공학석사학위논문

히스톤 변이 마크를 통한 인핸서 영역 분류 인공신경망 모델

Enhancer Prediction with Histone Modification Marks Using a Hybrid Neural Network Model

2018년 8월

서울대학교 대학원

공과대학 컴퓨터공학부

임 애 란

Abstract

Enhancer Prediction with Histone Modification Marks Using a Hybrid Neural Network Model

Aeran Lim
Department of Computer Science & Engineering
The Graduate School
Seoul National University

An enhancer is a regulatory region in DNA, increasing transcription of a gene by combining with transcription factors. Whereas a promoter is located near a transcription start site, an enhancer can often be located far from a target gene, making hard to identify enhancer regions in DNA. Therefore, many researches in Bioinformatics have challenged to classify enhancer region computationally. In this paper, a hybrid neural network, Convolutional Neural Network followed by Recurrent Neural Network, are used for classifying enhancer regions in DNA with histone modification marks input and my model showed high

performance in evaluation. With the trained model, optimizing virtual input matrix can give insight into how histone modification marks represent enhancer regions.

.....

Keywords : Enhancer, Convolutional Neural Network, Recurrent Neural Network, Histone modification mark

Student Number : 2016-28238

Contents

List of Figures

List of Tables

1. Introduction

Eukaryotic cells carry all the same genes, but their functions differentiate depending on how the genes are expressed. For example, in brain cell, neurotransmitter gene that sends signals is expressed, while the gene is repressed in liver cell. A malfunction in gene expression regulation can result in serious diseases like cancer. Thus, many researches aim to identify gene expression regulation mechanisms. Gene expression is regulated by various factors including gene rearrangement, gene mutations, and regulatory sequences. A promoter and an enhancer are regulatory sequences which have binding sites for proteins like coactivators. Both the promoter and the enhancer are short regions in DNA, and play a role in regulating gene expression involved in gene transcription. When the RNA polymerase and the transcription factors (TFs) bind to the promoter, the transcription of the target gene is controlled based on the type of the combined TF. When the TF is an activator, the promoter initiates the transcription of the target gene and in the case of a repressor, the

transcription is decreased. Similarly, the enhancer increases transcription of target gene with the help of TF. Whereas the promoter is located near transcription start site (TSS) of the target gene, the enhancer can often be located far from the TSS of the target gene, making hard to identify the enhancers. Moreover, existing experimentally validated enhancers sets are relatively small and biological experiments like GRO-seq to validate enhancer region is expensive and needs skilled technicians. As a result, many researches have challenged enhancer identification problem by various computational methods and biological input data.

One of the most commonly used biological input data is DNA sequence, given that enhancer region has TF binding motifs. kmer-SVM[1] employed Support Vector Machine (SVM) with kmer DNA sequence features and found the combination of kmers which specifies enhancer region. BiRen[2] used DNA sequence encoding from DeepSEA[3] model and adopted Recurrent Neural Network (RNN), showing the state-of-art performance only with sequence data.

Histone modification marks data is also commonly-used input in enhancer identification as the histone modification marks can reveal the state of DNA region. ENCODE project[4] annotated genomic state given histone modification marks using 2 computational tools- Segway[5] and ChromHMM[6]. Segway used dynamic Bayesian Network (DBN) and ChromHMM used Hidden Markov Model (HMM) to

segment genomic regions. ENCODE annotation is regarded as one of the reliable chromatin annotation. RFECS[7] employed random forest algorithm, and CSI-ANN[8] applied Time-Delay Neural Networks on the histone modification marks. DEEP[9] employed multiple SVMs and each SVM model is trained as cell line or tissue-specific and can be used with either histone modification marks or DNA sequence features.

Not only single type of input data but also multiple type of input data altogether can be integrated and used in this task. EnhancerFinder[10] exploits and combines information from DNA sequence, evolutionary history, and where proteins bind to DNA. PEDLA[11] has total 1,114 heterogeneous features including DNA sequence, histone modification marks, and chromatin accessibility data and its model is composed of HMM and Artificial Neural Network (ANN).

