Hindawi
Computational and Mathematical Methods in Medicine
Volume 2022, Article ID 7518779, 11 pages
https://doi.org/10.1155/2022/7518779

Research Article

Identifying and Classifying Enhancers by Dinucleotide-Based Auto-Cross Covariance and Attention-Based Bi-LSTM

Shulin Zhao[1,2] **Qingfeng Pan**[3] **Quan Zou**[1,2] **Ying Ju**[4] **Lei Shi**[5] **and Xi Su**[6]

[1]*Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, China*
[2]*Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou, Zhejiang, China*
[3]*General Hospital of Heilongjiang Province Land Reclamation Bureau, Harbin, China*
[4]*School of Informatics, Xiamen University, Xiamen, China*
[5]*Department of Spine Surgery, Changzheng Hospital, Naval Medical University, Shanghai, China*
[6]*Foshan Maternal and Child Health Hospital, Foshan, Guangdong, China*

Correspondence should be addressed to Lei Shi; slspine@163.com and Xi Su; xisu_fsfy@163.com

Received 20 December 2021; Accepted 12 March 2022; Published 5 April 2022

Academic Editor: Chung-Min Liao

Enhancers are a class of noncoding DNA elements located near structural genes. In recent years, their identification and classification have been the focus of research in the field of bioinformatics. However, due to their high free scattering and position variability, although the performance of the prediction model has been continuously improved, there is still a lot of room for progress. In this paper, density-based spatial clustering of applications with noise (DBSCAN) was used to screen the physicochemical properties of dinucleotides to extract dinucleotide-based auto-cross covariance (DACC) features; then, the features are reduced by feature selection Python toolkit MRMD 2.0. The reduced features are input into the random forest to identify enhancers. The enhancer classification model was built by word2vec and attention-based Bi-LSTM. Finally, the accuracies of our enhancer identification and classification models were 77.25% and 73.50%, respectively, and the Matthews' correlation coefficients (MCCs) were 0.5470 and 0.4881, respectively, which were better than the performance of most predictors.

1. Introduction

Enhancers are short noncoding fragments of DNA sequences that can greatly enhance the activity of promoters [1]. After Benerji discovered the first 140 bp enhancer in SV40DNA in 1981, researchers attempted to find more enhancers on a genome-wide scale [2]. Among these attempts, some computer methods have been used to identify and classify the enhancers [3, 4]. For example, Jia and He extracted features using high-dimensional eigenvectors based on double-contour Bayes, nucleotide composition, and pseudonucleotide composition, realizing the distinction between enhancers and nonenhancers and strong and weak enhancers through a support vector machine (SVM) and developing a web server named EnhancerPred [5]. iEnhancer-2L [6] selected a feature extraction method, namely, pseudo K tuple nucleotide composition (PseKNC), and predicted them with SVM. iEnhancer-EL [7] adopted three feature extraction methods, namely, k-mers, subsequence profile, and PseKNC, and utilized SVM as an individual classifier for ensemble learning prediction. The Enhancer-5step [8] applied the word-embedded representation to biological sequences, specifically by using the FastText tool to extract the 100-dimensional features and then using the supervisory method SVM for predictive classification. Tan et al. [9] took six types of dinucleotide physical and chemical properties as input characteristics and employed a deep recursive neural network-based classifier integration model, which achieved good results. iEnhancer-ECNN [10] exploited convolutional neural network (CNN) integration, combined with one-hot coding and k-mers descriptors as sequence coding projects, and is an effective computing strategy. iEnhancer-CNN [11] extracted the features of enhancers from the original DNA sequence using word2vec and predicted them using CNN. These models and predictors continuously improve the performance of enhancer

identification and classification, but the performance is not good enough in general, and further research is needed, especially the classification of enhancers.

