



OPEN ACCESS

EDITED BY

Lei Chen,
Shanghai Maritime University, China

REVIEWED BY

Guoxian Yu,
Shandong University, China
Qi Dai,
Zhejiang Sci-Tech University, China
Hongtao Lu,
Shanghai Jiao Tong University, China

*CORRESPONDENCE

Jianhua Jia,
✉ jjh163yx@163.com
Rufeng Lei,
✉ rufeng_lei@163.com

SPECIALTY SECTION

This article was submitted to
Computational Genomics,
a section of the journal
Frontiers in Genetics

RECEIVED 26 December 2022

ACCEPTED 13 February 2023

PUBLISHED 01 March 2023

CITATION

Jia J, Lei R, Qin L, Wu G and Wei X (2023),
iEnhancer-DCSV: Predicting enhancers
and their strength based on DenseNet
and improved convolutional block
attention module.
Front. Genet. 14:1132018.
doi: 10.3389/fgene.2023.1132018

COPYRIGHT

© 2023 Jia, Lei, Qin, Wu and Wei. This is
an open-access article distributed under
the terms of the [Creative Commons
Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use,
distribution or reproduction in other
forums is permitted, provided the original
author(s) and the copyright owner(s) are
credited and that the original publication
in this journal is cited, in accordance with
accepted academic practice. No use,
distribution or reproduction is permitted
which does not comply with these terms.

iEnhancer-DCSV: Predicting enhancers and their strength based on DenseNet and improved convolutional block attention module

Jianhua Jia^{1*}, Rufeng Lei^{1*}, Lulu Qin¹, Genqiang Wu¹ and Xin Wei²

¹School of Information Engineering, Jingdezhen Ceramic University, Jingdezhen, China, ²Business School, Jiangxi Institute of Fashion Technology, Nanchang, China

Enhancers play a crucial role in controlling gene transcription and expression. Therefore, bioinformatics puts many emphases on predicting enhancers and their strength. It is vital to create quick and accurate calculating techniques because conventional biomedical tests take too long time and are too expensive. This paper proposed a new predictor called iEnhancer-DCSV built on a modified densely connected convolutional network (DenseNet) and an improved convolutional block attention module (CBAM). Coding was performed using one-hot and nucleotide chemical property (NCP). DenseNet was used to extract advanced features from raw coding. The channel attention and spatial attention modules were used to evaluate the significance of the advanced features and then input into a fully connected neural network to yield the prediction probabilities. Finally, ensemble learning was employed on the final categorization findings *via* voting. According to the experimental results on the test set, the first layer of enhancer recognition achieved an accuracy of 78.95%, and the Matthews correlation coefficient value was 0.5809. The second layer of enhancer strength prediction achieved an accuracy of 80.70%, and the Matthews correlation coefficient value was 0.6609. The iEnhancer-DCSV method can be found at <https://github.com/leirufeng/iEnhancer-DCSV>. It is easy to obtain the desired results without using the complex mathematical formulas involved.

KEYWORDS

enhancer, DenseNet, channel attention, spatial attention, ensemble learning

1 Introduction

Genes are functional areas of an organism's DNA (Dai et al., 2018; Kong et al., 2020) that hold genetic information. The gene is transferred to the protein through a sequence of transcription (Maston et al., 2006) and translation (Xiao et al., 2016), and proteins control the organism's exterior phenotypic shape (Buccitelli and Selbach, 2020). Transcription is one of the most crucial aspects of gene expression. The enhancer and promoter (Cvetesic and Lenhard, 2017) are the most significant sequence regions for transcriptional activity. An enhancer is a brief non-coding DNA fragment on DNA (Kim et al., 2010) and controls rapid and slow gene expression (Shrinivas et al., 2019). According to previous studies, several illnesses (Yang et al., 2022) are produced as a result of enhancer mutations and deletions (Emission et al., 2005; Liu G. et al., 2018; Boyd et al., 2018; Wu et al., 2019). In terms of the

activities they express, the enhancers may be categorized into groups, such as strong and weak enhancers, closed (balanced) enhancers, and latent enhancers (Shlyueva et al., 2014). Therefore, understanding and recognizing these specific gene sequence segments is an urgent problem (Pennacchio et al., 2013).

Traditional medical experimental methods (Yang et al., 2020) in bioinformatics are costly and time-consuming. Therefore, it is crucial to develop computational techniques and derive some excellent predictors (Firpi et al., 2010; Fernández and Miranda-Saavedra, 2012; Erwin et al., 2014; Ghandi et al., 2014; Kleftogiannis et al., 2015; Lu et al., 2015; Bu et al., 2017; Yang et al., 2017). However, these techniques have limitations in the prediction of strong and weak enhancers. Liu et al. (2015) developed a predictor called iEnhancer-2L based on the support vector machine (SVM) algorithm and used the sequence pseudo-K-tuple nucleotide composition (PseKNC) approach to encode features. Afterward, machine learning-based methods were applied to the prediction of enhancers, such as SVM (Jia and He, 2016; He and Jia, 2017), RF (Singh et al., 2013; Wang et al., 2021), and XGBoost (Cai et al., 2021), and many excellent predictors have been created. However, a single machine learning classifier has obvious performance drawbacks. A predictor based on an ensemble learning model (Liu B. et al., 2018) was developed to address this problem, which generally has a significantly better performance. The ensemble learning model has diversity and complexity in feature processing. For instance, Wang C. et al. (2022) developed a predictor called Enhancer-FRL, which used 10 feature methods for feature coding. The manual creation of feature coding is a relatively difficult problem, and the presence of many complex feature coding types can lead to dimensional disasters. Furthermore, the effectiveness of conventional machine learning models depends on the extracted complex features. Consequently, the development of a predictor that requires only simple features is crucial.

Nowadays, deep learning is becoming increasingly popular. Nguyen et al. (2019) proposed the iEnhancer-ECNN model based on convolutional neural networks (CNNs). Niu et al. (2021) proposed a model called the iEnhancer-EBLSTM based on bi-directional long short-term memory (Bi-LSTM). They used one-hot and K-mers coding techniques to encode the enhancer sequences and then fed these features into the deep learning network to get relatively good prediction results. For example, in the iEnhancer-ECNN model, the ACC and MCC of enhancer recognition results were 0.769 and 0.537, and the ACC and MCC of enhancer strength prediction results were 0.678 and 0.368, respectively. However, there is a wide gap in prediction precision using a better deep learning model.

In deep learning networks, CNNs with more convolutional layers extract more advanced local features but lead to the problem of gradient disappearance and network degradation. To solve this problem, the residual neural network (ResNet) (Li et al., 2022) uses a short-circuit connection structure, which allows the convolutional layers to be connected several layers apart and can solve the problem of network degradation to some extent. However, the densely connected convolutional network (DenseNet) (Huang et al., 2010) has been enhanced based on ResNet. DenseNet extracts richer feature information by reusing the features of each previous layer, and it is more effective than ResNet. The attention model is also increasingly used, and the essence of the attention model is to

TABLE 1 Specifics of the benchmark dataset.

Layer	Original dataset	Enhancer	Non-enhancer
First layer	Training dataset	1,484	1,484
	Testing dataset	200	200
Second layer	Original dataset	Strong enhancers	Weak enhancers
	Training dataset	742	742
	Testing dataset	100	100

focus on more useful feature information and suppress useless feature information. Convolutional block attention module (CBAM) (Zhang et al., 2022) can focus on more useful feature information from channel and spatial dimensions. The current computational method has the disadvantages of poor performance and complex features. For this purpose, we developed a new predictor called iEnhancer-DCSV. The predictor is conducted using a modified DenseNet and an improved CBAM attention module. The DenseNet framework makes it easier to extract more advanced features. Experimental results show that our model outperforms the existing models. The iEnhancer-DCSV model is currently the optimal choice for predicting enhancers and their strengths.

