Journal of Intelligent Learning Systems and Applications, 2020, 12, 1-13
https://www.scirp.org/journal/jilsa
ISSN Online: 2150-8410
ISSN Print: 2150-8402

Transcription Factor Bound Regions Prediction: Word2Vec Technique with Convolutional Neural Network

Rixin Chen[1*], Ruoxi Dai[2*], Mingye Wang[3#*]

[1]University of Electronic Science and Technology of China, Chengdu, China
[2]Northeastern University, Shenyang, China
[3]Sun Yat-sen University, Guangzhou, China
Email: [#]wangmy57@mail2.sysu.edu.cn

How to cite this paper: Chen, R.X., Dai, R.X. and Wang, M.Y. (2020) Transcription Factor Bound Regions Prediction: Word2Vec Technique with Convolutional Neural Network. *Journal of Intelligent Learning Systems and Applications*, **12**, 1-13.
https://doi.org/10.4236/jilsa.2020.121001

Received: September 2, 2019
Accepted: December 2, 2019
Published: December 5, 2019

Abstract

Genome-wide epigenomic datasets allow us to validate the biological function of motifs and understand the regulatory mechanisms more comprehensively. How different motifs determine whether transcription factors (TFs) can bind to DNA at a specific position is a critical research question. In this project, we apply computational techniques that were used in Natural Language Processing (NLP) to predict the Transcription Factor Bound Regions (TFBRs) given motif instances. Most existing motif prediction methods using deep neural network apply base sequences with one-hot encoding as an input feature to realize TFBRs identification, contributing to low-resolution and indirect binding mechanisms. However, how the collective effect of motifs on binding sites is complicated to figure out. In our pipeline, we apply Word2Vec algorithm, with names of motifs as an input to predict TFBRs utilizing Convolutional Neural Network (CNN) to realize binary classification, based on the ENCODE dataset. In this regard, we consider different types of motifs as separate "words", and their corresponding TFBR as the meanings of "sentences". One "sentence" itself is merely the combination of these motifs, and all "sentences" compose of the whole "passage". For each binding site, we do the binary classification within different cell types to show the performance of our model in different binding sites and cell types. Each "word" has a corresponding vector in high dimensions, and the distances between each vector can be figured out, so we can extract the similarity between each motif, and the explicit binding mechanism from our model. We apply Convolutional Neural Network (CNN) to extract features in the process of mapping and pooling from motif vectors extracted by Word2Vec Algorithm and gain the result of 87% accuracy at the peak.

*Three authors (listed according to alphabetical order of last name) contributed to this work equally.
[#]Corresponding author: wangmy57@mail2.sysu.edu.cn.

Keywords

Motifs, Bound Regions, Word2Vec, CNN

1. Introduction

Motifs are significant in gene regulation, constructing genetic regulatory networks, and help us identify certain functions of each gene, which is also helpful for discovering the causes for human diseases. TFs can control the expression process of genes with various mechanisms, and motifs summarize TF sequence specificity. The same motif contained in the genes suggests that these genes may have similar functions. The motif discovery has been challenging, and there are several *de novo* tools for this, for example, AlignACE, MEME, Weeder, and Trawler, giving us access to TF binding information in whole-genome scale, which allows us to predict TFBRs, and capture more TF binding features. There are more papers to provide datasets and methodologies to predict TFBRs [1] [2] [3], providing us more information.

The word2vec algorithm transform words to vectors in high dimensions, which helps to cluster the feature of similarity, then the distance between each word can be presented. It has been a common way applied in the natural language processing. One paper [4] uses word2vec to find the semantic feature in Chinese language. In our paper, it is also utilized to identify the correlation between each motif. If two motifs appear jointly in different binding sites more frequently, then the distance between their corresponding vectors is shorter.

With recent advances in technology, deep neural network has also achieved outstanding performance in solving computational problems, and it has shown priority in dealing with large datasets, enabling us to find the hidden structure within them and thus making proper predictions [5]. In the field of NLP, a paper proposed a method of text classification combining CNN with BiLSTM [6]. In addition to this model, a recent study concerned with TFBR prediction is *DESSO,* which implements the CNN-RNN hybrid model to predict the possible sites with both base sequences and DNA shape features [2]. Another paper [7] makes predictions of binding proteins by deep learning framework, and identifies bound positions.

Our project is to predict TFBRs, given the motif instances from the dataset as an input in our model. We applied Word2Vec algorithm in our model to transform motifs into vectors, and use the vectors as inputs of our deep learning model. We use CNN to fulfill the task of binary classification, which is to predict whether the combination of given motifs will be one certain TF bound region. When the size of our positive samples and negative samples are approximately equal (On the TF Bound Region CTCF), the accuracy reaches 87% when it is optimal. However, when we apply the same model on other different TF Bound Regions, our model would have lower credibility due to the insufficient number of positive samples.

2. Method and Materials

2.1. Datasets

Our dataset comes from the paper of Pouya Kheradpour and Manolis Kellis [8], and we use the exp-regions-motif datafile on this website, which records all TFBRs and their corresponding motifs, for each human chromosome (1 to 22, X and Y). The same transcription factors (TF) can have various binding sites. After we remove TFs without matching motif, there are 71481 lines of data in total, and each line in our dataset contains the information for name of the TF, its binding site, location, and motif instances appearing in this region. In our experiment, we only apply the names of TFs and motifs in the corresponding TFBR. For each TF name and motif name, the mapping scheme, and the motif occurring order on the positive strand of DNA sequence are followed by underscores respectively. To simplify the data, we only reserve the part before the first underscore of these names, based on cell types and motif patterns, and finally we have 244 motif patterns.

