Weakly-Supervised Convolutional Neural Network Architecture for Predicting Protein-DNA Binding

Qinhu Zhang, Lin Zhu, Wenzheng Bao, and De-Shuang Huang

Abstract—Although convolutional neural networks (CNN) have outperformed conventional methods in predicting the sequence specificities of protein-DNA binding in recent years, they do not take full advantage of the intrinsic weakly-supervised information of DNA sequences that a bound sequence may contain multiple TFBS(s). Here, we propose a weakly-supervised convolutional neural network architecture (WSCNN), combining multiple-instance learning (MIL) with CNN, to further boost the performance of predicting protein-DNA binding. WSCNN first divides each DNA sequence into multiple overlapping subsequences (instances) with a sliding window, and then separately models each instance using CNN, and finally fuses the predicted scores of all instances in the same bag using four fusion methods, including *Max*, *Average*, *Linear Regression*, and *Top-Bottom Instances*. The experimental results on *in vivo* and *in vitro* datasets illustrate the performance of the proposed approach. Moreover, models built on *in vitro* data using WSCNN can predict *in vivo* protein-DNA binding with good accuracy. In addition, we give a quantitative analysis of the importance of the reverse-complement mode in predicting *in vivo* protein-DNA binding, and explain why not directly use advanced pooling layers to combine MIL with CNN, through a series of experiments.

Index Terms—Weakly-supervised information, multiple-instance learning, convolutional neural network, reverse-complement mode, fusion method, transcription factor binding site prediction

1 Introduction

Discovering transcription factor binding site (TFBS), also called motif discovery, is essential for further understanding the transcriptional regulatory mechanism in the process of gene expression. In the past decade, with the development of high-throughput sequencing technology, a variety of experimental methods have been proposed to extract binding regions. Particularly, ChIP-seq [1], combining chromatin immunoprecipitation with high-throughput sequencing, greatly increases the amount of available data, which is beneficial for the study of protein-DNA binding *in vivo*. On the other hand, protein binding microarray (PBM) [2] can measure *in vitro* binding of a transcription factor to all possible DNA sequence variants of a given length k, which delivers an excellent information source to develop TFBS models in a direct manner. However, the sequencing reads directly obtained from ChIP-seq or PBM do not precisely represent TFBS due to the low resolution of high-throughput sequencing experiments [3]. Therefore, a series of computational methods have been proposed for precisely modeling protein-DNA binding, which are roughly categorized into unsupervised and supervised algorithms. Given a set of DNA sequences bound by a specified transcription factor (TF), unsupervised algorithms usually attempt to model TF binding preferences by a position weight matrix (PWM) [4], [5], or a consensus sequence [6], [7]. In contrast, supervised algorithms collect a number of positive sequences (TF-bound sequences) and negative sequences (non-bound sequences), and model TF binding preferences that can discriminate between positive and negative sequences by using classification or regression models. Not surprisingly, supervised algorithms have been proven to be more accurate at predicting TF-DNA binding [8].

However, these conventional unsupervised or supervised methods usually suffer considerable limitations, e.g., poor ability of handling large-scale sequencing data, poor generalization ability, time-consuming, and so on. Recently, deep learning has been successfully applied to modeling TF binding preferences, which overcomes the above limitations in conventional methods. DeepBind [9] (Alipanahi et al.) and DeepSea [10] (Zhou et al.) both used deep convolutional neural network (CNN), which is a variant of multilayer artificial neural network [11], [12] and specialized for processing images, to predict the sequence specificities of DNA-binding proteins with a performance superior to the best existing conventional motif discovery methods. Furthermore, Zeng et al. [13] replicated DeepBind using the Caffe platform, which achieved slightly lower median AUC but fewer AUCs close to 0.5 than DeepBind, and conducted a systematic exploration of the performance of different CNN architectures by

- *The authors are with the Institute of Machine Learning and Systems Biology, School of Electronics and Information Engineering, Tongji University, No. 4800 Caoan Road, Shanghai 201804, China.*
E-mail: zhangqinhu1@qq.com, lizhonyx@163.com, baowz55555@126.com, dshuang@tongji.edu.cn.

Manuscript received 7 Feb. 2018; revised 15 June 2018; accepted 1 Aug. 2018. Date of publication 7 Aug. 2018; date of current version 1 Apr. 2020.
(Corresponding author: De-Shuang Huang.)
Digital Object Identifier no. 10.1109/TCBB.2018.2864203

varying CNN width, depth and pooling designs. Subsequently, variants of DeepBind or DeepSea are put forward one after another [14], [15], [16]. Despite the success of these deep-learning based methods, they usually follow the fully supervised learning paradigm, and ignore the intrinsic weakly-supervised information of DNA sequences.

An important paradigm for handling such weakly-supervised information is multiple-instance learning (MIL), which was initially proposed for drug activity prediction in the middle of 1990s, by Dietterich et al. [17]. Nowadays MIL is a commonly-used trick to boost the classification accuracy in the field of images [18], [19], since images have the intrinsic weakly-supervised information that each image may contain multiple objects of interest. MIL-based models first extract candidate regions (instances) from images with existing segmentation techniques [20], [21], and then separately model each instance using deep CNN, and finally fuse the scores of all instances in the same bag using the frequently-used *Max* method. Similarly, DNA sequences also have the intrinsic weakly-supervised information that a TF-bound sequence may contain multiple TFBS(s), thus it is reasonable to use MIL to handle DNA sequences. Recently, Gao et al. [22] proposed a MIL-based algorithm for modeling TF-DNA binding, which first divided DNA sequences into multiple overlapping subsequences (instances) with a sliding window, and then modeled all possible instances using the conventional method TeamD [23], and finally fused the predicted scores of all instances in the same bag using the *Average* method. However, the author only combines MIL with conventional methods, which have to suffer the above mentioned limitations.

In this paper, first, we conduct an in-depth study of the reason why the performance of the CNN model in [13] is worse than DeepBind, and we find that it ignores the reverse-complement mode, which is perhaps the main reason that leads to poor performance. Based on the CNN model, therefore, we offer a simple method for simultaneously taking into account the input sequence and its reverse complement. Second, we find that the frequently-used *Max* method only takes into consideration the most informative instance but ignores other instances containing useful information, thus we attempt to employ other three fusion methods: *Average*, *Linear Regression*, and *Top-Bottom Instances*. Motivated by the aforementioned observations, finally, we propose a weakly-supervised convolutional neural network architecture (WSCNN), which is a first attempt to combine MIL with CNN, to further boost the performance of predicting protein-DNA binding. WSCNN first divides each DNA sequence into multiple overlapping subsequences (instances), and then separately models each instance using CNN, and finally fuses the predicted scores of all instances in the same bag using the four fusion methods. Therefore WSCNN explicitly takes into account the weakly-supervised information of DNA sequences, resulting in higher accuracy in the task of predicting protein-DNA binding. The experimental results on *in vivo* and *in vitro* datasets show that the proposed method WSCNN outperforms single-instance learning (SIL) based CNN methods and MIL-based conventional algorithms. Moreover, models built on *in vitro* data using WSCNN can predict *in vivo* protein-DNA binding with good accuracy. In addition, we give a quantitative analysis of the importance of the reverse-complement mode in predicting *in vivo* protein-DNA binding, and demonstrate that WSCNN has a good generalization ability if provided with enough data, and compare the performance of WSCNN when the four fusion methods are separately used, and explain why not directly use advanced pooling layers to combine MIL with CNN, through a series of experiments.

The rest of the paper is organized as follows. In Section 2, we give a brief description of MIL and CNN. In Section 3, we detail the proposed method WSCNN, and introduce the four fusion methods. Experimental settings and results are given in Section 4.