2 Materials and methods

2.1 Benchmark dataset

The benchmark dataset was created by Liu et al. (2015). They took the enhancer fragments from nine cell lines, removed 80% of the redundant sequences with the CD-HIT (Huang et al., 2010) and then calculated the ideal fragment length of 200 bp for each enhancer sequence to create the final dataset. The dataset is split into two sections: a training dataset for the model's training and an independent test dataset for model testing. The independent test dataset is made up of 200 enhancer samples (with 100 strongly and 100 weakly enhancer samples) and 200 non-enhancer samples, whereas the training dataset is made up of 1,484 enhancer samples (with 742 strongly and 742 weakly enhancer samples) and 1,484 non-enhancer samples. All enhancer samples in the independent test dataset were different from the training dataset to guarantee that the samples are independent. The benchmark dataset is described in Table 1 and may be downloaded conveniently from the website: <https://github.com/leirufeng/iEnhancer-DCSV>.

2.2 Feature coding schemes

Two simple and effective coding techniques are used in this study: one-hot and NCP. Notably, these two coding techniques produce columns with a dimension of 200, so they can be feature-combined. For instance, an enhancer sequence with a length of 200 bp can obtain a 4×200 feature matrix and a 3×200 feature

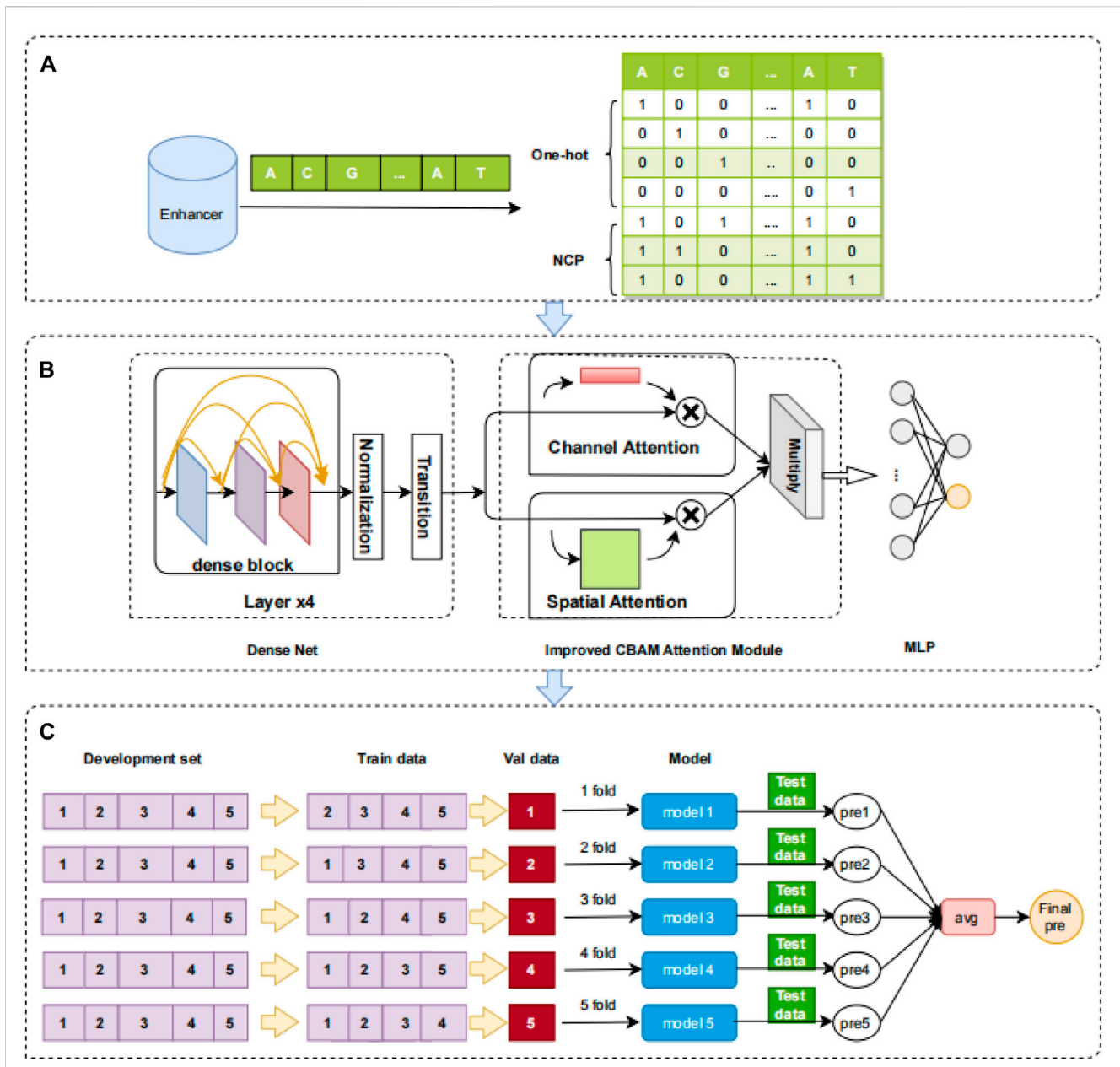


FIGURE 1 Overview of the iEnhancer-DCSV model. **(A)** Feature coding. One-hot and NCP are used to encode the enhancer sequence, and a 7×200 matrix is produced. **(B)** Framework of the iEnhancer-DCSV model. The original features are input directly to the modified DenseNet structure (which includes four dense blocks, normalized layers, and transition layers), and the improved structure is used to extract advanced features. Modules for spatial and channel attention are introduced to assess the extracted advanced features' importance. The two evaluated advanced feature maps are multiplied together at the corresponding positions. The fully connected neural network is used to output the prediction probabilities. **(C)** Ensemble model. The model uses fivefold cross-validation, where each fold is tested using an independent test set, each test enhancer sequence generates five prediction probabilities, and the final classification is voted using ensemble learning.

matrix after one-hot and NCP coding, respectively. Finally, combining these two matrices through feature fusion can yield a 7×200 feature matrix. In this study, the enhancer sequence is considered a gray image by the feature coding matrix. The 7×200 matrix is directly used as the original feature input.

2.2.1 One-hot coding

In the field of bioinformatics, one-hot coding is one of the most used coding techniques. The advantages of this coding technique are

its feasibility, efficiency, and ability to assure that each nucleotide letter is coded independently. The method is effective in avoiding the expression of interdependencies. This coding technique is particularly popular in bioinformatics. The double helix structure (Sinden et al., 1998) of DNA is widely known, and it is made up of four nucleotides: A (adenine deoxyribonucleotide), C (cytosine deoxyribonucleotide), G (guanine deoxyribonucleotide), and T (thymine deoxyribonucleotide) (Chou, 1984). The enhancer sequences are DNA sequences designated "0,1,2,3" in the order

TABLE 2 Nucleotide chemical property.

Chemical property	Category	Nucleotide
Ring structure	Purine	A, G
	Pyrimidine	C, T
Functional group	Amino	A, C
	Keto	G, T
Hydrogen bonding	Strong	C, G
	Weak	A, T

“ACGT.” The nucleotides in the sequences are then coded, and the coding length is four nucleotides. The coding elements are 0 and 1. The position corresponding to the nucleotide letter marker is coded as 1, and the other positions are coded as 0. For instance, “A” is coded as (1,0,0,0), “C” is coded as (0,1,0,0), “G” is coded as (0,0,1,0), and “T” is coded as (0,0,0,1) (Zhang et al., 2022). The one-hot coding is shown in Figure 1A.

