Analytical Biochemistry 656 (2022) 114878

Contents lists available at ScienceDirect

Analytical Biochemistry

journal homepage: www.elsevier.com/locate/yabio

Improving the prediction of DNA-protein binding by integrating multi-scale dense convolutional network with fault-tolerant coding

Yu-Hang Yin [a,1], Long-Chen Shen [b,1], Yuanhao Jiang [a], Shang Gao [a,**], Jiangning Song [c,d,***], Dong-Jun Yu [b,*]

[a] School of Computer, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu, 212100, PR China
[b] School of Computer Science and Engineering, Nanjing University of Science and Technology, Xiaolingwei 200, Nanjing, 210094, PR China
[c] Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, Monash University, Melbourne, VIC, 3800, Australia
[d] Monash Data Futures Institute, Monash University, Melbourne, VIC, 3800, Australia

ARTICLE INFO

Keywords:
DNA-Protein binding
Fault-tolerant coding
Dense connectional network
Multi-scale convolution
Sequence analysis

ABSTRACT

Accurate prediction of DNA-protein binding (DPB) is of great biological significance for studying the regulatory mechanism of gene expression. In recent years, with the rapid development of deep learning techniques, advanced deep neural networks have been introduced into the field and shown to significantly improve the prediction performance of DPB. However, these methods are primarily based on the DNA sequences measured by the ChIP-seq technology, failing to consider the possible partial variations of the motif sequences and errors of the sequencing technology itself. To address this, we propose a novel computational method, termed MSDenseNet, which combines a new fault-tolerant coding (FTC) scheme with the dense connectional deep neural networks. Three important factors can be attributed to the success of MSDenseNet: First, MSDenseNet utilizes a powerful feature representation approach, which transforms the raw DNA sequence into fusion coding using the fault-tolerant feature sequence; Second, in terms of network structure, MSDenseNet uses a multi-scale convolution within the dense layer and the multi-scale convolution preceding the dense block. This is shown to be able to significantly improve the network performance and accelerate the network convergence speed, and third, building upon the advanced deep neural network, MSDenseNet is capable of effectively mining the hidden complex relationship between the internal attributes of fusion sequence features to enhance the prediction of DPB. Benchmarking experiments on 690 ChIP-seq datasets show that MSDenseNet achieves an average AUC of 0.933 and outperforms the state-of-the-art method. The source code of MSDenseNet is available at https://github.com/csbio-njust-edu/msdensenet. The results show that MSDenseNet can effectively predict DPB. We anticipate that MSDenseNet will be exploited as a powerful tool to facilitate a more exhaustive understanding of DNA-binding proteins and help toward their functional characterization.

1. Introduction

Proteins that can bind to specific nucleotide sequences in the upstream of a gene are called transcription factors (TFs). Transcription factor binding site (TFBS) refers to a DNA fragment that binds to specific TFs. It is called motif, which is often located in the upstream region of the gene. The length of motif is generally in the range of 4-30bp [1–3], it usually regulates multiple genes at the same time. To some extent, its binding sites on different genes are conservative, but not identical [4,5]. Therefore, they often appear in similar forms, but some variation is allowed. TFBS interact with TFs to regulate the transcription process of genes. These binding regions in the recognition sequence, namely TFBS recognition, play a key role in gene regulation and biomolecular function [6,7]. The accurate identification of TFBS also provides technical

* Corresponding author. School of Computer Science and Engineering, Nanjing University of Science and Technology, PR China.
** Corresponding author. School of Computer, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu, 212100, PR China.
*** Corresponding author. Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, Monash University, Melbourne, Victoria, 3800, Australia.
E-mail addresses: gao_shang@just.edu.cn (S. Gao), jiangning.song@monash.edu (J. Song), njyudj@njust.edu.cn (D.-J. Yu).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.ab.2022.114878
Received 20 July 2022; Received in revised form 18 August 2022; Accepted 23 August 2022
Available online 29 August 2022
0003-2697/

support for analyzing protein function, discovering diseases and designing new drugs [8,9]. With the development of high-throughput sequencing technology [10], such as ChIP-seq [11], ChIP-exo [12] and ChIP-nexus [13], a great amount of experimentally verified TFBS has been accumulated. The sequencing cost has decreased significantly compared with the past, and various biological data have increased explosively, including high-quality TFBS experimental data such as TRANSFAC [14] and Jaspar [15]. High throughput sequencing technology has laid the foundation of "big data" in bioinformatics. Facing the tide of gene sequence data, it has brought great challenges to the research of follow-up genome analysis methods and the development of tools. In recent years, with the rapid development of bioinformatics technology, more and more experts in the field of computer and mathematics have joined the team of bioinformatics research, and many calculation methods have been applied to the task of identifying DPB [16–24].

Using traditional machine learning technology, researchers have developed many methods for predicting DPB. For example, Nitin et al. [22] used support vector machine (SVM) to combine different features to build a model that can recognize DNA binding proteins. Wong et al. [23] combined Hidden Markov Model (HMM) and belief propagation to predict DPB. Ghandi et al. [24] used gaped k-mers and support vector machine, developed an efficient tree data structure for calculating kernel matrix to predict DNA-binding sites. However, with the development of next-generation high-throughput DNA sequencing technology, DNA sequences are amplified in large quantities. Traditional machine learning algorithms cannot meet the current needs in efficiency and precision because they rely on artificial feature extraction.

In recent years, with the continuous development of deep learning technology, there have been many breakthroughs in computer vision [25,26] and natural language processing [27,28]. Because of their efficient performance, scientists studying bioinformatics and computational biology also use these advanced deep learning technologies to solve many related problems [29–31]. For example, in the DPB problem, the deep learning methods [16,17,20] have achieved better results than the traditional machine learning method. Alipanahi et al. [16] pioneering developed a deep convolution neural network model called DeepBind, which can be used to predict the sequence specificity of DNA and RNA binding proteins. Zeng et al. [17] determined the architecture with the best performance by changing the width, depth and pool design of CNN, and discussed the method of matching the CNN architecture with a given task. Luo et al. [18] effectively combined probabilistic model with CNN to improve DPB prediction. However, the nucleotide dependence and different binding length of different TFs can affect the prediction effect. HOCNN [32] used high-order coding method to establish high-order dependence between nucleotides, and used multi-scale convolution layer to capture motif features of different lengths. Du et al. [33] further considered the complementarity of DNA sequences, proposed a method to fuse different sequence features, and systematically analyzed them through multi-scale CNN. The above algorithms are developed based on CNN. While they have achieved good performance, they are also limited by the characteristics of convolution operation, that is, convolution can only focus on the extraction of local information. Such characteristics make it have obvious defects in processing long sequences. KEGRU [34] constructed a deep bidirectional Gated Recursive Unit (GRU) model for feature learning and classification. This method identifies TFBS by combining bidirectional GRU with k-mer embedding. Since then, some researchers combined the respective advantages of CNN and RNN to design hybrid models such as DeepSite [35], DeepTF [36] and DeepRAM [37] to predict DPB. Zhang et al. [38] proposed a novel motif discovery method, namely FCNA which incorporates a FCN, a global average pooling, and a hard negative mining loss. In contrast to predicting sequence specificity for DPB (i.e. sequence-level binary classification task), FCNA achieves localization of TFBS and prediction of DPB motifs at the nucleotide-level. He et al. [39] analyzed and compared some deep learning methods, and experiments proved that more complex models tended to perform better than simpler models on large-scale datasets. Shen et al. [20] recently proposed a deep learning method called SAResNet, which combines self-attention and residual structure, and uses the method of transfer learning to predict DPB from DNA sequence. After that, they further proposed MAResNet [21], which combines multi-scale bottom-up and top-down attention with residual networks to further improve the prediction performance. Although these advanced deep learning methods have achieved excellent results, they suffer from the following two drawbacks: First, in the feature representation stage, most of them only used the 4 one-hot vectors to encode four independent nucleotides, ignoring the dependence between two adjacent nucleotides. However, the high-order dependences among nucleotides within TFBS can not only improve the discriminative capability, but also produce better motif representation [40–42]. Furthermore, such methods also ignored the possible partial variation of nucleotide sequence [43] and the error of sequencing technology itself; Second, in the model design stage, most methods only considered using a fixed motif length to capture the binding features in the genome sequence, which is inadequate and might lose important contextual information. To address this, we can design and add multi-scale convolution layers to the network structure to improve the prediction accuracy and robustness of the model.

In this study, we propose a Multi-Scale Dense Convolutional Network-based approach, termed MSDenseNet, for improving the prediction of DPB. An important hallmark of MSDenseNet is that it combines the raw DNA sequence with the fault-tolerant feature sequence for fusion encoding, and in this manner, it can better integrate the high-order dependence between the nucleotides and multi-scale motif features into the original DenseNet [26]. Extensive benchmarking experiments show that compared with the most advanced methods, our developed model can achieve the best predictive performance with an average AUC value of 0.933 on 690 ChIP-seq datasets. To facilitate the community-wide exploration of this new method, we have built an online webserver of MSDenseNet at http://csbio.njust.edu.cn/bioinf/msdensenet. In addition, we have also released the source code of MSDenseNet at https://github.com/csbio-njust-edu/msdensenet.