2 Related Works

2.1 Multiple-Instance Learning (MIL)

In machine learning, MIL is a variation on supervised learning. Instead of receiving a set of instances with individual labels, the MIL-based methods receive a set of labeled bags, each of which contains many instances with unknown labels. In the past decades, MIL has been widely applied to a variety of domains, including computer vision [18], [19], [24], computational biology [22], and so on. Taking image classification for example, given a set of labeled images and based on the fact that each image may contain multiple objects of the same class, MIL-based methods often regard each image as a labeled bag, and all candidate objects extracted from that image as instances in the bag, without individual labels, and then model each instance in the bag using deep CNN, and finally fuse the predicted scores of all instances in the bag using the frequently-used *Max* method. Similarly, DNA sequences have the same weakly-supervised information that a TF-bound sequence may contain multiple TFBS(s). Therefore it is reasonable to use MIL to handle DNA sequences, where a DNA sequence can be described as a positive bag if there is at least one instance containing TFBS, or a negative bag if there are no any instances containing TFBS.

Recently, Gao et al. [22] proposed a MIL-based algorithm for modeling TF-DNA binding, which can freely use any SIL-based learners as base learners. Here, we give a brief description of the approach. *Step 1*: each DNA sequence (with length l) is divided into multiple overlapping subsequences (with starting point shift s) by using a sliding window (with length c), and all possible subsequences (instances) in the whole sequence are considered as a bag labeled positive or negative according to the binding signal value, thus the number of instances per bag is $\lceil (l-c)/s \rceil + 1$, where c and s are two hyper-parameters. *Step 2*: each instance is mapped to a feature vector representing k-mer appearances, and initially assigned the label of the bag that they belong to and a weight proportional to the inverse of the size of its bag. *Step 3*: the conventional classification algorithm (TeamD) is used to learn a model using all instances. *Step 4*: the predicted scores of all instances in the same bag are fused (averaged) to give the score of a bag.

2.2 Convolutional Neural Network (CNN)

In recent years, CNN has made great achievements in various application scenarios, which is perhaps the main reason why so many researchers attempt to apply it to TFBS prediction. Deep CNN, which tries to capture local patterns by utilizing a weight-sharing strategy, is well suited to genomics [25], since convolutional kernels can be thought of as motif scanners like PWM. A genomic sequence composed of four nucleotides $\{\mathrm{A, C, G, T}\}$ can be transformed into an image-like matrix by using one-hot encoding [9], [10], [13]. This task, therefore, is analogous to a two-class image classification task in computer vision. As far as we know, DeepBind is the first attempt to apply deep CNN to generating TFBS prediction from raw genomic sequences, which provides a base framework for other variants. Furthermore, Zeng et al. [13] replicated DeepBind using the Caffe platform, and conducted a systematic exploration of the performance of different CNN architectures. Here we give a generic description of CNN-based framework for TFBS prediction. A CNN-based framework usually comprises four computational stages, including encoding, convolution, pooling and neural network, and details are as follows:

1) *Encoding.* The main role of this stage is to encode DNA sequences into image-like inputs by using one-hot encoding. Given a DNA sequence $\tilde{s} = (s_1, s_2, \cdots, s_l)$ of length l and a fixed motif scanner length m, we first pad $(m-1)$ 'zero'(s) on both sides of the sequence, and then get the encoded matrix $S_{i,j}$ through the following equation:

$$S_{i,j} = \begin{cases} 1 & \text{if } s_{i-m+1} = j\text{th base in } \{\mathrm{A, C, G, T}\} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Therefore, the columns of matrix *S* correspond to the one-hot vectors of A, C, G, or T, which are denoted by $[1,\ 0,\ 0,\ 0]^{\mathrm{T}}$, $[0,\ 1,\ 0,\ 0]^{\mathrm{T}}$, $[0,\ 0,\ 1,\ 0]^{\mathrm{T}}$, and $[0,\ 0,\ 0,\ 1]^{\mathrm{T}}$ respectively.

2) *Convolution.* This stage contains a convolutional layer followed by a rectified linear unit [26] (ReLU) layer. The outputs (X) of this stage are the convolution of convolutional kernels *M* and the matrix *S*, and the computation of this stage is as follows:

$$X_{i,k} = \max\left(0, \sum_{j=1}^{m} \sum_{h=1}^{4} S_{i+j,h} M_{k,j,h} + b_k\right), \tag{2}$$

where, $i \in [1, l+m-1]$ and $k \in [1, d]$, d denotes the number of motif scanners (convolutional kernels), b_k denotes the bias term.

3) *Pooling.* With consideration of the ZOOPS assumption [27] that zero or one motif occurrences per sequence, this stage adopts a max-pooling strategy which picks out the maximum value over the outputs of the previous stage, and the computation of this stage is as follows:

$$z_k = \max\big(X_{1,k}, \cdots, X_{l,k}\big) \tag{3}$$

where $k \in [1, d]$, *d* is the number of motif scanners.

4) *Neural Network.* This stage contains two fully-connected layers where the first layer is followed by a ReLU layer, and the second layer is followed by a softmax layer which generates a probability distribution over two labels. In addition, dropout strategy [28] is used to avoid overfitting in this stage, which randomly masks the outputs of the first fully-connected layer to zero. The computation of this stage is as following:

$$\begin{aligned} h_j &= \mathrm{dropout}\left(\max\left(0, \sum_{k=1}^{d} W_{j,k}^{1} z_k + b_j\right)\right), \quad \text{for } j \in [1, 32] \\ f_i &= \mathrm{softmax}\left(\sum_{j=1}^{32} W_{i,j}^{2} h_j + b_i\right), \quad \text{for } i \in [1, 2]. \end{aligned} \tag{4}$$

where h_j denotes the outputs of the first fully-connected layer, f_i denotes the probability distribution over two labels.

3 Methods

3.1 Weakly-Supervised Convolutional Neural Network (WSCNN))

As we known, when training a model on double-stranded DNA sequences, it is unknown whether the protein binds to the input strand or the opposite strand. To handle this ambiguity, DeepBind took into consideration the reverse-complement mode, which simultaneously evaluated a score for both the input sequence and its reverse complement, and then took the maximum score as the final prediction. However, the CNN model proposed by [13] did not mention the reverse-complement mode, which is perhaps the main reason why the performance of it is worse than DeepBind. Based on the CNN model, we offer a simple method for simultaneously taking into account the input sequence and its reverse complement, which reconstructs an input by padding *m* 'zero'(s) between the input sequence and its reverse complement like this:

$$\tilde{s}^{new} = (s_1, s_2, \cdots, s_l) \oplus (m *' \mathrm{zero}') \oplus \left(s_l^{rc}, \cdots, s_2^{rc}, s_1^{rc}\right), \tag{5}$$

where s_i and s_i^{rc} denote any one nucleotide from $\{\mathrm{A, C, G, T}\}$ and its reverse complement respectively, *m* denotes the length of motif scanners ($m = 24$ in this paper), 'zero' can be denoted by a vector $[0,\ 0,\ 0,\ 0]^{\mathrm{T}}$ when encoded, $\oplus$ denotes the concatenation operation. This method not only considers the reverse-complement mode, but also shares the same parameters of filters.

Then, we propose a weakly-supervised convolutional neural network architecture (WSCNN), which is a first attempt to combine MIL with CNN, to further boost the performance of predicting protein-DNA binding. As shown in Fig. 1, WSCNN first divides a DNA sequence into multiple overlapping subsequences (instances) using the sliding window method in [22], where *c* and *s* are two hyper-parameters in this paper, and then encodes each instance into an image-like matrix using (1) and evaluates a score for all instances using CNN with the reverse-complement mode, and finally fuses the predicted scores of all instances in the same bag as the score of a bag using a certain fusion method.

