Briefings in Bioinformatics, 22(3), 2021, 1–11

doi: 10.1093/bib/bbaa159
Problem Solving Protocol

DeepATT: a hybrid category attention neural network for identifying functional effects of DNA sequences

Jiawei Li, Yuqian Pu, Jijun Tang, Quan Zou and Fei Guo

Corresponding authors: Fei Guo, School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin, China. Tel.: +86 18222586975; Fax: +86 18222586975; Email: fguo@tju.edu.cn; Quan Zou, Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, China. Tel.: +86 18222586975; Fax: +86 18222586975; Email: zouquan@nclab.net; Jijun Tang, School of Computational Science and Engineering, University of South Carolina, Columbia, U.S. Tel.: +86 18222586975; Fax: +86 18222586975; Email: jtang@cse.sc.edu.

Abstract

Quantifying DNA properties is a challenging task in the broad field of human genomics. Since the vast majority of non-coding DNA is still poorly understood in terms of function, this task is particularly important to have enormous benefit for biology research. Various DNA sequences should have a great variety of representations, and specific functions may focus on corresponding features in the front part of learning model. Currently, however, for multi-class prediction of non-coding DNA regulatory functions, most powerful predictive models do not have appropriate feature extraction and selection approaches for specific functional effects, so that it is difficult to gain a better insight into their internal correlations. Hence, we design a category attention layer and category dense layer in order to select efficient features and distinguish different DNA functions. In this study, we propose a hybrid deep neural network method, called DeepATT, for identifying 919 regulatory functions on nearly 5 million DNA sequences. Our model has four built-in neural network constructions: convolution layer captures regulatory motifs, recurrent layer captures a regulatory grammar, category attention layer selects corresponding valid features for different functions and category dense layer classifies predictive labels with selected features of regulatory functions. Importantly, we compare our novel method, DeepATT, with existing outstanding prediction tools, DeepSEA and DanQ. DeepATT performs significantly better than other existing tools for identifying DNA functions, at least increasing 1.6% area under precision recall. Furthermore, we can mine the important correlation among different DNA functions according to the category attention module. Moreover, our novel model can greatly reduce the number of parameters by the mechanism of attention and locally connected, on the basis of ensuring accuracy.

Key words: DNA function; deep neural network; category attention

Introduction

Identifying functions of DNA sequences is a major challenge in the broad field of human genomics. Non-coding genetic variations constitute the majority of diseases; however, characterizing their functional effects remains a major challenge. Transcription factor (TF) binding sites are influenced by cofactor binding sequences, chromatin accessibility and structural flexibility of binding site DNA [1]. DNase I-hypersensitive

Jiawei Li is currently a master degree candidate in Tianjin University. His research interests include genomics analysis and deep learning.
Yuqian Pu is currently a master degree candidate in Tianjin University. Her research interests include proteomic analysis and deep learning.
Jijun Tang is a professor in University of South Carolina. His main research interests include computational biology and algorithm.
Quan Zou is a professor in University of Electronic Science and Technology of China. His main research interests include bioinformatics, machine learning and parallel computing.
Fei Guo is an associate professor in Tianjin University. Her research interests include bioinformatics and computational biology.
Submitted: 29 April 2020; **Received (in revised form):** 5 June 2020

sites (DHSs) and histone marks are expected to have even more complex underlying mechanisms, involving multiple chromatin proteins [2, 3]. The regulatory sequence information for non-coding variant function prediction is particularly important to have enormous benefits for functional genomics.

Nowadays, how to identify functional effects of large-scale chromatin-profiling data on TF binding, DNase I sensitivity and histone-mark profile of DNA sequences is of great significance in the current biology research. DeepSEA [4] proposed converlutional neuron network (CNN) for predicting effects of non-coding variants across multiple cell types. It only captures regulatory motifs in order to learn tissue-specific functions. DanQ [5] employed a hybrid convolutional and recurrent deep neural network (DNN) for quantifying DNA functions. It captures both regulatory motifs and the regulatory grammar. High-throughput genome sequencing data have prompted the development of novel bioinformatics tools that can integrate the large and feature-rich dataset. Thus, it can be seen that the deep learning technology is especially formidable in handling mass biomedical data and also achieves great success in a wide variety of human genomics applications.

Deep learning models are attractive and effective in identifying complex patterns from feature-rich data [6]. DNNs have already been adapted for some genomics problems, such as motif discovery [7], deleteriousness prediction of genetic variants [8], gene expression inference [9], DNA/RNA sequence binding specificities [10], DNA methylation detection [11], enhancers prediction [12] and RNA subcellular [13]. CNNs are one variant of DNNs being appropriate for this task [14]. CNNs use a weight-sharing strategy to capture local patterns in initial data such as genome sequences. This weight-sharing strategy is especially useful for studying DNA sequences because the convolution filter can capture sequence motifs with short and recurring patterns that are presumed to have a biological function. Recurrent neural networks (RNNs) are another variant of DNNs that capture sequence information through the directed connection between RNN units. This creates an internal state of network that allows to exhibit dynamic temporal or spatial behavior. Here, biRNNs [15] combine outputs of two RNNs, one processing the sequence from left to right, the other one from right to left. Instead of regular hidden units, two RNNs containing LSTM blocks are smart network units that can remember a value for various length of time [16]. One more important variation of DNNs is the attention mechanism [17] that was inspired by the brain signal processing mechanism for human. It can quickly screen out high-value information from a large amount of initial data using limited attention resources. What is more, it can solve the long-term dependency problem in the RNN. Self-attention [18] is a general form of attention mechanism. We integrate the self-attention module and make some modification to solve DNA multi-label classification problem.

In this study, we propose a hybrid DNN method, called DeepATT, for identifying non-coding DNA sequence regulatory functions. Many various DNA functions should have corresponding different representations. However, most powerful predictive models do not have specific analysis for DNA functions. Therefore, we design a category attention layer and a category dense layer in order to select corresponding features and distinguish specific representations of different DNA functions. Our DNN framework has four built-in neural network constructions, including CNN, bi-direction long-term memory RNN, category attention neural network and category dense neural network. According to the category attention layer, we can mine the correlation among different non-coding DNA functions. We compare different DNN constructions in various hyper-parameters, which are implemented or replicated on our own platform. Also, our novel method is compared with the original results of existing outstanding prediction tools.

Materials and methods

In this study, we analyze DNA sequences to predict regulatory functions through our novel hybrid DNN method, called DeepATT. We implement a DNN framework with four built-in neural network constructions, including CNN, bi-direction long-term memory RNN, category attention neural network and category dense neural network. The framework of our DNN model is shown in Figure 1.

Dataset

We apply DeepATT on the same dataset as DeepSEA and DanQ. The human GRCh37 reference genome was segmented into non-overlapping 200 bp bins for training and evaluating chromatin feature prediction performance. Targets were computed by intersecting 919 ChIP-seq and DNase-seq peak sets from uniformly processed ENCODE [19] and Roadmap Epigenomics [20] data releases. It yields a length 919 binary target vector for each sample, which consists of a 1000 bp sequence centered on each 200 bp bin overlapping at least one TF binding peak.