Counting the times of each motif pattern appearing in each TFBR, we then have a matrix recording the information. To visualize, we plot motif instances with Bound Regions (**Figure 1**) in the resulting matrix. Every column stands for one TFBR, while every row of the figure stands for the distribution of every motif in all TFBRs. From the figure, we find some columns are in a redder color, which implies that some TFBRs have more binding sites, so that they can be appropriate positive samples. Some rows are in a redder color, which implies that these motifs almost interact with other motifs in almost every TFBR. After statistical selection, we choose CTCF to start with, which has the most enrichment.

2.2. Feature Engineering

We regard motifs as "words", and the combination of motifs in one specific TFBR as one "sentence" and concatenate these "sentences" as a "passage" to form our corpus. Then we apply Word2Vec algorithm to obtain the word vectors for each motif to describe the features.

From the heat map, intuitively we notice that the distribution of motifs for each TFBR varies, so we apply the method in Skip-Gram model [9] [10] to generate motif vectors because appearance frequency of certain motifs is very low (**Figure 2**). With Skip-gram model to find representation of motif patterns in the vector space, we can predict surrounding motif patterns in context. Given a sequence of training motif patterns $m_1, m_2, m_3, \cdots, m_T$ and the size of training context c, the goal of the Skip-gram model is to maximize the average log probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\ne 0}\log p\left(m_{t+j} \mid m_t\right) \tag{1}$$

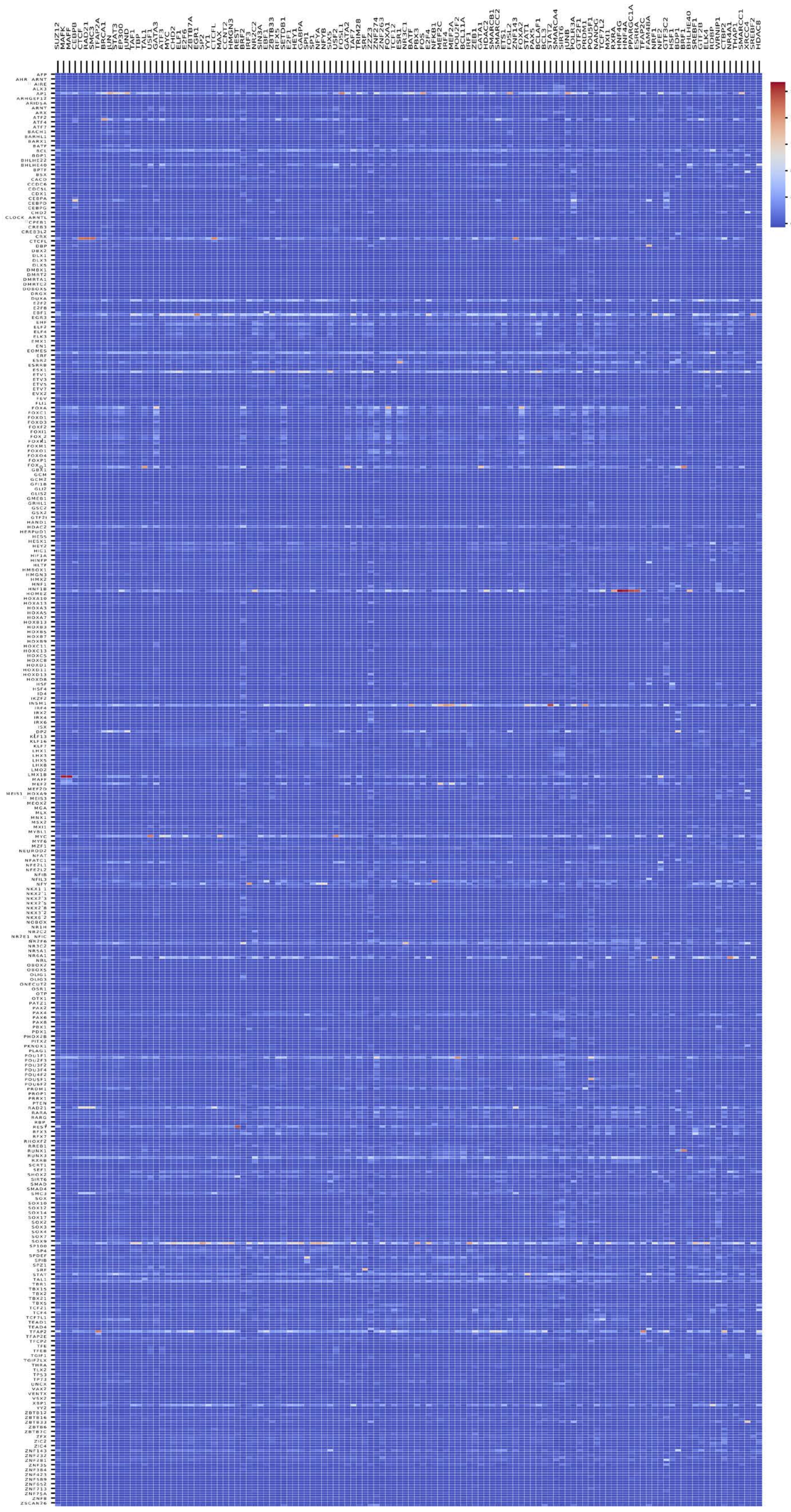

Figure 1. Distribution of Motifs in each TFBR.