2.2.2 NCP coding

The four DNA nucleotides are structurally different from each other and have different chemical molecular structures (Zhang et al., 2022). For instance, C and T contain one loop each, whereas A and G have two loops between the four nucleotides. G and T may be classified as ketone groups from the standpoint of chemical composition, whereas A and C can be classified as amino groups. A and T have two hydrogen bonds, but C and G have three hydrogen bonds. The strength between C and G is more powerful than that between A and T. The specific chemical properties between nucleotides are shown in Table 2.

Then, coding is performed based on the chemical characteristics. The nucleotide N_i is located in position i in the sequence. Three chemical characteristics of nucleotide N_i are “ring structure,” “functional group,” and “hydrogen bond strength” (Xiao et al., 2019). The vector representation of $N_i = (x_i, y_i, z_i)$, x_i, y_i, z_i is expressed as

$$\begin{cases} x_i = \begin{cases} 1, & \text{if } N_i \in \{A, G\}, \\ 0, & \text{if } N_i \in \{C, T\}, \end{cases} \\ y_i = \begin{cases} 1, & \text{if } N_i \in \{A, C\}, \\ 0, & \text{if } N_i \in \{G, T\}, \end{cases} \\ z_i = \begin{cases} 1, & \text{if } N_i \in \{A, T\}, \\ 0, & \text{if } N_i \in \{C, G\}. \end{cases} \end{cases} \quad (1)$$

A, C, G, and T may be encoded using this approach as (1,1,1), (0,1,0), (1,0,0), and (0,0,1). NCP coding is shown in Figure 1A.

2.3 Model construction

In this study, we constructed a network framework to automatically learn advanced features called iEnhancer-DCSV. The framework of iEnhancer-DCSV is divided into three parts: (A) feature coding, (B) framework of iEnhancer-DCSV model, and (C) ensemble model. The details are shown in Figure 1.

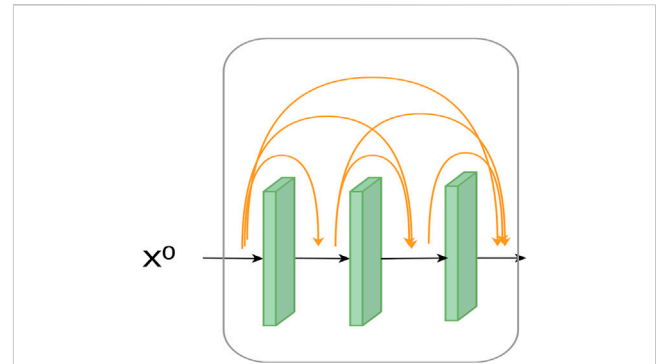


FIGURE 2 Structure of a dense block.

2.3.1 DenseNet

In this study, we modified the initial DenseNet structure. The original DenseNet consists of a convolutional layer, a dense block layer, and a transition layer. First, convolution is applied to the original features. Then, the convolution features are processed by the dense block and transition layers. The dense block layer is a dense connection of all the preceding layers to the following layers. In particular, each layer accepts all its preceding layers as its additional input, enabling feature reuse. The transition layer, which mainly connects two adjacent dense blocks, reduces the feature map size. Instead, we deleted the first convolutional layer and added a batch normalization layer between the dense block layer and the transition layer. This processing method can extract better-quality feature information and reduce the risk of overfitting.

2.3.1.1 Dense block

The traditional CNN network does not perform very well in extracting feature information. A convolutional structure called dense convolutional block extracts richer feature information by reusing previous features. Experimentally, the dense convolutional network feature extraction is proven better than traditional CNN. The structure diagram is shown in Figure 2.

In the dense block, the input of layer i is related to not only the output of layer $i - 1$, but the output of all the previous layers. The X_l level is represented as follows:

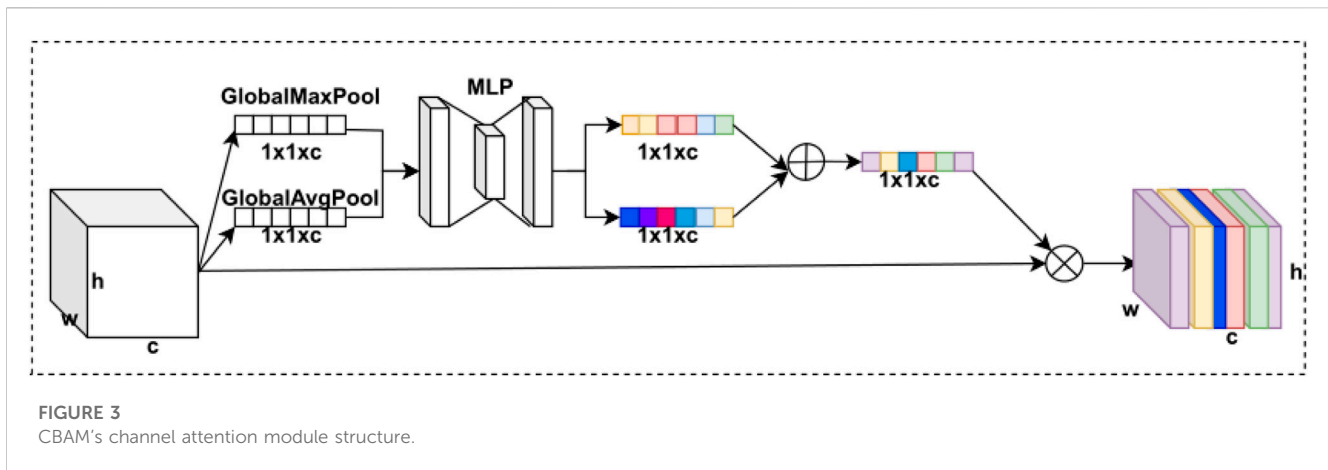
$$X_l = H_l ([X_0, X_1, X_2, \dots, X_{l-1}]), \quad (2)$$

where is denoted as layers X_0 to X_{l-1} stitched together by the dimension of the channel. H is a non-linear combinatorial function. It is a combination of batch normalization, ReLU activation function, and convolution (3×3).

In this study, we used four dense blocks, each containing three layers of convolution. The final extraction of features was $X^{(seq)}$.

2.3.1.2 Transition layer

The $l-1$ layers in front of the dense block are combined by channel dimension. As the number of channels in the l -layer becomes larger, it leads to an explosion of parameters, along with a slow training speed. We can improve the efficiency by connecting a transition layer with the dense block layer. The transition layer



consists of a 1×1 convolution and a 2×2 average pooling. It is a function of reducing the number of channels and parameters in the dense block layer by downsampling to compress the model.

2.3.2 Batch normalization

Gradient explosion and gradient disappearance are serious problems in deep learning training, and this phenomenon tends to occur more likely in the deeper network structure. If the shallow parameters are changed, their fluctuations during backpropagation may be significant, resulting in significant variable shifts in the deeper network. Batch normalization (Min et al., 2016) has been shown to improve the generalization ability of the model. The batch normalization is expressed as follows:

$$\tilde{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 - \epsilon}}, \tag{3}$$

$$y_i = \gamma \tilde{x}_i + \beta, \tag{4}$$

where A is the set of the feature dataset $[x_1, x_2, \dots, x_i]$, μ is the mean of dataset A , and σ^2 is the variance of dataset A . γ and β are trainable parameters.