Training, validation and testing sets were downloaded from DeepSEA website. Samples were stratified by chromosomes into strictly non-overlapping training, validation and testing sets. The predicted probability for each sequence was computed as average of probability predictions for the forward and complementary sequence pairs. Reverse complements effectively double the size of dataset. There are 4 400 000 sequences in the training set, 8000 sequences in the validation set and 455 024 sequences in the test set. Each 1000 bp DNA sequence is represented by a 1000×4 binary matrix, with columns corresponding to A, G, C and T.

Novel model architecture

Our DNN framework has four built-in neural network constructions, including CNN, bi-direction long-term memory RNN, category attention neural network and category dense neural network. The convolution layer captures regulatory motifs, and the recurrent layer captures a regulatory grammar. Furthermore, the category attention layer captures corresponding valid feature representations for different DNA functions, and category dense layer classifies predictive labels with feature vectors selected by query vectors of different non-coding DNA functions. We propose two novel model constructions, DeepATT and DeepATT_Plus, on the basis of different weight modes in category dense layer.

CNN module

CNN [14] is designed to extract features from high-dimensional data, while keeping the number of model parameters tractable by applying a series of convolutional and pooling operations.

Here, we apply a convolution operation to above one-hot encoding representation, the formulation can be expressed as follows:

$$ConV(X)_{i,k} = ReLU\left\{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} W_{m,n}^{k} X_{i*step+m,n}\right\}, \quad (1)$$

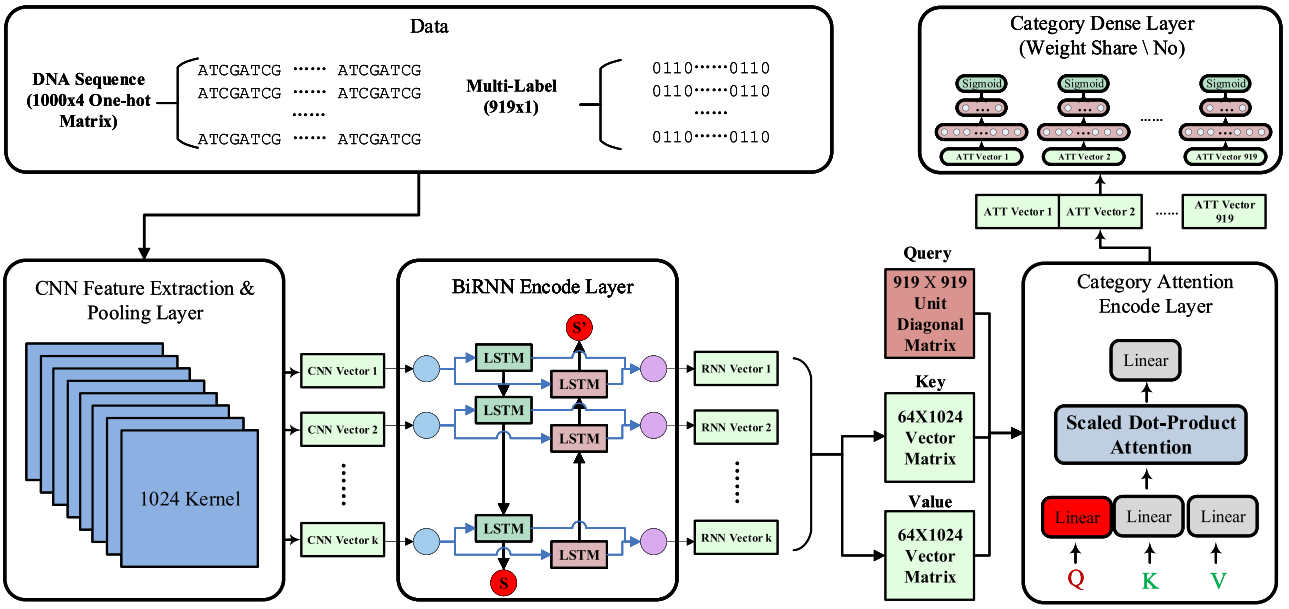

Figure 1. DeepATT: a hybrid DNN model for identifying functions of DNA sequences. Our deep learning model has four built-in neural network constructions: convolution layer captures regulatory motifs, recurrent layer captures a regulatory grammar, category attention layer selects corresponding valid features for different functions and category dense layer classifies predictive labels with selected features of regulatory functions.

where X is one-hot encoding input data, W is weight matrix, $M = 30$ and $N = 4$ and the *step* equals to 1.

The *ReLU* represents rectified linear function as follows:

$$ReLU(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}. \quad (2)$$

Furthermore, we add a max-pooling layer after the convolutional layer and also apply the dropout regularization technique. Dropout is a technique where randomly selected neurons are ignored during training. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to neurons on the backward pass. Functional effect is that the network becomes less sensitive to specific weights of neurons. This, in turn, results in a network that is capable of better generalization and is less likely to over-fit the training data. Here, keep-prob can be used as probability of keeping a neuron active during dropout.

The outputs of this module are 64 CNN vectors with 1024 dimensions. Also, we add a dropout layer after the output to prevent the over-fitting, as setting the keep-prob as 0.2.

RNN module

Long short-term memory (LSTM) [15] is an artificial RNN architecture [16], where connections between units form a directed cycle. It creates an internal state of the network that allows to exhibit dynamic temporal or spatial behavior.

Here, we build a bi-directional LSTM network, which is a variant of RNN combining outputs of two RNNs, one processing the sequence from left to right, the other one from right to left. Two RNNs contain some LSTM blocks, which can remember a value for an arbitrary length of sequence data. We get the fixed length output of time steps in two directions and merge the output of two directions to one feature vector.

The unit of LSTM is dynamically adjusted by the input sequence length. Each LSTM unit is comprised of the input gate, the forget gate and the output gate; the formulation can be expressed as follows:

$$f_t = \sigma(W_f x_t + U_f b_{t-1} + b_f) \quad (3)$$
$$i_t = \sigma(W_i x_t + U_i b_{t-1} + b_i) \quad (4)$$
$$\tilde{C} = \tanh(W_c x_t + U_c b_{t-1} + b_c) \quad (5)$$
$$C_t = i_t \circ \tilde{C}_t + f_t \circ C_{t-1} \quad (6)$$
$$o_t = \sigma(W_o x_t + U_o b_{t-1} + b_o) \quad (7)$$
$$h_t = o_t \circ \tanh(C_t) \quad (8)$$

where W and U are parameter matrices and b is a bias vector, x is the input at that particular time step, C is the cell state, h is the hidden state from previous cell or the output of previous cell, f is forget gate, i is input gate and o is output gate.

The outputs of this module are 64 RNN vectors with 1024 dimensions.

Category-attention neural network module

Self-attention mechanism [18] is built to extract global information by query, key and value form. It can solve long-term dependence problem. The attention mechanism is an improvement over the encoder–decoder-based neural machine translation system in natural language processing and other applications.

Here, we propose category-attention neural network (ATT) improved from the self-attention mechanism. We create category query code with a 919 × 919 diagonal matrix to represent the 1st stage query vector of 919 non-coding DNA sequence functions, generate the 2nd stage query vector by the linear transformation from the 1st stage query vector and achieve the multi-head attention to capture more complicated information. The multi-head attention mechanism is just split the 2nd stage query, key and value vectors into multiple pieces. Different heads

can pay attention to different kinds of information from various presentation spaces.