Although many approaches already exist, there is still much room for improvement in terms of performance and biological interpretation in enhancer identification problem. In this paper, I present a hybrid neural network which is composed of CNN and RNN and takes histone modification marks as input. The performance of my model is shown to be comparable with other existing tools and biological insights in enhancer identification are given by virtual optimized histone modification marks.

2. Methods

2.1 Data

Histone modification marks

Various kinds of biological data have been used for enhancer identification task, including DNA sequence, chromatin accessibility, and TF ChIP-Seq. In this paper, histone modification marks are used as input data since they represent the state of DNA region. DNA sequence is wrapped around 8 histone proteins-2 of each H2A, H2B, H3, and H4, making up a nucleosome. The histones have long tails which can be modified by epigenetic events such as methylation, phosphorylation, or acetylation. Since it is found that the degree of compression and accessibility of chromosome are influenced by the variation of each histone tail, histone modification marks are known to be an indicator of the state of the DNA region. For instance, H3K4me3 represents Trimethylation of H3 lysine 4, indicating active promoter

regions. Therefore, the histone modification marks are also frequently used in enhancer identification.

The ChIP-seq data of the histone modification marks used in this paper are available from ENCODE database. For IMR90 cell line, total 28 histone modification marks are downloaded-H2A.Z, H2AK5ac, H2AK9ac, H2AK9ac, H2BK12ac, H2BK15ac, H2BK20ac, H2BK5ac, H3K14ac, H3K18ac, H3K23ac, H3K27ac, H3K27me3, H3K36me3, H3K4ac, H3K4me1, H3K4me2, H3K4me3, H3K56ac, H3K79me1, H3K79me2, H3K9ac, H3K9me1, H3K9me3, H4K20me1, H4K5ac, H4K8ac and H4K91ac. For H1hesc cell line, total 17 histone modification marks are downloaded-H3K14ac, H4K5ac, H3K4ac, H3K4me3, H4K8ac, H3K56ac, H4K91ac, H3K9ac, H2A.Z, H3K27me3, H4K20me1, H3K9me3, H2BK120ac, H3K36me3, H3K23ac, H3K27ac, and H3K4me1. However, as the number of available histone marks vary in each cell line, it is hard to apply the same model architecture to all cell lines or see the model's transferability in other cell lines. ChromImpute[12] generated imputed signal tracks for the histone modification marks and DNase in various cell lines. Therefore, 28 imputed histone modification marks, the same kinds of marks with IMR90, are also downloaded from ChromImpute for 4 cell lines-H1hesc, GM12878, HeLaS3, and HepG2 cell lines, maintaining the same number of input marks.

Preprocessing histone modification marks

The histone modification marks are preprocessed in the same way as RFECS, except the interval for binning is 50 base pair (bp)[1], not 100 bp and the number of input bins is 60 bins, not 20 bins. More precise model can be obtained with smaller bin size and wider input window. The ChIP-seq reads and the corresponding input (control) are binned in the 50 bp interval, and each bin is normalized in RPKM (Reads per kilobase per million). Peak calling for each ChIP-seq data is carried out through MACS2[13] software.

Positive set and negative set definition

Unfortunately, no gold standard for enhancer identification exists since there are few experimentally validated enhancer sets. As a result, a positive set for enhancer varies from study to study. In this paper, I adopted one of the most commonly used criteria in previous researches. The positive enhancer set was defined as a set of regions with a P300 peak, a transcription activation factor which is known to be bound with enhancers, and DHS (DNase I hypersensitive sites) and at least 2.5kbp away from the TSS (Figure 1). Because histone modification marks are input in my model, histone modification marks

[1] A base pair (bp) is a pair of nucleobase consisting of DNA sequence. A bp is used as a unit of DNA sequence length.

should be excluded in defining positive set, though many researches adopt H3K27ac peaks as enhancer regions. The negative set was defined as background regions not overlapped with P300 peaks and distal to TSS (Figure 1).