2. Materials and methods

2.1. Benchmark datasets

In this study, we used the 690 ChIP-seq experimental datasets provided by the Encyclopedia of DNA Elements (ENCODE) project [44]. The 690 ChIP-seq datasets covered the DNA sequences of 91 human cell types bound to 161 unique regulatory factors, some of which were under various treatment conditions. For each of the 690 ChIP-seq datasets, Zeng et al. [17] divided it into the corresponding training subsets and testing subsets, in which the training subsets accounted for 80% while the testing subsets accounted for 20%, respectively. Each training subset and testing subset includes a positive subset and a negative subset respectively. The positive subset consists of the centering 101 bp region of each ChIP-seq peak, and the negative subset consists of shuffled positive sequences with matching dinucleotide composition. The 'fasta-dinucleotide-shuffle' package in MEME [45] was used for shuffling. These datasets can be downloaded at http://cnn.csail.mit.edu/motif_discovery/.

In order to meet the pre-training requirements for transfer learning, Shen et al. [20] constructed a set of global datasets based on 690 ChIP-seq datasets. They used the under-sampling strategy to construct 4, 614,580 training sequences based on 690 training subsets and 800,000 testing sequences based on 690 testing subsets. This partition could effectively ensure the independence of training and testing subsets. 4, 614,580 training sequences were divided into a global training set and a global validation set in a ratio of 9:1, and 800,000 testing sequences were valued as a global testing set. In addition, they combined training and testing sets of several typical cell lines based on 690 ChIP-seq

Table 1
A statistical summary of the five benchmark datasets.

Dataset	Subset	Number of positive samples	Number of negative samples	Total number of samples
global datasets	global-TR[a]/global-VL[b] (90%/10%)	2,307,290	2,307,290	4,614,580
	global-TS[c]	400,000	400,000	800,000
A549	TR[d]/VL[e] (80%/20%)	459,740	459,472	919,212
	TS[f]	114,777	115,045	229,822
H1-hESC	TR/VL (80%/20%)	607,774	608,088	1,215,862
	TS	152,155	151,841	303,996
HUVEC	TR/VL (80%/20%)	255,931	255,812	511,743
	TS	63,912	64,031	127,943
MCF-7	TR/VL (80%/20%)	433,823	434,368	868,191
	TS	108,801	108,256	217,057

[a,b] global-TR and global-VL denote the training set and validation set of the global datasets, respectively.
[c] global-TS denotes the testing set of the global datasets.
[d,e] TR and VL denote the training set and validation set of the related dataset, respectively.
[f] TS denotes the testing set of the relevant dataset.

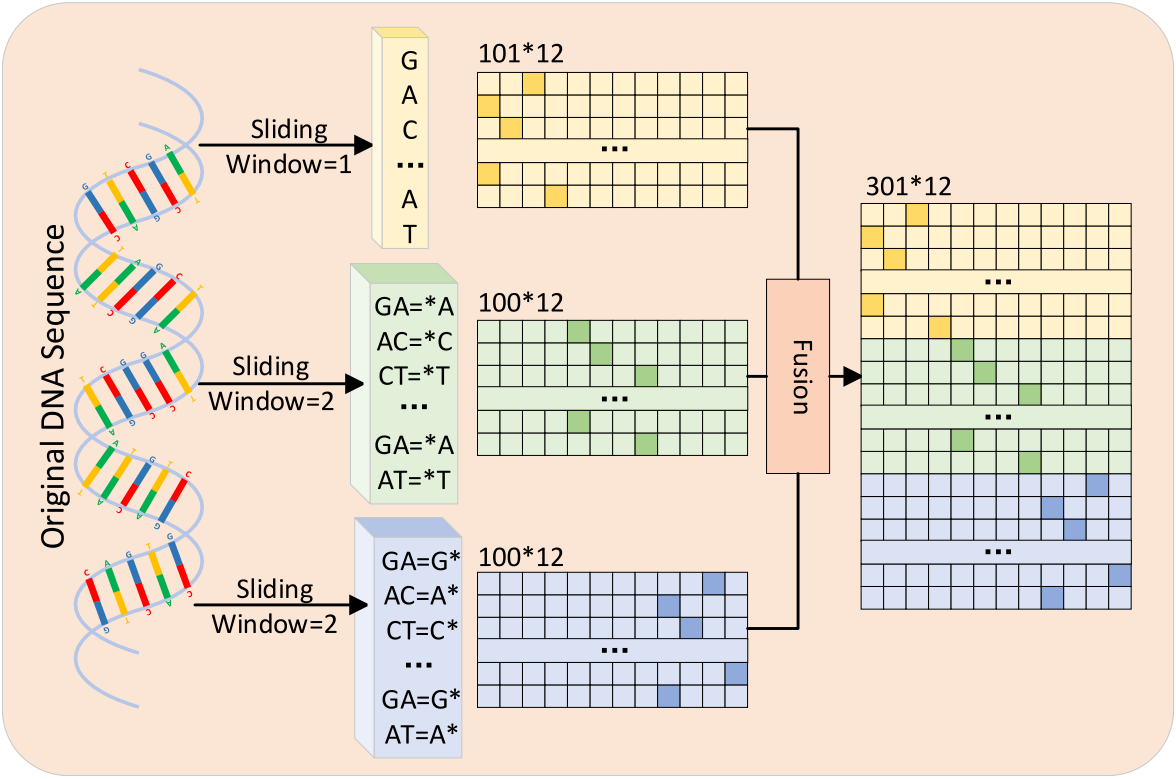

Fig. 1. Graphical illustration of the coding mechanism of FTC.

datasets, respectively, and split them into training and testing data in a ratio of 8:2. The training data was further divided into training set and validation set in a ratio of 8:2. Four cell line datasets were generated, namely A549, H1-hESC, HUVEC and MCF-7 [21]. During the construction of these datasets, the 'cd-hit-est-2d' [46] tool was used to remove the sequence redundancy and ensure the independence of the testing set. All these datasets are publicly available at http://csbio.njust.edu.cn/bioinf/maresnet/. A statistical summary of these five benchmark datasets is provided in Table 1.

2.2. Feature representation

Different from other methods that used the raw DNA sequence for feature coding directly, the input of MSDenseNet included the possible partial variation of nucleotide sequence [43] and the error of sequencing technology to a certain extent. The DNA sequence was composed of four different bases [A, C, G, T]. For two adjacent bases, considering the possibility of variation (or sequencing error), we proposed a new sequence encoding scheme, termed Fault-Tolerant Coding (FTC), in order to encode more informative features.

2.2.1. Fault-tolerant coding

Specifically, for a given raw DNA sequence, we scanned it using sliding windows of sizes 1 and 2, respectively. After scanning the sequence with a sliding window of size 1, a sequence Seq_1 of length L (L = 101bp) consisting of the $Alphabet_1$ = {A, C, G, T} was obtained. After scanning the sequence with a sliding window of size 2, a sequence Seq_2 of length L-1 was obtained, which consisted of the $Alphabet_2$ = {AA, AC, AG, AT, CA, CC, CG, CT, GA, GC, GG, GT, TA, TC, TG, TT}. Considering the possible partial variation of the nucleotide sequence [43], we used the wildcard "*" to represent the possible variation of nucleotides, that is, "*" represents any one of A, C, G, or T. For example, we treated the dinucleotides "AA", "CA", "GA", and "TA" as "*A", which represents a mutation at the position 1 in the dinucleotide. Similarly, if the position 2

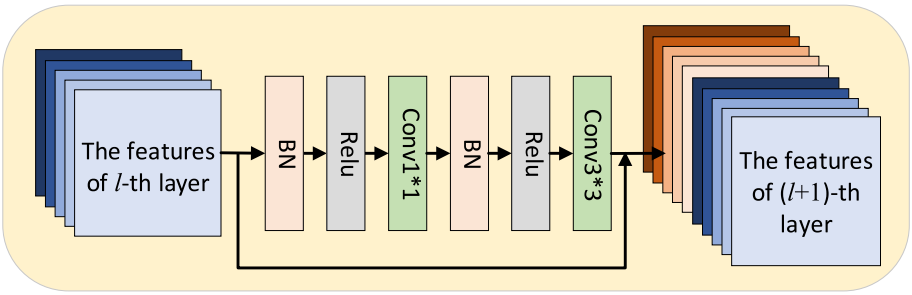

Fig. 2. The structure of the dense layer.

in the dinucleotide was mutated, then "AA", "AC", "AG", and "AT" would be regarded as "A*". According to the sequence Seq_2 composed of dinucleotides, we considered the possibility of mutation at the positions 1 and 2, respectively, and accordingly generated the sequence Seq_3 consisting of the $Alphabet_3$ = {*A, *C, *G, *T} and the sequence Seq_4 consisting of the $Alphabet_4$ = {A*, C*, G*, T*}. Finally, we added the sequences Seq_1, Seq_3 and Seq_4 to generate a fault-tolerant sequence Seq of length 3L-2, consisting of the $Alphabet$ = {A, C, G, T, *A, *C, *G, *T, A*, C*, G*, T*}. Fig. 1 provides a graphical illustration of the coding mechanism of our proposed FTC encoding method. In Fig. 1, each alphabet in the sequence Seq is encoded as a feature vector of size 12, that is, A→ [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], …, T*→[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]. The input matrix $S_{i,j}$ of the deep network can be encoded by the following equation:

$$S_{i,j} = \begin{cases} 1, & if \ \ Seq_i = j^{th} \ \ base \ \ in \ \ Alphabet \\ 0, & otherwise \end{cases} \quad (1)$$

where i∈[1, 301], j∈ [1,12].