The self-attention layer has three inputs including query, key and value vectors; the formulation can be expressed as follows:

$$\begin{aligned} q^i &= W^q a_q^i \\ k^i &= W^k a_k^i \\ v^i &= W^v a_v^i \end{aligned} \tag{9}$$

$$\alpha_{1,i} = q^1 \cdot k^i / \sqrt{d} \tag{10}$$

$$\hat{a}_{1,i} = \exp\left(\alpha_{1,i}\right) / \Sigma_j \exp\left(\alpha_{1,j}\right) \tag{11}$$

$$b^1 = \sum_i \hat{a}_{1,i} v^i \tag{12}$$

where q is the 1st stage query vector, k is the key vector and v is the value vector, α is scaled dot-product attention, a is softmax of predefined alignment score and b is context vector for output.

For the category attention layer, vectors of q, k and v are generated from different inputs. The 1st stage query vector is a 919 × 919 diagonal matrix. The 2nd stage query vector is generated by linear transformation with weights from the 1st stage query vector. The value and key vectors are the same as above RNN vectors. We split query vector to four heads.

The outputs of this module are 919 ATT vectors with 400 dimensions. Also, we add a dropout layer after the output to prevent the over-fitting, as setting the keep-prob as 0.2.

Category-dense neural network module

Locally connected dense layer produces each output vector at each different patch of the input. We can just regard it as multiple dense layers with different inputs.

Here, we propose category dense neural network improved from the locally connected dense layer. We assign 919 dense layers to 919 ATT vectors, respectively. Also, there are two category dense modes, such as shared weight mode and unshared weight mode. For shared weight mode, different dense layers have the same weight. For unshared weight mode, different dense layers have different weights.

The normal dense neural network comes about by the non-linear combination of all extracted features, formulation can be expressed as follows:

$$Dense(X) = ReLU(WX + b), \tag{13}$$

where W is a weight matrix and b is a bias vector.

Then, we use the sigmoid output layer to obtain different probabilities of 919 non-coding DNA functions. The prediction is scaled into the 0-1 range by the sigmoid function, formulation can be expressed as follows:

$$Sigmoid(x) = \frac{1}{1 + e^{-x}}. \tag{14}$$

Different model architectures

The DNN model is organized by a sequential layer-by-layer structure executing a sequence of functional transformation. Here, we replicate two state-of-the-art models including DeepSEA and DanQ. We compare different model architectures with our novel model DeepATT. Three constructions are shown in Figure 2.

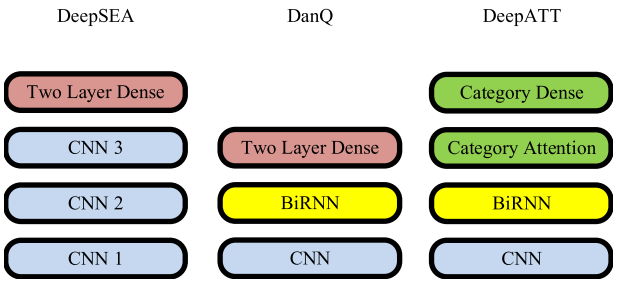

Figure 2. Different DNN constructions of DeepSEA, DanQ and DeepATT. DeepSEA has three convolution-pooling layers and two dense layers; DanQ has convolutional layer, bi-directional recurrent layer and two dense layers; DeepATT has convolutional layer, bi-directional recurrent layer, category attention layer and category dense layer.

DeepSEA

We replicate the neural network framework of DeepSEA [4] that is put forward to predict effects of non-coding variants across multiple cell types. It consists of three convolution-pooling layers and two dense layers in series, where the convolutional layer can capture regulatory motifs in order to learn tissue-specific functions. Three convolutional layers extract sequence features at different spatial scales, followed by two dense layers that can integrate information from all features extracted by front layers. In this model, we also use the regularization terms to prevent over-fitting, including dropout, L1 regularization and L2 regularization.

DanQ

We also replicate the neural network framework of DanQ and DanQ-JASPAR [5], which are hybrid convolutional and recurrent DNNs for quantifying DNA functions.

For DanQ model, it includes convolutional layer, bi-directional recurrent layer and two dense layers, which can capture both regulatory motifs and the regulatory grammar. A convolution layer with rectifier activation acts as a motif scanner across the input matrix as DeepSEA model. However, there is only one convolutional layer in DanQ model, which is different from DeepSEA model. The subsequent bi-LSTM layer considers orientations and spatial distances between various motifs.

For DanQ-JASPAR model, we just change the kernel number of convolution layer into 1024 and half of these kernels are initialized with known motifs from JASPAR [21].

Loss function

We utilize two loss functions to train the model. One is the binary cross entropy loss (NLL Loss), and the other one is focal loss.

BCE loss:

$$BCE = \begin{cases} -\log\left(y'\right) & y = 1 \\ -\log\left(1 - y'\right) & y = 0 \end{cases}, \tag{15}$$

where $y = y_{truth} \in \{0, 1\}$ and $y' = y_{pred} \in [0, 1]$.

Focal loss:

$$Focal = \begin{cases} -\alpha\left(1 - y'\right)^{\gamma} \log y' & y = 1 \\ -(1 - \alpha) y' \log\left(1 - y'\right) & y = 0 \end{cases}, \tag{16}$$

where $y = y_{truth} \in \{0, 1\}$, $y' = y_{pred} \in [0, 1]$, the balancing parameter $\alpha \in (0, 1)$ and the focusing parameter $\gamma > 0$.

It is worth noting that, focal loss is more concerned with samples that are difficult to classify and less concerned with samples that are easy to classify, which makes the proportion of difficult-to-classify samples increased. Furthermore, focal loss introduces the balance factor to solve the problem of imbalance between (0, 1) labels.

Model training and validation

All models are implemented by using Tensorflow-2.0 framework [22]. Moreover, we use GTX-2080ti to train our models. The dataset consists of training set, validation set and independent test set. We calculate the sigmoid output for 919 labels and perform the Adam algorithm [23] to minimize loss function. Here, we set learning rate and add a learning rate scheduler. Above of all, our model is fitted on the training set, hyper-parameters are optimized on the validation set and final performance and interpretation are exclusively reported on the independent test set. Because of the model complexity and the large amount of data, it takes 1–2 h for an epoch and 1–2 days for each model. In this study, we have trained 28 models to compare the predictive performance, which takes about 1 month to get experimental results.

Results

In this section, we evaluate DeepATT with other existing deep learning models in the same platform. And also, we analyze the learned motif in the convolution layer and the trained 2nd stage query vector in the category attention layer, in order to mine the correlation among 919 DNA non-coding regulatory functions. Furthermore, we compare the performance of DeepATT with original DeepSEA and DanQ.