In our task, we assign each motif pattern with 200 features, so the corresponding vector has 200 dimensions. Therefore, the hidden layer can be represented by a weighted matrix with 244 rows and 200 columns, one column for each hidden neuron. The output is a one-dimensional vector with 244 entries, containing the probability for each motif pattern having the same type as a randomly selected nearby motif pattern (see in Figure 2).

We use hierarchical softmax to calculate the probability:

$$p(m \mid m_I) = \prod_{j=1}^{L(m)-1} \sigma\left[\!\left[n(m, j+1) = ch(n(m, j)) \right]\!\right] \cdot {v'}_{n(m,j)}^{\mathrm{T}} v_{mI} \tag{2}$$

where $n(m, j)$ is the j-th node on the path from the root to m, and $L(m)$ is the length of this path. Besides, for any inner node n, $ch(n)$ is an arbitrary fixed child of n and $[\![x]\!]$ is 1 if x is true and -1 otherwise. $\sigma(x) = 1/(1+\exp(-x))$.

Every row in hidden layer weight matrix is a word vector that represents a specific motif pattern (see in Figure 3).

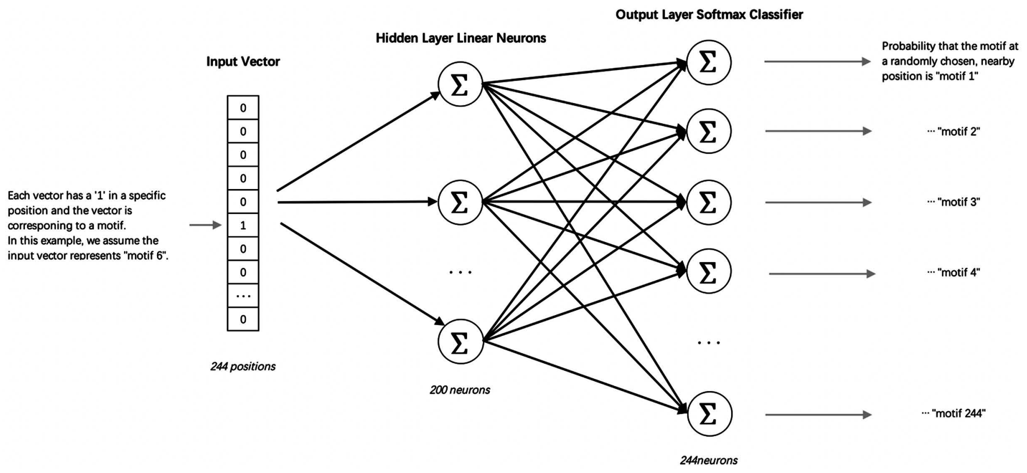
Output Layer Softmax Classifier
Hidden Layer Linear Neurons
Input Vector
Probability that the motif at a randomly chosen, nearby position is "motif 1"
··· "motif 2"
··· "motif 3"
··· "motif 4"
··· "motif 244"
Each vector has a '1' in a specific position and the vector is corresponding to a motif. In this example, we assume the input vector represents "motif 6".
0
0
0
0
0
1
0
0
...
0
244 positions
200 neurons
244neurons

Figure 2. Illustration of the Skip-gram model architecture.

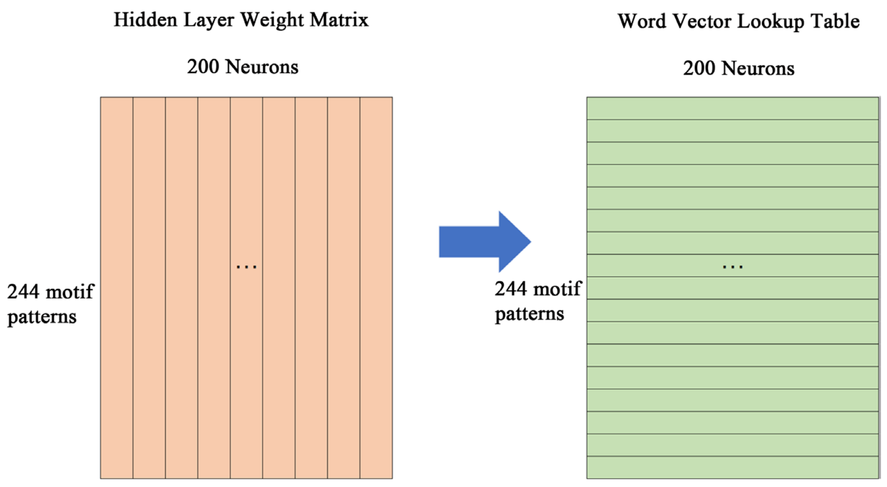

Figure 3. In the hidden layer weight matrix, each column represents the hidden layer neurons' weights of one motif pattern. In the word vector lookup table, each row represents each motif pattern in the vector space.

Then we apply Principal Components Analysis (PCA) to reduce the dimensions of vectors from 200 to 2. Assume that our motif pattern vector is $\boldsymbol{x}_l$, and we combine all the vectors (Assume we have n vectors) into one matrix $\boldsymbol{X}(n, 200)$ and we want to project x in a certain direction with a unit vector $\boldsymbol{v}$. The vector after PCA is $\boldsymbol{z}$, and all the word vectors are matrix $\boldsymbol{Z}(n, 2)$. We can get from the assumption that for each z_i,

$$z_i = \frac{\boldsymbol{x}_l^{\mathrm{T}} \boldsymbol{v}}{\|\boldsymbol{v}\|} = \boldsymbol{x}_l^{\mathrm{T}} \boldsymbol{v} \tag{3}$$

We could have different outcomes z after PCA, due to the different direction of vector $\boldsymbol{v}$. Now before we start, we also need to mean-center our word vectors.

$$\boldsymbol{X}^* = \boldsymbol{X} - \bar{\boldsymbol{X}} \tag{4}$$

So that now we have $E(\boldsymbol{X}^*) = E(\boldsymbol{Z}) = 0$.