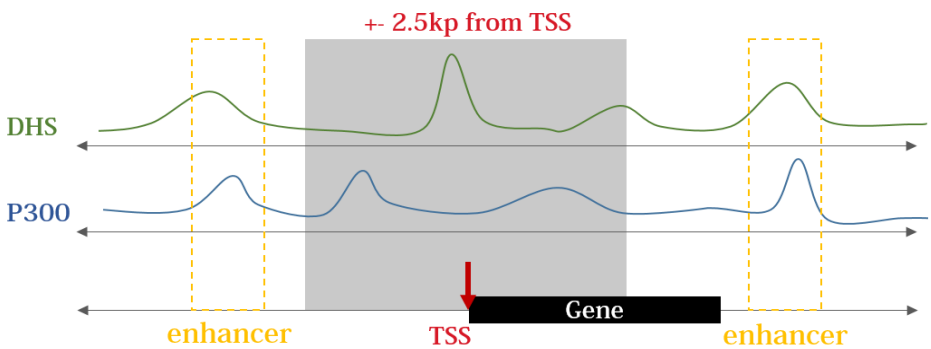

Figure 1. The definition of a positive sample. Enhancer samples are defined as P300 peaks overlapped with DHS and distal to TSS.

2.2 Convolutional Neural Network

Convolution Neural Network (CNN) is a neural network with convolution kernels, inspired by receptive fields in biological vision system. CNN has been applied and shown to be very successful in a wide range of research areas, especially in computer vision. The name, CNN, comes from its convolution layers which have a number

of convolution kernels whose size is smaller than its input. Each kernel produces a feature map when applied convolution function on input. Typically, a convolution layer is followed by a pooling layer which down-samples the information, the feature map, in convolution kernels beforehand. The resulted feature map from convolution layer follows below formula,

$$y_{ij} = \sum_{a,b\in[0,k)} W_{ab} x_{(i+a)(j+b)} \qquad (1)$$

where y_{ij} is the feature map element at position (i, j), k is the width and height of the convolution kernel. The kernel W is element-wise multiplied with corresponding input position x (1). (a, b) indicates the offset from the convolution kernel. The convolution kernel W is shared, reducing the number of parameters and training burden of the network.

CNN got popular in various Bioinformatics research areas including bio-image processing and motif finding. CNN is employed to classify malignant lung nodules given CT scan images [14]. The trained convolution kernels can work as motif finders when applied to biological sequence data. DeepBIND[15] successfully predicted the sequence specificities of DNA- and RNA-binding proteins with CNN.

2.3 Recurrent Neural Network

Recurrent Neural Network (RNN) is expected to model long-range dependency by introducing recurrent edges in hidden layers of a neural network. RNN is designed to model sequential data. There is no limitation on input length. As input length gets long, the model is regarded as deep model. Vanilla RNN suffers from vanishing gradients problem as other deep neural network has. Long Short Term Memory (LSTM)[16] and Gated Recurrent Unit (GRU)[17] cells were introduced to alleviate vanishing gradients problem by adding an additive path through time steps.

In my model, I chose LSTM cell which has 3 gates-input gate i_t, forget gate f, and output gate o_t. Each gate is given input at current step t, x_t, and previous hidden state, h_{t-1}, and outputs a number ranging from 0 to 1 by applying sigmoid function. If the gate output is closed to 0, it means that the gate is "closed", blocking input flow to the corresponding gate. Basically, LSTM works as whether cell state, C_t, is maintained or modified is learned by gating system. LSTM can be presented as below,

$$f_t = \sigma(W_f[x_t, h_{t-1}] + b_f) \qquad (2)$$

$$i_t = \sigma(W_i[x_t, h_{t-1}] + b_i) \qquad (3)$$

$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_o) \quad (4)$$

$$\widetilde{C_t} = \tanh(W_C[x_t, h_{t-1}] + b_c) \quad (5)$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \quad (6)$$

$$h_t = o_t * \tanh(C_t) \quad (7)$$

where W is a weight matrix, b is s bias term and σ represents a sigmoid function. The cell state C_t is a weighted sum of previous cell state C_{t-1} and current candidate value $\widetilde{C_t}$, where the weights are the output of forget gate and input gate (6). The final output of LSTM, the hidden state h_t, is obtained by multiplying output gate value and activation function value of the cell state (7). The cell state of LSTM can remain informative by adding required information for the task of the model and removing needless past information.