2.3.3 Improved CBAM attention module

The CBAM attention module comprises channel attention and spatial attention modules (Chen et al., 2017). First, we use the channel attention module to evaluate the original features. Second, we take the feature map output from the channel attention module and feed it back into the spatial attention module. Finally, we output the final feature maps from the spatial attention module. This serial connection of CBAM attention modules has the disadvantage that the attention modules are all computed in a specific way, and the computation of weights destroys the feature shape of the input. This leads to inaccurate weight calculation of the spatial attention modules and loss of channel weighting information in the final feature map. We change the original serial approach in the CBAM attention module to a parallel method. The principle is to input the original features into the channel attention module and the spatial attention module and let the output features be multiplied by their corresponding positions. By this method, the effect of each attention model after evaluation can be maximally preserved and the expressiveness of the features can be improved.

2.3.3.1 Channel attention module

In deep learning, the degree of importance varies between different feature map channels, so we use the channel attention module to calculate different weights for each channel. By weighting each channel of the feature map, the model automatically pays attention to the more useful channel information to achieve the fixation of channel dimension and compression of spatial dimension. The channel attention module comprises the max pooling layer, the average pooling layer, the MLP module, and the sigmoid activation function. The CBAM's channel attention module structure is shown in Figure 3.

The channel attention module starts with the feature map passing through two parallel max pooling and average pooling layers, which are input into the fully connected neural network (MLP) module separately. Second, the two results of the MLP output are summed element by element, and the channel attention module weights are obtained using the sigmoid activation function. Finally, these weights are multiplied by the feature map to obtain the feature map of the channel attention model weighting. The CBAM's channel attention model is expressed as follows:

$$M_c(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))), \tag{5}$$

$$F = F_{scale}(F, M_c(F)) = M_c(F) \cdot F, \tag{6}$$

where pooling here is the global max pooling and the global average pooling. $F_{scale}(F, M_c(F))$ denotes each channel-specific value of F multiplied by the weight $M_c(F)$.

2.3.3.2 Spatial attention module

In deep learning, different receptive fields have different degrees of value to the feature map, so we use a spatial attention model to calculate the weights between receptive fields. By weighting the receptive fields, we allow the model to focus on the more useful target location information to achieve a constant spatial dimension and a compressed channel dimension. The spatial attention model is implemented through a max pooling layer, an average pooling layer, a CNN module, and a sigmoid activation function. The CBAM's spatial attention module structure is shown in Figure 4.

The spatial attention model first passes the feature maps through two parallel max pooling and average pooling layers and performs a

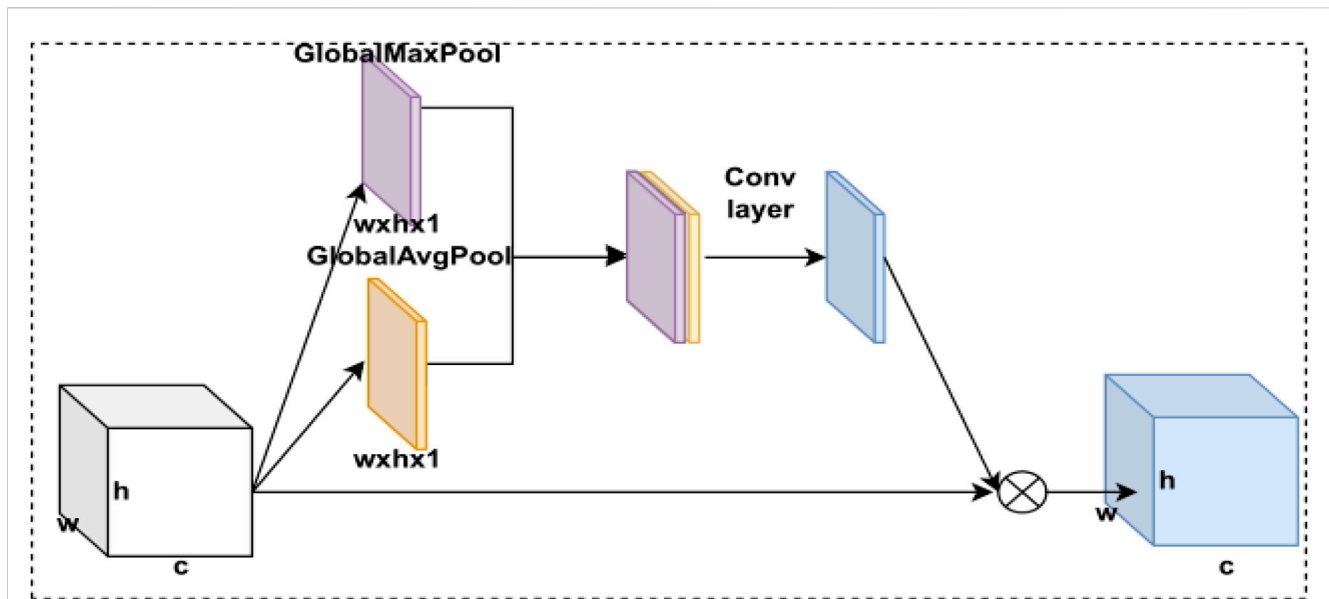


FIGURE 4
CBAM's spatial attention module structure.

stitching operation on the two pooling feature maps. Then, the newly obtained features are input into the CNN module to be transformed into a feature map with channel number 1, and the spatial attention module weights are obtained by the sigmoid activation function. Finally, this weight is multiplied by the feature map to obtain the weighted feature map of the spatial attention model. The CBAM's spatial attention model is expressed as follows:

$$M_s(F) = \sigma(f^{7 \times 7}([AvgPool(F); MaxPool(F)])), \quad (7)$$

$$F = F_{scale}(F, Ms(F)) = Ms(F) \cdot F. \quad (8)$$

Pooling here is the global max pooling and global average pooling. The size of the convolutional kernel used in the CNN module is 7×7 . Finally, $F_{scale}(F, Mc(F))$ denotes each receptive field of F multiplied by the weight $Mc(F)$.

2.3.4 Fully connected neural network

We used a fully connected neural network (Wang, et al., 2022b) to predict the enhancers and their strength. After we extracted the advanced features, the size of the advanced features was reduced using a pooling layer. Then, these features are flattened into vectors, which are later input into the fully connected neural network. Finally, the softmax function is used to calculate the predicted probability of the enhancers. The softmax formula is expressed as

$$P(y = i|x) = \frac{e^{W_i^s \cdot X}}{\sum_{j=1}^C e^{W_j^s \cdot X}}, \quad (9)$$

where W_i^s and W_j^s denote the weights in the fully connected neural network, X denotes the sample, and C is the number of categories. $P(y = i|x)$ denotes the probability that x is predicted to be i . This is a dichotomous problem, $i = 0$ or $i = 1$.

2.3.5 Ensemble model

There is an ensemble method called bagging (Bauer and Kohavi, 1999). It is accomplished by training several different models, allowing independent test data to calculate the predicted results using different models and then averaging them. This ensemble learning approach is called model averaging. The advantage of model averaging is that different models do not usually produce the same error on the test data, and it is a very powerful method for reducing generalization errors.

In this study, we used a fivefold cross-validation method (Shang et al., 2022). The training dataset was divided into five parts: four for training and one for validation. We used an independent test set put into each fold in cross-validation, by which five predictions are obtained. Finally, the final prediction results are obtained by the voting method. The ensemble method is shown in Figure 1C.