Now we need to find an appropriate direction of unit vector $\boldsymbol{v}$ that maximizes the variance of $\boldsymbol{Z}$. The reason why we want to maximize the variance is that we need to keep information as much as possible after the reduction of dimensions, and make our data points as scattered as possible.

$$\operatorname{var}(\boldsymbol{Z}) = \frac{1}{n}\sum_{i=1}^{n}\left(z_i - E(\boldsymbol{Z})\right)^2 = \frac{1}{n}\sum_{i=1}^{n}(z_i)^2 = \frac{1}{n}\boldsymbol{z}^{\mathrm{T}}\boldsymbol{z} = \frac{1}{n}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{v} \tag{5}$$

Hence, the maximization problem becomes,

$$\max_v \left\{\frac{1}{n}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{v}\right\} \ \text{s.t}\ \boldsymbol{v}^{\mathrm{T}}\boldsymbol{v} = 1 \tag{6}$$

We use the method of Lagrange Multipliers. Assume:

$$L = \frac{1}{n}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{v} - \lambda\left(\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v} - 1\right) \tag{7}$$

To have maximum value of L:

$$\frac{\partial L}{\partial \boldsymbol{v}} = \frac{2}{n}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{v} - 2\lambda\boldsymbol{v} = 0 \Rightarrow \frac{1}{n}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{v} = \lambda\boldsymbol{v} \tag{8}$$

If we assume $S = \frac{1}{n}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$, then we can have $S\boldsymbol{v} = \lambda\boldsymbol{v}$.

Apparently, $\boldsymbol{v}$ is the eigenvector while variance of $\boldsymbol{Z}$ is maximized, and after the eigenvalue decomposition of matrix $\frac{1}{n}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$, we can have our result matrix $\boldsymbol{Z}$. Then we use the 2-dimensional vectors z to draw the scatter plot (**Figure 2**). The figure illustrates ability of the Skip-gram model to automatically organize concepts and learn the implicit relationships between different types of motif instances.

Result in the scatter plot (**Figure 4**) below shows the distance representing how often the motifs are bound in one specific TFBR, or CTCF in our case. If vectors for two motif instances have less distance from each other, they are more likely to combine and work at the same time. We choose some motifs and show their DNA sequences, and we draw the conclusion that motifs with closer distance also have similar structures.

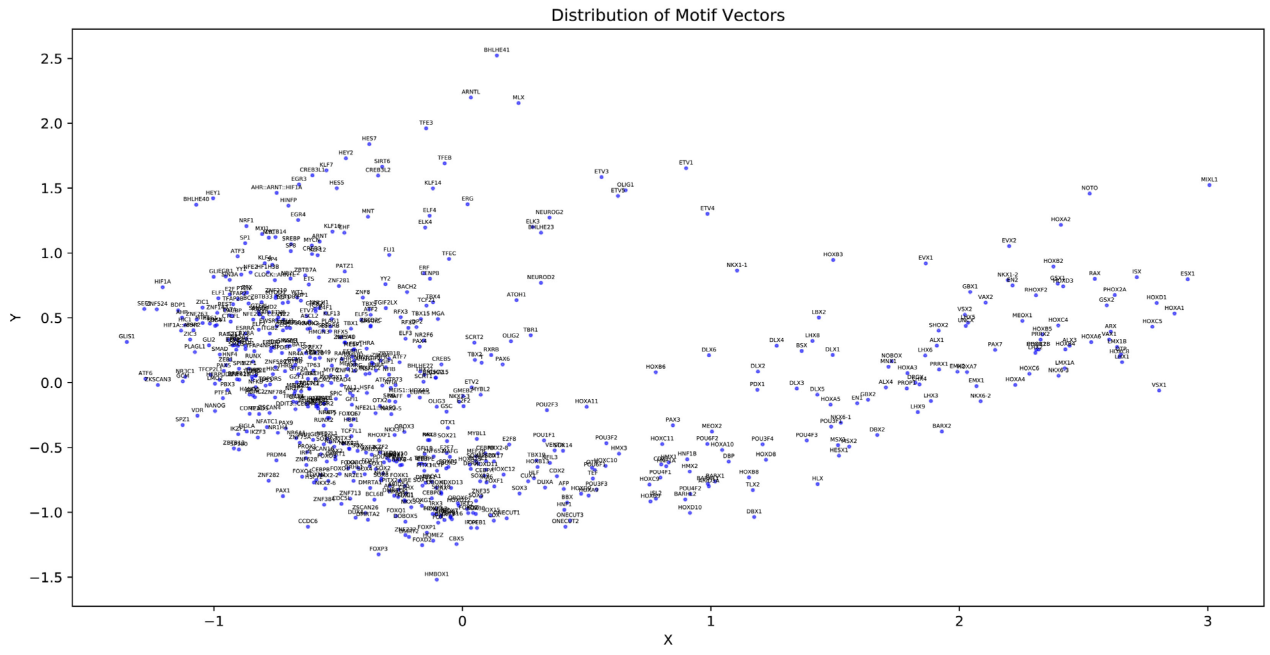

Figure 4. Two-dimensional PCA projection of the 200-dimensional Skip-gram vectors of motif instances.