2.3. Model architecture and training procedures

In this study, we processed the input feature of neural network as a 12-channel image feature with the length of 301. In this way, the DPB prediction task in bioinformatics can be transformed as an image binary classification problem in computer vision. A number of advanced deep learning algorithms [18,20,37] have been applied to this field and shown excellent performance. In recent years, in order to further improve the model performance in real-world scenarios, convolutional neural networks have been designed with a more complex deep structure [25,26]. Use of multiple convolution kernels to extract richer features has also been widely used in the fields of computer vision [47].

2.4. The structure of dense block and multi-scale convolution

Huang et al. proposed DenseNet [26] in 2017 by leveraging the advantages of ResNet [25]. Compared with residual block of ResNet, Huang et al. creatively proposed dense block. Each dense block consists of multiple dense layers. Fig. 2 shows the structure of the dense layer. As can be seen, there exists a direct connection between any two dense layers, that is to say, the input of each layer of the network is the set of the outputs of all preceding layers; further, the features learned by this layer will also be directly passed as the input to all the subsequent layers. Through such a dense connection structure, sufficient reuse of features can be achieved, and accordingly, the number of parameters is reduced to a certain extent. In this way, the problem of gradient disappearance is effectively alleviated. The dense connections are given by the following equation:

$$x_l = H_l([x_0, x_1, \ldots, x_{l-1}]) \quad (2)$$

where x_l is the output of the l-th layer, [x_0, x_1, …, x_{l-1}] is the splicing of the characteristic diagrams generated by each layer, while $H_l(\cdot)$ denotes the nonlinear conversion function.

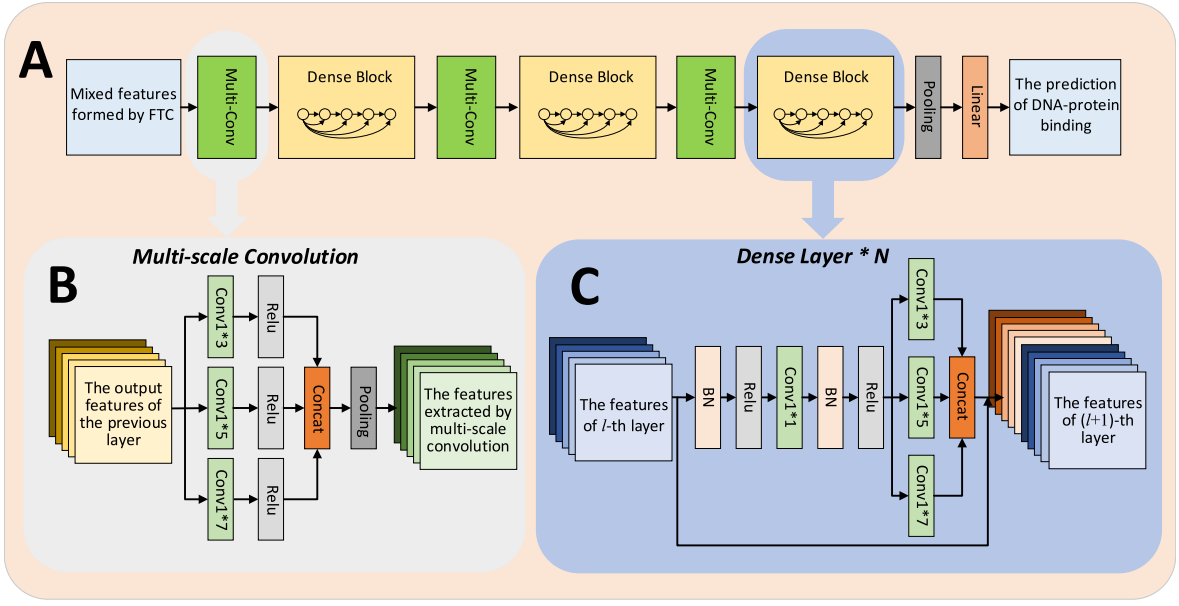

Fig. 3. The network structure of MSDenseNet: (A) The overall structure of the network. (B) The multi-scale convolution network preceding the dense block. (C) The multi-scale convolution network within the dense layer, where N denotes the number of the dense layers.

Considering that different TFs have different binding lengths [48, 49], it is difficult to fully capture the binding characteristics in genomic sequences using fixed motif lengths. In this work, refer to the Inception [47] module in computer vision and the HOCNN [32], we added a series of fixed filters (convolution kernels) of different sizes to the original DenseNet [26] to capture multi-scale features. However, in order to balance the performance and computational complexity of the model, we chose convolution kernels with the sizes of 3, 5 and 7, respectively. Fig. 3B and C shows the detailed structure of multi-scale convolution before the dense block and within the dense layer, respectively. We can get the output X through the convolution of kernel M and the vectors S. The multi-scale convolution can be realized by the following equation:

$$X_{i,k} = \underset{m\in\Phi}{Conactenate}\left(\max\left(0, \sum_{j=1}^{m}\sum_{c=1}^{12} S_{i+j,c}M_{k,j,c} + b_k\right)\right) \quad (3)$$

where $i\in[1, l]$, l represents the length of the input sequence, $k\in[1, d]$, d represents the number of the convolution kernels, Φ represents a set of convolution kernels sizes, m represents the size of the convolution kernel, c represents the number of channels, b_k represents the bias term, respectively.

The added multi-scale convolution is able to extract informative features of different scales, thereby providing richer features. Theoretically, the more diverse the features, the better the predictive performance of the model. In addition, the convolution kernels of different sizes were used for feature extraction, and then the features with strong correlation were gathered together. In this way, the convolution of each size only output a part of all features, which can gather the features with strong correlation in advance to accelerate the convergence. At the same time, compared with the sparse feature set output by a single convolution kernel, the output of multi-scale convolution can be exploited as multiple densely distributed feature subsets. In light of the principle of decomposing sparse matrix into dense matrix [50], the convergence speed of the model can be accelerated.

2.5. Implementation of the neural network architecture of MSDenseNet

Multi-scale convolution is capable of capturing rich features to improve the predictive performance of the model and accelerate its convergence. Therefore, we proposed and implemented the MSDenseNet pipeline. Fig. 3 shows the main structure of MSDenseNet. Its structure contains three dense block modules, each of which comprises a different number of dense layers. The numbers of the dense layers contained in the three dense blocks were 6, 12, and 8, respectively. The channel hyperparameters were set to 96 while the channel growth rate was set to 16, respectively.

Firstly, we used the FTC encoded DNA sequence as the input of the network. Then, we applied the three convolution kernels of sizes 1×3, 1×5, and 1×7, respectively for feature extraction. After that, we used the ReLU activation function, spliced the three branches and then applied the max-pooling layer for further down sampling. Each dense block consisted of multiple dense layers. In this work, we used three dense blocks, each of which contained 6, 12 and 8 dense layers respectively. In each dense layer, the output of the previous layer first passed through the batch normalization layer and the ReLU layer respectively. Secondly, through 1×1 convolution, it cannot only reduce the dimension and reduce the amount of calculation, but also integrate the characteristics of each channel. Again, through the batch normalization layer and ReLU layer, three convolution layers with the sizes of 1×3, 1×5 and 1×7 were followed. Finally, we combined the extracted features of the three convolution kernels with the original input features to form the output of this layer. In addition, between each dense block, we used the batch normalization and ReLU layers, and also used 1×3, 1×5 and 1×7 multi-scale convolution layers to extract the features. Afterwards, we used the average pooling layer to reduce the number of parameters and save the computing power, which also helped control the overfitting [51] to a certain extent. At the end of the last dense block, we employed the global average pooling to regularize the structure of the whole network to prevent overfitting, and then connected the softmax classifier to generate the probability distribution of the two tags.

