at the promoter region of the gene (the "Switch" of gene transcription), thus affecting the binding of RNA polymerase to the promoter and in turn controlling gene transcription [1]. Although transcription factors binding data obtained by experimental methods are of high accuracy, they are time-consuming, laborious and expensive to obtain. Therefore, we hope to use computational methods to assist experimental screening and reduce research costs.

With the continuous development of machine learning and deep learning, more and more researchers use these approaches to predict the binding sites of transcription factors. For example, DeepBind was the first deep learning method to predict protein DNA binding

sites using convolutional neural networks [2]. Dilated [3] is a deep learning approach to model long-distance sequences by using dilated convolutional neural networks. KEGRU [4] combines the Bidirectional Gated Recurrent Unit (GRU) network with k-mer embedding to identify transcription factor binding sites. DanQ, a hybrid convolution and recurrent network framework, is a valuable resource for exploring the utility of non-coding DNA [5]. MAResNet is a deep learning method that combines bottom-up and top-down attention mechanisms and a state-of-the-art feed-forward network (ResNet), which was built by stacking attention modules that generate attention-aware features [6]. Machine learning and deep learning have obtained significant achievements in the prediction of binding sites.

There are also approaches focusing on the prediction of transcription factors. DeepTFactor employs a convolutional neural network to extract protein features for the purpose of predicting transcription factors [7]. By constructing a bacterial transcription factors database, Oliveira Monteiro et al. trained the deep learning model PredicTF, which provided the first pipeline for predicting and annotating TF within complex microbial communities [8]. However, these models only considered the prediction and classification of bacterial transcription factors, and did not take their binding sites into account.

In short, there are numerous methods to predict transcription factors and binding sites. Then, there is a natural question: for, e.g., bacteria, given a transcription factor and a binding site, how to determine whether they bind or not? This paper constructs a prediction model using deep learning to give an answer to this question.

As we know, the core idea of deep learning is to extract features and build models by training large-scale data. Data are the cornerstone of deep learning, containing rich information and rules, and are the original material for model learning and optimization. Data of high quality can provide more accurate and comprehensive information, thereby enhancing the performance and generalization ability of the model. For the prediction of transcription factor binding sites, positive samples can be obtained from experimental papers and from some databases; however, unfortunately, these do not provide negative samples. Therefore, the selection of negative samples plays a crucial role in the prediction performance of the model. Some methods, such as DeepBind [2], MAResNet [6], and deepRAM [9], use the shuffling method to obtain negative samples. However, the principle of this method is to shuffle the positive sequences while matching the dinucleotide composition [9] and the negative samples are probably not in the original whole genome sequences. To overcome this limitation, we use two negative sample sampling methods to select the negative samples from the whole genome sequences of the corresponding bacteria.

In general, based on convolutional neural networks (CNNs), ordinal positional encoding [10] and MultiheadAttention, we construct a deep learning model named BTFBS.

2. Materials and Methods

2.1. Datasets

At present, prokaryotes lack a comprehensive database of transcriptional regulation as eukaryotes. CollecTF [11] is an experimentally verified database of transcription factor binding sites in bacteria. As CollecTF provides the correspondence between transcription factor (TF) accession numbers and the binding sites, it is necessary to retrieve the amino acid sequences based on the TF accession numbers. It is regrettable that CollecTF has only provided data up to 2015. PRODORIC is one of the largest collections of prokaryotic transcription factor binding sites from a multitude of bacterial sources [12]. Our data from 2016 to 2021 is obtained from this database. As a result, 5159 non-redundant positive sequences are obtained. Approximately 10% of the sequences are randomly selected from the datasets as the independent dataset, while the remaining 90% of the sequences are used as the training and test datasets.

For negative samples, we adopt two methods. The first method introduced in [13] (denoted as RS hereafter) is to randomly select negative samples from the whole genome sequences of the bacteria to which the binding sites belong. Given that the binding regions of bacterial transcription factors are usually located in the promoter regions of genes that are considered non-coding regions [14], we only randomly intercept the non-coding regions from the whole genome sequences of the corresponding bacteria. Difflib (a flexible Python library) is employed for the comparison of sequence similarity between the intercepted sequences and the binding sites, thus avoiding the inadvertent capture of the latter. The sequence is excluded if the similarity score between the two sequences exceeds 90%. Subsequently, 27,236 non-redundant sequences are obtained, and 19% of the data are randomly selected as negative samples, matching the number of positive samples. Approximately 10% of the sequences are randomly selected as the independent dataset, while the remaining 90% are used as the training and test datasets. Figure 1A illustrates the process of RS.

The second method (denoted as EE hereafter) is based on experimental experience, which shows that the adjacent positions of the binding sites are not combined with their corresponding transcription factors. CollecTF provides the location information of binding sites. The DNA sequences are obtained by intercepting the adjacent positions of the binding sites, resulting in sequences of equal length to the binding sites themselves. Consequently, 9112 non-redundant sequences are obtained, with 57% of the data randomly selected as negative samples, in order to match the number of positive samples. Additionally, 10% of the sequences are randomly selected as the independent datasets, while the remaining 90% are utilized as the training and test datasets. Figure 1B illustrates the process of EE.

The positive and negative samples are merged to form the complete datasets, which are then divided into 80% training sets and 20% test sets. Five-fold cross-validation experiments are conducted on the training sets (Figure 1C).