Evaluation criteria

We calculate two metrics to evaluate the performances of different models on the test set, which is the same as DeepSEA and DanQ. One is the area under receiver operating characteristic curve (AUROC) and another is the area under precision recall (AUPR) curve. AUROC is the area under receiver operating characteristic (ROC) curve, which is created by plotting true positive rate against false positive rate at various threshold settings. AUPR is the area under curve that is created by plotting precision against recall at various threshold settings. Moreover, the AUPR statistic is a much more balanced metric than the AUROC statistic to assess performance, due to the massive class imbalance. Moreover, AV-AUROC and AV-AUPR perform overall values of AUROC and AUPR for 919 binary targets.

AUROC and AUPR provide more comprehensive and alternative measures for machine learning algorithms by being more adaptive to selected decision criterion and prior probabilities. Therefore, as same in previous methods, we choose these two metrics to evaluate the performance of different models.

Performance of various architectures

We compare the performance of five different model architectures with different hyper-parameters. We replicate the neural network framework of three existing models on our own platform, including DeepSEA, DanQ and DanQ-JASPAR. It needs to be noted that we only compare the performance of our model architectures to three existing neural network frameworks, not from previous literatures. DeepSEA [4] utilized CNN with three convolution layers and two dense layers. DanQ [5] built a hybrid convolutional and recurrent DNN with convolutional layer, bidirectional recurrent layer and two dense layers. DanQ-JASPAR only modified the convolutional layer from 512 kernels to 1024 kernels. Our DNN method has four built-in neural network constructions: convolution layer, recurrent layer, category attention layer and multiple category dense layers. DeepATT uses shared weight mode to build two category dense layers. DeepATT_Plus uses shared weight mode to build first category dense layer and uses unshared weight mode to build last category dense layer. The performance of five models are implemented or replicated on our own platform with different loss function, learning rate and scheduler.

Settings of hyper-parameters

We regard learning rate, learning rate scheduler and loss function as hyper-parameters. We discover the influence of these hyper-parameters on five model architectures. We improve the performance of different models through adjusting the learning rate and the learning rate scheduler, as shown in Table 1. It is worth noting that DeepATT can achieve the best performance with 0.39619 AV-AUPR and 0.94486 AV-AUROC on BCE loss, and 0.39522 AV-AUPR and 0.94519 AV-AUROC on focal loss, under same hyper-parameter settings 0.0005 learning rate and StepLR scheduler. Above all, comparison results demonstrate that small learning rate, StepLR learning rate scheduler and focal loss function can achieve better performance in most cases.

Numbers of parameters

More importantly, we compare the number of parameters for each neural network layer on our two novel models and three existing models, as shown in Table 2. DeepATT can reduce the number of parameters by the mechanism of attention, locally connected and weight-sharing strategy, from 10 million to million. The way to achieve this is that the attention mechanism determines relevant characteristics for each binary target and then the locally connected layer eliminates all unnecessary connections for each specific binary target. We can take advantage of shared weights; let each local connection share same weights, which greatly reduces the amount of weights. Actually, the local connection and sharing parameters are two main reasons for decreasing the number of parameters. However, DeepATT_Plus does not use weight-sharing strategy in the last layer that increases the number of parameters and causes the over-fitting phenomenon.

AUROC and AUPR

We analyze overall performance of five different models under various hyper-parameters, as shown in Figure 3. DeepSEA can achieve the best result with 0.29214 AV-AUPR and 0.90847 AV-AUROC via BCE loss. DanQ can get the best result with 0.35921 AV-AUPR and 0.93399 AV-AUROC via BCE loss. However, DanQ-JASPAR can get the best result with 0.38441 AV-AUPR and 0.94171 AV-AUROC via focal loss. In addition, focal loss can improve the performance of DanQ-JASPAR for all learning rates. DeepATT can obtain the best result with 0.39522 AV-AUPR and 0.94519 AV-AUROC, via 0.0005 learning rate, StepLR scheduler and focal loss. DeepATT_Plus can obtain the best result with 0.39324 AV-AUPR and 0.94432 AV-AUROC via BCE loss. It is noted that, small learning rate and StepLR scheduler can improve the performance of DeepATT and DeepATT_Plus. Also, focal loss can really improve

Table 1. Performance of five model architectures under different hyper-parameter settings on two loss functions

Model	Learning rate	Scheduler	BCE loss		Focal loss	
			AV-AUPR	AV-AUROC	AV-AUPR	AV-AUROC
DeepSEA*	0.0010	None	0.26140	0.89225	0.24434	0.87009
	0.0005	None	0.29214	0.90847	0.25994	0.88411
DanQ*	0.0010	None	0.33254	0.92363	0.34454	0.92875
	0.0005	None	0.35921	0.93399	0.34962	0.93160
DanQ_JASPAR*	0.0010	None	0.37443	0.93827	0.37692	0.93954
	0.0005	None	0.37872	0.94001	0.38441	0.94171
DeepATT	0.0010	None	0.38519	0.94232	0.39303	0.94332
	0.0010	StepLR	0.39304	0.94422	0.39246	0.94432
	0.0005	None	0.39267	0.94436	0.39488	0.94491
	0.0005	StepLR	**0.39619**	**0.94486**	**0.39522**	**0.94519**
DeepATT_Plus	0.0010	None	0.37768	0.93932	0.38711	0.94274
	0.0001	StepLR	0.38595	0.94271	0.38772	0.94266
	0.0005	None	0.38406	0.94293	0.38797	0.94308
	0.0005	StepLR	0.38125	0.94196	0.39324	0.94432

*Three existing deep learning constructions of DeepSEA, DanQ and DanQ_JASPAR are replicated on our own platform.

Table 2. Numbers of parameters for each neural network layer on our two novel models and three existing models

Model	Convolution layer	bi-RNN layer	Category attention layer	Category dense layer 1	Category dense layer 2	Summation
DeepATT	123 904	6 295 552	1 348 400	40 100 (weight share)	101 (weight share)	**7,808,057**
DeepATT_Plus	123 904	6 295 552	1,348 400	40 100 (weight share)	92 819	**7,900,775**
Model	Convolution layer	bi-RNN layer	Dense layer 1	Dense layer 2	Summation	
DanQ*	33 600	6 295 552	44 400 925	850 994	**46 926 479**	
DanQ-JASPAR*	123 904	6 295 552	60 621 725	850 994	**67 892 175**	
Model	Convolution layer 1	Convolution layer 2	Convolution layer 3	Dense layer 1	Dense layer 2	Summation
DeepSEA*	10 560	1 229 280	3 687 360	55 944 925	850 994	**61,723,119**

*Three existing deep learning constructions of DeepSEA, DanQ and DanQ_JASPAR are replicated on our own platform.

the performance of label-unbalanced multi-classification problem. DeepATT performs much better than other models when using same hyper-parameter settings.

What is more, we perform the statistical analysis on five different models. We calculate the mean and standard deviation of predictive results, and plot charts to represent distribution state and probability density on AUROC and AUPR values. Our best method, DeepATT, achieves 0.94519 ± 0.0456 AUROC and 0.39521 ± 0.1897 AUPR. However, previous best method, DanQ-JASPAR, obtains 0.94174 ± 0.0465 AUROC and 0.37935 ± 0.1914 AUPR. Our proposed method achieves low variability and high mean value, which effectively avoid inconsistent over-fitting.

