RNAprotein binding sites and motifs prediction

LUO Lu
11155107885
1155107885@link.cuhk.edu.hk

Han Chenyu
1155124364
1155124364@link.cuhk.edu.hk

December 2, 2020

Abstract

RNA Binding proteins(RBPs) are a class of proteins in eukaryotes and play important roles in many biological process, including gene splicing. Unfortunately, reseaches about RBPs, especially the binding preferences, are far from sufficient, so the prediction performance of traditional bioinfomatics algorithms is not satisfactory due to the deficiency of prior information. Thanks to the development of high-throughput technology, vast experiment results are generated in recent few years, providing the precondition of using data-driven approaching like machine learning and CNNs. In this report, we design a model which consists of two CNNs, train it for each RBPs in the dataset, test it performance and compare it with other similar solutions finally. We also discuss difference between our model and other solutions in both the structure and performance, and try to give some potential improvements.

I. Introduction

RNA-binding proteins (RBPs) play key roles in many biological processes. However, detecting the RBP binding sites by experiments is very time-intensive. Although high-throughput technologies, e.g. CLIP-seq [5] have boosted the process of detecting RBP binding sites, and have gained a large amount of experiment results, they are still time-consuming and high-costly. Thus computational methods to predict the RBP binding sites with the help of deep learning is essential.

Computational methods will save a lot of time and will reduce costs. Currently, many computational methods have been proposed to predict RBP binding sites [6], including methods based on support vector machines (SVMs). With the development of deep learning, methods based on convolutional neural networks (CNNs) are also proposed.

However, most current methods with deep learning consider only global representations of entire RNA sequences, while specific RBPs recognize local structure context derived from local RNA sequences from the biological point of view. Therefore, a model which considers both global and local sequences is needed.

II. Background

i. RNA-binding proteins(RBPs)

RNA-binding proteins (RBPs) play key roles in many biological processes. As introduced by [7], RBPs take more than 5-10% of the eukaryotic proteome. RBPs have great impact on important biological process like gene regulation [8] and mRNA localization [9]. Mutations of RBPs may lead to various diseases. Thus, to predict the RBP binding sites may provide a better understanding of some important biological processes.

ii. CNN

CNN is a kind of supervised learning that use multiple convolutional layer to extract hidden feature from input data. CNN is widely used in image recognition and classification algorithm. A typical CNN consists of convolutional, pooling and full connected layers. Convolutional layers are used to extract features, pooling layers are used to decrease the dimensions of data and full connected layers can choose which features are more relative to which class.

III. Methods

The key of our method is that a specific RBP can only recognize one class of RNA sequences which have similar subsequence in somewhere of the rna, called motif. Therefore, our goal is to recognize the similar subsequence, motif, from the given RNA sequence.

One method is based on the experiment results. As long as we get some motifs from experiments, we can use traditional bioinfomatics tools like k-mers to calculate the similarity of input sequence and known motifs. However, the high-throughput technologies are time-intensive and expensive [1], and results are not satisfactory, so the number of accurately known motif is small compared to the real situation. And other drawback is that this method may not be able to make a accurate prediction if the corresponding motif havn't been found.

The second method is to use data-driven approaching like machine learning. Our project is based on this method. In our method, we don't need the knowledge of motifs, and the network will try to find the features we need. CNN, as a powerful feature extracting tool, has potential to find similar structure from giving data, and make a more accurate prediction compared to the first method.

IV. Implementation

There are lots of reseaches showing that deep learning can perform much better than the previous method [2]. There are several network has promising performance, including CNNs and LSTMs. In our project, we decide to use two kinds of CNNs, global CNN and local CNN, to make the prediction, and here are the implementation details:

- **Global CNNs**: The purpose of this CNN is to extract useful feature from RNA sequence. In this CNN, we will feed the whole RNA sequence to the network, so that the network will find features no matter where the subsequence locate. Since the CNN can only receive fixed length input, we need to read and preprocess the data before training and test models(Details about the data preprocess can be found in the third points).
  For the network, we set 2 convolutional layers, 2 pooling layers and 2 dense layers. The kernel size of the first layer is [10, 4], because as Deepbind suggested, the best kernel length is about 1.5 times of the average motifs' length which 7 [3]. The kernel size of the second layer is [10, 1] due to the same reason.
- **Local CNNs**: The purpose of the CNN is to find the relationship of consecutive features. In this network, we will feed the multi-channel tensor to the network, so that the relationship between different channel can make difference. Since the raw data contain only one channel, we also need to do the preprocessing.
  For this network, we set 3 convolutional layers, 3 pooling layer and 2 dense layers. This networks is more complicated because the dimensions of input data are greater, and the complexity of data is higher.
- **Data preprocessing**: We downloaded the RBP-24 dataset from the website of GraphProt http://www.bioinf.uni-

freiburg.de/Software/GraphProt. The origin data type in this dataset is nucleotide sequence. In order to feed the data to CNNs, we need to transform the string into one-hot tensor. After transforming, every sequence can be seen as a graph with width of 4.