Multiple overlapping instances ensure that most of them may contains TFBS. However, the frequently-used *Max* method, which is a standard fusion method in MIL-based models, only takes into consideration the most informative instance but ignores other instances containing useful

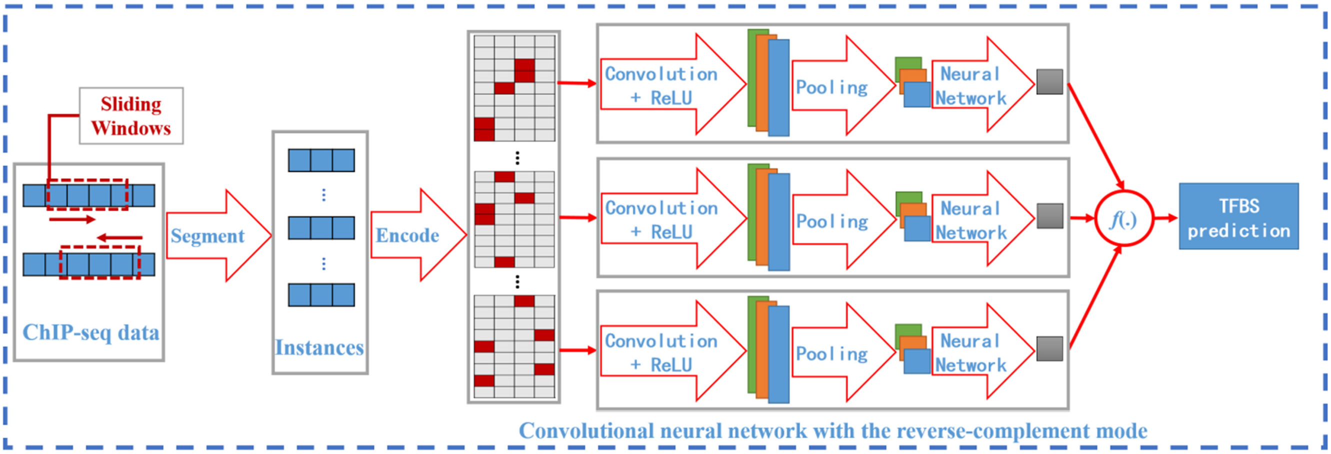

Fig. 1. A graphical illustration of WSCNN.

information. Therefore, we attempt to employ other three fusion methods: *Average*, *Linear Regression*, and *Top-Bottom Instances*, in this paper. These fusion methods are detailed as follows:

1) *Max.* This fusion method takes the score of the most informative instance as the final TFBS prediction.

$$prediction = \max(score_1, score_2, \cdots, score_n). \tag{6}$$

2) *Average (Ave).* This fusion method takes a weighted sum of the scores of all instances in the same bag as the final TFBS prediction, where the weights are equal.

$$prediction = \sum_{i=1}^{n} score_i/n \tag{7}$$

3) *Linear Regression (LR).* This fusion method takes a weighted sum of the scores of all instances in the same bag as the final TFBS prediction, where the weights are automatically learned during training.

$$prediction = \sum_{i=1}^{n} w_i \times score_i \tag{8}$$

where the coefficient w_i means the effect of the ith instance to the final prediction.

4) *Top-Bottom Instances (TBI).* In the computer vision community, Durand et al. [29] proposed a weakly-supervised prediction (WSP) module for image prediction, incorporating top instances and bottom instances as the final prediction. The intuition behind top instances is to provide a more robust selection strategy, since using a single best instance for training deep CNN model necessarily increases the risk of selecting outliers, guiding the training towards bad local minima. The intuition behind bottom instances is that they can be effectively combined with top instances for improving classification performance [30]. In view of this idea, TBI is used as an advanced fusion method in the framework of WSCNN, which includes two parts: top instances and bottom instances.

Top instances. The q highest scoring instances are selected as follows:

$$top = \max_{h} \sum_{i=1}^{n} h_i \times score_i, \text{ s.t.} \sum_{i=1}^{n} h_i = q, \tag{9}$$

where $h_i \in \{0,1\}$ denotes whether or not the ith instance is selected as one of the q highest scoring instances.

Bottom instances. The p lowest scoring instances are selected as follows:

$$bottom = \min_{h} \sum_{i=1}^{n} h_i \times score_i, \text{ s.t.} \sum_{i=1}^{n} h_i = p, \tag{10}$$

where $h_i \in \{0,1\}$ denotes whether or not the ith instance is selected as one of the p lowest scoring instances.

At last, the scores of top instances and bottom instances in the same bag are combined (averaged) as the final TFBS prediction:

$$prediction = (top + bottom)/(q + p) \tag{11}$$

where q and p are two hyper-parameters in this paper.

Regarding training, the model parameters are optimized using back-propagation with standard classification losses or regression losses.

4 Experiments

In this section, we carried out a series of experiments to demonstrate the effectiveness of the proposed method WSCNN.

4.1 Experimental Setup

4.1.1 ChIP-Seq Data Preparation

In order to evaluate the performance of the proposed method WSCNN, we collected 48 public transcription factor ChIP-seq datasets from the ENCODE project, where each ChIP-seq dataset has an average of ~45000 positive sequences, and each sequence consists of 199 bps. Positive sequences were extracted from the hg19 genome centered at the reported ChIP-Seq peak. On the other hand, the way of generating negative sequences is also crucial. It is widely recognized that the negative sequences should be selected to match the statistical properties of the positive set [31], [32],

[33], otherwise the elicited motifs could be biased [34]. To satisfy such requirements, Ghandi et al. [35] provided a R package for generating negative sequences by matching the length, GC content and repeat fraction of the positive set. Therefore, we employed the R package to build an imbalanced dataset, where the number of generated negative sequences is 1~5 times more than that of positive sequences.

Following [36], [37], to accurately evaluate the performance of our proposed method, here we adopted three-fold cross-validation strategy. In other words, each ChIP-seq dataset is randomly partitioned into 3 sets (folds) of roughly equal size, and two of them are used as the training set while the rest is used as the test set. During training, we randomly sampled $1/8$ of the training set as the validation set.

4.1.2 PBM Data Preparation

PBM data were collected from *Weirauch* et al. [8], corresponding to eighty-six mouse TFs. Each TF dataset comprises a complete set of PBM probe intensities for two distinct microarray designs named HK and ME, each of which contains ~40,000 unique 35bp probe sequences. In our study, HK arrays were used to train models, and their corresponding ME arrays were used to evaluate models. We first preprocessed PBM data by following [9]: computing each probe's median intensity across all 86 experiments for that microarray design, and then dividing them by median intensity.

4.1.3 Competing Methods

The competing methods include the basic CNN architecture in [13], the MIL-TeamD approach in [22], and the CNN model with the reverse-complement mode, abbreviated as RC-CNN.

The basic CNN architecture, a single-instance learning based CNN method (SIL-CNN), has the same architecture as DeepBind. SIL-CNN consists of a convolutional layer (followed by a ReLU layer), a global max-pooling layer, and two fully-connected layers (followed by a ReLU layer and a softmax layer respectively). The convolutional layer has 16 convolutional kernels (motif scanners), each of which separately scans the input sequence with stride size 1. The global max-pooling layer only outputs the maximum value of all of its respective convolutional layer outputs. The first fully-connected layer consists of 32 neurons, and the second one consists of 2 neurons.

MIL-TeamD, a multiple-instance learning based conventional algorithm, converts TeamD into its corresponding MIL version. TeamD uses *k*-mer appearances as features, where $k = [4,\ 8]$, and adopts linear-regression as the base learner. Moreover, TeamD efficiently selects features by filtering out the 6mer to 8mer features with low variance, and performs normalization and a few transformation tricks on the data.

RC-CNN, a single-instance learning based CNN method, incorporates the reverse-complement mode for simultaneously taking into account the input sequence and its reverse complement, and the details are in Section 3.

4.1.4 Evaluation Metrics

ChIP-seq provides a ranked list of putatively bound sequences, whereas PBM provides a specificity coefficient for each probe sequence. Therefore three evaluation metrics were used in this paper.

Area under receiver operating characteristic curve (ROC AUC), a widely-used evaluation metric in both machine learning and motif discovery [8], [9], [33], [38], is equal to the probability that a classifier will rank a randomly chosen positive sequence higher than a randomly chosen negative one [39].