RNN is frequently used in natural language processing area and also in Bioinformatics dealing with biological sequential data such as DNA sequence. Protein sequences are used as input of bi-directional RNN for prediction protein secondary structure [18].

2.4 Hybrid Neural Network

Both the pattern of histone modification marks and local dependency in spatial sequence is important in identifying enhancers. To reflect

these factors, my model is composed of ahybrid neural network in which CNN followed by RNN, dense layer and final softmax layer. CNN is expected to capture histone modification marks patterns, and RNN is expected to handle local dependency.

The histone modification marks input x_{input} is given as a $N_{input} \times N_{bin}$ matrix. A convolution layer has N_{kernel} convolution kernels whose size is $N_{input} \times k$. After convolution, there are N_{kernel} feature maps with the size of $1 \times (N_{bin} - k + 1)$. A 1D pooling with stride m is applied upon the feature maps, making smaller feature maps with the size of $1 \times \lceil (N_{bin} - k + 1)/m \rceil$. The 1D pooling is employed to maintain histone modification marks information. The RNN has h hidden states and take the output from the CNN as a sequential data with $\lceil (N_{bin} - k + 1)/m \rceil$ steps. The last output from the RNN is given to a fully connected layer followed by a softmax layer, determining whether the input is enhancer or not.

In this paper, the number of input histone marks, N_{input}, is 17 for the H1hesc cell line, and 28 for the other cell lines. The length of binned sequence, N_{bin}, is 60, which means total 3kbp (+ −1.5kbp) around the query position is given as input of the network. 1,000 convolution kernels (N_{kernel}=1,000) with size 60×5 (k=5) are used. The pooling stride, m, is 3 and, therefore, the input for the RNN has 19 steps. The RNN has 200 hidden states (h=200), resulting in a 200×2 weight matrix in the fully connected layer. In other words, there exist total

$N_{bin} \times k \times N_{kernel} + 4\left(\left\lceil \frac{N_{bin}-k+1}{m} \right\rceil \times h + h^2\right) + h \times 2$ learnable parameters (except biases) in my model.

The model is implemented with the help of TensorFlow. A loss function is defined as cross entropy loss combined with a l2-regularization term. The l2-regularization is applied on the parameters in the RNN and a final fully-connected layer with coefficient $\lambda = 0.01$. Dropout with probability 0.5 is also applied in the RNN hidden layer for regularization. The loss function l:

$$\hat{y} = \mathrm{f}(\mathrm{x}_{\mathrm{input}}), \qquad f: the\ hybrid\ neural\ networks$$

$$l = \sum_{x_{input}\ in\ mini-batch} cross\ entropy(y, \hat{y}) + \lambda \cdot l2 - normalization(f)$$

(y: a true label)

Adam optimizer of initial learning rate=0.001 is used for learning the network parameters with mini-batch size 200. Each mini-batch is shuffled before every use for effective training. The training procedure stops if the loss gets saturated.

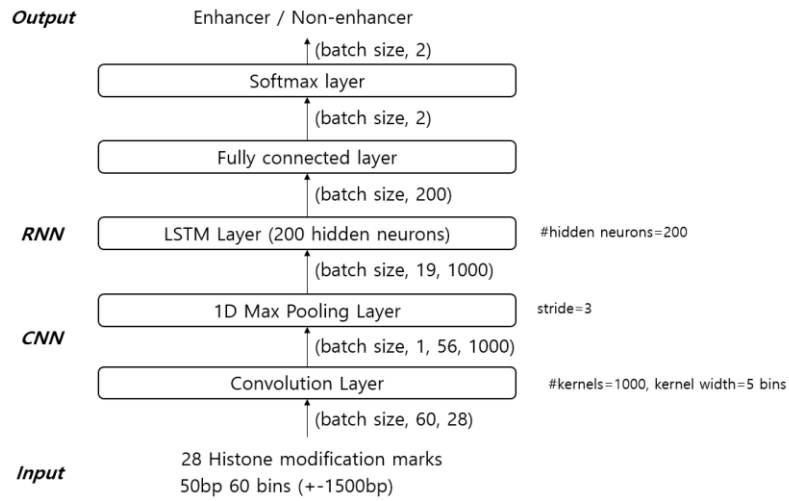

Figure 2. The hybrid neural network architecture.