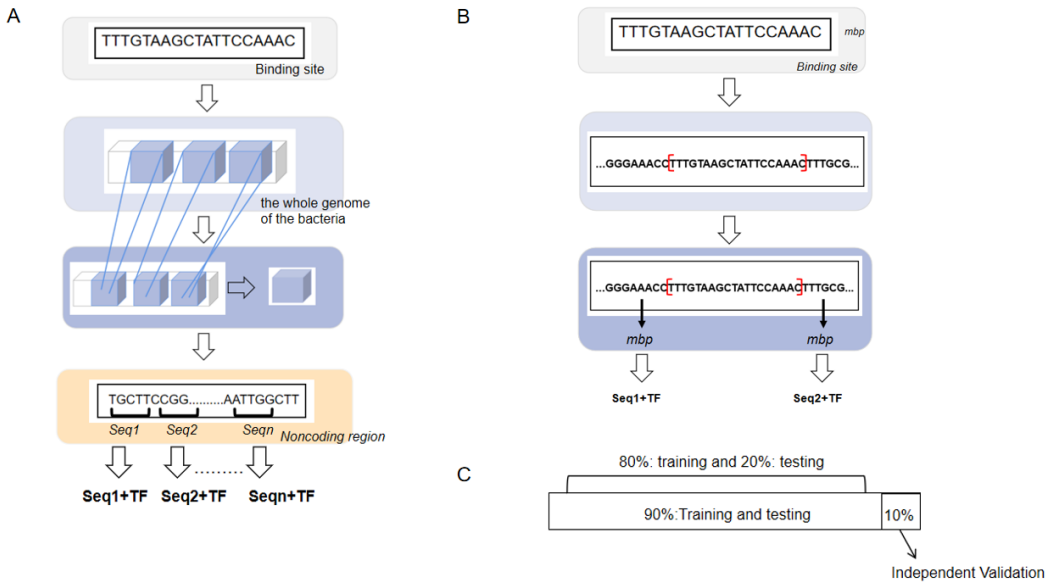


Figure 1. Negative sample extraction methods. (**A**) Flowchart of RS. (**B**) Flowchart of EE. (**C**). Schematic representation of the datasets.

2.2. Model Construction

In this work, we use convolutional neural networks as well as MultiheadAttention to construct a predictive model, BTFBS. The construction of BTFBS for bacteria is illustrated in Figure 2. The model has four main parts. The first part is the encoding module, which converts the sequence into dense vectors of fixed size. The second part is the convolutional

neural network module, which extracts features. The third part is the MultiheadAttention module, which captures connections and differences in the sequences. The fourth part is the classification module, which gives the classification results. The pipeline of BTFBS is shown in Algorithm 1.

Algorithm 1: The pipeline of BTFBS.

Input: Dseq, Pseq	DNA and protein sequences
1: $D_{OPE} \leftarrow Ordinal\ positional\ encoding(Dseq)$	
2: $P_{OPE} \leftarrow Ordinal\ positional\ encoding(Pseq)$	
3: $D_{embed} \leftarrow Embedding\ Layer1(D_{OPE})$	
4: $P_{embed} \leftarrow Embedding\ Layer2(P_{OPE})$	
5: $D_{cnn} \leftarrow CNN\ Block1(D_{embed})$	
6: $P_{cnn} \leftarrow CNN\ Block2(P_{embed})$	
7: D_f,$P_f \leftarrow MultiheadAttention(D_{cnn}$,$P_{cnn})$	
8: label $\leftarrow Classification(D_f$,$P_f)$	
Output:label(0/1)	0:non-binding 1:binding

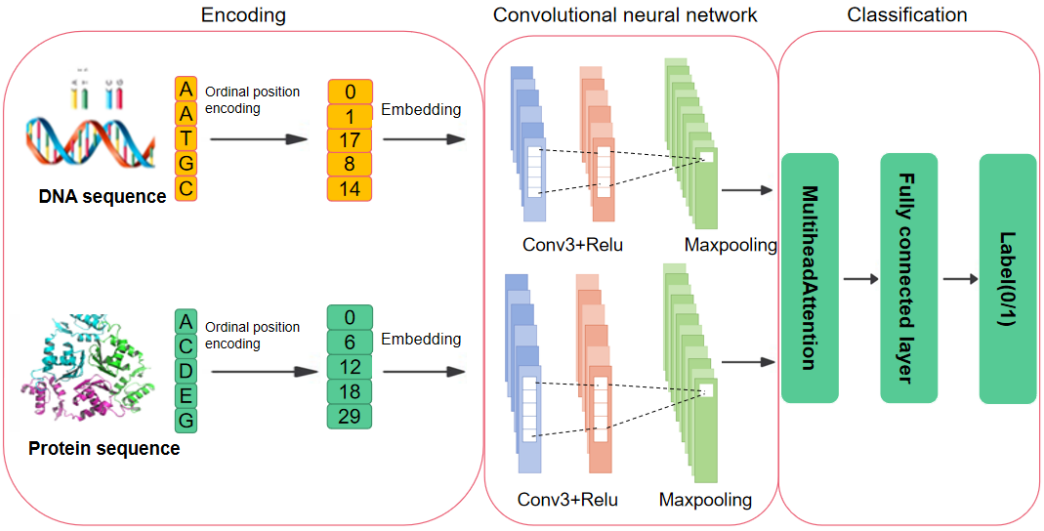


Figure 2. Flowchart of bacteria transcription factors and its binding sites BTFBS model based on convolutional neural network and MultiheadAttention.

2.2.1. Encoding

Before embedding, we first carry out ordinal positional encoding on the sequences. It is known from the literature that ordinal positional encoding [10] can improve the accuracy of CNN models and cross-validation, so that each sequence is represented as integers. In order to ensure consistency in sequence length, the maximum length of a DNA sequence is set to 100 and the maximum length of a protein sequence is set to 600. Sequences shorter than the maximum length are padded with zeros to match the fixed length. Then, the embedding layer receives these fixed length vectors and converts them into dense continuous feature vectors with a fixed size. As the first layer of embedding training, the weight matrix is constantly updated during the training process [15], and eventually an embedding matrix is learned to represent the sequence.

2.2.2. CNN

The previously learned embedding matrix is successively fed into convolution layer with three convolution kernels, and the rectified linear unit (ReLU) is used as the activation function. Finally, the obtained eigenvectors are processed by the maximum pooling layer.

2.2.3. MultiheadAttention

Following Bian et al. [16], the architecture of MultiheadAttention is described as follows. The DNA attention component first passes a DNA feature map D_{cnn} through a linear layer to compute a DNA query vector Q_D^i , a DNA key vector K_D^i and a DNA value vector V_D^i. The queries, keys and values for the DNA are given by

$$\begin{cases} Q_D^i = D_{cnn} \times W_q^i, \\ K_D^i = D_{cnn} \times W_k^i, \\ V_D^i = D_{cnn} \times W_v^i \qquad i = 1, 2, 3, \ldots, heads \end{cases} \tag{1}$$

where W_q^i, W_k^i, W_v^i are the weight matrices of three linear layers, and heads is the number of attention heads.