In this paper, we propose a new model building strategy; the process is shown in Figure 1. First, we divided the task into the identification and classification of enhancers. In enhancer identification, we used the density-based spatial clustering of applications with noise (DBSCAN) [12] algorithm to cluster the physicochemical properties of the original 148 dinucleotides and extract 47 of them, as detailed in Supplementary Materials (available here). Then, 11,045 ($47 \times 47 \times 5$) dimensional features were obtained by the dinucleotide-based auto-cross covariance (DACC) [13] feature extraction method. To prevent overfitting, the dimension was reduced to 791 using MRMD2.0 [14], a Python toolkit that combines seven commonly used feature ranking algorithms with the PageRank strategy. After CNN, RNN, etc., failed to achieve ideal results, the use of random forest achieved good results. In the final independent test, an accuracy of 77.5% and MCC of 0.552 were achieved. In the process of enhancer classification, we used 3-mers to split sequences and CBOW as word embedding models to transform biological sequences into 198×200 dimension word sequences. Then, we used attention-based bidirectional long short-term memory (Bi-LSTM) [15] to carry out predictive classification, and in independent tests, the accuracy was 65%, and the MCC was 0.3824.

Finally, we give a general introduction to the structure and organization of this work. In Results, we compared and discussed the prediction performance achieved by the enhancer identification and classification models proposed in this paper with existing models or predictors, and summarize the paper. Then, in Discussion, we introduced our models in detail and discussed the dimensionality reduction and dimension selection experiment in enhancer identification and the word2vec model parameter selection experiment in enhancer classification. Finally, in Material and Methods, the datasets, DACC feature extraction algorithm, the selection rules of physicochemical properties using DBSCAN algorithm, the principle of attention-based Bi-LSTM, and the model evaluation metrics are described, respectively.

2. Results

In this study, we proposed different models for enhancer identification and enhancer classification. In enhancer identification, the physicochemical properties of dinucleotides obtained by clustering screening were used for DACC feature extraction, and then, we performed feature dimension reduction. Finally, random forest was used for prediction. In enhancer classification, we used 3-mers and CBOW models to obtain word vectors and then used attention-based Bi-LSTM for classification. The model proposed in this paper finally achieved excellent performance in the independent test. Specifically, the model had 77.25%, 77.30%, 77.20%, and 0.5470 values for enhancer identification, accuracy, sensitivity, specificity, and MCC, respectively. For the enhancer classification, the performances were 73.50%, 87.00%, 60.00%, and 0.4881, respectively. Table 1 gives a detailed comparison of the performance of the model presented in this paper and the previous models. In terms of enhancer identification, we are slightly inferior to Enhancer-5Step and iEnhancer-CNN but superior to other models. Although the performance is not absolutely excellent, we hope that the construction idea of the model has some inspiration to others. In the enhancer classification, the MCC of the model presented in this paper was significantly higher than the MCC of other models, with an increase of 0.1201 compared with the highest MCC of 0.3680, and its sensitivity was also the highest, reaching 87.00%. Both models have achieved preeminent performance.

The contribution of this paper is to use the DBSCAN clustering algorithm to select representative physical and chemical properties, and then extracted DACC features, which avoids overfitting to a certain extent. And we experimentally compared the effects of word2vec model parameters and different types of LSTM on performance. The ideas of model construction can also be applied to other bioinformatics datasets or computational biology directions [16–22] such as enhancer-promoter interaction identification [23, 24], disease biomarker mining [25–31], and drug discovery [32–34].

In the future research, we will try to optimize the DBSCAN algorithm in terms of adaptive selection of parameters to improve its processing of different density datasets. And deep learning can indeed achieve better results than ordinary machine learning algorithms in enhancer classification. We will try hot deep learning technologies such as graph neural networks to further improve prediction performance.

3. Discussion

3.1. Enhancer Identification. Feature extraction is a vital link in building an excellent classification model. In this paper, to obtain DACC feature vectors, we use iLearn [35] to extract them. A total of 148 dinucleotide physicochemical properties were provided by iLearn [35]. If the DACC in the form of all physicochemical properties is adopted, a total of 109,520 dimensions of features will be obtained, but the sample size is relatively small, and overfitting is easily generated. Therefore, in this study, our solution was to use DBSCAN to conduct cluster screening for physical and chemical property indexes.

DBSCAN is a commonly used density-based clustering method. Compared with K-means, the DBSCAN algorithm does not need to predefine the number of clusters and DBSCAN can find clusters of arbitrary shapes. In addition, DBSCAN can also identify "outliers", and the "outliers" are the special physical and chemical properties we want to find. At present, many studies have improved DBSCAN to enable it to process large datasets at a high speed.