2.5 Optimizing virtual histone modification marks data

After finishing training the network for the enhancer identification, a new $N_{input} \times N_{bin}$ matrix as learnable parameters can replace the input histone modification marks matrix to find an optimized input matrix for an enhancer. Other parameters are fixed except the new input matrix and the class label is also fixed as an enhancer. The new input matrix can be trained with the same loss function, cross-entropy loss and l2-regularization term, instead of feeding input samples and training the network. When the optimized virtual histone modification marks matrix was obtained, negative values were replaced with 0, and min-max

normalization was performed at the matrix. The normalized matrix is analyzed and visualized to find biological meaning in the network. This technique was used in DeepChrome[19], which models gene expression.

3. Results

3.1 Classification results

Results from the ChIP-seq data in IMR90 and H1hesc cell lines

The collected data was divided into 9:1 training set and test set, respectively. With the training set, 10-fold cross validation was performed and the validation accuracy was also examined and the model with the highest validation accuracy was chosen as a best model. Both the training set and the test set are consisted of 1:1 positive and negative samples, making them balanced set.

In both IMR90 and H1hesc cell lines, the model achieved state-of-the-art performance as presented in Table 1. The performance is shown in each training set, validation set (in the test set of 10-fold cross validation), and test set. The IMR90 dataset has 45,050 training samples 5,004 test samples. The H1hesc dataset has 10,590 training samples and 1,167 test samples.

	IMR90	H1hesc
Training Accuracy	98.00	95.50
Validation Accuracy	97.98	96.60
Validation Precision	97.15	96.30
Validation Recall	98.90	97.33
Test Accuracy	97.34	95.20
Test Precision	96.32	95.84
Test Recall	98.44	94.53

Table 1. Performance in IMR90 and H1hesc data

Results from the imputed data of H1hesc, HeLaS3. GM12878, and HepG2 cells

Another experiment with imputed histone modification marks from ChromImpute is performed to introduce the same number of input marks in H1hesc, HeLaS3. GM12878, and HepG2 cell lines. H1hesc has 12,002 training samples and 1,320 test samples. HeLaS3 has 29,636 training samples and 3,292 test samples. HepG2 has 44,586 training samples and 4,954 test samples. Lastly, GM12878 has 30,516 training samples and 3,390 test samples. The results with convolution kernel width 5 bins is presented in Table 2. The performance in both precision and recall is superb with range 95~98%.

	H1hesc	HeLaS3	GM12878	HepG2
Training Accuracy	98.00	95.50	96.50	97.50
Validation Accuracy	97.98	96.60	95.90	96.75
Validation Precision	97.15	96.30	95.66	96.52
Validation Recall	98.90	97.33	96.22	97.08
Test Accuracy	97.34	95.20	95.55	96.79
Test Precision	96.32	95.84	95.41	97.38
Test Recall	98.44	94.53	95.69	96.16

Table 2. Performance in ChromImpute samples with k=5

The results with convolution kernel width 30 bins is shown in Table 3. When the convolution kernel width is increased as 30 bins, the RNN take sequential data of 11 steps. As the number of the network parameters grows, the model showed the highest but slightly improved training accuracy and test accuracy compared to when the kernels with width of 5 bins are used. About 1~2% of improvement was observed as extra 700,000 parameters are added in the model.