Table 2
The hyper-parameter settings of MSDenseNet.

Hyper-parameter	Choice	Sampling
dropout ratio (dense layer)	0.1, 0.2, 0.3	all evaluation
kernel size	$3\oplus5\oplus7$[a]	Fixed
learning rate (pre-training)	1×10^{-3}, 2×10^{-3}, 3×10^{-3}	all evaluation
learning rate (transfer learning)	2×10^{-4}, 4×10^{-4}, 6×10^{-4}	all evaluation
batch size (pre-training)	32, 64, 128	all evaluation
batch size (transfer learning)	32, 64, 128	all evaluation
optimizer	SGD	Fixed
loss	softmax cross entropy	Fixed

[a] $3\oplus5\oplus7$ denotes the concatenation of the three kernel sizes.

2.6. Hyper-parameter settings

We implemented MSDenseNet using PyTorch (v1.8.1) [52] and conducted the experiments on the computing resources of 1/4 NVIDIA Tesla A100 Graphics Card. During our experiments, the softmax cross entropy function and SGD method [53] were used to optimize the model. The detailed settings of the model's hyperparameters are listed in Table 2. First, we searched a group of hyper-parameters that could ensure a high performance of the model by enumerating all the possible values of each hyper-parameter listed in Table 2 on the datasets of A549, H1-hESC, HUVEC and MCF-7 cell lines. Then, we applied such a set of hyper-parameters to the training global dataset to construct an excellent pre-training model. Finally, we applied the pre-training model to perform transfer learning on 690 ChIP-seq datasets and evaluated the performance of the model in each of the datasets.

2.7. Assessing predictive ability

DPB prediction is a typical binary classification problem. As such, the metrics used to evaluate the classification performance of binary classes are also suitable for evaluating the prediction performance of DPB. In this study, we used accuracy, precision, recall and F1 score as the main performance metrics to evaluate the performance of different predictors. These metrics are defined as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (7)$$

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

However, with the change of the prediction cut-off threshold, the values of the above four evaluation indicators will also accordingly change. For example, assume that the predicted protein binding probability is 0.67. If the cut-off threshold is 0.5, the sequence will be identified as containing TFBS (i.e., the label is 1), and if the cut-off threshold is 0.7, the sequence will be identified as not containing TFBS (i.e., the label is 0). Therefore, a metric that does not change with the cut-off threshold and can still measure the prediction performance is needed to evaluate the predictors. The area under receiver operating characteristic (ROC) curve (AUC) meets such requirements, which is used as another major performance index in this study. The value of AUC

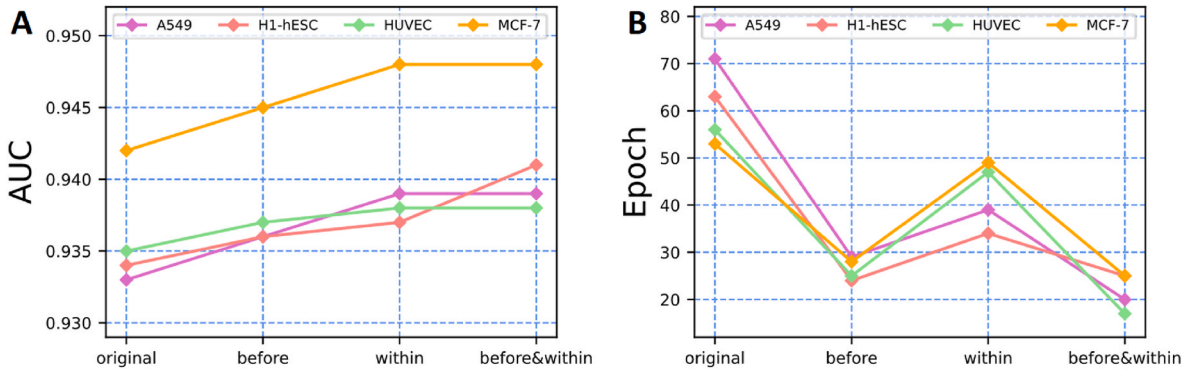

Fig. 4. Experimental results of the networks with different structures on the A549, H1-hESC, HUVEC and MCF-7 datasets. (A) displays the AUC values of the networks trained with different structures. (B) displays the training epochs required by training networks with different structures. 'original' indicates that the original DenseNet is used, 'before' indicates that multi-scale convolution is added before the dense block, 'within' indicates that multi-scale convolution is added within the dense layer, and 'before&within' indicates that multi-scale convolution is added before the dense block and within the dense layer concurrently.

Table 3
Experimental results of ablation comparisons.[a]

Dataset	Structure	Epoch[b]	Accuracy	Precision	Recall	F1 score	AUC
A590	original[c]	71	0.833	0.940	0.711	0.810	0.933
	before[d]	29	0.860	0.894	0.817	0.853	0.936
	within[e]	39	0.826	0.957	0.681	0.796	**0.939**
	before&within[f]	**20**	0.837	0.785	0.928	0.850	**0.939**
H1hesc	original	63	0.859	0.880	0.831	0.855	0.934
	before	**24**	0.862	0.888	0.828	0.857	0.936
	within	34	0.848	0.933	0.749	0.831	0.937
	before&within	25	0.834	0.780	0.933	0.849	**0.941**
Huvec	original	56	0.801	0.968	0.623	0.758	0.935
	before	25	0.859	0.846	0.877	0.861	0.937
	within	47	0.848	0.935	0.747	0.830	**0.938**
	before&within	**17**	0.860	0.913	0.796	0.850	**0.938**
Mcf7	original	53	0.867	0.908	0.818	0.861	0.942
	before	28	0.873	0.886	0.856	0.871	0.945
	within	49	0.862	0.948	0.768	0.848	**0.948**
	before&within	**25**	0.867	0.842	0.906	0.873	**0.948**

[a] FTC was used in the ablation experiments for network structure.
[b] Terminate the training when the AUC of the validation set dropped for ten consecutive times, record the best AUC and epoch.
[c] 'original' indicates that the original DenseNet is used.
[d] 'before' indicates that multi-scale convolution is added before the dense block.
[e] 'within' indicates that multi-scale convolution is added within the dense layer.
[f] 'before&within' indicates that multi-scale convolution is added before the dense block and within the dense layer concurrently.

is between 0 and 1. The closer its value is to 1, the better the predictive performance of the model.

3. Results and discussion

3.1. Multi-scale convolution improves the model performance

In this study, we added the multi-scale convolutions (1×3, 1×5, 1×7) within the dense layers and before each dense block to extract the features. Such operation will extract features of different scales, which makes the features more abundant and also means that the prediction performance of the model might be improved. We compared the performance of different network structures on four different cell line datasets.

3.1.1. Ablation experiments

Various structures in MSDenseNet were investigated to provide insights into MSDenseNet's performance. We have evaluated four main structures, i.e., the original DenseNet, adding multi-scale convolution before the dense block, adding multi-scale convolution within the dense layer, and adding multi-scale convolution before the dense block and within the dense layer at the same time. Fig. 4A shows the performance (AUC) of different structural networks on four cell line datasets in the form of a line chart. From Table 3, we can see that adding the multi-scale convolution within the dense layer achieved the better performance compared with the original DenseNet, across all four cell line datasets under the evaluation of the performance metric AUC.

However, due to its densely connected structure, the output of each layer was used as the input of the subsequent layers. As a consequence, the computational complexity would be considerably increased, with the model becoming complex and the convergence speed not being improved. To solve this problem, we added a structure similar to Inception [47] before each dense block module, and applied the principle of decomposing sparse matrix into dense matrix calculation, thereby making the convergence speed of the model much faster. The input data of the traditional convolution layer is only convolved with a convolution kernel of one scale, and the data of a fixed dimension is the output. All the output features are basically evenly distributed in this scale range, which can be considered as the output of a sparse distributed feature set.