Area under precision-recall curve (PR AUC) is a better metric to measure the classification performance of the proposed method. Neither the precision nor recall takes into consideration the number of true negatives, thus the PR AUC metric is less prone to inflation by the class imbalance than the ROC AUC metric is [14], [40].

Pearson correlation coefficient (PCC) is used to measure the correlation between the predicted probe intensities and the actual intensities.

4.1.5 Implementation and Hyper-Parameter Settings

Both RC-CNN and WSCNN are based on SIL-CNN, and implemented on the Caffe platform, which are freely available at: https://github.com/turningpoint1988/WSCNN. The implementation has three stages as described in [13], including hyper-parameter search stage, training stage and testing stage. In the hyper-parameter search stage, for each ChIP-seq experiment, 30 randomly sampled hyper-parameter settings were used to train our proposed methods on the training set with a mini-batch size of 100. Then, a hyper-parameter setting with the best validation accuracy which best fits the corresponding ChIP-seq dataset was obtained. In the training stage, the best hyper-parameter setting was used to train our proposed methods on the whole training set (covering the validation set) with a mini-batch size of 300. In the testing stage, the test set was tested using the final trained model.

Recall that WSCNN has two pairs of additional hyper-parameters, including $(c,\ s)$ which controls the length and number of instances, and $(q,\ p)$ which controls the number of top instances and bottom instances in the *Top-Bottom Instances* fusion method. For ChIP-seq data, c was set to 79, and s was set to 10, thus the number of instances per bag is 13 ($\lceil(199-79)/10\rceil+1$), and, q and p were set to 4 and 2 respectively. For PBM data, c was set to 25, and s was set to 1, thus the number of instances per bag is 11 ($\lceil(35-25)/1\rceil+1$), and, q and p were set to 3 and 1 respectively. As will be shown later, varying these hyper-parameters does not affect the main conclusion of the comparison. Other hyper-parameters are the same as those of SIL-CNN, e.g., learning rate, weight decay, and iterations, and the details are shown in Table 1.

For MIL-TeamD, we used the default settings where $(c,\ s)$ were set to (79, 10) for ChIP-seq data, and $(c,\ s)$ were set to (8, 2) for PBM data.

4.2 Performance Comparison on ChIP-Seq Data

4.2.1 WSCNN Outperforms the Competing Methods on ChIP-Seq Data

A comparison of the proposed method WSCNN and the competing methods on 48 ChIP-seq datasets is shown in Supplementary Figures 1, 2 and Supplementary Tables 1, 2, which

TABLE 1
Hyper-Parameter Settings

Hyper-parameters	For ChIP-seq data	For PBM data
Dropout ratio	0.75, 0.5, 0.1	0.75, 0.5, 0.1
Momentum in AdaDelta	0.999, 0.99, 0.9	0.999, 0.99, 0.9
Delta in AdaDelta	1e-4, 1e-6, 1e-8	1e-4, 1e-6, 1e-8
Learning rate	1	1
Weight decay	0.0005	0.0005
Kernel size (motif length)	24	8
Convolutional kernels	16	16
Neurons (two fully-connected layers)	32 and 2	32 and 1
Iterations	4000	4000
Losses	Binary cross entropy loss	Euclidean (L2) loss

can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2018.2864203. Evaluation is done with three-fold cross-validation, and prediction accuracy is measured by the ROC AUC and PR AUC metrics.

Fig. 2a shows the average ROC AUC scores of WSCNN and the competing methods. We find that WSCNN performs better than MIL-TeamD (average ROC AUC 0.869 vs. 0.801 with a gain of 0.068, $p-\text{value} \ < \ 2.2\text{e}-16$, paired t-test), SIL-CNN (average ROC AUC 0.869 vs. 0.846 with a gain of 0.023, $p-\text{value} \ < \ 2.2\text{e}-16$, paired t-test), and RC-CNN (average ROC AUC 0.869 vs. 0.859 with a gain of 0.01, $p-\text{value} \ \leq 2.573\text{e}-12$, paired t-test). Fig. 2b shows the average PR AUC scores of WSCNN and the competing methods. We have the same conclusion that WSCNN performs better than MIL-TeamD (average PR AUC 0.772 vs. 0.653 with a gain of 0.119, $p-\text{value} \ < \ 2.2\text{e}-16$, paired t-test), SIL-CNN (average PR AUC 0.772 vs. 0.735 with a gain of 0.037, $p-\text{value} \ \leq 6.342\text{e}-16$, paired t-test), and RC-CNN (average PR AUC 0.772 vs. 0.755 with a gain of 0.017, $p-\text{value} \ \leq 6.794\text{e}-11$, paired t-test). Moreover, we find that the performance gap between the proposed method and the competing methods is much more pronounced under the PR AUC metric than under the ROC AUC metric, suggesting that the PR AUC metric is more balanced for imbalanced data.

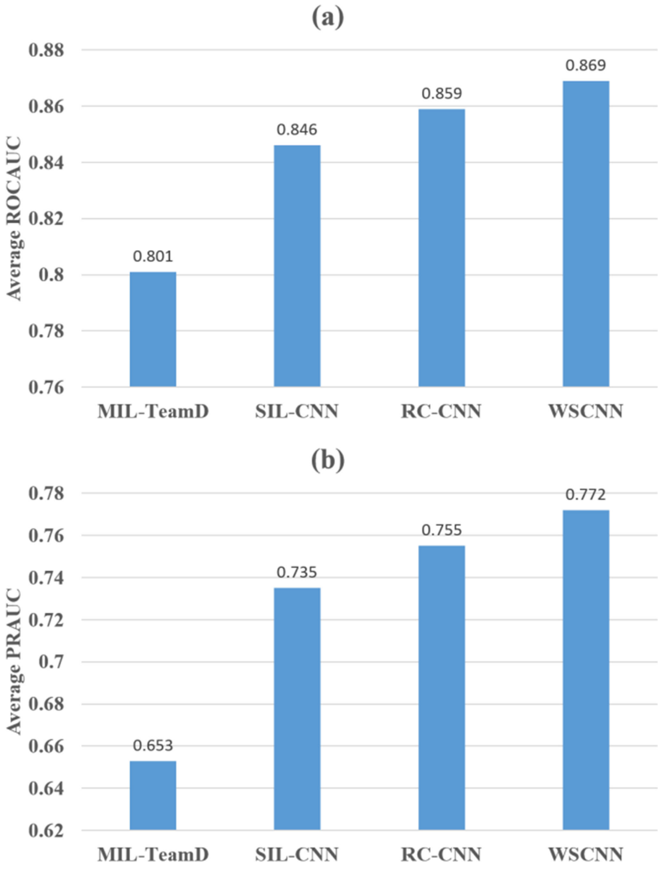

Fig. 2. (a) ROC AUCs and (b) PR AUCs of all methods on *in vivo* data.