2.3. Model Structure

After we have 200-dimensional vectors, we apply Convolutional Neural Network (CNN) in one dimension for the sequential information and to fulfill the task of binary classification by extracting more features from the input. Firstly, we use an embedding layer to map each motif with its corresponding 200-dimensional vector, and input is all types of motif patterns (244 in total). For the second layer and before the last layer, we apply dropout to reduce calculating cost and avoid overfitting. Then, to extract main features and correlations between motif patterns and reduce parameters in calculation, we employ one-dimensional convolutional layer and max pooling layer respectively for three times. We also use flatten layer as a connection to transform the data into the form of input for the dense layer, which is also known as fully-connected layer. In order to output one-bit value for the binary classification, we choose sigmoid function as the activation function of last layer, and ReLU unit in other layers (see in Figure 5).

3. Results and Discussion

Our CNN model examines whether the bound region is of one particular TF or not, therefore it helps to predict bound regions based on motif patterns. We apply our model first on the bound region of transcription factor CTCF, then on all other TFBRs in our dataset.

3.1. Experiment Setup

In our project, our dataset are randomly split for training (90%) and test (10%). The batch size is set to 128.Ourmodel is implemented using Python 3.6 and Keras framework. We used Graphics Processing Unit (NVIDIA GeForce GTX

1050Ti) to accelerate the computation.

3.2. Binary Classification on CTCF

First, we start to do the binary classification on binding region of CTCF, which accounts for the largest portion of all TFs (Figure 6). As expected, our CNN

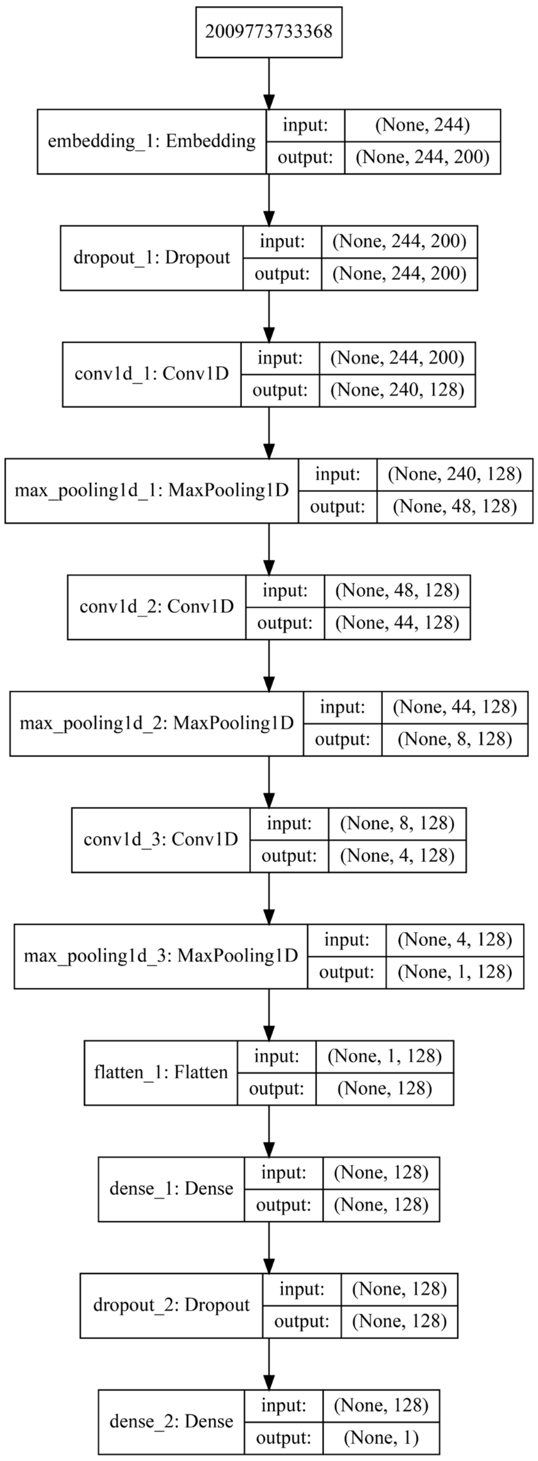

Figure 5. CNN architecture with the specific layer configuration above each layer.