Nevertheless, in this study, by extracting the features at multiple scales (1×3, 1×5, 1×7), the output features were no longer uniformly distributed; instead, highly correlated features were clustered together. These can be regarded as multiple densely distributed feature subsets. In

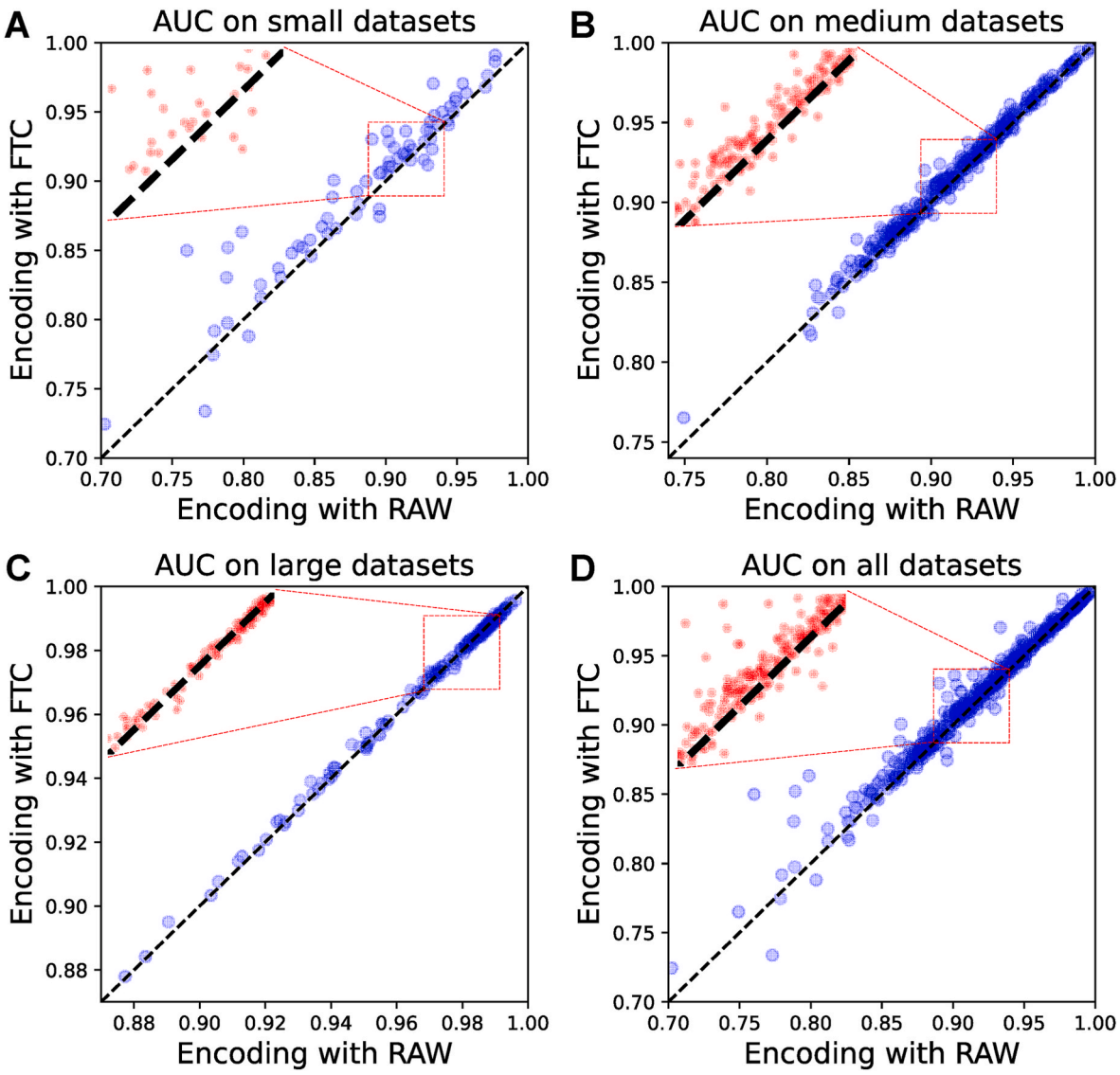

Fig. 5. Effect of feature representations on the performance of MSDenseNet in terms of the AUC score across the datasets of different scales. (A) displays the AUC on small datasets of different feature representations. (B) displays the AUC on medium datasets of different feature representations. (C) displays the AUC on large datasets of different feature representations. (D) displays the AUC on all 690 ChIP-seq datasets of different feature representations.

such a feature set, because the features with strong correlation are clustered together, the irrelevant features are weakened, and accordingly the features output by such a method tend to have less redundant information. Using such a feature set to pass on to the next layer and finally as the input to the reverse calculation, the network will converge faster. When conducting the experiments, we took the AUC of the verification set of each training epoch as the standard, and would terminate the training when the AUC of the verification set dropped for ten consecutive times. From Table 3, we can see that by adding the multi-scale convolution before each dense block, the model indeed converged significantly faster when being trained on the four cell line datasets. Meanwhile, Fig. 4B visually shows, in line chart form, the number of epochs required to achieve optimal results for different structural networks on the four cell line datasets. Combining the data in Table 3 and Fig. 4, we conclude that on the four cell line datasets, multi-scale dense connection networks with different structures can all achieve excellent performance. Among them, when the multi-scale convolution was added before the dense block and within the dense layer concurrently, the network with this structure can obtain the best performance with significantly less epochs, which is the MSDenseNet proposed by us.

Overall, the ablation experimental results showed that adding the multi-scale convolution within the dense layer could capture more features and accordingly improve the predictive performance. Meanwhile, adding the multi-scale convolution before dense block could also reduce the number of training epochs. When the multi-scale convolution was added before the dense block and within the dense layer concurrently, such a network structure could not only achieve high predictive performance, but also converge quickly.

3.2. FTC is an effective new sequence encoding method and improves the performance

In a previous study, Shen et al. [20] divided the 690 datasets into three different scale datasets, i.e. small datasets with the scale of less than 3000 samples, medium datasets with the scale between 3000 and 30,000 samples, and large datasets with the scale of greater than 30,000 samples. To investigate the effect of feature representation methods on model performance, we compared the predictive performance of models trained using raw DNA sequence and FTC on different scale datasets to

Table 4
Performance comparison of the MSDenseNet trained with different feature representation methods with respect to different dataset scales.

Dataset scale	Encoding	Accuracy	Precision	Recall	F1 score	AUC
Small	RAW[a]	0.724	0.729	0.714	0.721	0.886
	FTC[b]	0.791	0.792	0.788	0.789	**0.897**
Medium	RAW	0.833	0.847	0.814	0.830	0.918
	FTC	0.844	0.849	0.837	0.843	**0.921**
Large	RAW	0.924	0.935	0.910	0.922	**0.973**
	FTC	0.925	0.933	0.916	0.924	**0.973**
All	RAW	0.846	0.858	0.830	0.844	0.929
	FTC	0.860	0.866	0.853	0.859	**0.933**

[a] RAW denotes that the raw DNA sequence is used as the input feature.
[b] FTC denotes that the fault-tolerant coding is used as the input feature.

examine the effectiveness of the proposed FTC encoding method. When the raw DNA sequence is used, each base in it will be denoted as one of the 4 one-hot vectors (i.e. A→ [1, 0, 0, 0], C→[0, 1, 0, 0], G→[0, 0, 1, 0], T→[0, 0, 0, 1]).

In terms of the performance evaluation indicators, here we paid more attention to the AUC metric. In contrast, all other performance metrics will change with the change of the prediction cut-off threshold. Therefore, the metric AUC that does not change with the threshold can reflect the comprehensive performance of the model. In terms of AUC, Fig. 5 shows the effect of feature representations on the performance of MSDenseNet across datasets of different scales in the form of scatter diagram, where each of 690 testing subsets corresponds to a point whose X and Y coordinates indicate the AUC scores of the corresponding feature representation method. We can see that the majority of the points fell above the diagonal line, indicating that in different scale datasets, the AUC of MSDenseNet using FTC is higher than that using only the raw DNA sequence, which shows that our proposed feature representation method (FTC) can improve the model performance to a certain extent. From Table 4, we can see that in most cases, the average predictive performance of various metrics is improved by using FTC. It is worth mentioning that the various average performances on large datasets maintain very high values, and FTC brings significant improvements to the various average performances on small and medium datasets.

3.3. Performance comparison between MSDenseNet and other existing methods

The majority of existing prediction methods of DPB are developed based on the human ChIP-seq datasets from the ENCODE project. In particular, HOCNN [32], KEGRU [34], and DeepRAM [37] used 214, 125, and 83 ChIP-seq datasets from the ENCODE project, respectively, to evaluate the performance of their respective methods. Herein, to enable an objective comparative analysis, we compared with gkm-SVM [54], DeepBind [16], CNN-Zeng [17], DeepTF [36], Expectation-Lou [18], HOCNN [32], SAResNet [20] and MAResNet [21] using all the 690 ChIP-seq datasets, to ensure the integrity and fairness of the experiments.