2.4 Performance evaluation

Scientific evaluation metrics are a measure of model performance. In this study, the evaluation of model performance contains four metrics: sensitivity (Sn), specificity (Sp), accuracy (Acc), and Mathew's correlation coefficient (MCC) (Sokolova and Lapalme, 2009). The specific calculation formula is shown as follows:

$$\left\{ \begin{aligned} Sp &= \frac{TN}{TN + FP}, \\ Sn &= \frac{TP}{TP + FN}, \\ Acc &= \frac{TP + TN}{TP + TN + FP + FN}, \\ MCC &= \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \end{aligned} \right. \quad (10)$$

TABLE 3 Comparison results of different coding schemes.

Layer	Coding	SN (%)	SP (%)	Acc (%)	MCC	AUC
First layer	One-hot	81.70	75.50	78.60	0.5737	0.8275
	NCP	83.25	70.50	76.88	0.5428	0.8168
	One-hot + NCP	80.25	77.65	78.95	0.5809	0.8527
Second layer	One-hot	60.30	72.80	66.55	0.3418	0.7491
	NCP	90.50	53.40	71.95	0.4780	0.7666
	One-hot + NCP	99.10	62.30	80.70	0.6609	0.8686

TABLE 4 Comparison with different architecture methods at layer 1 (enhancer recognition).

Model framework	SN (%)	SP (%)	Acc (%)	MCC	AUC
ResNet	69.80	77.90	73.85	0.4927	0.8211
DenseNet	83.10	68.50	75.80	0.5219	0.8108
DenseNet + channel attention	78.20	78.35	78.27	0.5686	0.8316
DenseNet + spatial attention	78.75	78.20	78.48	0.5717	0.8304
DenseNet + CBAM attention	83.70	67.25	75.48	0.5183	0.8046
DenseNet + improved CBAM attention	80.25	77.65	78.95	0.5809	0.8527

where TP, TN, FP, and FN are the four metrics in the confusion matrix, representing true positive, true negative, false positive, and false negative, respectively (Niu et al., 2021). In addition, we added the ROC curve area AUC metric (Vacic et al., 2006) to evaluate the model, and higher values of these metrics indicate better model performance.

3 Results and discussion

3.1 Construction of the first layer (enhancer recognition) model

The recognition of enhancers in the first layer is very important to complete the prediction mission. For the first layer of enhancer recognition, we used the iEnhancer-DCSV network framework. The advanced feature extraction and weight assignment are performed automatically by the model's iEnhancer-DCSV network framework. First, the enhancer sequences are encoded using the one-hot and NCP methods, and then feature coding is fed into the DenseNet to extract advanced features. These advanced features are input into the channel attention module and the spatial attention module, respectively. The two evaluated advanced feature maps are multiplied at the corresponding positions, and then the pooling layer is used to compress the feature size. Finally, a fully connected neural network is used to derive the predicted probabilities. We validate the model by putting independent test sets into each fold of the fivefold cross-validation. The aforementioned five-time results are passed through a soft voting mechanism to arrive at the final prediction. The whole process was cycled 10 times to verify the stability of the model, and the obtained individual performance

metrics were averaged. The experimental results for SN, SP, Acc, and MCC were 80.25%, 77.65%, 78.95%, and 0.5809, respectively.

3.2 Construction of the second layer (strong and weak enhancer prediction) model

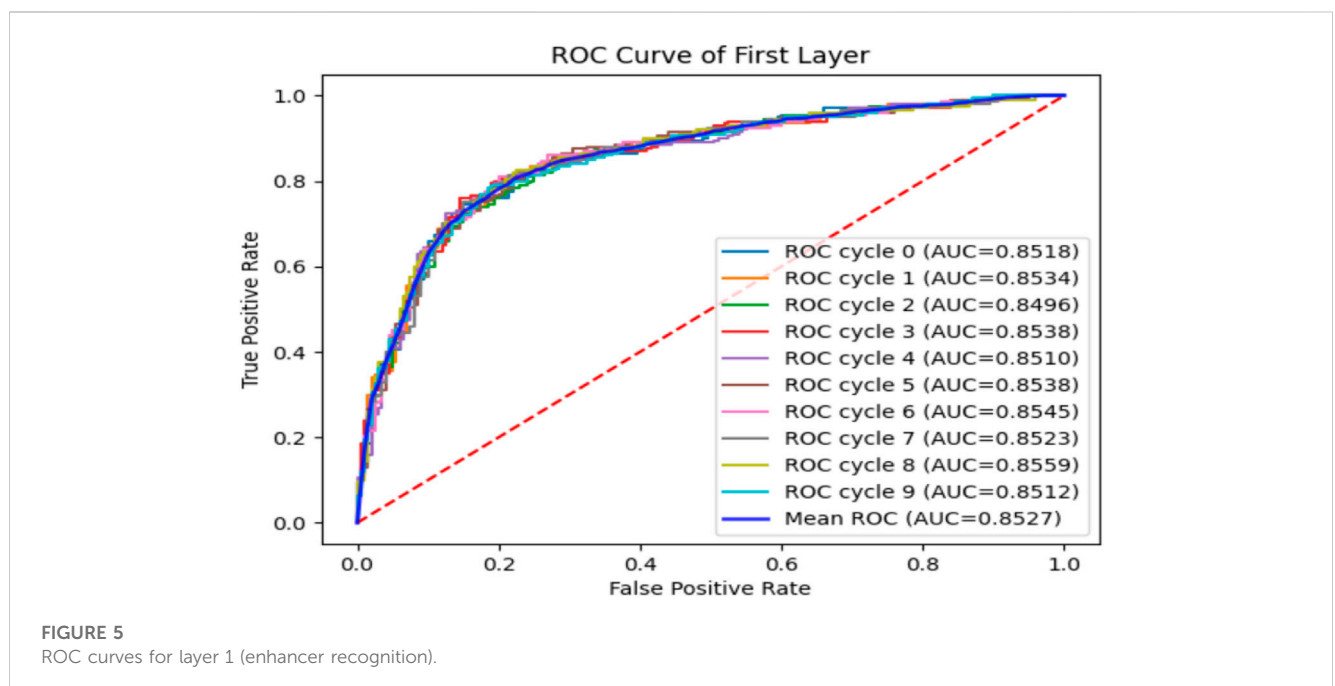
On the basis of the correct identification of enhancers in the first layer, the second layer predicts the strengths and weaknesses of enhancers. As the second layer has less training data and the complex network structure can lead to overfitting, we removed the attention module from the iEnhancer-DCSV network framework and used the same training as the first layer, with experimental results of 99.10%, 62.30%, 80.70%, and 0.6609 for SN, SP, Acc, and MCC, respectively.

3.3 Comparison of different coding methods

Currently, feature engineering has been a very important part of the process because building a model and using a simple and efficient coding method is crucial. In this study, we compared the one-hot + NCP coding, one-hot coding, and NCP coding to determine the final coding method. We input the three encoding methods into the two network frameworks, layer 1 and layer 2, respectively, and the results of the experiment are shown in Table 3. In the first layer (enhancer recognition), the one-hot + NCP coding was slightly better than the one-hot coding and better than the NCP coding. In the second layer (strong and weak enhancer prediction), the one-hot + NCP coding was much better than these two coding types. Therefore, we adopted one-hot + NCP coding as the final coding method in this study.

TABLE 5 Performance of iEnhancer-DCSV in 10 trials.