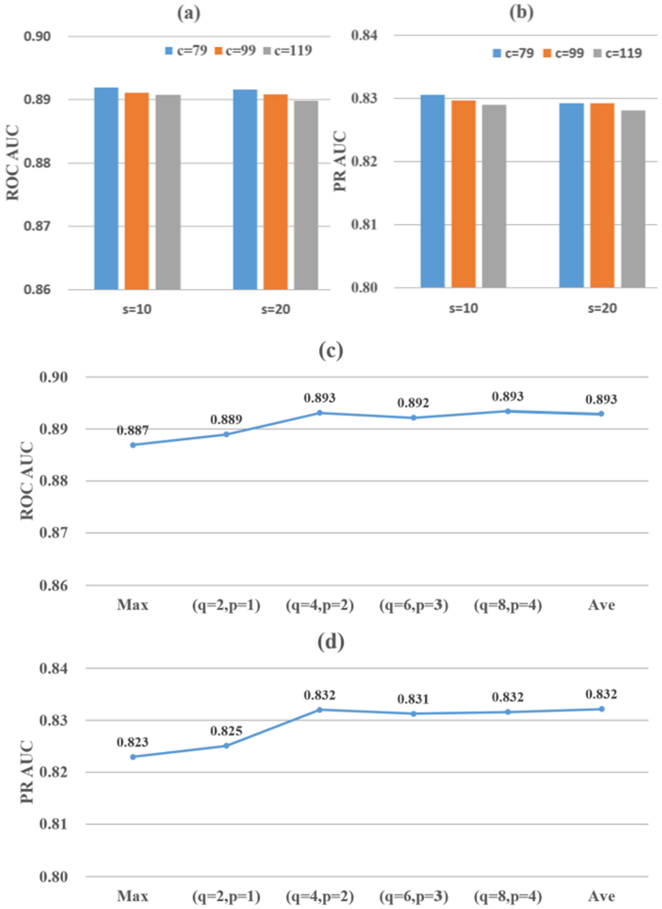

Fig. 3. Effect of hyper-parameter pairs $(c,\ s)$ and $(q,\ p)$ on the performance of WSCNN. (a) ROC AUC and (b) PR AUC of WSCNN which uses the *Linear Regression* fusion method under different $(c,\ s)$ settings. (c) ROC AUC and (d) PR AUC of WSCNN which uses the *Top-Bottom Instances* fusion method under different $(q,\ p)$ settings, while keeping $(c=79, s=10)$ fixed. The number of instances per bag is 13.

10 TF datasets from the whole datasets were randomly selected to investigate the effect of the hyper-parameter pair $(c,\ s)$ on the performance of WSCNN, as shown in Fig. 3. Figs. 3a, 3b separately show the ROC AUC and PR AUC of WSCNN which uses the *Linear Regression* fusion method under different $(c,\ s)$ settings. We observe that the $(c=79,\ s=10)$ setting is the best one although their performance difference is relatively small. Similarly, the 10 TF datasets were used to investigate the impact of the hyper-parameter pair $(q,\ p)$ on the performance of WSCNN, while $(c=79,\ s=10)$ was kept fixed. Figs. 3c, 3d separately show the ROC AUC and PR AUC of WSCNN which uses the *Top-Bottom Instances* fusion method under different $(q,\ p)$ settings. We observe

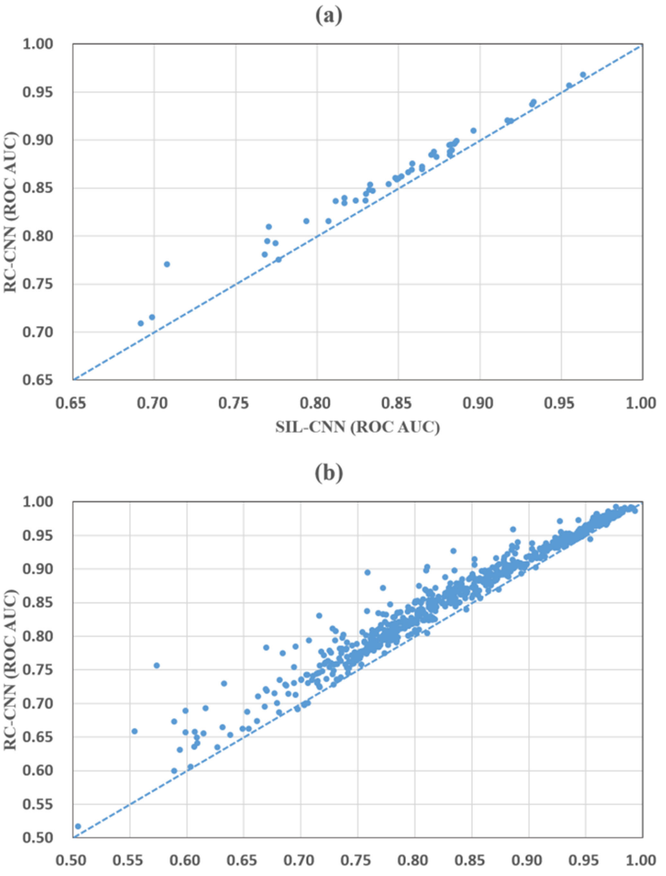

Fig. 4. ROC AUC comparison of SIL-CNN and RC-CNN. (a) ROC AUC comparison of them on 48 TF datasets. (b) ROC AUC comparison of them on 690 TF datasets.

that the $(q = 4, p = 2)$ setting is best suitable for the *Top-Bottom Instances* fusion method. Moreover, the *Max* and *Average* methods are two special cases of the *Top-Bottom Instances* method, corresponding to $(q = 1, p = 0)$ and $(q + p =$ the number of all instances) respectively.

4.2.2 The Importance of the Reverse-Complement Mode in Predicting In Vivo Protein-DNA Binding

In the context of double-stranded DNA sequences, it is reasonable to take into account the reverse-complement mode when training a model, since it is unknown whether the protein binds to the input strand or the opposite strand. Here we give a quantitative analysis of the importance of the reverse-complement mode in predicting *in vivo* protein-DNA binding through a series of experiments. Except the 48 datasets in this paper, 690 ChIP-seq datasets in [13] were used to compare RC-CNN which incorporates the reverse-complement mode with SIL-CNN which ignores the reverse-complement mode, as shown in Fig. 4. From the comparison results, we find that RC-CNN performs better than SIL-CNN (average ROC AUC 0.859 vs. 0.846 with a gain of 0.013 on 48 TF datasets, $p\text{-value} \leq 8.674\text{e}-12$, paired t-test, and average ROC AUC 0.869 vs. 0.845 with a gain of 0.024 on 690 TF datasets, $p\text{-value} < 2.2\text{e}-16$, paired t-test), suggesting the importance of the reverse-complement mode. The reason of performance gain may lie in: both the DNA sequence and its reverse-complement sequence are integrated into one vector so that they can share the same parameters of filters. This strategy is similar to the one proposed by *Shrikumar* et al. [41], which designed a reverse-complement convolutional layer to explicitly share parameters between forward and reverse-complement representations in the model.

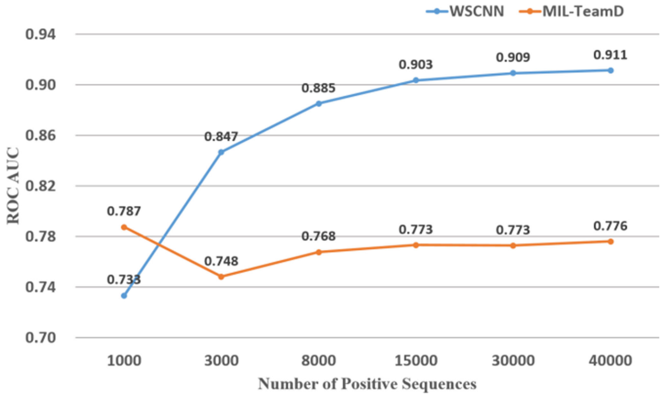

Fig. 5. ROC AUC comparison of MIL-TeamD and WSCNN as the number of positive sequences increases.

4.2.3 WSCNN Has a Good Generalization Ability if Provided with Enough Data

Although MIL-TeamD achieved much higher ROC AUC scores than SIL-TeamD in [22], Supplementary Tables 1, 2, available online show that it suffers poor performance when the number of positive sequences is tremendous. Top 3000 peaks were selected as the positive set in [22], while an average of ~ 45000 top peaks were selected as the positive set in our study. Compared with MIL-TeamD, therefore, we make an assumption that WSCNN has a good generalization ability if provided with enough data. In order to verify this assumption, we performed cross-validation experiments on the above 10 TF datasets where up to top m $(m \in \{1000, 3000, 8000, 15000, 30000, 40000\})$ peaks were selected as the positive set for each TF, while keeping the other parameters fixed. As shown in Fig. 5, ROC AUC of WSCNN is lower than MIL-TeamD when the number of positive sequences is insufficient (less than 1000), while ROC AUC of WSCNN is higher than MIL-TeamD when the number of positive sequences exceeds 3000. Moreover, MIL-TeamD suffers low ROC AUC as the number of positive sequences increases, while ROC AUC of WSCNN is improved as the number of positive sequences increases. Thus WSCNN has a good generalization ability if provided with enough data.