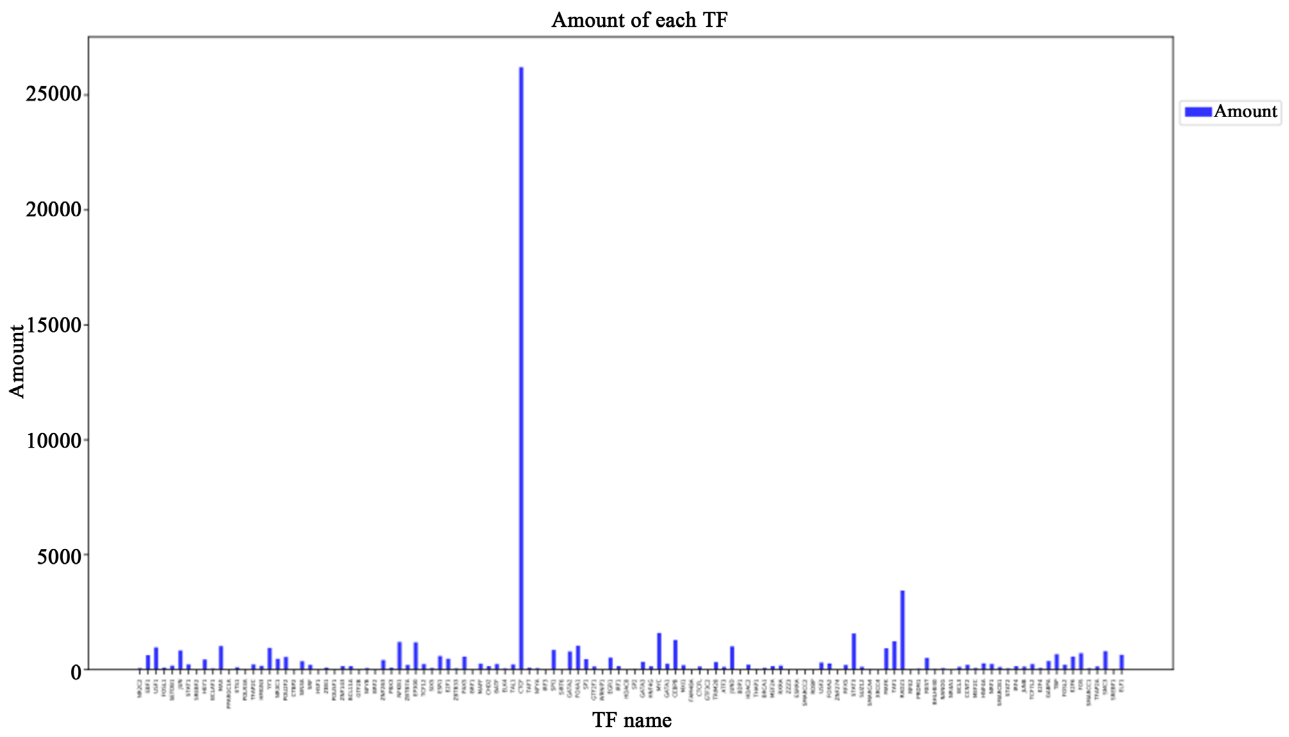

Figure 6. Amount of each transcription factor.

model extracts feature of motif patterns easily. When the epoch reaches 40, our model accuracy on the training set stabilizes, higher than 88%. The accuracy on our test reaches optimal at around the 20th epoch (in 100 epochs), which is 87%, and the loss increases after this point (see from Figure 7 and Figure 8). Similar results on both the accuracy and loss (Figure 9 and Figure 10) are achieved by RNN model for comparison (Figure 11). However, compared with the CNN model, the RNN model is more time-consuming for the same number of epochs (100 epochs), about 10 times of CNN. The reason for this is that every time after mapping and pooling, size of the data to be processed by the network reduces.

3.3. Binary Classification on All TF Bound Regions

We also do binary classifications for other TF bound regions (Figure 12). RAD21 takes the second largest portion, and the corresponding number is about 4000, less than the 0.2 times of CTCF. The training performance is evaluated by the accuracy of CNN binary classification model on each TFBR. Under this metric, our model performs least satisfactory on CTCF. The appearance of this binding site is more frequent, which means more combinations of different motifs, leading to the increasing difficulty of prediction. The appearing frequencies of other TFBRs are greatly less, compared with that of CTCF, generally. For each TFBR, the accuracy of our model exceeds 80%. For some regions with less data, for example, RDBP and ESRRA, the accuracy reaches nearly 100%.

Our CNN model for binary classification can reach weighted average accuracy to 92.7165% for all 122 TFBRs. Although with larger data, the accuracy of our

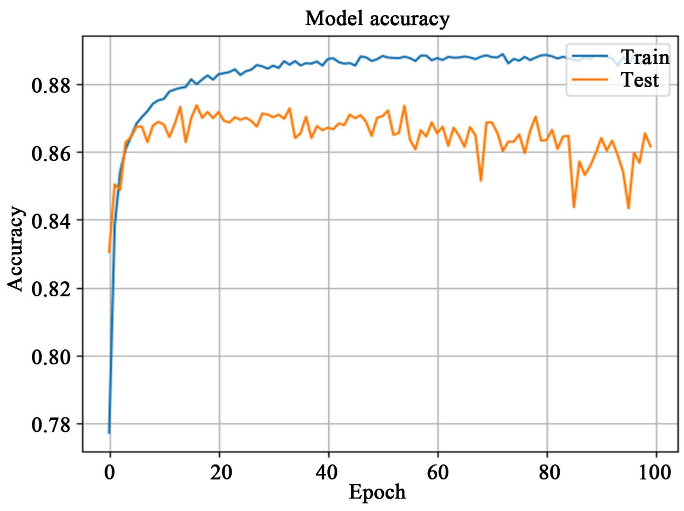

Figure 7. Accuracy of CNN Modelon CTCF dataset.

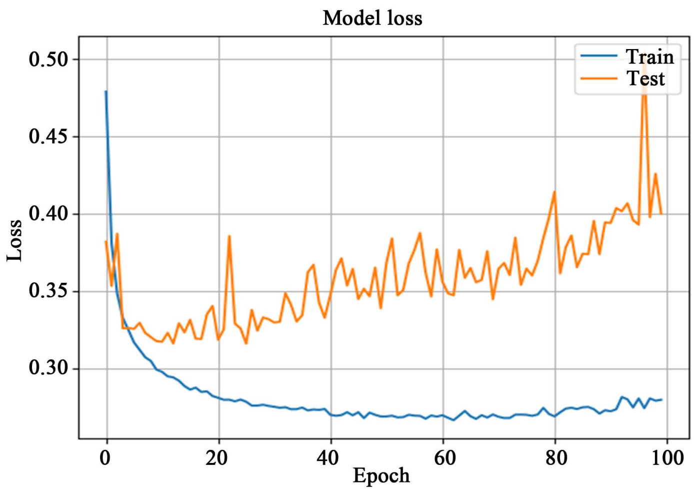

Figure 8. Loss of CNN Modelon CTCF dataset.