Layer	Cycle index	Sn (%)	Sp (%)	Acc (%)	MCC
First layer	0	78.50	78.00	78.25	0.5650
	1	83.00	75.50	79.25	0.5866
	2	80.50	75.00	77.75	0.5558
	3	74.00	85.50	79.75	0.5989
	4	77.00	81.50	79.25	0.5855
	5	87.50	68.00	77.75	0.5658
	6	80.50	80.00	80.25	0.6050
	7	79.50	80.00	79.75	0.5950
	8	80.50	78.00	79.25	0.5851
	9	81.50	75.00	78.25	0.5661
	Mean ± STD	80.25 ± 3.39	77.65 ± 4.47	78.95 ± 0.84	0.5809 ± 0.0158
Second layer	0	99.99	61.99	81.00	0.6702
	1	95.99	62.99	79.50	0.6250
	2	95.99	64.99	80.50	0.6416
	3	99.99	60.99	80.50	0.6624
	4	99.99	56.99	78.50	0.6313
	5	99.99	61.99	81.00	0.6702
	6	99.99	60.99	80.50	0.6624
	7	99.99	64.99	82.50	0.6938
	8	99.99	58.99	79.50	0.6468
	9	98.99	67.99	83.50	0.7047
	Mean ± STD	99.10 ± 1.58	62.30 ± 3.00	80.70 ± 1.38	0.6609 ± 0.0243



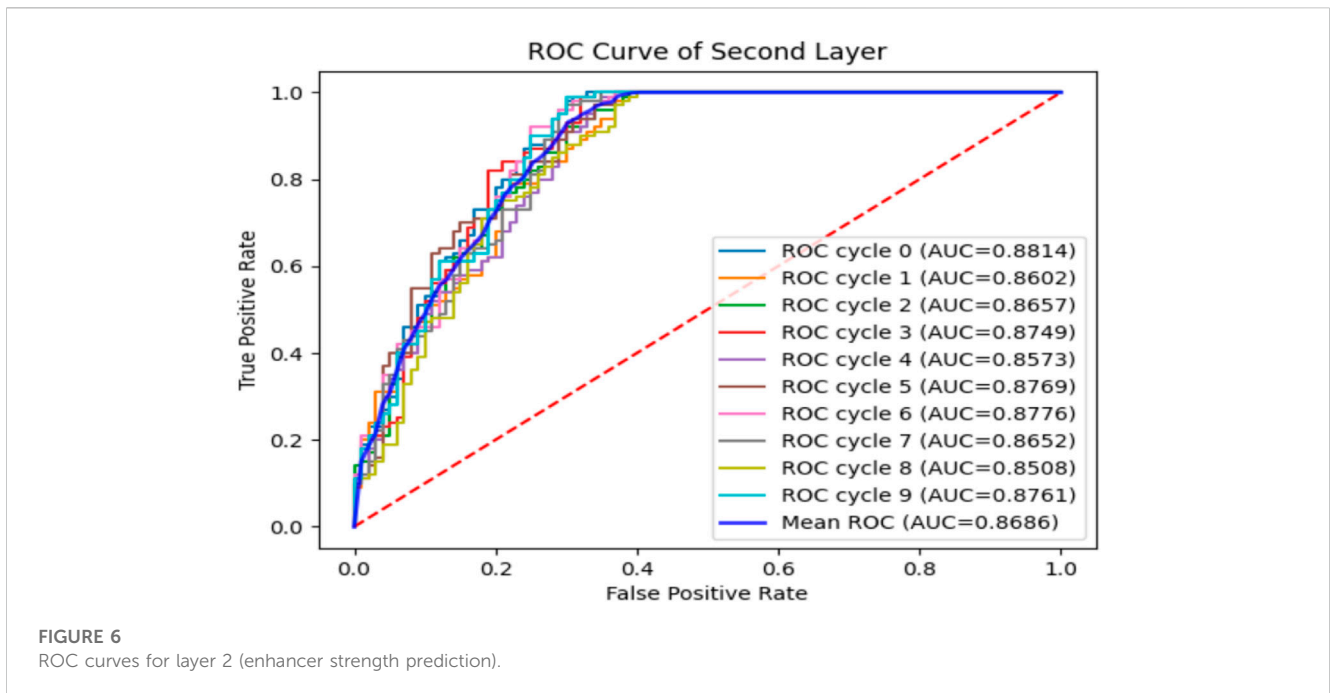


TABLE 6 Comparison with other methods on the same independent datasets.

Layer	Predictor	SN	SP	Acc	MCC	AUC	Source
First layer	iEnhancer-2L	71.00	75.00	73.00	0.4604	0.8062	Liu et al. (2015)
	EnhancerPred	73.50	74.50	74.00	0.4800	0.8013	Jia and He (2016)
	iEnhancer-EL	71.00	78.50	74.75	0.4964	0.8173	Liu et al. (2018a)
	iEnhancer-ECNN	78.50	75.20	76.90	0.5370	0.8320	Nguyen et (2019)
	iEnhancer-XG	75.75	74.00	77.50	0.5150	—	Cai et al. (2021)
	iEnhancer-EBLSTM	75.50	79.50	77.20	0.5340	0.7720	Niu et al. (2021)
	Enhancer-FRL	80.50	75.50	78.00	0.5607	0.8573	Wang et al. (2022a)
	iEnhancer-DCSV	80.25	77.65	78.95	0.5809	0.8527	This study
Second layer	iEnhancer-2L	47.00	74.00	60.50	0.2181	0.6678	Liu et al. (2015)
	EnhancerPred	45.00	65.00	55.00	0.1021	0.5790	Jia and He (2016)
	iEnhancer-EL	54.00	68.00	61.00	0.2222	0.6801	Liu et al. (2018b)
	iEnhancer-ECNN	79.10	56.40	67.80	0.3680	0.7480	Nguyen et (2019)
	iEnhancer-XG	70.00	57.00	63.50	0.2720	—	Cai et al. (2021)
	iEnhancer-EBLSTM	81.20	53.60	65.80	0.3240	0.6580	Niu et al. (2021)
	Enhancer-FRL	98.00	49.00	73.50	0.5391	0.8723	Wang et al. (2022b)
	iEnhancer-DCSV	99.10	62.30	80.70	0.6609	0.8686	This study

3.4 Comparison of different model frameworks

In this study, we used six network frameworks: ResNet, DenseNet, DenseNet + channel attention model, DenseNet + spatial attention model, DenseNet + CBAM attention model, and DenseNet + improved CBAM attention model. We tested these five

network frameworks in the first layer (enhancer recognition) task because the amount of data for the second layer (enhancer strength prediction) task was too small. The original features were extracted using each of these five network frameworks for the high-level features, and the best-performing network framework was selected based on the experimental results. The experimental comparison results are shown in Table 4. Adding an attention

model behind the DenseNet is already very effective, and the improved CBAM attention model integrates the advantages of both attention models. However, the improved effect is limited because the shape of the feature map is too small. The results show that the DenseNet + improved CBAM attention network framework works better. Therefore, we finally chose the DenseNet + improved CBAM attentional network framework model.

3.5 Performance of iEnhancer-DCSV on the training dataset

To verify the performance of the iEnhancer-DCSV classifier, we cycled through 10 times of fivefold cross-validation, and the experimental results are shown in Table 5. We found that the values of the evaluation metrics fluctuated relatively steadily on the first (enhancer recognition) and second (enhancer strength prediction) layer tasks, indicating that the iEnhancer-DCSV model has good generalization capability. Figure 5 shows the ROC curves of the first layer (enhancer recognition) with a mean AUC value of 0.8527 in 10 experiments, and Figure 6 shows the ROC curves of the second layer (enhancer strength prediction) with a mean AUC value of 0.8686 in 10 experiments. The results show that our proposed iEnhancer-DCSV has good performance.