For global CNNs, we will find the max sequence length among data of each RBPs, and then padding the rest to that length by adding [[0.25],[0.25],[0.25],[0.25]] to the tail of each one-hot matrix.

For local CNNs, we will divide the sequence into fixed length subsequence with fixed shifting length, and then append the later subsequence to the channel of previous subsequence, until the channels are full or there is no subsequence left. In our project, the window size has been set to 101, shifting size is 20 and the channel size is 7.

- **Models ensembling**: Model ensembling is common because it's hard to get a high performance predictor by using only one model. In our situation, we also meet this problem that the prediction is not accurate if we only use one of the CNN models, because both the sequence character and the features independence are important for binding. Therefore, we train those 2 CNN separately, and use both in the prediction phase, by just average the output probabilities(probabilities of positive and negative). In our experiment, this method can increase the accuracy.
- **Other details**: Due to the limitation of experiments, the number of positive data and negative data is unbalanced. In order to ensure the balance between positive input and negative input, we use over-sampling strategy, which means the sample in the small size class can be used much more times in training. This method can significantly increase the training efficient [4].

V. Results

Results are gained after training and testing the model. In this section, we will first introduce the parameter of our model, then we compare our methods with other currently proposed methods. Additionally, we evaluate the performance of combining the local CNN and global CNN, by comparing their performance independently.

i. Parameter

Due to the limitation of time, we have only tried a small amount of parameter options, including hyperparmeters of CNNs (dropout probability = 0.5, learning rate = 0.0001, other parameters have been listed in secion IV.), thus the parameters may not be the optimal, but we still gained satisfactory results.

ii. Performance

As shown in Table 2, our method performs better than Pse-SVM in general, and is close to other state-of-art methods. The main defect of our method appears in the first few datasets, including ALKBH5, C17ORF85, C22ORF28, etc. We've analyze these datasets, and found that these datasets are smaller than the others, with fewer sequences for training. This may lead to overfitting, and thus reduce the testing accuracy. In datasets with sufficient training sequences, e.g. QKI, ELAVL1C and TAF15, our method performs similarly to the state-of-art methods. Consider that the parameters we have used may not be the optimal, our method have huge potential.

iii. Local and global CNNs

Our method combines the local CNN and global CNN together. It's naturally to ask whether this combination is effective. As shown in Table 1, in most cases, the combination of two networks results in better performance of both independent network.

Table 1: *Average performance of local and global CNN, and combination*

local CNN	global CNN	combination
0.852	0.847	0.871

VI. Discussion

i. Unsatisfactory prediction result for dataset *Baltz2012*

From the result part showed, the prediction for first 3 RBPs is not accurate enough. We discussed about why these performance are different, and give some potential reason:

- **Overfitting:** Actually, we have found this problem in the early phase of our project, and the reason we gave at that time is overfitting because the training accuracy was not bad but the testing accuracy is super low, even lower than 50 in that time. High training accuracy and low testing accuracy are the character of overfitting. However, we have cut down the number of layers and neurons per layer since finding that situation, and the final version of our networks is not complex. However, the problem have been improved but not really solved. Therefore, overfitting may be parts of the reason, but we don't think it's the main reason.
- **Small size of datasets**: We also found that the datasize for those 3 RBPs is significantly smaller than others, especially the first dataset. We natually guessed that the size is too small to represent all the features to recognize the special structures.
- **Quality of Experiment result**: As we mentioned before, the high-throughput technology is an expensive and time-intensive new technology, so the quality of a experiment may not as reliable as some mature technology. We found that not only our project, others also meet the similar problem. Maybe there are too many false result in those datasets, which cannot be distinguished by our project.

ii. Potential improvements

The performance of our predictor is not bad, but it should be better since the accuracy of other similar predictor can reach 0.9 or even higher. There must be some potential improvements, and we will give our ideas here:

- **Adjust the parameters of CNNs**: The parameters of CNN can effect the performance of a model, so they should be carefully tuned to get the best result. However, we use same parameters(including kernel size, number of nodes) for all RBPs, which will definitely decease the accuracy. Therefore, we can adjust the parameters for every RBPs.
  Additionally, the prior knowledge about the motif can be helpful for parameters adjustment. If we know the average length of motifs for each RBPs, we can change the kernel size to 1.5 times of that length to get a better feature extraction performance.
- **Try to rearrange the testing and training set**: In our project, the dataset which we chose have been separated by other researchers. However, we can rearrange the training and testing set. In this way, we can have multiple pairs of training and testing dataset. We can train and test CNN by using these different pairs, and choose the one with best accuracy.

References

[1] Hafner, Markus, et al. "Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP." *Cell* 141.1 (2010): 129-141.

[2] Pan, Xiaoyong, et al. "Prediction of RNA-protein sequence and structure binding preferences using deep convolutional and

recurrent neural networks." *BMC genomics* 19.1 (2018): 511.

[3] Alipanahi, Babak, et al. "Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning." *Nature biotechnology* 33.8 (2015): 831-838.

[4] Ando, Shin, and Chun Yuan Huang. "Deep over-sampling framework for classifying imbalanced data." *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, Cham, 2017.

[5] Anders G. et al. "doRiNA: a database of RNA interactions in post-transcriptional regulation." *Nucleic Acids Res.*, 40, D180D186, 2012.

[6] Corrado G. et al. "RNAcommender: genome-wide recommendation of RNAprotein interactions." *Bioinformatics*, 32, 36273634, 2016.

[7] Castello A. et al. "Insights into RNA biology from an atlas of mammalian mRNA-binding proteins." *Cell*, 149, 13931406, 2012.

[8] Gerstberger S. et al. "A census of human RNA-binding proteins." *Nat. Rev. Genet.*, 15, 829845, 2014.

[9] Dictenberg J.B. et al. "A direct role for FMRP in activity-dependent dendritic mRNA transport links filopodial-spine morphogenesis to fragile X syndrome." *Dev. Cell*, 14, 926939, 2008.

Table 2: *performance of different methods across different datasets*

method RBP	Our method	iDeepE	CNN-LSTM-E	ResNet-E	Pse-SVM	Graph Prot	Deepnet-rbp
ALKBH5	0.609	0.758	0.653	0.656	0.648	0.680	0.714
C17ORF85	0.762	0.830	0.822	0.756	0.734	0.800	0.820
C22ORF28	0.73	0.837	0.801	0.829	0.764	0.751	0.792
CAPRIN1	0.824	0.893	0.871	0.891	0.728	0.855	0.834
Ago2	0.833	0.884	0.851	0.854	0.746	0.765	0.809
ELAVL1H	0.932	0.979	0.975	0.975	0.816	0.955	0.966
SFRS1	0.891	0.946	0.929	0.945	0.746	0.898	0.931
HNRNPC	0.943	0.976	0.973	0.975	0.824	0.952	0.962
TDP43	0.901	0.945	0.928	0.937	0.840	0.874	0.876
TIA1	0.853	0.937	0.911	0.929	0.784	0.861	0.891
TIAL1	0.842	0.934	0.901	0.930	0.724	0.833	0.870
Ago1-4	0.84	0.915	0.873	0.911	0.728	0.895	0.881
ELAVL1B	0.938	0.971	0.963	0.970	0.837	0.935	0.961
ELAVL1A	0.912	0.964	0.962	0.961	0.830	0.959	0.966
EWSR1	0.901	0.969	0.965	0.967	0.753	0.935	0.966
FUS	0.922	0.985	0.980	0.977	0.762	0.968	0.980
ELAVL1C	0.952	0.988	0.986	0.988	0.853	0.991	0.994
IGF2BP1-3	0.89	0.947	0.940	0.952	0.753	0.889	0.879
MOV10	0.857	0.916	0.899	0.911	0.783	0.863	0.854
PUM2	0.927	0.967	0.963	0.965	0.840	0.954	0.971
QKI	0.968	0.970	0.966	0.969	0.809	0.957	0.983
TAF15	0.947	0.976	0.974	0.971	0.769	0.970	0.983
PTB	0.902	0.944	0.929	0.943	0.867	0.937	0.983
ZC3H7B	0.817	0.907	0.879	0.906	0.743	0.820	0.796
Mean	0.871	0.931	0.912	0.919	0.778	0.887	0.902