The protein attention component follows a similar process to the DNA attention component, and both attention components shares these three linear layers. The protein attention component first delivers a protein feature map P_{cnn} through a linear layer to compute a protein query vector Q_P^i, a protein key vector K_P^i and a protein value vector V_P^i . The queries, keys and values for the protein are given by

$$\begin{cases} Q_P^i = P_{cnn} \times W_q^i, \\ K_P^i = P_{cnn} \times W_k^i, \\ V_P^i = P_{cnn} \times W_v^i \qquad i = 1, 2, 3, \ldots, heads \end{cases} \tag{2}$$

where W_q^i, W_k^i, W_v^i are the weight matrices of three linear layers, and heads is the number of attention heads.

The attention matrix for DNAs and proteins is calculated as follows by using the softmax function:

$$\begin{cases} S_D^i = softmax(Q_D^i {K_D^i}^T / \sqrt{d_{K_D^i}}), \\ S_P^i = softmax(Q_P^i {K_P^i}^T / \sqrt{d_{K_P^i}}), \qquad i = 1, 2, 3, \ldots, heads \end{cases} \tag{3}$$

where $d_{K_D^i} = d_{K_P^i} = d_{head} = d_{cnn}/\text{heads}$ is the dimension of queries and keys.

The DNA attention matrix for each head is multiplied by the DNA value matrix for each head to obtain the attended DNA feature map for each head. Then the attended DNA feature maps of all attention heads are concatenated along the channel dimension and fed into a linear layer to obtain the final attended DNA feature map Z_{DC}:

$$Z_{DC} = Concat(S_D^i \times V_D^i) \times W_0 \qquad i = 1, 2, 3, \ldots, heads \tag{4}$$

where W_0 is the weight matrix.

The protein attention matrix S_P^i and protein value matrix V_P^i are operated on in the same way to obtain the final protein feature map Z_{PC}:

$$Z_{PC} = Concat(S_P^i \times V_P^i) \times W_0 \qquad i = 1, 2, 3, \ldots, heads \tag{5}$$

where W_0 is the weight matrix.

Finally, in order to obtain the final feature maps, the attended feature maps are combined with the original feature maps:

$$\begin{cases} D_f = 0.5 \times Z_{DC} + 0.5 \times D_{cnn}, \\ p_f = 0.5 \times Z_{PC} + 0.5 \times P_{cnn}, \qquad i = 1, 2, 3, \ldots, heads \end{cases} \tag{6}$$

where D_f denotes the DNA feature map and P_f denotes the protein feature map.

2.2.4. Classification

We use the fully connected layer as the classification module, and concatenate the feature vectors obtained by the convolutional neural network into the classification module. The classification module consists of a dropout layer, a fully connected layer and a ReLU activation function. The dropout layer [16] is used to avoid overfitting during training.

2.3. Evaluation Metrics

Accuracy can be interpreted as the ratio of correctly predicted samples to all samples, sensitivity as the ratio of all positive samples correctly predicted to all actual positive samples, specificity as the ratio of all negative samples correctly predicted to all actual negative samples, and the F1-score takes into account the precision and recall of a classification model and is defined as the reconciled mean of the model precision and recall. These values range from 0 to 1, and larger values represent better model performance. The Matthews correlation coefficient takes into account the counts of all four categories in the confusion matrix and evaluates them collectively, with a range of $[-1, 1]$. Where -1 means that the model's prediction is completely opposite to the real situation, 0 means that the model's performance is equivalent to random prediction, and 1 means perfect prediction. The Matthews correlation coefficient is particularly suitable for binary classification problems with unbalanced samples, as it takes into account the prediction performance of each category simultaneously. In order to measure the performance of the models, we calculate the accuracy (ACC), sensitivity, specificity, F1-score, and Matthews correlation coefficient (MCC), which are defined below [17]:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{8}$$

$$Specificity = \frac{TN}{TN + FP} \tag{9}$$

$$F1\text{-}score = \frac{2TP}{2TP + FN + FP} \tag{10}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{11}$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

2.4. Experimental Settings

2.4.1. Hyperparameters

Appropriate hyperparameters, such as batch size, learning rate, and convolutional kernel size, can help the model produce more accurate results and converge quickly [16]. Since DNA sequences are usually shorter than protein sequences, our CNN blocks for proteins and DNA use different hyperparameter configurations. In the CNN block for DNA, the kernel sizes of the three convolutions are set to 3, 5, and 6, while in the CNN block for proteins, the kernel sizes of the three convolutions are set to 5, 8, and 10. Figure 3 lists the details of these parameters. During training, we choose the cross entropy loss function as the loss function. the AdamW optimizer is utilized to update the network weights. It has two hyperparameters, weight decay and learning rate, both of which are set to 0.0001. Batch size is set to 16. An early-stop approach is taken during model validation,

i.e., the training process is stopped when the model's performance on top of the validation set is no longer consistently improving.

Block	Layer	Parameters
DNA CNN Block	1-CNN	size = 3, channel = 40
	2-CNN	size = 5, channel = 80
	3-CNN	size = 6, channel = 160
Protein CNN Block	1-CNN	size = 5, channel = 40
	2-CNN	size = 8, channel = 80
	3-CNN	size = 10, channel = 160
Multihead Attention	Multihead Attention	heads = 5
Classification Block	Concatenate	
	1-dropout	ratio = 0.1
	1-Fully Connected Layer	in = 320, out = 1024
	2-dropout	ratio = 0.1
	2-Fully Connected Layer	in = 1024, out = 1024
	3-dropout	ratio = 0.1
	3-Fully Connected Layer	in = 1024, out = 512
	4-Fully Connected Layer	in = 512, out = 2

Figure 3. BTFBS parameters.

2.4.2. Training Settings and Experimental Environment

The entire model is implemented in the Pytorch (https://pytorch.org, accessed on 1 October 2022) architecture and is trained on the following hardware configurations:

CPU: Intel(R) Xeon(R) Gold 5218 CPU @ 2.30 GHz;
GPU: NVIDIA-SMI 455.45.01;
Training time: 1400.78 s;
Memory Usage: 2576.02 MB.