3.6 Comparison of iEnhancer-DCSV with existing predictors

The iEnhancer-DCSV predictor proposed in this study is compared with seven existing predictors. The performance of independent datasets under different methods is shown in Table 6. The iEnhancer-DCSV predictor has better Acc and MCC metrics compared with others. The improvement ranges for ACC and MCC in the first layer (enhancer recognition) were 1.95%–5.95% and 0.0202–0.1205, respectively, and the improvement ranges for ACC and MCC in the second layer (enhancer strength prediction) were 7.2%–25.7% and 0.1218–0.5588, respectively. Meanwhile, in the first and second layers, the SN and SP metrics also have some advantages, indicating that iEnhancer-DCSV is more balanced and has more stable and superior performance in identifying positive and negative samples. The iEnhancer-DCSV predictor is expected to be the most advanced and representative tool for predicting enhancement and its strengths and weaknesses.

4 Conclusion

In this study, we propose a new predictor of enhancer recognition and its strength called iEnhancer-DCSV. It is based on DenseNet and an improved CBAM attention module approach. The experimental results demonstrate that the MCC value for enhancer identification on the independent test set is 0.5809, and the MCC value for enhancer strength prediction is 0.6609. This indicates that the iEnhancer-DCSV predictor has good performance and generalization ability, which is better than the existing prediction tools. We combine deep learning methods with enhancer research to innovate computational methods in the

field of bioinformatics and enrich enhancer research. In the future, the iEnhancer-DCSV predictor not only is applicable to enhancer classification tasks but can also be used in different prediction tasks, making its use convenient for researchers.

Of course, some deficiencies must be overcome in our proposed model. The current enhancer sample of data is small and fails to sufficiently promote the performance of the iEnhancer-DCSV model using a big data-driven approach. In addition, data enhancement strategies were not employed to augment our data samples, such as generative adversarial networks (GANs) (Li and Zhang, 2021). This will be our future work issue to address. However, as the research on enhancers progresses, the disadvantage of a small amount of data will gradually disappear, and better deep learning methods will be used in the research, creating more possibilities for future enhancer recognition and strength prediction.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding authors.

Author contributions

JJ and RL conceived and designed the experiments. RL implemented feature extraction, model construction, model training, and performance evaluation. RL, LQ, and XW drafted the manuscript and revised the manuscript. JJ supervised this study. All authors contributed to the content of this paper and approved the final manuscript.

Funding

This work was partially supported by the National Natural Science Foundation of China (nos 61761023, 62162032, and 31760315), the Natural Science Foundation of Jiangxi Province, China (nos 20202BABL202004 and 20202BAB202007), and the Scientific Research Plan of the Department of Education of Jiangxi Province, China (GJJ190695 and GJJ212419). These funders had no role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Acknowledgments

The authors are grateful for the constructive comments and suggestions made by the reviewers.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated

organizations or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Bauer, E., and Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Mach. Learn.* 36, 105–139. doi:10.1023/a:1007515423169
- Boyd, M., Thodberg, M., Vitezic, M., Bornholdt, J., Vitting-Seerup, K., Chen, Y., et al. (2018). Characterization of the enhancer and promoter landscape of inflammatory bowel disease from human colon biopsies. *Nat. Commun.* 9, 1661. doi:10.1038/s41467-018-03766-z
- Bu, H., Gan, Y., Wang, Y., Zhou, S., and Guan, J. (2017). A new method for enhancer prediction based on deep belief network. *BMC Bioinforma.* 18, 418. doi:10.1186/s12859-017-1828-0
- Buccitelli, C., and Selbach, M. (2020). mRNAs, proteins and the emerging principles of gene expression control. *Nat. Rev. Genet.* 21, 630–644. doi:10.1038/s41576-020-0258-4
- Cai, L., Ren, X., Fu, X., Peng, L., Gao, M., and Zeng, X. (2021). iEnhancer-XG: interpretable sequence-based enhancers and their strength predictor. *Bioinformatics* 37, 1060–1067. doi:10.1093/bioinformatics/btaa914
- Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., et al. (Year). "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning", in: 2017 Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 6298–6306.
- Chou, K. C. (1984). Low-frequency vibrations of DNA molecules. *Biochem. J.* 221, 27–31. doi:10.1042/bj2210027
- Cvetesic, N., and Lenhard, B. (2017). Core promoters across the genome. *Nat. Biotechnol.* 35, 123–124. doi:10.1038/nbt.3788
- Dai, Q., Bao, C., Hai, Y., Ma, S., Zhou, T., Wang, C., et al. (2018). MTGPick allows robust identification of genomic islands from a single genome. *Brief. Bioinform* 19, 361–373. doi:10.1093/bib/bbw118
- Emison, E. S., Mccallion, A. S., Kashuk, C. S., Bush, R. T., Grice, E., Lin, S., et al. (2005). A common sex-dependent mutation in a RET enhancer underlies Hirschsprung disease risk. *Nature* 434, 857–863. doi:10.1038/nature03467
- Erwin, G. D., Oksenberg, N., Truty, R. M., Kostka, D., Murphy, K. K., Ahituv, N., et al. (2014). Integrating diverse datasets improves developmental enhancer prediction. *PLoS Comput. Biol.* 10, e1003677. doi:10.1371/journal.pcbi.1003677
- Fernández, M., and Miranda-Saavedra, D. (2012). Genome-wide enhancer prediction from epigenetic signatures using genetic algorithm-optimized support vector machines. *Nucleic Acids Res.* 40, e77. doi:10.1093/nar/gks149
- Firpi, H. A., Ucar, D., and Tan, K. (2010). Discover regulatory DNA elements using chromatin signatures and artificial neural network. *Bioinformatics* 26, 1579–1586. doi:10.1093/bioinformatics/btq248
- Ghandi, M., Lee, D., Mohammad-Noori, M., and Beer, M. A. (2014). Enhanced regulatory sequence prediction using gapped k-mer features. *PLoS Comput. Biol.* 10, e1003711. doi:10.1371/journal.pcbi.1003711
- He, W., and Jia, C. (2017). EnhancerPred2.0: Predicting enhancers and their strength based on position-specific trinucleotide propensity and electron-ion interaction potential feature selection. *Mol. Biosyst.* 13, 767–774. doi:10.1039/c7mb00054e
- Huang, Y., Niu, B., Gao, Y., Fu, L., and Li, W. (2010). CD-HIT suite: A web server for clustering and comparing biological sequences. *Bioinformatics* 26, 680–682. doi:10.1093/bioinformatics/btq003
- Jia, C., and He, W. (2016). EnhancerPred: A predictor for discovering enhancers based on the combination and selection of multiple features. *Sci. Rep.* 6, 38741. doi:10.1038/srep38741
- Kim, T.-K., Hemberg, M., Gray, J. M., Costa, A. M., Bear, D. M., Wu, J., et al. (2010). Widespread transcription at neuronal activity-regulated enhancers. *Nature* 465, 182–187. doi:10.1038/nature09033
- Kleftogiannis, D., Kalnis, P., and Bajic, V. B. (2015). Deep: A general computational framework for predicting enhancers. *Nucleic Acids Res.* 43, e6. doi:10.1093/nar/gku1058
- Kong, R., Xu, X., Liu, X., He, P., Zhang, M. Q., and Dai, Q. (2020). 2SigFinder: The combined use of small-scale and large-scale statistical testing for genomic island detection from a single genome. *BMC Bioinforma.* 21, 159. doi:10.1186/s12859-020-3501-2
- Li, M., and Zhang, W. (2021). Phiaf: Prediction of phage-host interactions with GAN-based data augmentation and sequence-based feature fusion. *Briefings Bioinforma.* 23, bbab348. doi:10.1093/bib/bbab348
- Li, X., Han, P., Chen, W., Gao, C., Wang, S., Song, T., et al. (2022). Marppi: Boosting prediction of protein-protein interactions with multi-scale architecture residual network. *Briefings Bioinforma.* 24, bbac524. doi:10.1093/bib/bbac524
- Liu, B., Fang, L., Long, R., Lan, X., and Chou, K.-C. (2015). iEnhancer-2L: a two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition. *Bioinformatics* 32, 362–369. doi:10.1093/bioinformatics/btv604
- Liu, B., Li, K., Huang, D.-S., and Chou, K.-C. (2018a). iEnhancer-EL: identifying enhancers and their strength with ensemble learning approach. *Bioinformatics* 34, 3835–3842. doi:10.1093/bioinformatics/bty458
- Liu, G., Zhang, Y., Wang, L., Xu, J., Chen, X., Bao, Y., et al. (2018b). Alzheimer's disease rs11767557 variant regulates EPHA1 gene expression specifically in human whole blood. *J. Alzheimer's Dis.* 61, 1077–1088. doi:10.3233/JAD-170468
- Lu, Y., Qu, W., Shan, G., and Zhang, C. (2015). Delta: A distal enhancer locating tool based on AdaBoost algorithm and shape features of chromatin modifications. *PLoS One* 10, e0130622. doi:10.1371/journal.pone.0130622
- Maston, G. A., Evans, S. K., and Green, M. R. (2006). Transcriptional regulatory elements in the human genome. *Annu. Rev. Genomics Hum. Genet.* 7, 29–59. doi:10.1146/annurev.genom.7.080505.115623
- Min, S., Lee, B., and Yoon, S. (2016). Deep learning in bioinformatics. *Briefings Bioinforma.* 18, bbw068–869. doi:10.1093/bib/bbw068
- Nguyen, Q. H., Nguyen-Vo, T. H., Le, N. Q. K., Do, T. T. T., Rahardja, S., and Nguyen, B. P. (2019). iEnhancer-ECNN: identifying enhancers and their strength using ensembles of convolutional neural networks. *BMC Genomics* 20, 951. doi:10.1186/s12864-019-6336-3
- Niu, K., Luo, X., Zhang, S., Teng, Z., Zhang, T., and Zhao, Y. (2021). iEnhancer-EBLSTM: Identifying enhancers and strengths by ensembles of bidirectional long short-term memory. *Front. Genet.* 12, 665498. doi:10.3389/fgene.2021.665498
- Pennacchio, L. A., Bickmore, W., Dean, A., Nobrega, M. A., and Bejerano, G. (2013). Enhancers: Five essential questions. *Nat. Rev. Genet.* 14, 288–295. doi:10.1038/nrg3458
- Shang, Y., Ye, X., Futamura, Y., Yu, L., and Sakurai, T. (2022). Multiview network embedding for drug-target Interactions prediction by consistent and complementary information preserving. *Briefings Bioinforma.* 23, bbac059. doi:10.1093/bib/bbac059
- Shlyueva, D., Stampfel, G., and Stark, A. (2014). Transcriptional enhancers: From properties to genome-wide predictions. *Nat. Rev. Genet.* 15, 272–286. doi:10.1038/nrg3682
- Shrinivas, K., Sabari, B. R., Coffey, E. L., Klein, I. A., Boija, A., Zamudio, A. V., et al. (2019). Enhancer features that drive formation of transcriptional condensates. *Mol. Cell* 75, 549–561.e7. doi:10.1016/j.molcel.2019.07.009
- Senden, R. R., Pearson, C. E., Potaman, V. N., and Ussery, D. W. (1998). "Dna: Structure and function," in *Advances in genome biology*. Editor R. S. Verma[AI, 1–141.
- Singh, M., Rajagopal, N., Xie, W., Li, Y., Wagner, U., Wang, W., et al. (2013). Rfecs: A random-forest based algorithm for enhancer identification from chromatin state. *PLoS Comput. Biol.* 9, e1002968. doi:10.1371/journal.pcbi.1002968
- Sokolova, M., and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* 45, 427–437. doi:10.1016/j.ipm.2009.03.002
- Vacic, V., Iakoucheva, L. M., and Radivojac, P. (2006). Two sample logo: A graphical representation of the differences between two sets of sequence alignments. *Bioinformatics* 22, 1536–1537. doi:10.1093/bioinformatics/btl151
- Wang, C., Zou, Q., Ju, Y., and Shi, H. (2022a). Enhancer-FRL: Improved and robust identification of enhancers and their activities using feature representation learning. *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, 1–9. doi:10.1109/TCBB.2022.3204365
- Wang, Y., Peng, Q., Mou, X., Wang, X., Li, H., Han, T., et al. (2022b). A successful hybrid deep learning model aiming at promoter identification. *BMC Bioinforma.* 23, 206. doi:10.1186/s12859-022-04735-6
- Wang, Y., Xu, Y., Yang, Z., Liu, X., and Dai, Q. (2021). Using recursive feature selection with random forest to improve protein structural class prediction for low-

similarity sequences. *Comput. Math. Methods Med.* 2021, 5529389. doi:10.1155/2021/5529389

Wu, S., Ou, T., Xing, N., Lu, J., Wan, S., Wang, C., et al. (2019). Whole-genome sequencing identifies ADGRG6 enhancer mutations and FRS2 duplications as angiogenesis-related drivers in bladder cancer. *Nat. Commun.* 10, 720. doi:10.1038/s41467-019-08576-5

Xiao, X., Xu, Z. C., Qiu, W. R., Wang, P., Ge, H. T., and Chou, K. C. (2019). iPSW(2L)-PseKNC: A two-layer predictor for identifying promoters and their strength by hybrid features via pseudo K-tuple nucleotide composition. *Genomics* 111, 1785–1793. doi:10.1016/j.ygeno.2018.12.001

Xiao, Z., Zou, Q., Liu, Y., and Yang, X. (2016). Genome-wide assessment of differential translations with ribosome profiling data. *Nat. Commun.* 7, 11194. doi:10.1038/ncomms11194

Yang, B., Liu, F., Ren, C., Ouyang, Z., Xie, Z., Bo, X., et al. (2017). BiRen: Predicting enhancers with a deep-learning-based model using the DNA sequence alone. *Bioinformatics* 33, 1930–1936. doi:10.1093/bioinformatics/btx105

Yang, S., Wang, Y., Chen, Y., and Dai, Q. (2020). Masqc: Next generation sequencing assists third generation sequencing for quality control in N6-methyladenine DNA identification. *Front. Genet.* 11, 269. doi:10.3389/fgene.2020.00269

Yang, Z., Yi, W., Tao, J., Liu, X., Zhang, M. Q., Chen, G., et al. (2022). HPVMD-C: A disease-based mutation database of human papillomavirus in China. *Database J. Biol. Databases Curation* 2022. doi:10.1093/database/baac018

Zhang, T., Tang, Q., Nie, F., Zhao, Q., and Chen, W. (2022). DeepLncPro: An interpretable convolutional neural network model for identifying long non-coding RNA promoters. *Briefings Bioinforma.* 23, bbac447. doi:10.1093/bib/bbac447