	H1hesc	HeLaS3	GM12878	HepG2
Training Accuracy	98.00	99.50	98.00	99.00
Validation Accuracy	98.42	98.25	96.82	97.71
Validation Precision	99.50	97.54	96.57	96.99
Validation Recall	97.37	98.89	97.13	98.53
Test Accuracy	96.74	97.80	96.55	97.34
Test Precision	98.43	98.00	96.41	96.96
Test Recall	95.00	98.00	96.69	97.74

Table 3. Performance in ChromImpute samples with k=30

With the same number of the input marks, transferability between different cell lines can be examined, and the result is given in Table 4. Overall, the transferability seems to exist between different cell lines except when trained on HeLaS3 data. This result may be because the HeLaS3 best model was under more training iterations compared to others, getting overfitted on the HeLaS3 dataset.

	H1hesc	HeLaS3	GM12878	HepG2
H1hesc	96.74	81.06	96.97	96.06
HeLaS3	95.90	97.80	97.02	96.90
GM12878	92.71	88.88	96.55	96.11
HepG2	93.08	92.45	96.75	97.34

Table 4. Test accuracy in different cell line data. A column is a training dataset and a row corresponds to a test dataset.

3.2 Comparison with existing tools

Since each approach has different criteria for enhancers, it is hard to compare those tools altogether. DENdb[20] is a database containing putative enhancer predicted by five different computational methods- RFECS, CSIANN, ENCODE, ChromHMM, and Segway. It has genome-wide and cell line specific enhancer annotation for those 5 tools and 4 cell lines-H1hesc, HeLaS3, HepG2, and GM12878 are selected for comparison. With this database, an experiment that checks the intersection between my model and other tools has done and the result is shown in Table 5. It is already known that the overlap of different

computational enhancer prediction tools is relatively small [21]. The result with CSIANN showed the highest overlap rate. A small set of enhancers annotated by CSIANN may lead to the highest overlapping result. My model also showed considerable correspondence with other tools including ENCODE annotation which is one of the reliable chromatin annotations. In contrast, the result with ChromHMM seems to be poor. I suspect that ChromHMM has relatively large number of enhancers, resulting in a low overlapping rate. In H1hesc, where the intersection rate is only 5%, ChromHMM has 1,423,364 enhancers annotated whereas CSIANN has 8,434 enhancers annotated.

	RFECS	CSIANN	ENCODE	ChromHMM	Segway
H1hesc	42.22 (24,814/58,774)	93.13 (7,855/8,434)	48.30 (52,449/108,590)	5.08 (72,358/1,424,364)	32.25 (31,902/98,922)
HeLaS3	61.08 (24,039/39,356)	75.45 (8,409/11,145)	70.36 (44,083/62,654)	34.64 (72,101/208,145)	67.15 (35,107/52,281)
HepG2	30.27 (35,138/116,081)	84.14 (13,744/16,335)	79.41 (29,679/37,374)	13.91 (83,979/603,732)	52.21 (31,601/60,527)
GM12878	25.83 (41,933/162,344)	86.22 (3,965/4,599)	74.01 (38,776/52,393)	41.04 (68,359/166,569)	59.00 (36,162/61,292)

Table 5. Overlap enhancer prediction rate between my model and existing tools. The number of the overlapped enhancers and the total number of enhancer prediction from corresponding tools are shown in parentheses

3.3 Biological interpretation

Instead of feeding input samples and training the network, a 28x60 input matrix was trained with the fixed network and given positive output. Even though I tried many times with different model, similar optimized virtual input matrix is obtained. Two examples of the optimized input matrix are shown in Figure 3.

In the optimized virtual input marks, H2AK9ac, H3K27ac, and H3K4me2 appeared to have a peak in the middle H2AK9ac is known as an indicator for a promoter and H3K27ac is an active enhancer mark. H3K4me2 high regions represent TF binding sites. In addition, H3K27ac and H3K4me2 together is known to be a powerful evidence in predicting TF binding sites [22]. H3K27me3 is considered as inhibitory to transcription [23], and enhancers is found to transcribe eRNAs. The optimized virtual H3K27me3 has no active bins in the middle, the query position, which agrees well with the known fact.