Another factor may be the way of generating negative sequences. [22] generated negative sequences by employing a second order Markov model or shifting the location of the peak sequences on the genome by 5000bp, while WSCNN generated them by matching the length, GC content and repeat fraction of the positive set. But this factor is not our focus in this paper.

4.2.4 Is the Max Fusion Method Suitable for WSCNN?

In this section, we compare the performance of WSCNN when the four fusion methods are separately used. To the best of our knowledge, *Max* is a frequently-used fusion method in MIL-based algorithms, which selects the best instance from a bag. However, it only takes into consideration

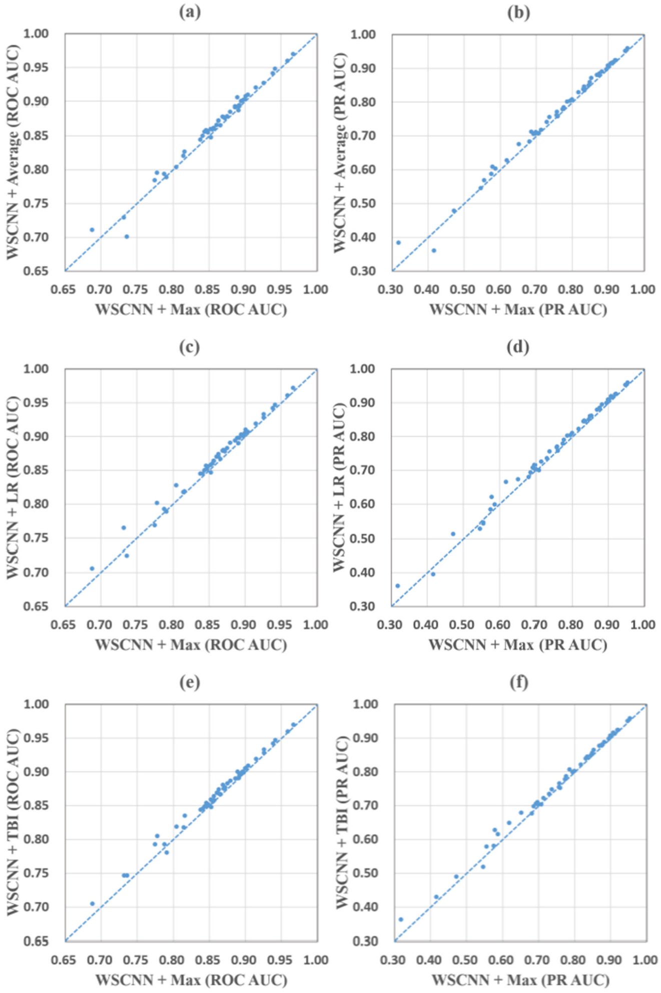

Fig. 6. ROC AUC comparison of WSCNN when using the four fusion methods. (a) ROC AUC and (b) PR AUC comparison of *Max* and *Average*. (c) ROC AUC and (d) PR AUC comparison of *Max* and *Linear Regression*. (e) ROC AUC and (f) PR AUC comparison of *Max* and *Top-Bottom Instances*.

the most informative instance but ignores other instances containing useful information. Therefore, we attempt to employ other three fusion methods, including *Average*, *Linear Regression*, and *Top-Bottom Instances*, which can make the best of the instances containing useful information. From Supplementary Tables 1, 2 available online, we find that the performance of the *Max* method has little improvement compared with other three fusion methods. Moreover, a comparison of the *Max* method and the other three methods is shown in Fig. 6, which shows that the three methods are more robust than the *Max* method, and better suitable for WSCNN.

4.2.5 Why Not Directly Use Advanced Pooling Layers to Combine MIL with CNN

Kraus et al. [42] explored the similarity between the pooling layer of CNN and the MIL aggregation functions, which implies that a regular CNN would perform some kind of MIL. However, with regard to DNA sequences of limited length, unlike microscopy images which contain enough objects of interest, it is hard to guarantee that each sequence contains enough TFBS (s), which may lead to little effect when using advanced pooling functions. Therefore WSCNN divides each DNA sequence into multiple overlapping subsequences (instances) using a sliding window, which guarantees that 1) the weakly-supervised information can be maintained, and that 2) enough instances containing TFBS are generated, and that 3) implicitly taking into consideration binding site locations (which has been explored in [22]). We conducted some experiments on the first 10 ChIP-seq datasets in K562 cell line by comparing RC-CNNs with three pooling layers (*Max*, *Top-Bottom Instances*, and *Noisy-and* [42] pooling functions) with WSCNN, and the detailed results are shown in Supplementary Table 4 available online. From Fig. 7, we can find that WSCNN outperforms RC-CNNs with three pooling layers, and that RC-CNN with the *Max* pooling layer outperforms RC-CNNs with advanced pooling layers (*Top-Bottom Instances* and *Noisy-and*). On the other hand, we also tested the performance of WSCNN with the *Noisy-and* pooling function (fusion method) on the 10 ChIP-seq datasets, and the detailed results are shown in Supplementary Table 5 available online. From Fig. 7, we can find that WSCNN with the *Noisy-and* pooling function has competitive performance to WSCNN under the ROC AUC metric, but outperforms RC-CNN (average PR AUC 0.814 vs. 0.758 with a gain of 0.056, p-value ≤ 0.0057, paired t-test) and WSCNN (average PR AUC 0.814 vs. 0.775 with a gain of 0.039, p-value ≤ 0.0126, paired t-test) under the PR AUC metric (The PR AUC metric is less prone to inflation by the class imbalance than the ROC AUC metric is [14], [40]). In particular, WSCNN with the *Noisy-and* pooling function outperforms RC-CNN with the same function by a large margin (average ROC AUC 0.855 vs. 0.786 with a gain of 0.069, p-value ≤ 0.0003, paired t-test; average PR AUC 0.814 vs. 0.722 with a gain of 0.092, p-value ≤ 0.002, paired t-test), and

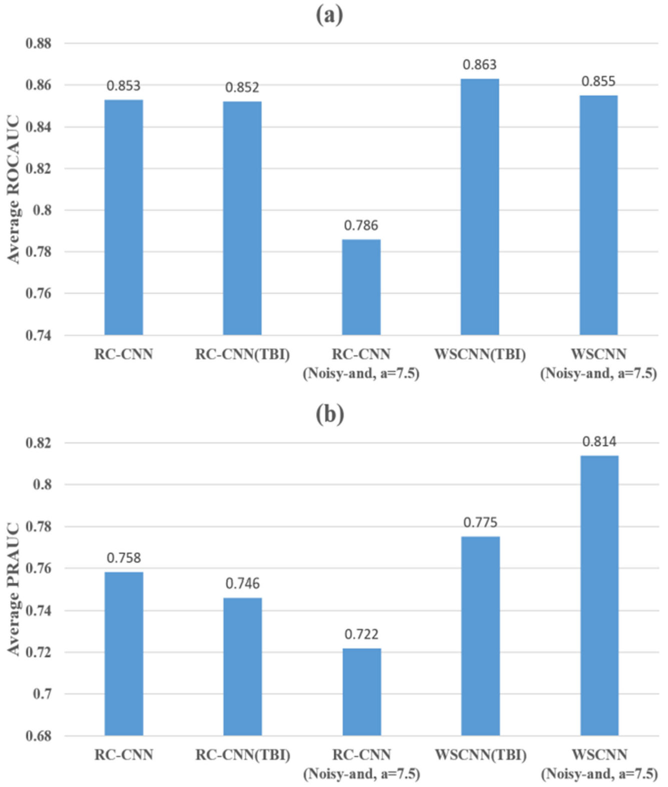

Fig. 7. (a) ROC AUCs and (b) PR AUCs of various methods on the first 10 ChIP-seq datasets in K562 cell line.