Motif analysis

We study some important non-coding DNA functional effects for discovering functionally related motifs. Using a similar approach described in the DeepBind method [7], we convert kernels from convolution layer of DeepATT model to position-specific weight matrix (PSWM) or motifs. Then, we aligned these potential motifs to some known motifs using the TOMTOM algorithm [24]. It is a commonly used representation of patterns in biological sequences.

From 1024 motifs learned by DeepATT, hundreds of significantly potential motifs can match known motifs ($E < 0.01$). We align all motifs together into various clusters and confirm that our model can learn a large variety of informative motifs. We select three important functional effects, NRSF, EZH2 and P300, in order to demonstrate functionally related motifs. We visualize three convolution kernels with NRSF, EZH2 and P300 motif logos and display significance values of matching motif names, as shown in Figure 4. Furthermore, NRSF, EZH2 and P300 obtain 259, 242 and 254 significantly matching known motifs ($E < 0.01$) from the JASPAR2018_CORE_vertebrates_non-redundant database, respectively.

According to motif logos, we observe that kernels of different non-coding DNA functions hold various position-specific preference, which can help us to extract functionally related motifs.

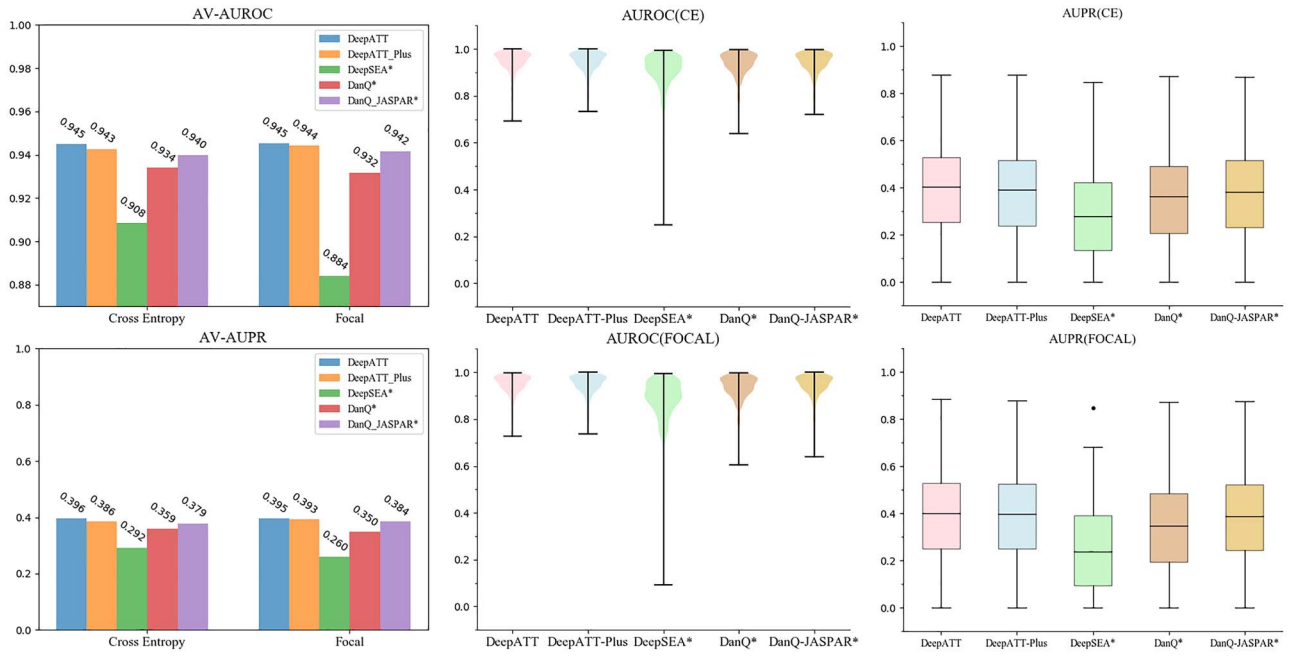

Figure 3. Overall AUPR and AUROC values of five different models. Left: bar charts represent AV-AUROC and AV-AUPR values of five models on two loss functions; medium: violin charts represent distribution state and probability density on AUROC values of five models with different loss functions; right: box plot charts represent distribution state on AUPR values of five models with different loss functions.

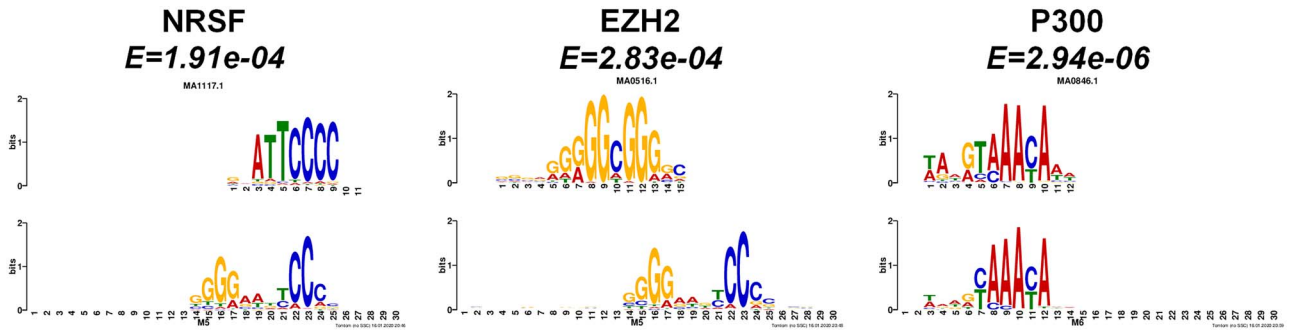

Figure 4. Three convolution kernels visualized with NRSF, EZH2 and P300 motif logos and significance values of matching motif names. NRSF, EZH2 and P300 obtain 259, 242 and 254 significantly matching known motifs ($E < 0.01$) from the JASPAR2018 database, respectively.

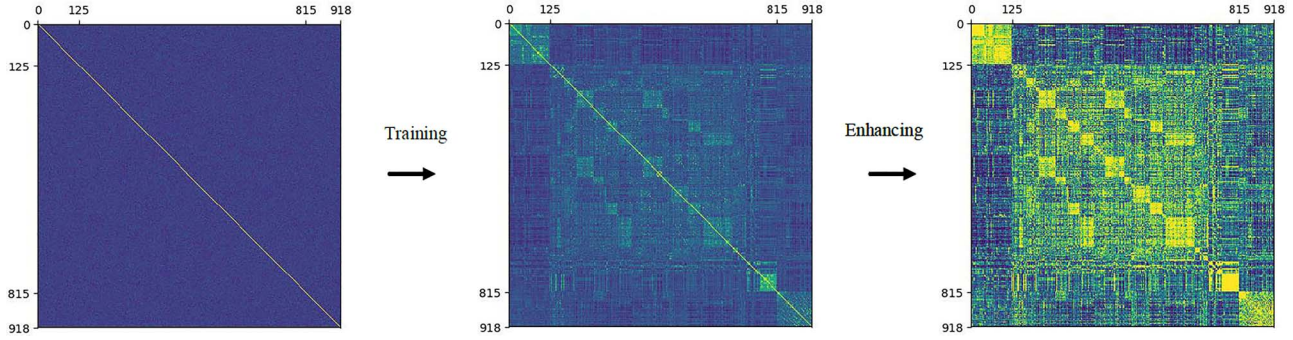

Figure 5. Annotated heatmap charts in different stages of the cosine similarity matrix between all query vectors in the category attention layer. Visualization of cosine similarity matrix for 919 chromatin features: 125 DNase features, 690 TF features and 104 histone features.