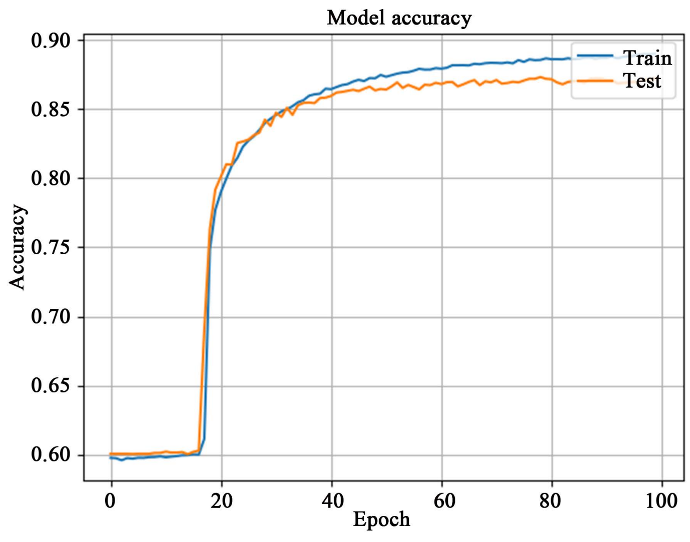

Figure 9. Accuracy of RNN model on CTCF dataset.

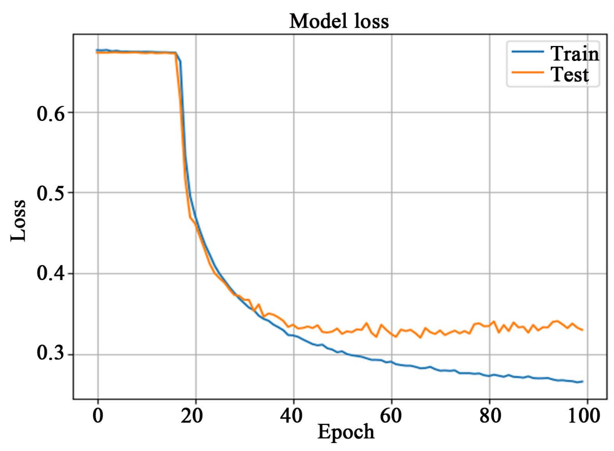

Figure 10. Loss of RNN model on CTCF dataset.

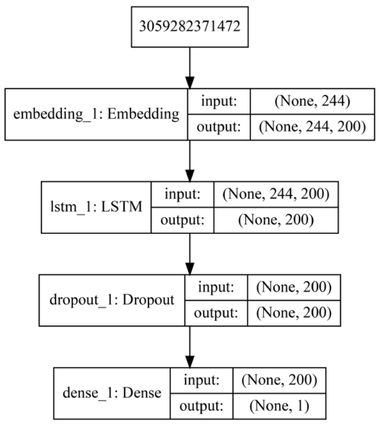

Figure 11. Structure of the RNN model in our contrast experiment.

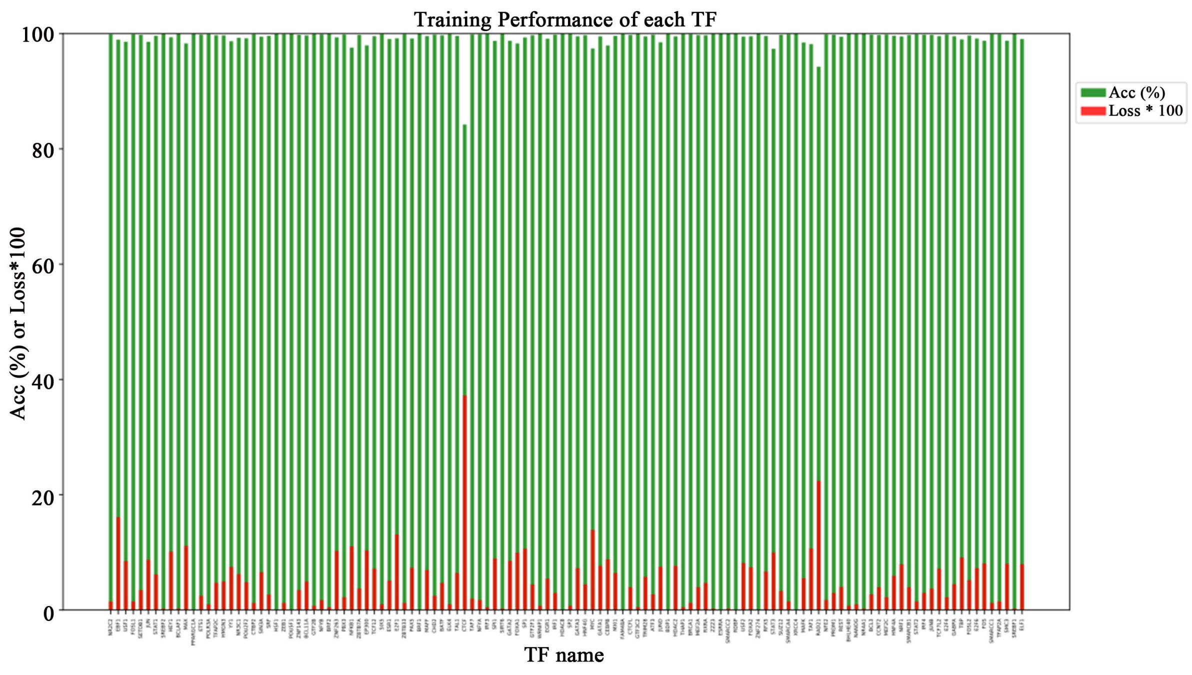

Figure 12. Training performance of each TF.