3. Results

3.1. Comparison of Models Based on the Two Negative Sample Sampling Methods

This paper employs two distinct negative sampling methods: RS and EE. As shown in Tables 1 and 2, the results on the test dataset for RS are found to be ACC = 0.91446, Sensitivity = 0.89746, Specificity = 0.93134, MCC = 0.82946, and F1-score = 0.91264, while the results for EE are that ACC = 0.87868, Sensitivity = 0.89354, Specificity = 0.86394, MCC = 0.75796, and F1-score = 0.87996. As seen in Figure 4A,B, the performance of RS is better than that of EE, and the accuracies of independent validation and testing are similar, indicating no bias for the prediction models. However, for the merged independent datasets (RS + EE), EE shows better prediction accuracy than RS, indicating the prediction model trained by EE has better generalization ability than RS.

Table 1. The performance of the model on the various datasets (RS).

Model	F1-Score	ACC	Sensitivity	Specificity	MCC
CNN3 + MHA [1]	0.91264	0.91446	0.89746	0.93134	0.82946
CNN3 + MHA(ID) [2]	0.91	0.91	0.904	0.912	0.82
CNN3 + MHA(ID (RS + EE))	0.6934	0.737	0.9008	0.6566	0.5256

[1] MultiheadAttention. [2] independent datasets.

Table 2. The performance of the model on the various datasets (EE).

Model	F1-Score	ACC	Sensitivity	Specificity	MCC
CNN3 + MHA [1]	0.87996	0.87868	0.89354	0.86394	0.75796
CNN3 + MHA(ID) [2]	0.876	0.876	0.888	0.864	0.76
CNN3 + MHA(ID (RS + EE))	0.7608	0.818	0.8754	0.7898	0.6338

[1] MultiheadAttention. [2] independent datasets.

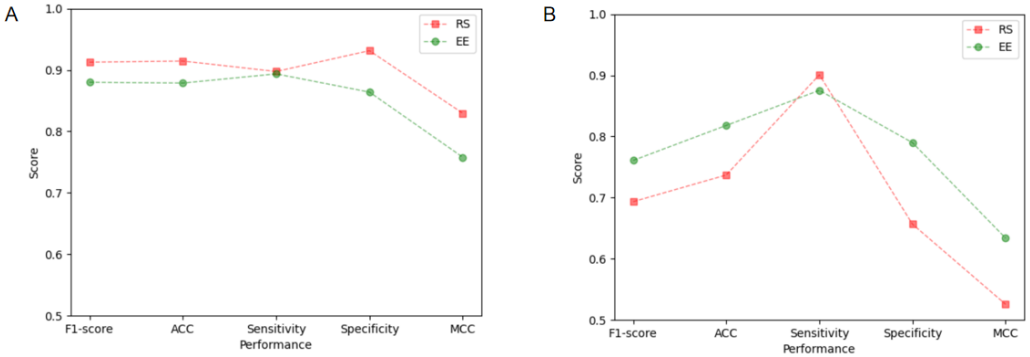


Figure 4. (**A**) Comparison of models based on the two negative sample sampling methods (RS and EE) on test data set. (**B**) Comparison of models based on the two negative sample sampling methods on merged independent datasets (RS + EE).

3.2. Ordinal Positional Encoding Can Improve the Performance of the Model

Ordinal positional encoding is a coding method that adds positional information [10]. To test whether this coding method improves the performance of the model, a comparative test was conducted. The results for RS and EE are shown in Figure 5A and Figure 5B, respectively. It is clear that a model with ordinal positional encoding performs better than the one without it.

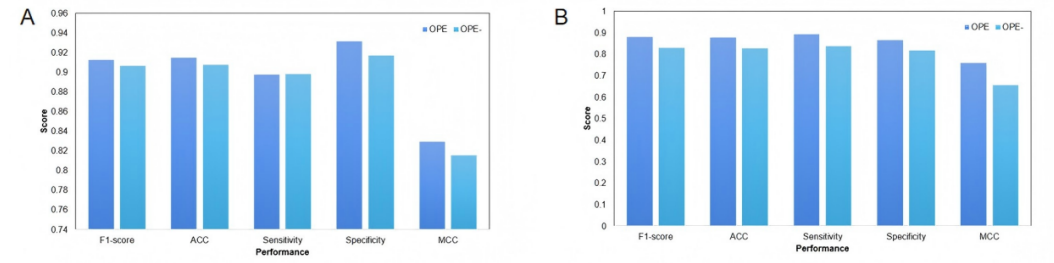


Figure 5. Results on test sets with and without ordinal positional encoding on the two negative sample sampling methods (RS and EE). (**A**) Results for RS. (**B**) Results for EE.

3.3. Comparison with Other Methods

The performance of BTFBS is compared with several conventional methods for predicting protein-DNA-binding sites [18], namely, DeepBind [2], Dilated [3], KEGRU [4], DanQ [5], and MAResNet [6]. DeepBind, Dilated, DanQ, and KEGRU were implemented based on the deepRAM [9] framework. Since our model has two input parts, to ensure fairness, we modify BTFBS to take only one input of DNA sequence for comparison with these models. Bacterial binding sites are used as positive samples (while negative samples are obtained through deepRAM), and they are randomly divided into a training set comprising 80% and a test set comprising 20% of the samples. The aforementioned data are then used to train DeepBind, Dilated, DanQ, KEGRU, and MAResNet, respectively. Following the completion of three-fold cross-validation on the training set and subsequent evaluation of the models' performance on the test dataset, the two metrics for the different

models on the test datset are as presented in Figure 6, and Table 3 lists additional performance metrics for these models. The accuracy and AUC of BTFBS are 0.8065 and 0.8854, representing an improvement of 0.86% in accuracy and 3.72% in AUC compared to the second-highest-performing model.