Therefore, the convolution layer can capture different position-specific information for many non-coding DNA function categories. We can take advantage of similar functionally related motifs to mine the correlation among non-coding DNA sequence regulatory functions. Given the large scope of data, we could learn the large variety of motifs on important functional effects, in order to exhaust the entire space of functionally related motifs.

Table 3. Comparison of five outstanding prediction tools for identifying functional effects of DNA sequences in the same dataset

	DeepSEA	DanQ	DanQ_J	DeepATT	DeepATT_P
AVAUPR	0.34163	0.37089	0.37936	**0.39522**	0.39324
AVAUROC	0.93260	0.93837	0.94174	**0.94519**	0.94432

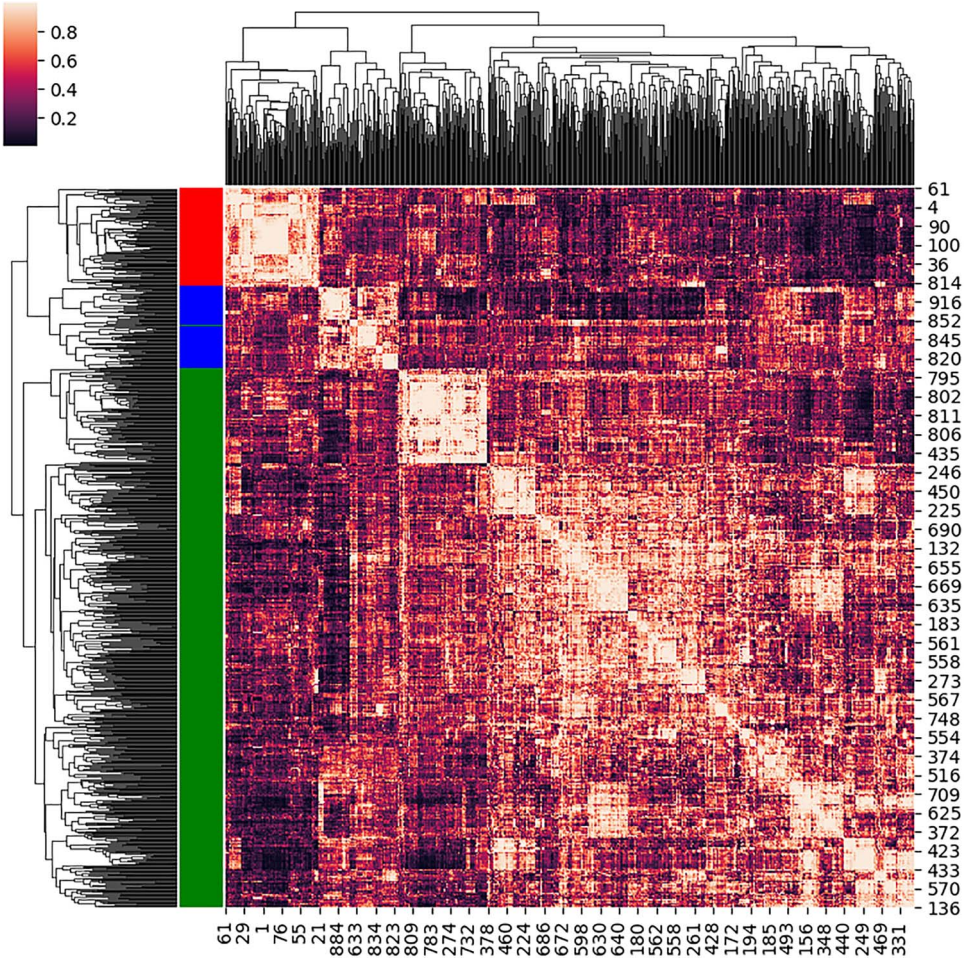

Figure 6. Dendrogram and heatmap charts for unsupervised hierarchical clustering in 919 chromatin features by corresponding 2nd stage query vectors. The red labels represent 125 DNase I sensitivity features, the green labels represent 690 TF binding features and the blue labels represent 104 histone-mark features.

Attention analysis

Since the 2nd stage query vector is trainable, we analyze all trained 2nd stage query vector in the category attention layer, in order to mine correlations among 919 DNA non-coding regulatory functions. In the category attention module, we generate a 919 × 919 diagonal matrix as the first-stage query vector to train the attention layer. First, we make use of 919 randomly generated independent 2nd stage query vector within 400 length because the kernel of linear transformation for query vector in the category attention layer is generated by Glorot uniform. We calculate the cosine similarity matrix of these randomly query vectors; however, we obtain no valid correlation information. Then, we effectively train the 2nd staged query vector in the category attention layer and calculate the cosine similarity matrix of 919 trained the 2nd stage query vector for 919 chromatin features (125 DNase features, 690 TF features and 104 histone features). Basically, we can find out some subtle correlations among the same function category. Moreover, we enhance the cosine similarity matrix by the sigmoid function. Some obvious small blocks indicate a lot of learned correlation information between 919 DNA non-coding regulatory functions. It needs to be stated that three major categories of various non-coding functions are quantified as DNase I sensitivity for 0–124 items, TF binding for 125–814 items and histone-mark profile for 815–918 items. It illustrates that the cosine similarity matrix reveals some sub-categories in the TF binding functions. We visualize the cosine similarity matrix via annotated heatmaps in different stages, as shown in Figure 5.

Furthermore, we make use of statistical data analysis to explore that various query vectors in the same group are more similar to each other, rather than to those in other groups. We plot dendrogram and heatmap for unsupervised hierarchical clustering in 919 chromatin features, as shown in Figure 6. It can be clearly seen that query vectors with the same function