model decreases, its performance in CTCF also reaches 84.233% (using early stop), and the optimal accuracy is 87% (in 100 epoch) which has the most numeral data.

4. Conclusion and Future Goals

We present our dataset, the motif enrichment of each binding site (in **Figure 1**). Redder columns imply more binding sites, as positive samples, and redder rows stand for more interactive motifs in every TFBR. To downsize our raw data, we group motif instances to motif patterns, and divide TFBRs according to the cell type. We use Word2Vec algorithm to transfer each motif pattern in TFBRs to a 200-dimensional vector, and their relative distance is shown in **Figure 4**. From the distribution of motif vectors, we can see the structure and function of motifs within shorter distance are more similar, and they are expected to work together. Compared with RNN model, our CNN model achieves the similar performance on our dataset, but saves more calculating resources. Our study result clearly elucidates relationship between different motif patterns, and distribution of these patterns in various TFBRs. Another contribution is that after our model learns the relationship between motif patterns and TFBRs, it successfully predicts whether the TF will bind with the DNA sequence or not.

In this project, to reduce the data, we utilize datasets of Chromosome 21, the shortest chromosome of human. In the future, we hope to expand the datasets to all chromosomes of human. Limitation of our model is that some information may be lost in the process of mapping and pooling of CNN, so we hope to continue adjusting the structure of our model or applying a more advanced structure to prevent loss of information from datasets. We will also try to implement RNN model to all other TFBRs. For the input of feature, we only consider the motif patterns, and we hope to add more factors, for example, epigenomics and comparative genomics, so that other decisive factors of TFBRs can be figured out.

Contribution

MY Wang, RX Dai, and RX Chen conceived the method and designed the neural network model. MY Wang preprocessed the dataset, generated the heat map, optimized the CNN model and visualized training results. RX Dai and RX Chen designed the feature extraction and visualized the dataset, and wrote the manuscript. RX Dai implemented the models and also helped optimize the CNN model.

Acknowledgements

We are grateful to Ying Zhang for useful discussions. This work was instructed by Professor Manolis Kellis, and supported by CIS.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Pique-Regi, R., Degner, J.F., Pai, A.A., Gaffney, D.J., Gilad, Y. and Pritchard, J.K. (2011) Accurate Inference of Transcription Factor Binding from DNA Sequence and Chromatin Accessibility Data. *Genome Research*, **21**, 447-455. https://doi.org/10.1101/gr.112623.110

[2] Lanchantin, J., Singh, R., Wang, B. and Yanjun, Q.I. (2017) Deep Motif Dashboard: Visualizing and Understanding Genomic Sequences Using Deep Neural Networks. Proceedings of the Pacific Symposium.

[3] Lanchantin, J., Singh, R., Wang, B. and Qi, Y. (2016) Deep Motif Dashboard: Visualizing and Understanding Genomic Sequences Using Deep Neural Networks. *Pacific Symposium on Biocomputing*, **22**, 254-265. https://doi.org/10.1142/9789813207813_0025

[4] Gers, F.A. and Schmidhuber, E. (2001) LSTM Recurrent Networks Learn Simple Context-Free and Context-Sensitive Languages. *IEEE Transactions on Neural Networks*, **12**, 1333-1340. https://doi.org/10.1109/72.963769

[5] Angermueller, C., PRnamaa, T., Parts, L. and Stegle, O. (2016) Deep Learning for Computational Biology. *Molecular Systems Biology*, **12**, 878. https://doi.org/10.15252/msb.20156651

[6] Li, C., Zhan, G. and Li, Z. (2018) News Text Classification Based on Improved Bi-LSTM-CNN. 2018 9*th International Conference on Information Technology in Medicine and Education*, Hangzhou, 19-21 October 2018, 890-893. https://doi.org/10.1109/ITME.2018.00199

[7] Alipanahi, B., Delong, A., Weirauch, M.T. and Frey, B.J. (2015) Predicting the Sequence Specificities of DNA- and RNA-Binding Proteins by Deep Learning. *Nature Biotechnology*, **33**, 831-838. https://doi.org/10.1038/nbt.3300

[8] Pouya, K. and Manolis, K. (2014) Systematic Discovery and Characterization of Regulatory Motifs in ENCODE TF Binding Experiments. *Nucleic Acids Research*, **42**, 2976-2987. http://compbio.mit.edu/encode-motifs https://doi.org/10.1093/nar/gkt1249

[9] Tomas, M., Kai, C., Greg, C. and Jeffrey, D. (2013) Efficient Estimation of Word Representations in Vector Space. From Cornell University, arXiv:1301.3781.

[10] Tomas, M., Kai, C., Greg, C., Ilya, S. and Jeffrey, D. (2013) Distributed Representations of Words and Phrases and Their Compositionality. From Cornell University, arXiv:1310.4546.