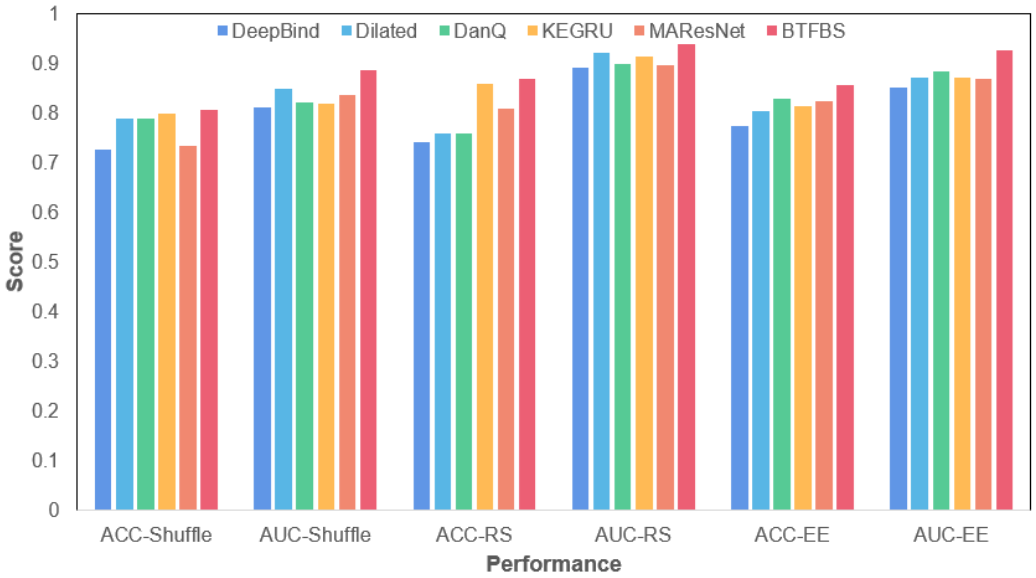


Figure 6. The results of 3-fold cross-validation of BTFBS and some other existing methods. ACC-Shuffle and AUC-Shuffle are obtained by shuffling the positive sample. ACC-RS and AUC-RS are obtained from RS. ACC-EE and AUC-EE are obtained from EE.

Table 3. The performance of BTFBS and other methods on the test set (shuffle).

Model	Accuracy	Sensitivity	Specificity	MCC	F1	AUC
DeepBind	0.7272	0.7560	0.7068	0.4563	0.6966	0.8105
Dilated	0.7878	0.7884	0.7872	0.5752	0.7961	0.8482
DanQ	0.7878	0.8163	0.7600	0.5770	0.7920	0.8207
KEGRU	0.7979	0.7454	0.8636	0.6060	0.8039	0.8176
MAResNet	0.7337	0.6993	0.7575	0.4572	0.7407	0.8368
BTFBS	0.8065	0.8004	0.8124	0.6131	0.8012	0.8554

Subsequently, the negative samples were replaced with data from RS and EE, respectively, to ascertain whether these two negative sample schemes are more effective than the shuffling method. The training of the model is conducted in accordance with the same procedure as that employed for the shuffling method. Figure 6 depicts the two metrics (RS) for the various models on the test dataset, while Table 4 enumerates supplementary performance metrics for these models. Figure 6 illustrates the two metrics (EE) for the disparate models on the test set, with Table 5 providing additional performance metrics for these models. We can see from the above results that (1) our model outperforms other models in all three negative sample sampling methods and (2) when using negative sample data in RS or EE, each model performs better than when using the shuffling method. A possible reason is that the data obtained by the shuffling method may be not pieces of the whole genome sequences of the corresponding bacteria.

Table 4. The performance of BTFBS and other methods on the test set (RS).

Model	Accuracy	Sensitivity	Specificity	MCC	F1	AUC
DeepBind	0.7421	0.746	0.7384	0.4844	0.7401	0.8902
Dilated	0.7578	0.8833	0.647	0.5406	0.7737	0.921
DanQ	0.7578	0.6984	0.8153	0.5177	0.7394	0.8988
KEGRU	0.8593	0.8281	0.8906	0.7201	0.8548	0.9145
MAResNet	0.8095	0.8644	0.7461	0.626	0.8111	0.8967
BTFBS	0.8681	0.8651	0.871	0.7362	0.8666	0.938

Table 5. The performance of BTFBS and other methods on the test set (EE).

Model	Accuracy	Sensitivity	Specificity	MCC	F1	AUC
DeepBind	0.7734	0.7966	0.7536	0.5485	0.7642	0.8513
Dilated	0.8046	0.7777	0.8243	0.6006	0.7706	0.8702
DanQ	0.8281	0.8088	0.8500	0.6575	0.8333	0.8824
KEGRU	0.8125	0.8548	0.7727	0.6284	0.8153	0.8719
MAResNet	0.8226	0.8800	0.7564	0.6469	0.7946	0.8678
BTFBS	0.8568	0.8495	0.8639	0.7137	0.8546	0.9262

3.4. Case Study

In order to demonstrate the validity and generalization ability of our proposed method, a search of the literature from 2022 to 2024 on the Web of Science was conducted using the keywords "bacteria", "transcription factor", and "binding sites". A total of 11 pieces of data were retrieved, which were used to identify protein sequences of transcription factors and DNA sequences of binding sites; these were FabT [19], TrcR [20], HP1021 [21], LitR [22], and CodY [23]. In trying to predict the binding for these TFs and binding sites, with the RS approach, two prediction errors were identified in the positive samples (Table 6), while with the EE approach, there were two instances of incorrect prediction for positive samples (Table 7).

Table 6. The prediction results of RS.

TF	TFBS	Prediction	Real
FabT	AGTTTGAACATCAAATG	False	True
TrcR	ATACTACACTTGTATTAC	True	True
HP1021	TGTTAAT	True	True
HP1021	AGTAAATCA	True	True
HP1021	TGCTACA	True	True
LitR	CAACTTATTATTTTATTAA TATTAT	True	True
LitR	CTGATATTATTTTCGTTA ATATTG	True	True
LitR	TAATAGGTTGATCTACAA AGATAAATAT	True	True
LitR	AGCCTTATCTCAGA	False	True
CodY	ATATTTCGTAGTA	True	True
CodY	AATTTTGACAATC	True	True

Table 7. The prediction results of EE.