TABLE 2
Average And Median Pearson Correlation Coefficient (PPC) of Various Methods on *in vitro* Data

Methods PCC	MIL-TeamD	SIL-CNN	WSCNN
Average	0.414	0.491	0.534
Median	0.413	0.514	0.569

the reason of performance gain may lie in: the number of positive instances (containing TFBS) in the raw sequences cannot activate a positive bag level probability, since the *Noisy-and* function is under the assumption that a bag is positive if the number of positive instances in the bag surpasses a certain threshold, but the framework of WSCNN can explicitly generate enough positive instances, resulting in better performance when combining with the *Noisy-and* function. The above results not only demonstrate the necessity of segmenting DNA sequences into overlapping instances, but also indirectly reflect that the framework of WSCNN can combine better fusion methods.

4.3 Performance Comparison on PBM Data

4.3.1 *WSCNN Outperforms the Competing Methods on PBM Data*

A comparison of the proposed method WSCNN and the competing methods on 86 PBM datasets is shown in Supplementary Figure 3 and Supplementary Table 3 available online. Pearson correlation coefficient (PCC) is used as the evaluation metric. In the experiments of PBM data, we found that the reverse-complement mode had little improvement in performance, so we did not take into consideration the reverse-complement mode.

Table 2 shows the average and median PCCs of WSCNN and the competing methods. We find that WSCNN performs better than MIL-TeamD (average PPC 0.534 vs. 0.414 with a gain of 0.12, and median PPC 0.569 vs. 0.413 with a gain of 0.156, $p\text{-value} \leq 1.017\text{e}-11$, paired t-test) and SIL-CNN (average PPC 0.534 vs. 0.491 with a gain of 0.043, and median PPC 0.569 vs. 0.514 with a gain of 0.055, $p\text{-value} \leq 4.038\text{e}-06$, paired t-test).

4.3.2 *Models Built on In Vitro Data Can Predict In Vivo Binding*

To test whether Models built on *in vitro* data can predict *in vivo* binding, we collected three mouse ChIP-seq datasets (gata4, tbx5, tbx20) from [8], which have corresponding PBM experiments. In each experiment, 3000 top ranking peaks were chosen as the positive set in which each sequence consists of 50 bps and 100 bps, and the corresponding negative set consists of shuffled positive sequences with dinucleotide frequency maintained. Table 3 shows that WSCNN outperforms SIL-CNN, and has competitive performance to MIL-TeamD, which implies that models built on *in vitro* data using WSCNN can predict *in vivo* protein-DNA binding with good accuracy.

TABLE 3
ROC AUC of Using Models Built on *in vitro* Data to Predict *in vivo* Binding

Methods	MIL-TeamD		SIL-CNN		WSCNN	
TFs	50bps	100bps	50bps	100bps	50bps	100bps
Gata4	0.651	0.732	0.643	0.724	0.66	0.751
Tbx20	0.563	0.571	0.55	0.548	0.561	0.564
Tbx5	0.574	0.58	0.562	0.57	0.573	0.578

TABLE 4
Average Runtime of Various Methods

Data Methods	ChIP-seq (s)	PBM (s)
MIL-TeamD	1706	1354
SIL-CNN	32	11
RC-CNN	35	–
WSCNN (MAX)	95	20
WSCNN (AVE)	98	20
WSCNN (LR)	95	19
WSCNN (TBI)	96	20

4.4 Runtime Evaluation

Table 4 lists the average runtime for all methods in this paper. Since MIL-TeamD needs to map each instance to a feature vector representing k-mer appearances, it will cost a great deal of runtime and memory size although it uses a parallel technique in Matlab. However, SIL-CNN, RC-CNN, and WSCNN are all based on deep learning which benefits from the sophisticated training algorithms, and implemented on the Caffe platform which benefits from the power of parallel and distributed computing, thus the average runtime of them is much less than that of MIL-TeamD. Moreover, the average runtime of them is sorted from small to large: $\mathrm{SIL-CNN} < \mathrm{RC-CNN} < \mathrm{WSCNN}$, suggesting that complex networks are much more time-consuming to train. For WSCNN, the average runtime of using the four fusion methods is roughly identical.

5 Conclusions

In this article, we propose a weakly-supervised convolutional neural network architecture (WSCNN) to further boost the performance of predicting protein-DNA binding, which explicitly takes into account the weakly-supervised information of DNA sequences. WSCNN first divides each DNA sequence into multiple overlapping subsequences (instances) with a sliding window, and then models each instance using CNN, and fuses the predicted scores of all instances in the same bag using four fusion methods in the end. The experimental results on *in vivo* and *in vitro* datasets show that WSCNN significantly outperforms the competing methods, suggesting the effectiveness of WSCNN. In addition, we give a quantitative analysis of the importance of the reverse-complement mode in predicting *in vivo* protein-DNA binding, and demonstrate that WSCNN has a good generalization ability if provided with enough data, and compare the performance of WSCNN when the four fusion methods are separately used, and explain why not use advanced pooling layers to combine MIL with CNN.

There are several future directions where we intend to extend this work. 1) Although ‘one-hot’ is the most commonly-used encoding way in CNN-based motif discovery, it has a significant limitation that it assumes that all nucleotides in the binding sites are statistically independent. Thus it would be an interesting work to find other advanced encoding methods for WSCNN. 2) Recurrent neural network (RNN) [43] is an alternative or complementary to CNN for predicting protein-DNA binding, which can capture long-term dependencies among nucleotides. Thus it would be another interesting work to combine WSCNN with RNN for predicting protein-DNA binding. 3) From the experimental results, WSCNN with the *Noisy-and* pooling function has achieved higher performance than other fusion methods, which means the framework of WSCNN can combine better fusion methods. So we will design some application-specific pooling functions to improve the performance of predicting protein-DNA binding.

Acknowledgments

This work was supported by the grants of the National Science Foundation of China, Nos. 61732012, 61520106006, 31571364, 61672382, U1611265, 61532008, 61472280, 61472173, 61572447, and 61772357, China Postdoctoral Science Foundation Grant, Nos. 2016M601646 and 2017M611619, and supported by the "BAGUI Scholar" Program of Guangxi Province of China.

References

[1] T. S. Furey, "ChIP–seq and beyond: New and improved methodologies to detect and characterize protein–DNA interactions," *Nature Rev. Genetics*, vol. 13, pp. 840–852, 2012.

[2] M. F. Berger, A. A. Philippakis, A. M. Qureshi, F. S. He, P. W. Estep III, and M. L. Bulyk, "Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities," *Nature Biotechnology*, vol. 24, 2006, Art. no. 1429.

[3] R. Jothi, S. Cuddapah, A. Barski, K. Cui, and K. Zhao, "Genome-wide identification of in vivo protein–DNA binding sites from ChIP-Seq data," *Nucleic Acids Res.*, vol. 36, pp. 5221–5231, 2008.

[4] G. D. Stormo, "Consensus patterns in DNA," *Methods Enzymology*, vol. 183, pp. 211–221, 1990.

[5] G. D. Stormo, "DNA binding sites: Representation and discovery," *Bioinf.*, vol. 16, pp. 16–23, 2000.

[6] X. Zhao, H. Huang, and T. P. Speed, "Finding short DNA motifs using permuted Markov models," *J. Comput. Biol.*, vol. 12, pp. 894–906, 2005.

[7] G. Badis, M. F. Berger, A. A. Philippakis, S. Talukder, A. R. Gehrke, S. A. Jaeger, et al., "Diversity and complexity in DNA recognition by transcription factors," *Sci.*, vol. 324, pp. 1720–1723, 2009.