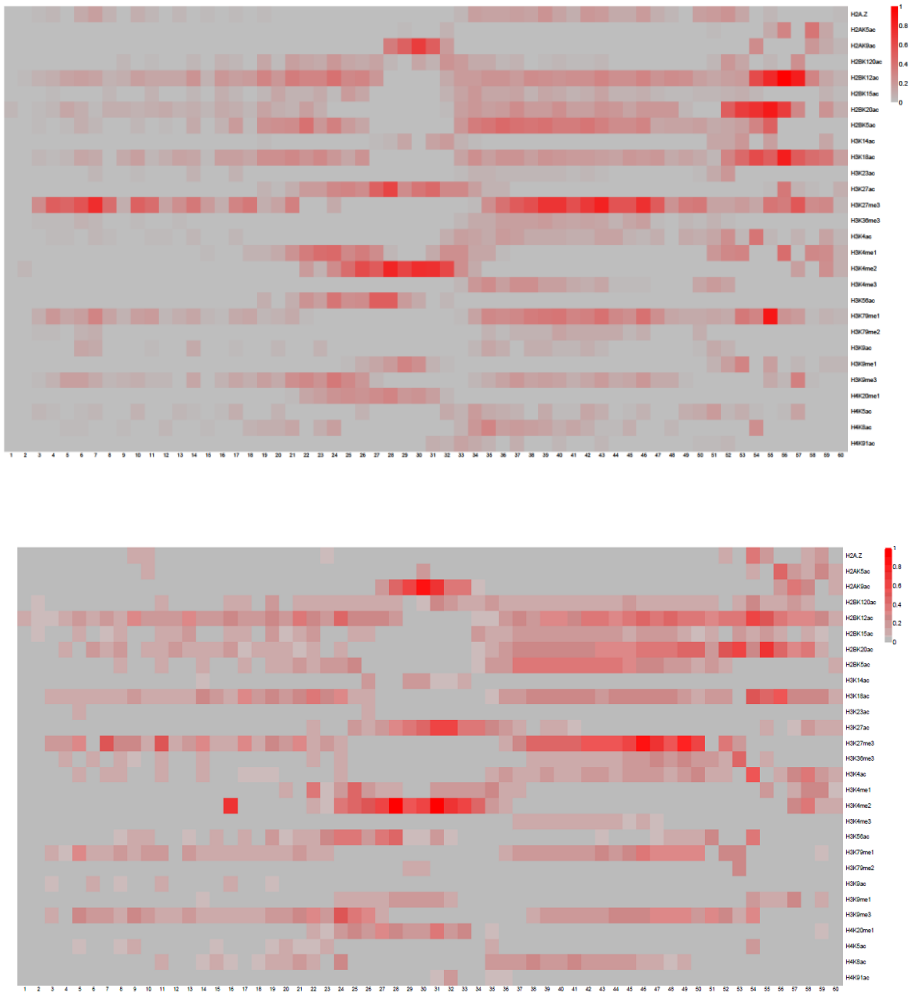

Figure 3. Examples of the optimized virtual histone modification marks matrix

4. Conclusion

In this paper, the hybrid neural network with CNN and RNN is used to model both histone modification marks patterns and local dependencies in histone spatial sequence for the enhancer identification. The model has shown superb performance with real ChIP-seq data of IMR90 and H1hesc cell lines. With imputed histone modification marks, the model achieved state-of-art accuracy and showed transferability among different cell lines as well. My model is found to corresponds ENCODE annotation well. Moreover, by training the virtual input matrix, it is confirmed that it agrees with the known traits of the histone modification marks.

Reference

[1] Fletez-Brant, Christopher, et al. "kmer-SVM: a web server for identifying predictive regulatory sequence features in genomic data sets." Nucleic acids research 41.W1 (2013): W544-W556.

[2] Yang, Bite, et al. "BiRen: predicting enhancers with a deep-learning-based model using the DNA sequence alone." Bioinformatics 33.13 (2017): 1930-1936.

[3] Zhou, Jian, and Olga G. Troyanskaya. "Predicting effects of noncoding variants with deep learning-based sequence model." Nature methods 12.10 (2015): 931.