TF	TFBS	Prediction	Real
FabT	AGTTTGAACATCAAATG	False	True
TrcR	ATACTACACTTGTATTAC	True	True
HP1021	TGTTAAT	True	True
HP1021	AGTAAATCA	True	True

Table 7. *Cont.*

TF	TFBS	Prediction	Real
HP1021	TGCTACA	True	True
LitR	CAACTTATTATTTTATTAA TATTAT	True	True
LitR	CTGATATTATTTTCGTTA ATATTG	True	True
LitR	TAATAGGTTGATCTACAA AGATAAATAT	True	True
LitR	AGCCTTATCTCAGA	False	True
CodY	ATATTTCGTAGTA	True	True
CodY	AATTTTGACAATC	True	True

3.5. BTFBS Provides an Experimental Guide

While experimental methods for obtaining transcription factor binding data are highly accurate, these methods are also time-consuming, laborious, and costly. Our model BTFBS can guide experiments and help researchers to narrow down the scope of their work. The WOAH Reference Lab for Swine Streptococcosis, College of Veterinary Medicine, Nanjing Agricultural University provided two 100 bp DNA fragments that had been demonstrated to bind to the transcription factors XtrSs and ArgR, respectively. In order to select an appropriate sliding window length, we counted the lengths of all binding sites in CollecTF (Figure 7A) as well as those of experimentally studied Streptococcus (Figure 7B). Figure 7 illustrates that the majority of binding sites are between 6 and 30 bp in length, whereas the binding sites of Streptococcus are predominantly between 6 and 25 bp. Therefore, the aforementioned DNA sequences were intercepted using a sliding window of length 25 and a step size of 1. For XtrSs, the RS-predicted fragment containing the binding site scored 0.99009, ranking third out of 76 DNA fragments (Table 8), while the EE-predicted fragment scored 0.9874, ranking second out of 76 DNA fragments (Table 9). For ArgR, the RS-predicted fragment containing the binding site scored 0.9994, ranking second out of 76 DNA fragments (Table 10), while the EE-predicted fragment scored 0.9172, ranking twelfth out of 76 DNA fragments (Table 11). Specific binding site information is available in [24].

Table 8. DNA prediction score (XtrSs) by using RS.

DNA	Score
AGAACGTTACCGCTTCTTTAGTTTT	0.997107
CTGTTGAAGAACGTTACCGCTTCTT	0.9908368
TATAAACATGCTGTTGAAGAACGTT [1]	0.9900971
TTTTGGGGATGCAATGTTTATTCAA	0.98833066
TAGTTTTGGGGATGCAATGTTTATT	0.9861426
TTGCAGGTAGAGAATTTACCTTAGA	0.9858835
TTGGGGATGCAATGTTTATTCAATA	0.97582114
ATAAACATGCTGTTGAAGAACGTTA	0.97555196
GAAGAACGTTACCGCTTCTTTAGTT	0.9666874
TGTTGAAGAACGTTACCGCTTCTTT	0.9654274

[1] The binding site that has been validated so far is highlighted in red.

Table 9. DNA prediction score (XtrSs) by using EE.

DNA	Score
CTGTTGAAGAACGTTACCGCTTCTT	0.98748213
TATAAACATGCTGTTGAAGAACGTT [1]	0.9874292
TTTTGGGGATGCAATGTTTATTCAA	0.97213477
TTGAAGAACGTTACCGCTTCTTTAG	0.9494212

Table 9. *Cont.*

DNA	Score
AGAACGTTACCGCTTCTTTAGTTTT	0.9382387
GCCTATAAACATGCTGTTGAAGAAC	0.9351411
TAGTTTTGGGGATGCAATGTTTATT	0.92027825
ATAAACATGCTGTTGAAGAACGTTA	0.914299
TGTTGAAGAACGTTACCGCTTCTTT	0.9132359
TCTTTAGTTTTGGGGATGCAATGTT	0.9065796

[1] The binding site that has been validated so far is highlighted in red.

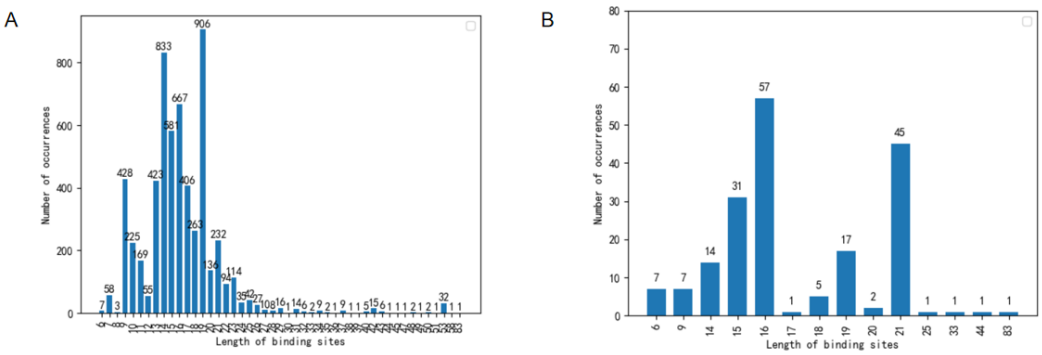


Figure 7. Binding site length distribution map. (**A**) Distribution of all binding site lengths. (**B**) Streptococcus binding site length distribution.

Table 10. DNA prediction score (ArgR) by using RS.

DNA	Score
TATTTATAAATAATTATCGATTATA	0.9995259
ATTATGCATAATTCTTTCTATTATA [1]	0.9994281
TGTGAATTTTTATTTTTTTATAAAA	0.999033
TATCATTATGTGAATTTTTATTTTT	0.9986633
TTATGTGAATTTTTATTTTTTTATA	0.99777675
CTATTATATCATTATGTGAATTTTT	0.9971888
GCATAATTCTTTCTATTATATCATT	0.99718213
CATTATGTGAATTTTTATTTTTTTA	0.99665123
TCATTATGTGAATTTTTATTTTTTT	0.99468386
ATCATTATGTGAATTTTTATTTTTT	0.99411714

[1] The binding site that has been validated so far is highlighted in red.

Table 11. DNA prediction score (ArgR) by using EE.

DNA	Score
TATTTATAAATAATTATCGATTATA	0.99634826
TGTGAATTTTTATTTTTTTATAAAA	0.9824623
TATCATTATGTGAATTTTTATTTTT	0.97935367
TATTATGCATAATTCTTTCTATTAT	0.96274704
CATTATGTGAATTTTTATTTTTTTA	0.96082777
AATTATCGATTATATCGCCTATATT	0.95746976
CCTATATTATGCATAATTCTTTCTA	0.9566975
TTATATCATTATGTGAATTTTTATT	0.9526178
ATTATCGATTATATCGCCTATATTA	0.9460073
ATCGCCTATATTATGCATAATTCTT	0.9447763
TATATCATTATGTGAATTTTTATTT	0.92416495
ATTATGCATAATTCTTTCTATTATA [1]	0.9172764

[1] The binding site that has been validated so far is highlighted in red.