To test and compare with the performance of the gkm-SVM method, we downloaded the gkm-SVM R package (https://cran.r-project.org/web/packages/gkmSVM) and replicated their experiments with the default parameters. The relevant experimental data of the two models, DeepBind and CNN-Zeng, were obtained from http://cnn.csail.mit.edu/. The authors of DeepTF provided its AUC results. The source code of Expectation-Luo was downloaded from https://github.com/gao-lab/ePooling, we used the default parameters to train and test each of the 690 ChIP-seq datasets. According to the description of HOCNN by authors, we reproduced their model with PyTorch and carried out experiments on 690 ChIP-seq datasets. Furthermore, the authors of SAResNet and MAResNet published their experimental data. Fig. 6 shows the performance of MSDenseNet on 690 ChIP-seq datasets in comparison with gkm-SVM, DeepBind, CNN Zeng, DeepTF, Expectation-Luo, HOCNN, SAResNet and MAResNet. From Fig. 6, it can be seen that MSDenseNet outperformed all the other methods on the 690 datasets in terms of AUC. Moreover, the median AUC of MSDenseNet was 0.937, which was better than that of the suboptimal method MAResNet with the median AUC of 0.931. In addition, we can also see that MSDenseNet

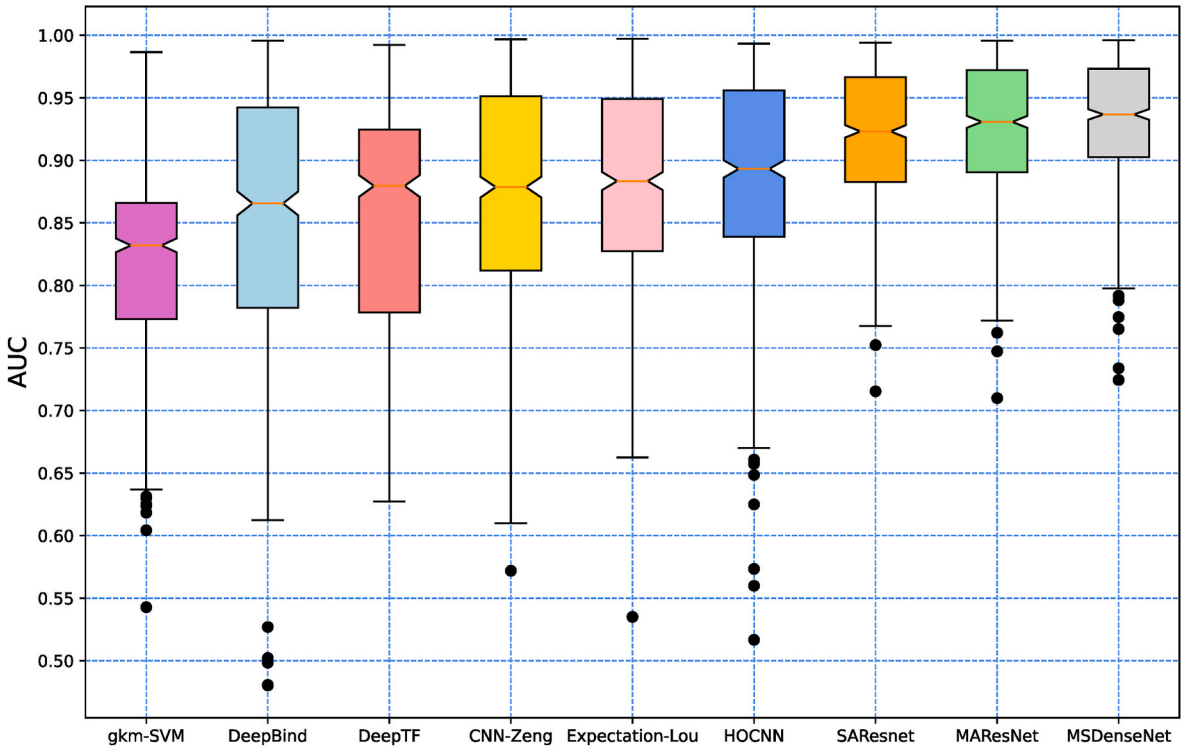

Fig. 6. Boxplots of the AUCs achieved by different methods with 690 ChIP-seq datasets.

Table 5
Performance comparison of MSDenseNet and the other existing methods in terms of AUC on different scale datasets.

Method	All datasets	Small datasets	Medium datasets	Large datasets	P-value[a]
MSDenseNet	**0.933**	**0.897**	**0.921**	**0.973**	-[b]
MAResNet	0.927	0.883	0.914	0.972	2.0×10^{-2}
SAResNet	0.920	0.876	0.907	0.966	8.4×10^{-7}
HOCNN	0.887	0.821	0.868	0.957	1.7×10^{-38}
Expectation-Luo	0.881	0.835	0.859	0.947	4.7×10^{-52}
CNN-Zeng	0.875	0.818	0.850	0.953	6.2×10^{-54}
DeepTF	0.845	0.809	0.818	0.919	4.0×10^{-100}
DeepBind	0.830	0.785	0.809	0.896	2.0×10^{-64}
gkm-SVM	0.818	0.798	0.809	0.856	4.3×10^{-208}

[a] P-value of the student's t-test was performed to evaluate the statistical differences in the AUC values between MSDenseNet and the other prediction methods.
[b] '-' denotes that the relevant value was not applicable.

improved the AUC values in both the upper and lower quartiles averaged on all 690 datasets compared with the other methods (Fig. 6), highlighting the excellent generalization ability of MSDenseNet.

We further evaluated the performance of each model on the datasets of different scales. The average AUC of each model on these three scale datasets is provided in Table 5. As can be seen, MSDenseNet achieved the best AUC score on all the datasets with different scales, which illustrates the competitiveness and robustness of MSDenseNet. By a closer inspection at the last column of Table 5, we can see that MSDenseNet also achieved a statistically significant performance improvement in terms of AUC (student's t-test, $P < 2.0 \times 10^{-2}$) compared with other methods on 690 independent testing datasets.

In addition, we presented the performance of MSDenseNet and other seven methods on all evaluation metrics on datasets of different scales in the form of bar charts. We noted that in Fig. 7A, the precision of gkm-SVM and HOCNN is relatively high, but their recall is very low. In this work, in order to fairly compare the performance of MSDenseNet with other models on datasets of different scales, we focus more on the comprehensive performance metric AUC which is not affected by the cut-off threshold. As shown in Fig. 7, we can more intuitively see that MSDenseNet outperforms other existing models on almost all evaluation metrics on datasets of various scales. Overall, these results demonstrate that our model improved the predictive performance across all datasets with different scales, which was more pronounced when tested on the small and medium datasets.

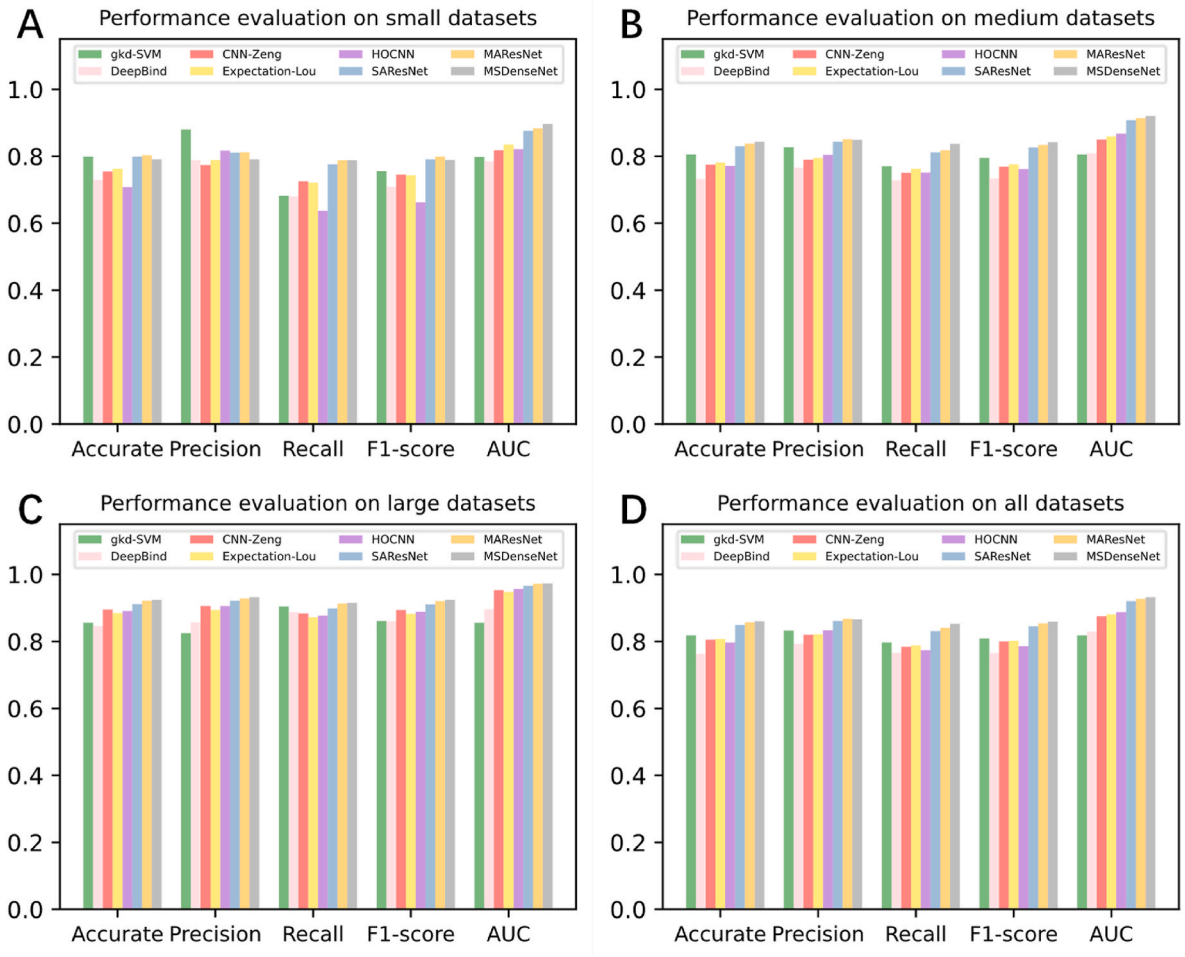

Fig. 7. Performance comparison between MSDenseNet and other methods on the testing datasets with different scales. (A) displays the performance of each model on the small datasets. (B) displays the performance of each model on the medium datasets. (C) displays the performance of each model on the large datasets. (D) shows the performance of each model on all 690 ChIP-seq datasets.