[4] ENCODE Project Consortium. "The ENCODE (ENCyclopedia of DNA elements) project." Science 306.5696 (2004): 636-640.

[5] Hoffman, Michael M., et al. "Unsupervised pattern discovery in

human chromatin structure through genomic segmentation." Nature methods 9.5 (2012): 473.

[6] Ernst, Jason, and Manolis Kellis. "ChromHMM: automating chromatin-state discovery and characterization." Nature methods 9.3 (2012): 215.

[7] Rajagopal, Nisha, et al. "RFECS: a random-forest based algorithm for enhancer identification from chromatin state." PLoS computational biology 9.3 (2013): e1002968.

[8] Firpi, Hiram A., et al. "Discover regulatory DNA elements using chromatin signatures and artificial neural network." Bioinformatics 26.13 (2010): 1579-1586.

[9] Kleftogiannis, Dimitrios, et al. "DEEP: a general computational framework for predicting enhancers." Nucleic acids research 43.1 (2014): e6-e6.

[10] Erwin, Genevieve D., et al. "Integrating diverse datasets improves developmental enhancer prediction." PLoS computational biology 10.6 (2014): e1003677.

[11] Liu, Feng, et al. "PEDLA: predicting enhancers with a deep learning-based algorithmic framework." Scientific reports 6 (2016): 28517.

[12] Ebert, Peter, and Christoph Bock. "Improving reference epigenome catalogs by computational prediction." Nature biotechnology 33.4 (2015): 354.

[13] Zhang, Yong, et al. "Model-based analysis of ChIP-Seq (MACS)." Genome biology 9.9 (2008): R137.

[14] Ciompi, Francesco, et al. "Towards automatic pulmonary nodule management in lung cancer screening with deep learning." Scientific reports 7 (2017): 46479.

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

[16] Hochreiter, Sepp, and Jurgen Schmidhuber. "Bridging long time lags by weight guessing and “Long Short-Term Memory”." Spatiotemporal models in biological and artificial systems 37 (1996): 65-72.

[17] Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).

[18] Pollastri, Gianluca, et al. "Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles." Proteins: Structure, Function, and Bioinformatics 47.2 (2002): 228-235.

[19] Singh, Ritambhara, et al. "Deepchrome: deep-learning for predicting gene expression from histone modifications." Bioinformatics 32.17 (2016): i639-i648.

[20] Ashoor, Haitham, et al. "DENdb: database of integrated human enhancers." Database 2015 (2015).

[21] Kleftogiannis, Dimitrios, et al. "Progress and challenges in bioinformatics approaches for enhancer identification." Briefings in bioinformatics 17.6 (2015): 967-979.

[22] Wang, Ying, et al. "H3K4me2 reliably defines transcription factor

binding regions in different cells." Genomics 103.2 (2014): 222-228.

[23] Young, Matthew D., et al. "ChIP-seq analysis reveals distinct H3K27me3 profiles that correlate with transcriptional activity." Nucleic acids research 39.17 (2011): 7415-7427.

요약

인핸서(Enhancer)는 DNA의 한 영역으로, 전사인자(Transcription factor)과 결합하여 유전자 전사를 향상시키는 DNA 영역이다. 조절하는 유전자와 인접한 프로모터와는 달리, 인핸서는 조절하는 유전자와 멀리 떨어져있기 때문에 정확한 위치를 찾기 어렵다. 인핸서 영역을 찾는 문제는 생물정보학 분야에서 다양한 전산적인 방법으로 연구되어왔다. 이 논문에서는 히스톤 변이 마크(Histone modification mark) 데이터를 입력으로 하여 컨볼루션 신경망(Convolutional Neural Network)과 순환 신경망(Recurrent Neural Network)의 혼합모델을 통해 DNA 상의 인핸서를 분류하고, 가상의 입력 매트릭스를 최적화하여 인핸서 영역 판별에 영향을 주는 히스톤 변이 마크를 찾아냈다.

.....

주요어 : 인핸서, 컨볼루션 신경망, 순환신경망, 히스톤 변이 마크

학 번 : 2016-28238