3.6. Discussion

In order to predict whether bacterial transcription factors and binding sites bind or not, we propose a model BTFBS, which takes protein sequences of transcription factors and nucleotide sequences of binding sites as inputs. BTFBS, based on deep learning, can

automatically extract features from training data and capture the internal relationship between protein and nucleotide sequences. The experimental results show that ordinal positional encoding can improve the performance of the model by adding positional information. Two negative sample sampling methods are adopted, and it is found that the choice of negative samples has an impact on the model's performance. The performance of RS is better than that of EE. However, the model trained by EE has better generalization ability than RS. One possible reason is that in RS, there is a difference between the randomly truncated DNA fragments and the binding sites, making it easier for the model to learn this difference, whereas in EE, the DNA fragments are truncated from the adjacent positions of the binding sites, which have some connection to the binding sites, making it difficult for the model to distinguish between them. It is worth noting that it is possible to target the whole genome sequence of a particular bacterium for prediction using a trained BTFBS; however, achieving this requires significant computational resources due to the sheer volume of data.

4. Conclusions

We put forward a model named BTFBS, which has two inputs: protein sequences of transcription factors and nucleotide sequences of binding sites. Our work mainly includes the following four points: (1) We provide a deep learning model with two inputs that can capture links between transcription factors and binding sites. (2) Ordinal positional encoding is introduced to add positional information, which can improve the performance of the model. (3) We use two negative sample sampling methods. For bacteria, the model performance of RS and EE compared to the shuffling method indicates that the negative samples should come from the whole genome sequences of the corresponding bacteria. (4) Our proposed method is superior to some conventional methods for predicting protein DNA binding sites.

5. Further Research

Although the results show that BTFBS has good performance, there is still room for further improvement: (1) Contrastive learning can maximize the agreement (disagreement) between similar (dissimilar) samples. Contrast learning is the process of learning a representation of input features by minimizing the output distance of samples from the same category and maximizing the output distance of samples from different categories. Therefore, by introducing contrast learning, the model may be able to better learn the differences between positive and negative samples [25]. Later, we will consider introducing contrast learning to further optimize the model performance. (2) TBCA is a new framework based on a convolution and attention mechanisms and it combines DNA sequence information and shape information to predict transcription factor binding sites. DNA shape is a comprehensive description of the morphological and functional features of DNA tertiary structure revealed by quantifying features such as helical twist (HelT), propeller twist (ProT), minor groove width (MGW), rolling, and minor groove electrostatic potential (EP), which are essential for understanding biological processes such as TFBS [26]. MultiTF, based on cross-attention network, is a multi-modal representation learning method for predicting transcription factor binding sites. With the help of the cross-attention module, DNA sequence, structural, and shape features were successfully combined to achieve interactive fusion [27]. Inspired by these models, we will consider taking the structure and shape information of DNA into account in future work to see if this can further improve the model performance. (3) We might consider studying only one type of bacteria, such as Escherichia coli. (4) Oliveira Monteiro et al. trained their deep learning model PredicTF by constructing a bacterial transcription factors database. PredicTF can realize the prediction of

bacterial transcription factors [8]. We plan to use PredicTF to predict the likely transcription factors of a bacteria and then split its whole genome sequence using sliding windows. We could use our model to generate prediction scores for predicted transcription factors and DNA fragments.

Author Contributions: Conceptualization, G.L. and T.Y.; methodology, B.J.; software, B.J.; validation, B.J.; formal analysis, B.J.; investigation, B.J., S.L., X.L., R.Z., Y.Z., Y.C., G.L. and T.Y.; resources, S.L.; data curation, B.J.; writing—original draft preparation, B.J.; writing—review and editing, B.J., S.L., X.L., R.Z., Y.Z., Y.C., G.L. and T.Y.; visualization, B.J.; supervision, G.L. and T.Y.; project administration, G.L. and T.Y.; funding acquisition, G.L. and T.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Research Funds for the Central Universities, Nanjing Agricultural University (Grant No. YDZX2024046), the Guidance Foundation of the Sanya Institute of Nanjing Agricultural University [NAUSY-MS21], the Fundamental Research Funds for the Central Universities (KYTZ2023002), and the National Natural Science Foundation of China (No. 32470199).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Codes and datasets are available at https://github.com/Vceternal/BTFBS, accessed on 1 December 2024.

Acknowledgments: All the authors would like to thank the anonymous reviewers for their very helpful suggestions and comments, which led to the improvement of our original manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

TF	Transcription factor
CNN	Convolutional neural network
RS	The first negative sample sampling method
EE	The second negative sample sampling method
BTFBS	Bacterial Transcription Factor and Binding Site
ReLU	Rectified linear unit
ACC	Accuracy
MCC	Matthews correlation coefficient

References

1. Ulrich, L.E.; Koonin, E.V.; Zhulin, I.B. One-component systems dominate signal transduction in prokaryotes. *Trends Microbiol.* **2005**, *13*, 52–56. [CrossRef] [PubMed]
2. Alipanahi, B.; Delong, A.; Weirauch, M.T.; Frey, B.J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.* **2015**, *33*, 831–838. [CrossRef]
3. Gupta, A.; Rush, A.M. Dilated convolutions for modeling long distance genomic dependencies. *bioRxiv* **2017**. [CrossRef]
4. Shen, Z.; Bao, W.; Huang, D.-S. Recurrent Neural Network for Predicting Transcription Factor Binding Sites. *Sci. Rep.* **2018**, *8*, 15270. [CrossRef] [PubMed]
5. Quang, D.; Xie, X. DanQ: A hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Res.* **2016**, *44*, e107. [CrossRef]
6. Han, K.; Shen, L.-C.; Zhu, Y.-H.; Xu, J.; Song, J.; Yu, D.-J. MAResNet: Predicting transcription factor binding sites by combining multi-scale bottom-up and top-down attention and residual network. *Brief. Bioinform.* **2022**, *23*, bbab445. [CrossRef] [PubMed]
7. Kim, G.B.; Gao, Y.; Palsson, B.O.; Lee, S.Y. DeepTFactor: A deep learning-based tool for the prediction of transcription factors. *Proc. Natl. Acad. Sci. USA* **2020**, *118*, e2021171118. [CrossRef]