[8] M. T. Weirauch, A. Cote, R. Norel, M. Annala, Y. Zhao, T. R. Riley, et al., "Evaluation of methods for modeling transcription factor sequence specificity," *Nature Biotechnology*, vol. 31, 2013, Art. no. 126.

[9] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning," *Nature Biotechnology*, vol. 33, pp. 831–838, 2015.

[10] J. Zhou and O. G. Troyanskaya, "Predicting effects of noncoding variants with deep learning-based sequence model," *Nature Methods*, vol. 12, pp. 931–934, 2015.

[11] D. S. Huang, "Systematic theory of neural networks for pattern recognition," Publishing House of Electronic Industry of China, Beijing, vol. 201, 1996.

[12] D. S. Huang, "Radial basis probabilistic neural networks: Model and application," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 13, pp. 1083–1101, 1999.

[13] H. Zeng, M. D. Edwards, G. Liu, and D. K. Gifford, "Convolutional neural network architectures for predicting DNA–protein binding," *Bioinf.*, vol. 32, pp. i121–i127, 2016.

[14] D. Quang and X. Xie, "DanQ: A hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences," *Nucleic Acids Res.*, vol. 44, pp. e107–e107, 2016.

[15] D. R. Kelley, J. Snoek, and J. L. Rinn, "Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks," *Genome Res.*, vol. 26, pp. 990–999, 2016.

[16] H. R. Hassanzadeh and M. D. Wang, "DeeperBind: Enhancing prediction of sequence specificities of DNA binding proteins," in *Proc. IEEE Int. Conf. Bioinf. Biomed.*, 2017, pp. 178–183.

[17] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, pp. 31–71, 1997.

[18] J. Amores, "Multiple instance classification: Review, taxonomy and comparative study," *Artif. Intell.*, vol. 201, pp. 81–105, 2013.

[19] J. Wu, Y. Yu, C. Huang, and K. Yu, "Deep multiple instance learning for image classification and auto-annotation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3460–3469.

[20] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders, "Segmentation as selective search for object recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1879–1886.

[21] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 391–405.

[22] G., Zhen and J. Ruan, "Computational modeling of in vivo and in vitro protein-DNA interactions by multiple instance learning," *Bioinf.* vol. 33, no. 14, pp. 2097–2105, 2017.

[23] M. Annala, K. Laurila, H. Lähdesmäki, and M. Nykter, "A linear model for transcription factor binding affinity prediction in protein binding microarrays," *PloS One*, vol. 6, 2011, Art. no. e20059.

[24] O. Maron and A. L. Ratan, "Multiple-instance learning for natural scene classification," in *Proc. 15th Int. Conf. Mach. Learn.*, 1998, pp. 341–349.

[25] Y. Park and M. Kellis, "Deep learning for regulatory genomics," *Nature Biotechnology*, vol. 33, pp. 825–826, 2015.

[26] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[27] T. L. Bailey and C. Elkan, "The value of prior knowledge in discovering motifs with MEME," in *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 1995, pp. 21–29.

[28] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[29] T. Durand, N. Thome, and M. Cord, "Weldon: Weakly supervised learning of deep convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4743–4752.

[30] T. Durand, N. Thome, and M. Cord, "Mantra: Minimum maximum latent structural svm for image classification and ranking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2713–2721.

[31] C. Fletez-Brant, D. Lee, A. S. McCallion, and M. A. Beer, "kmer-SVM: A web server for identifying predictive regulatory sequence features in genomic data sets," *Nucleic Acids Res.*, vol. 41, pp. W544–W556, 2013.

[32] D. Lee, D. U. Gorkin, M. Baker, B. J. Strober, A. L. Asoni, A. S. McCallion, et al., "A method to predict the impact of regulatory variants from DNA sequence," *Nature Genetics*, vol. 47, pp. 955–961, 2015.

[33] Y. Orenstein and R. Shamir, "A comparative analysis of transcription factor binding models learned from PBM, HT-SELEX and ChIP data," *Nucleic Acids Res.*, vol. 42, pp. e63–e63, 2014.

[34] Z. Yao, K. L. MacQuarrie, A. P. Fong, S. J. Tapscott, W. L. Ruzzo, and R. C. Gentleman, "Discriminative motif analysis of high-throughput dataset," *Bioinf.*, vol. 30, pp. 775–783, 2013.

[35] M. Ghandi, M. Mohammad-Noori, N. Ghareghani, D. Lee, L. Garraway, and M. A. Beer, "gkmSVM: An R package for gapped-kmer SVM," *Bioinf.*, vol. 32, pp. 2205–2207, 2016.

[36] H. Hartmann, E. W. Guthöhrlein, M. Siebert, S. Luehr, and J. Söding, "P-value-based regulatory motif discovery using positional weight matrices," *Genome Res.*, vol. 23, pp. 181–194, 2013.

[37] R. Y. Patel and G. D. Stormo, "Discriminative motif optimization based on perceptron training," *Bioinf.*, vol. 30, pp. 941–948, 2013.

[38] L. Zhu, W. L. Guo, S. P. Deng, and D.S. Huang, "ChIP-PIT: Enhancing the analysis of ChIP-Seq data using convex-relaxed pair-wise interaction tensor decomposition," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 13, no. 1, pp. 55–63, Jan./Feb. 2016.

[39] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Letters*, vol. 27, pp. 861–874, 2006.

[40] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 233–240.

[41] A. Shrikumar, P. Greenside, and A. Kundaje, "Reverse-complement parameter sharing improves deep learning models for genomics," bioRxiv, 2017, Art. no. 103663.

[42] O. Z. Kraus, J. L. Ba, and B. J. Frey, "Classifying and segmenting microscopy images with deep multiple instance learning," *Bioinf.*, vol. 32, pp. i52–i59, 2016.

[43] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, 1997.

Qinhu Zhang received the MS degree in communication and information system from Yunnan University, Kunming, China, in 2015. Now, he is working toward the PhD degree in computer science and technology at Tongji University, China. His research interests include bioinformatics, machine learning, and deep learning.

Lin Zhu received the PhD degree in pattern recognition and intelligent system from the University of Science and Technology of China (USTC), Hefei, China, in 2013. Now, he is an associate researcher in the College of Electronics and Information Engineering, Tongji University, China. His research interests include bioinformatics, latent feature learning, dimensionality reduction, and large-scale learning.

Wenzheng Bao received the master's degree in computer science from the University of Jinan, in 2015. He currently working toward the PhD degree in computer science in the School of Electronic and Information Engineering, Tongji University, Shanghai, China. His research interests include bioinformatics and machine learning.

De-Shuang Huang received the BSc degree from the Institute of Electronic Engineering, Hefei, China, in 1986, the MSc degree from the National Defense University of Science and Technology, Changsha, China, 1989, and the PhD degree from Xidian University, Xian, China, in 1993, all in electronic engineering. During the 1993-1997 period, he was a postdoctoral student, respectively, in the Beijing Institute of Technology and in the National Key Laboratory of Pattern Recognition, Chinese Academy of Sciences, Beijing, China. In September, 2000, he joined the Institute of Intelligent Machines, Chinese Academy of Sciences as the Recipient of "Hundred Talents Program of CAS". In September 2011, he entered into Tongji University as a chaired professor. From September 2000 to March 2001, he worked as a research associate in Hong Kong Polytechnic University. From August to September 2003, he visited George Washington University as a visiting professor, Washington DC, USA. From July to December 2004, he worked as the University Fellow in Hong Kong Baptist University. From March, 2005 to March, 2006, he worked as a research fellow in the Chinese University of Hong Kong. From March to July, 2006, he worked as a visiting professor in the Queen's University of Belfast, UK. In 2007, 2008, and 2009, he worked as a visiting professor in Inha University, Korea, respectively. At present, he is the director of the Institute of Machines Learning and Systems Biology, Tongji University. He is currently an IAPR Fellow and a senior member of the IEEE. He has published more than 180 journal papers. His current research interest includes bioinformatics, pattern recognition, and machine learning.