4. Conclusions

In this study, we have developed a novel deep learning method, termed MSDenseNet, which has improved sequence-based prediction of DPB. Evaluated by performing comprehensive and unbiased experiments, MSDenseNet clearly outperformed a variety of existing state-of-the-art methods on the 690 ChIP-seq datasets. Notably, our proposed MSDenseNet method achieved significant performance improvements on the small and medium datasets. Three critical factors can be attributed to its success: First, we introduced a fault-tolerance mechanism during the feature representation stage, and applied the FTC encoding method to generate the feature matrix input to the deep network, which has been shown to be able to effectively improve the predictive performance; Second, by adding the multi-scale convolution within the dense layer to extract the features of different scales, the extracted features were more abundant and informative. As such, the predictive performance of the resulting model could be further improved, and third, we leveraged a powerful neural network structure analogous to Inception before each dense block, and applied the principle of decomposing a sparse matrix into a dense matrix to accelerate the convergence of the model.

Despite the outstanding performance of MSDenseNet for DPB prediction, it has some limitations and there exist several aspects for further improvement: First, due to the restraints of time and computational resources, only the two-order FTC was used, which was shown to result in the model performance improvement. This implies that FTC performed well in terms of feature enrichment. However, the potential and impact of higher-order FTC will need to be further examined in the future work; Second, MSDenseNet is only designed for the prediction of DPB. However, with a little modification, it can be used to solve other sequence-based prediction problems in bioinformatics and computational biology [55]. For example, by performing the corresponding transformations in the feature representation stage, binding sites can be predicted from protein sequences [56–58], and other types of binding and functional sites can also be predicted [59–61], and finally, we hope to develop a suite of useful bioinformatics tools based on the MSDenseNet methodology, which should prove useful and valuable for identifying functional elements of gene regulation from the genomic sequence regions.

Data and software availability

The 690 ChIP-seq datasets [17] used in benchmarking experiments can be downloaded at http://cnn.csail.mit.edu/motif_discovery/. Four different cell line datasets [21] (i.e. A549, H1-hESC, HUVEC and MCF-7) and the global datasets [20] used in pre-training are publicly available at http://csbio.njust.edu.cn/bioinf/maresnet/.

We implemented MSDenseNet using PyTorch (v1.8.1), https://pytorch.org/. The source code of MSDenseNet has been released at https://github.com/csbio-njust-edu/msdensenet. In addition, we have built an online webserver of MSDenseNet at http://csbio.njust.edu.cn/bioinf/msdensenet.

Author contributions

Yu-Hang Yin: Designed research, Performed research, Formal analysisanalyzed data, Writing – original draftwrote the paper.

Long-Chen Shen: Designed research, Performed research, Formal analysisanalyzed data, Writing – original draftwrote the paper.

Yuanhao Jiang: Performed research, Formal analysisanalyzed data.

Shang Gao, Jiangning Song, Dong-Jun Yu: Conceptualization, Methodology, Writing – review & editing.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62072243, 61772273 and 62176107), the Natural Science Foundation of Jiangsu (BK20201304), the Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX21_3486), the National Health and Medical Research Council of Australia (NHMRC) (APP1127948 and APP1144652), the Australian Research Council (ARC) (LP110200333 and DP120104460), the National Institute of Allergy and Infectious Diseases of the National Institutes of Health (R01 AI111965) and a Major Inter-Disciplinary Research (IDR) project awarded by Monash University.

Abbreviations

DPB	DNA-protein binding
MSDenseNet	Multi-Scale Dense Convolutional Network
FTC	fault-tolerant coding
TFs	transcription factors
TFBS	transcription factor binding site
SVM	support vector machine
HMM	Hidden Markov Model
GRU	Gated Recursive Unit
ENCODE	Encyclopedia of DNA Elements
ROC	receiver operating characteristic
AUC	area under the receiver operating characteristic curve

References

[1] M. Tompa, N. Li, T.L. Bailey, G.M. Church, B. De Moor, E. Eskin, A.V. Favorov, M. C. Frith, Y. Fu, W.J. Kent, Assessing computational tools for the discovery of transcription factor binding sites, Nat. Biotechnol. 23 (2005) 137–144.
[2] G. Tan, B. Lenhard, TFBSTools: an R/bioconductor package for transcription factor binding site analysis, Bioinformatics 32 (2016) 1555–1556.
[3] K. Qu, L. Wei, Q. Zou, A review of DNA-binding proteins prediction methods, Curr. Bioinf. 14 (2019) 246–254.
[4] S.G. Kuntz, B.A. Williams, P.W. Sternberg, B.J. Wold, Transcription factor redundancy and tissue-specific regulation: evidence from functional and physical network connectivity, Genome Res. 22 (2012) 1907–1919.
[5] M. Kumar, M.M. Gromiha, G.P. Raghava, Identification of DNA-binding proteins using support vector machines and evolutionary profiles, BMC Bioinf. 8 (2007) 1–10.
[6] K.A. Aeling, N.R. Steffen, M. Johnson, G.W. Hatfield, R.H. Lathrop, D.F. Senear, DNA deformation energy as an indirect recognition mechanism in protein-DNA interactions, IEEE ACM Trans. Comput. Biol. Bioinf 4 (2007) 117–125.
[7] J.M. Gualberto, K. Kühn, DNA-binding proteins in plant mitochondria: implications for transcription, Mitochondrion 19 (2014) 323–328.
[8] P. Schmidtke, X. Barril, Understanding and predicting druggability. A high-throughput method for detection of drug binding sites, J. Med. Chem. 53 (2010) 5858–5867.
[9] D.J. Smyth, V. Plagnol, N.M. Walker, J.D. Cooper, K. Downes, J.H. Yang, J. M. Howson, H. Stevens, R. McManus, C. Wijmenga, Shared and distinct genetic variants in type 1 diabetes and celiac disease, N. Engl. J. Med. 359 (2008) 2767–2777.
[10] J. Shendure, H. Ji, Next-generation DNA sequencing, Nat. Biotechnol. 26 (2008) 1135–1145.
[11] T.S. Furey, ChIP–seq and beyond: new and improved methodologies to detect and characterize protein–DNA interactions, Nat. Rev. Genet. 13 (2012) 840–852.
[12] L. Wang, J. Chen, C. Wang, L. Uusküla-Reimand, K. Chen, A. Medina-Rivera, E. J. Young, M.T. Zimmermann, H. Yan, Z. Sun, MACE: model based analysis of ChIP-exo, Nucleic Acids Res. 42 (2014) e156-e156.
[13] Q. He, J. Johnston, J. Zeitlinger, ChIP-nexus enables improved detection of in vivo transcription factor binding footprints, Nat. Biotechnol. 33 (2015) 395–401.
[14] V. Matys, O.V. Kel-Margoulis, E. Fricke, I. Liebich, S. Land, A. Barre-Dirrie, I. Reuter, D. Chekmenev, M. Krull, K. Hornischer, TRANSFAC® and its module TRANSCompel®: transcriptional gene regulation in eukaryotes, Nucleic Acids Res. 34 (2006) D108–D110.
[15] O. Fornes, J.A. Castro-Mondragon, A. Khan, R. Van der Lee, X. Zhang, P. A. Richmond, B.P. Modi, S. Correard, M. Gheorghe, D. Baranašić, JASPAR 2020: update of the open-access database of transcription factor binding profiles, Nucleic Acids Res. 48 (2020) D87–D92.