8. Oliveira Monteiro, L.M.; Saraiva, J.P.; Brizola Toscan, R.; Stadler, P.F.; Silva-Rocha, R.; Nunes da Rocha, U. PredicTF: Prediction of bacterial transcription factors in complex microbial communities using deep learning. *Environ. Microbiome* **2022**, *17*, 7. [CrossRef]
9. 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–i277. [CrossRef] [PubMed]
10. Yuan, Q.; Chen, K.; Yu, Y.; Le, N.Q.K.; Chua, M.C.H. Prediction of anticancer peptides based on an ensemble model of deep learning and machine learning using ordinal positional encoding. *Brief. Bioinform.* **2023**, *24*, bbac630. [CrossRef]
11. Kılıç, S.; White, E.R.; Sagitova, D.M.; Cornish, J.P.; Erill, I. CollecTF: A database of experimentally validated transcription factor-binding sites in Bacteria. *Nucleic Acids Res.* **2014**, *42*, D156–D160. [CrossRef] [PubMed]
12. Dudek, C.-A.; Jahn, D. PRODORIC: State-of-the-art database of prokaryotic gene regulation. *Nucleic Acids Res.* **2022**, *50*, D295–D302. [CrossRef] [PubMed]
13. Khanal, J.; Lim, D.Y.; Tayara, H.; Chong, K.T. I6mA-stack: A stacking ensemble-based computational prediction of DNA N6-methyladenine (6mA) sites in the Rosaceae genome. *Genomics* **2021**, *113*, 582–592. [CrossRef]
14. Deng, X.; Lan, L.; Xiao, Y.; Kennelly, M.; Zhou, J.; Tang, X. Pseudomonas syringae Two-Component Response Regulator RhpR Regulates Promoters Carrying an Inverted Repeat Element. *Mol. Plant-Microbe Interact.* **2010**, *23*, 927–939. [CrossRef] [PubMed]
15. Tang, W.; Dai, R.; Yan, W.; Zhang, W.; Bin, Y.; Xia, E.; Xia, J. Identifying multi-functional bioactive peptide functions using multi-label deep learning. *Brief. Bioinform.* **2022**, *23*, bbab414. [CrossRef] [PubMed]
16. Bian, J.; Zhang, X.; Zhang, X.; Xu, D.; Wang, G. MCANet: Shared-weight-based MultiheadCrossAttention network for drug–target interaction prediction. *Brief. Bioinform.* **2023**, *24*, bbad082. [CrossRef] [PubMed]
17. Zhang, Y.; Hamada, M. DeepM6ASeq: Prediction and characterization of m6A-containing sequences using deep learning. *BMC Bioinform.* **2018**, *19*, 524. [CrossRef]
18. Zhang, Y.; Bao, W.; Cao, Y.; Cong, H.; Chen, B.; Chen, Y. A survey on protein–DNA-binding sites in computational biology. *Brief. Funct. Genom.* **2022**, *21*, 357–375. [CrossRef] [PubMed]
19. Zou, Q.; Dong, H.; Zhu, L.; Cronan, J.E. The Enterococcus faecalis FabT Transcription Factor Regulates Fatty Acid Biosynthesis in Response to Exogeneous Fatty Acids. *Front. Microbiol.* **2022**, *13*, 877582. [CrossRef] [PubMed]
20. Wang, Z.-Q.; Yang, Y.; Zhang, J.-Y.; Zeng, X.; Zhang, C.-C. Global translational control by the transcriptional repressor TrcR in the filamentous cyanobacterium Anabaena sp. PCC 7120. *Commun. Biol.* **2023**, *6*, 643. [CrossRef]
21. Yang, H.; Huang, X.; Zhang, X.; Zhang, X.; Xu, X.; She, F.; Wen, Y. AI-2 Induces Urease Expression Through Downregulation of Orphan Response Regulator HP1021 in Helicobacter pylori. *Front. Med.* **2022**, *9*, 790994. [CrossRef] [PubMed]
22. Zhao, J.; Li, Y.; Huang, Y.; Jin, L.; Xu, Y.; Xu, M.; Quan, C.; Chen, M. Heterologous expression of quorum sensing transcriptional regulator LitR and its function in virulence-related gene regulation in foodborne pathogen Aeromonas hydrophila. *Mol. Biol. Rep.* **2022**, *50*, 2049–2060. [CrossRef]
23. Kang, D.Y.; Kim, A.; Kim, J.N. CcpA and CodY Regulate CRISPR-Cas System of Streptococcus mutans. *Microbiol. Spectr.* **2023**, *11*, e0182623. [CrossRef]
24. Zhang, Y.; Liang, S.; Zhang, S.; Bai, Q.; Dai, L.; Wang, J.; Yao, H.; Zhang, W.; Liu, G. Streptococcal arginine deiminase system defences macrophage bactericidal effect mediated by XRE family protein XtrSs. *Virulence* **2024**, *15*, 2306719. [CrossRef] [PubMed]
25. Zhang, X.; Wei, L.; Ye, X.; Zhang, K.; Teng, S.; Li, Z.; Jin, J.; Kim, M.J.; Sakurai, T.; Cui, L.; et al. SiameseCPP: A sequence-based Siamese network to predict cell-penetrating peptides by contrastive learning. *Brief. Bioinform.* **2023**, *24*, bbac545. [CrossRef] [PubMed]
26. Wang, X.; Lian, Q.; Qu, P.; Yang, Q. TBCA: Prediction of Transcription Factor Binding Sites Using a Deep Neural Network with Lightweight Attention Mechanism. *IEEE J. Biomed. Health Inform.* **2024**, *28*, 2397–2407. [CrossRef] [PubMed]
27. Wei, Y.; Zhang, Q.; Liu, L. Predicting transcription factor binding sites by a multi-modal representation learning method based on cross-attention network. *Appl. Soft Comput.* **2024**, *166*, 112134. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.