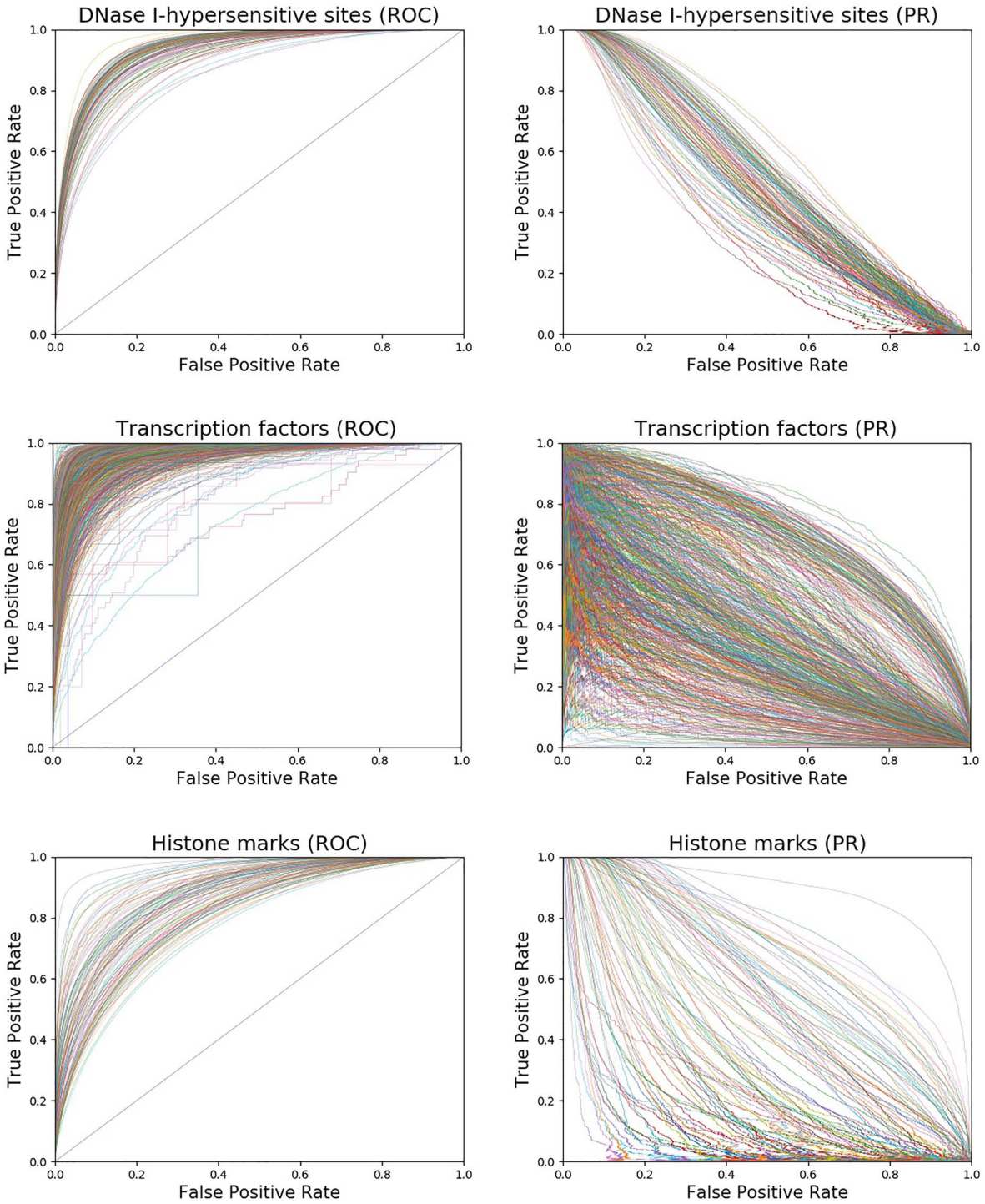

Figure 7. All AUROC and AUPR curves of DeepATT for identifying DNase I sensitivity, TF binding and histone-mark profile.

are clustered in the same category. It can be ignored that few TF binding functions are incorrectly clustered to histone-mark profile functions. To sum up, the category attention module can learn the correlation information between different DNA non-coding functions. Moreover, it will be useful to find the internal mechanism about various DNA functions. In addition, it is easy to achieve that the attention score of RNN vectors for one specific function can be calculated to estimate all functional targets for different regulatory functions.

Comparison of existing predictors

We compare our novel methods with three existing outstanding prediction tools for identifying non-coding functions of DNA sequences, as shown in Table 3. The result data of existing predictors are extracted from previous literatures [4, 5]. DeepATT achieves the best performance of 0.39522 AV-AUPR and 0.94519 AV-AUROC, which is far better than other existing non-coding DNA function prediction methods. Also, DeepATT_Plus

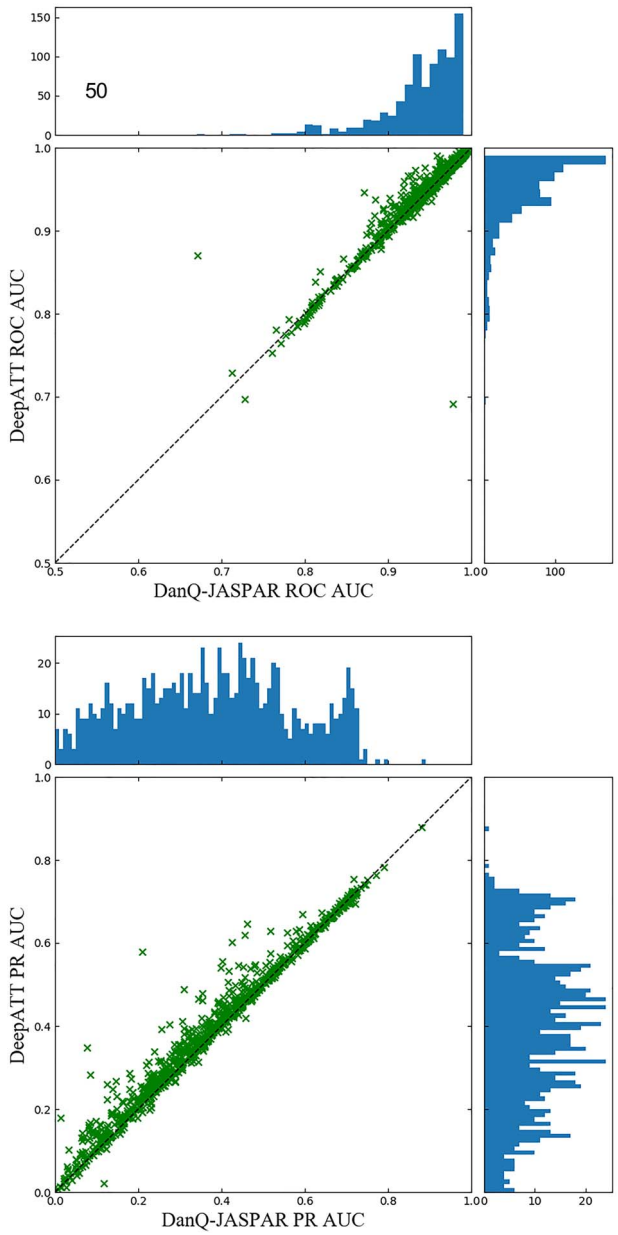

Figure 8. Scatter-plot charts for comparing AUPR and AUROC values between DeepATT and DanQ-JASPAR. The x-axis represents DanQ-JASPAR and the y-axis represents DeepATT.

obtains excellent performance of 0.39324 AV-AUPR and 0.94432 AV-AUROC. According to above results, DeepATT has a significantly improvement in the target task based on category attention layer and category dense layer.

ROC curve and PR curve

We calculate AUROC and AUPR curves of DeepATT for identifying DNase I sensitivity, TF binding and histone-mark profile, as shown in Figure 7. Obviously, DeepATT demonstrates state-of-the-art performance in identifying DNA regulatory functions, especially for identifying DHSs.