[16] B. Alipanahi, A. Delong, M.T. Weirauch, B.J. Frey, Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning, Nat. Biotechnol. 33 (2015) 831–838.
[17] H. Zeng, M.D. Edwards, G. Liu, D.K. Gifford, Convolutional neural network architectures for predicting DNA–protein binding, Bioinformatics 32 (2016) i121–i127.
[18] X. Luo, X. Tu, Y. Ding, G. Gao, M. Deng, Expectation pooling: an effective and interpretable pooling method for predicting DNA–protein binding, Bioinformatics 36 (2020) 1405–1412.
[19] D. Quang, X. Xie, DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences, Nucleic Acids Res. 44 (2016) e107-e107.
[20] L.-C. Shen, Y. Liu, J. Song, D.-J. Yu, SAResNet: self-attention residual network for predicting DNA-protein binding, Briefings Bioinf. 22 (2021) bbab101.
[21] K. Han, L.-C. Shen, Y.-H. Zhu, J. Xu, J. Song, D.-J. Yu, MAResNet: predicting transcription factor binding sites by combining multi-scale bottom-up and top-down attention and residual network, Briefings Bioinf. 23 (2022) bbab445.
[22] N. Bhardwaj, R.E. Langlois, G. Zhao, H. Lu, Kernel-based machine learning protocol for predicting DNA-binding proteins, Nucleic Acids Res. 33 (2005) 6486–6493.
[23] K.-C. Wong, T.-M. Chan, C. Peng, Y. Li, Z. Zhang, DNA motif elucidation using belief propagation, Nucleic Acids Res. 41 (2013) e153-e153.
[24] M. Ghandi, D. Lee, M. Mohammad-Noori, M.A. Beer, Enhanced regulatory sequence prediction using gapped k-mer features, PLoS Comput. Biol. 10 (2014), e1003711.
[25] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, Proc. IEEE Conf. Comput. Vis. Patt. Recog. (2016) 770–778.
[26] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, Proc. IEEE Conf. Comput. Vis. Patt. Recog. (2017) 4700–4708.
[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. (2017) 30.
[28] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018 arXiv preprint arXiv:1810.04805.
[29] H. Zhao, Z. Tu, Y. Liu, Z. Zong, J. Li, H. Liu, F. Xiong, J. Zhan, X. Hu, W. Xie, PlantDeepSEA, a deep learning-based web service to predict the regulatory effects of genomic variants in plants, Nucleic Acids Res. 49 (2021) W523–W529.
[30] S. Min, H. Kim, B. Lee, S. Yoon, Protein transfer learning improves identification of heat shock protein families, PLoS One 16 (2021), e0251865.
[31] Y. Liu, Y.-H. Zhu, X. Song, J. Song, D.-J. Yu, Why can deep convolutional neural networks improve protein fold recognition? A visual explanation by interpretation, Briefings Bioinf. 22 (2021), bbab001.
[32] Q. Zhang, L. Zhu, D.-S. Huang, High-order convolutional neural network architecture for predicting DNA-protein binding sites, IEEE ACM Trans. Comput. Biol. Bioinf 16 (2018) 1184–1192.
[33] X. Du, J. Hu, S. Li, Using chou's 5-step rule to predict DNA-protein binding with multi-scale complementary feature, J. Proteome Res. 20 (2021) 1639–1656.
[34] Z. Shen, W. Bao, D.-S. Huang, Recurrent neural network for predicting transcription factor binding sites, Sci. Rep. 8 (2018) 1–10.
[35] Y. Zhang, S. Qiao, S. Ji, Y. Li, DeepSite: bidirectional LSTM and CNN models for predicting DNA–protein binding, Int. J. Machine learn. Cyber. 11 (2020) 841–851.
[36] X.-R. Bao, Y.-H. Zhu, D.-J. Yu, DeepTF: Accurate Prediction of Transcription Factor Binding Sites by Combining Multi-Scale Convolution and Long Short-Term Memory Neural Network, International Conference on Intelligent Science and Big Data Engineering, Springer, 2019, pp. 126–138.
[37] A. Trabelsi, M. Chaabane, A. Ben-Hur, Comprehensive evaluation of deep learning architectures for prediction of DNA/RNA sequence binding specificities, Bioinformatics 35 (2019) i269–i277.
[38] Q. Zhang, S. Wang, Z. Chen, Y. He, Q. Liu, D.-S. Huang, Locating transcription factor binding sites by fully convolutional neural network, Briefings Bioinf. 22 (2021) bbaa435.
[39] Y. He, Z. Shen, Q. Zhang, S. Wang, D.-S. Huang, A survey on deep learning in DNA/RNA motif mining, Briefings Bioinf. 22 (2021) bbaa229.
[40] J. Keilwagen, J. Grau, Varying levels of complexity in transcription factor binding motifs, Nucleic Acids Res. 43 (2015) e119-e119.
[41] M. Siebert, J. Söding, Bayesian Markov models consistently outperform PWMs at predicting motifs in nucleotide sequences, Nucleic Acids Res. 44 (2016) 6055–6069.
[42] R. Eggeling, T. Roos, P. Myllymäki, I. Grosse, Inferring intra-motif dependencies of DNA binding sites from ChIP-seq data, BMC Bioinf. 16 (2015) 1–15.
[43] H. Kilpinen, S.M. Waszak, A.R. Gschwind, S.K. Raghav, R.M. Witwicki, A. Orioli, E. Migliavacca, M. Wiederkehr, M. Gutierrez-Arcelus, N.I. Panousis, Coordinated effects of sequence variation on DNA binding, chromatin structure, and transcription, Science 342 (2013) 744–747.
[44] E.P. Consortium, An integrated encyclopedia of DNA elements in the human genome, Nature 489 (2012) 57.
[45] T.L. Bailey, J. Johnson, C.E. Grant, W.S. Noble, The MEME suite, Nucleic Acids Res. 43 (2015) W39–W49.
[46] Y. Huang, B. Niu, Y. Gao, L. Fu, W. Li, C.D.-H.I.T. Suite, A web server for clustering and comparing biological sequences, Bioinformatics 26 (2010) 680–682.
[47] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, Proc. IEEE Conf. Comput. Vis. Patt. Recog. (2015) 1–9.
[48] I. Sela, D.B. Lukatsky, DNA sequence correlations shape nonspecific transcription factor-DNA binding affinity, Biophys. J. 101 (2011) 160–166.
[49] J. Telorac, S.V. Prykhozhij, S. Schöne, D. Meierhofer, S. Sauer, M. Thomas-Chollier, S.H. Meijsing, Identification and characterization of DNA sequences that prevent glucocorticoid receptor binding to nearby response elements, Nucleic Acids Res. 44 (2016) 6142–6156.
[50] Ü.i.t.V. Çatalyürek, C. Aykanat, B. Uçar, On two-dimensional sparse matrix partitioning: models, methods, and a recipe, SIAM J. Sci. Comput. 32 (2010) 656–683.
[51] H. Gholamalinezhad, H. Khosravi, Pooling Methods in Deep Neural Networks, a Review, 2020 arXiv preprint arXiv:2009.07485.
[52] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, Pytorch: an imperative style, high-performance deep learning library, Adv. Neural Inf. Process. Syst. (2019) 32.
[53] L. Bottou, Large-scale machine learning with stochastic gradient descent, Proc. COMPSTAT (2010) 177–186. Springer2010.
[54] M. Ghandi, M. Mohammad-Noori, N. Ghareghani, D. Lee, L. Garraway, M.A. Beer, gkmSVM: an R package for gapped-kmer SVM, Bioinformatics 32 (2016) 2205–2207.
[55] L. Xu, S. Jiang, J. Wu, Q. Zou, An in silico approach to identification, categorization and prediction of nucleic acid binding proteins, Briefings Bioinf. 22 (2021) bbaa171.
[56] R. Xu, J. Zhou, H. Wang, Y. He, X. Wang, B. Liu, Identifying DNA-Binding Proteins by Combining Support Vector Machine and PSSM Distance Transformation, BMC systems biology, BioMed Central, 2015, pp. 1–12.
[57] Y.-H. Zhu, J. Hu, X.-N. Song, D.-J. Yu, DNAPred: accurate identification of DNA-binding sites from protein sequence by ensembled hyperplane-distance-based support vector machines, J. Chem. Inf. Model. 59 (2019) 3057–3071.
[58] S. Adilina, D.M. Farid, S. Shatabda, Effective DNA binding protein prediction by using key features via Chou's general PseAAC, J. Theor. Biol. 460 (2019) 64–78.
[59] J. Hu, Y. Li, Y. Zhang, D.-J. Yu, ATPbind: accurate protein–ATP binding site prediction by combining sequence-profiling and structure-based comparisons, J. Chem. Inf. Model. 58 (2018) 501–510.
[60] J. Song, F. Li, K. Takemoto, G. Haffari, T. Akutsu, K.-C. Chou, G.I. Webb, PREvaIL, an integrative approach for inferring catalytic residues using sequence, structural, and network features in a machine-learning framework, J. Theor. Biol. 443 (2018) 125–137.
[61] J. Hu, L.-L. Zheng, Y.-S. Bai, K.-W. Zhang, D.-J. Yu, G.-J. Zhang, Accurate prediction of protein-ATP binding residues using position-specific frequency matrix, Anal. Biochem. 626 (2021), 114241.