DeepATT versus DanQ-JASPAR

Also, we analyze scatter plot for comparing AUPR and AUROC values between previous best method DanQ-JASPAR and our best method DeepATT, as shown in Figure 8. For most of DNA functional predictions, the performance of DeepATT is better than that of DanQ-JASPAR which represents the state-of-the-art method before. Moreover, AUPR value of DeepATT far surpasses that of DanQ-JASPAR. It demonstrates that the category attention neural network layer significantly improves robustness, versatility and precision of DeepATT for identifying functional effects of DNA sequences.

Discussion

The contribution of our novel model can be seen as follows. First, we design two modules for feature extraction, including convolution layer and bi-directional recurrent layer. According to this task, we can capture regulatory motifs. Moreover, we design the category attention module for feature selection. According to this task, we can capture corresponding valid feature representations for different DNA non-coding regulatory functions.

Importantly, category attention improved from multi-head self-attention layer is a novel module for multi-label classification. According to the category attention module, we can mine correlations among different non-coding DNA functions. Also, category dense improved from locally connected dense layer is a module to adapt the category attention module. It can enable specific functions only to depend upon corresponding features.

Moreover, our novel model can greatly reduce the number of parameters. Attention mechanism can determine relevant features for each binary target, and locally connected layer may eliminate all unnecessary connections for specific binary target. It is a great contribution to reduce the parameter size and also ensure the prediction accuracy.

Conclusion

We propose a computational approach based on a hybrid DNN, DeepATT, for modeling the source of DNA variability. In our model, the attention mechanism is really efficient for multi-label classification of DNA non-coding function prediction. Therefore, we design a category attention layer and a category dense layer in order to select corresponding valid features and distinguish specific representations of different DNA functions. DeepATT provides novel insights into non-coding genomic regions, which contributes to understand the potential function of complex disease- or trait-associated genetic variants. DeepATT performs significantly better than other existing outstanding prediction tools for identifying DNA functions and even more reduces the parameter size on the basis of ensuring the prediction accuracy.

What is more, there are several avenues of future interests to explore. First, the attention mechanism is a really novel module for DNA function prediction task. In our model, we can calculate the attention score to approximately find the functional site in DNA sequence that is very useful for wet experiments. Second, we can add an embedding layer in the front of existing model, like word2vec [25]. It can learn internal similarities of different sub-sequences, which may improve neural network performance. Third, we can do de-redundancy for the original dataset. Current data are too massive, so it takes a few days to train one model.

Key points

- We propose a hybrid DNN method with four built-in neural network layers, DeepATT, for identifying 919 regulatory functions on nearly 5 million DNA sequences. We firstly design a category attention layer and a category dense layer in order to distinguish specific representations of different DNA functions.
- We replicate two state-of-the-art models, DeepSEA and DanQ, in order to compare different model architectures with our novel model construction in various hyper-parameters. DeepATT performs significantly better than other prediction tools for identifying DNA functions.
- Our novel model mine important correlation among different DNA functions according to the category attention module. The attention score for one specific function can be used to estimate all functional targets for different regulatory functions.
- Our novel model reduces the number of parameters by attention mechanism, locally connected layer and weight-sharing strategy. The attention mechanism determines relevant characteristics for each binary target, and the locally connected layer eliminates all unnecessary connections for specific DNA functions.

Conflict of interest

The authors declare no competing financial interest.

Funding

National Natural Science Foundation of China (61772362 and 61972280); National Key R&D Program of China (2018YFC0910405 and 2017YFC0908400).

References

1. Slattery M, Zhou T, Yang L, *et al*. Absence of a simple code: how transcription factors read the genome. *Trends Biochem Sci* 2014; **39**:381–99.
2. Benveniste D, Sonntag HJ, Sanguinetti G, *et al*. Transcription factor binding predicts histone modifications in human cell lines. *Proc Natl Acad Sci U S A* 2014; **111**(37): 13367–72.
3. Whitaker J, Chen Z, Wang W. Predicting the human epigenome from DNA motifs. *Nat Methods* 2015; **12**: 265–72.
4. Zhou J, Troyanskaya OG. Predicting effects of noncoding variants with deep learning-based sequence model. *Nat Methods* 2015; **12**(10): 931.
5. Quang D, Xie X. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Res* 2016; **44**(11): e107–7.
6. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015; **521**(7553): 436–44.
7. Alipanahi B, Delong A, Weirauch MT, *et al*. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nat Biotechnol* 2015; **33**(8): 831.
8. Quang D, Chen Y, Xie X. DANN: a deep learning approach for annotating the pathogenicity of genetic variants. *Bioinformatics* 2014; **31**(5): 761–3.
9. Chen Y, Li Y, Narayan R, *et al*. Gene expression inference with deep learning. *Bioinformatics* 2016; **32**(12): 1832–9.
10. Trabelsi A, Chaabane M, Ben-Hur A. Comprehensive evaluation of deep learning architectures for prediction of DNA/RNA sequence binding specificities. *Bioinformatics* 2019; **35**(14): i269–77.
11. Ni P, Huang N, Zhang Z, *et al*. DeepSignal: detecting DNA methylation state from Nanopore sequencing reads using deep-learning. *Bioinformatics* 2019; **35**(22): 4586–95.
12. Yang B, Liu F, Ren C, *et al*. BiRen: predicting enhancers with a deep-learning-based model using the DNA sequence alone. *Bioinformatics* 2017; **33**(13): 1930–6.
13. Yan Z, Lécuyer E, Blanchette M. Prediction of mRNA subcellular localization using deep recurrent neural networks. *Bioinformatics* 2019; **35**(14): i333–42.
14. LeCun Y, Bottou L, Bengio Y, *et al*. Gradient-based learning applied to document recognition. *Proc IEEE* 1998; **86**(11): 2278–324.
15. Graves A, Fernández S, Schmidhuber J. Multi-dimensional recurrent neural networks. In: *International Conference on Artificial Neural Networks*. Porto, Portugal: Springer, 2007, 549–58.
16. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* 1997; **9**(8): 1735–80.
17. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:14090473. 2014.
18. Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. In: *Advances in Neural Information Processing Systems*. Long Beach, CA, USA: The MIT Press, 2017, 5998–6008.
19. The ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature* 2012; **489**(7414): 57.
20. Kundaje A, Meuleman W, Ernst J, *et al*. Integrative analysis of 111 reference human epigenomes. *Nature* 2015; **518**(7539): 317.
21. Mathelier A, Fornes O, Arenillas DJ, *et al*. JASPAR 2016: a major expansion and update of the open-access database of transcription factor binding profiles. *Nucleic Acids Res* 2015; **44**(D1): D110–5.
22. Abadi M, Agarwal A, Barham P, *et al*. Tensorflow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:160304467. 2016.
23. Kingma D, Ba J. Adam: a method for stochastic optimization. *ICLR 2015*. San Diego, CA: Ithaca, 2014.
24. Gupta S, Stamatoyannopoulos JA, Bailey TL, *et al*. Quantifying similarity between motifs. *Genome Biol* 2007; **8**(2): R24.
25. Mikolov T, Chen K, Corrado G, *et al*. Efficient estimation of word representations in vector space. arXiv preprint arXiv:13013781. 2013.