In this paper, clustering and processing of physicochemical dinucleotide indexes are carried out. After the treatment, we obtained 47 kinds of physical and chemical property indexes. Then, feature extraction was carried out through DACC. After executing the iLearn [35] command line, 47

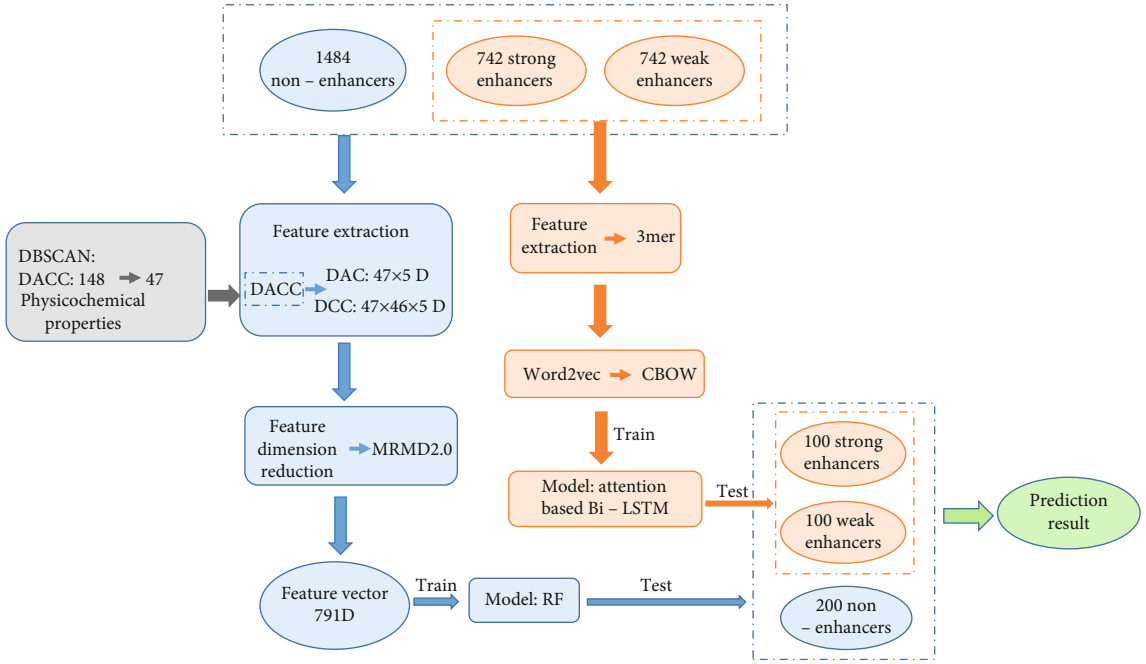

Figure 1: The main flow chart of the research process in this paper.

Table 1: The independent test performance comparison of this model with other models.

Method	ACC (%)	SN (%)	SP (%)	MCC
Enhancer identification				
EnhancerPred	74.00	73.50	74.50	0.4800
iEnhancer-2L	73.00	71.00	75.00	0.4604
iEnhancer-EL	74.75	71.00	78.50	0.4964
Enhancer-5Step	79.00	82.00	76.00	0.5800
Tan et al.	75.50	75.50	76.00	0.5100
iEnhancer-ECNN	76.90	78.50	75.20	0.5370
iEnhancer-CNN	77.50	78.25	79.00	0.5850
Our method	**77.25**	**77.30**	**77.20**	**0.5470**
Enhancer classification				
EnhancerPred	55.00	45.00	65.00	0.1021
iEnhancer-2L	60.50	47.00	74.00	0.2181
iEnhancer-EL	78.03	54.00	68.00	0.2222
Enhancer-5Step	63.50	74.00	53.00	0.2800
Tan et al.	68.49	83.15	45.61	0.3120
iEnhancer-ECNN	67.80	79.10	56.40	0.3680
iEnhancer-CNN	75.00	65.25	76.10	0.3232
Our method	**73.50**	**87.00**	**60.00**	**0.4881**

× 47 × 5 (11,045) feature dimensions were obtained: python iLearn-nucleotide-acc.py –file data. txt –method DACC –type DNA –lag 5.

Considering that there are still more features in 11,045 dimensions, we tried to use MRMD2.0 [36–38] for feature dimension reduction. MRMD2.0 integrates rich feature selection algorithms and feature ranking algorithms and is superior to the single feature selection algorithm. We conducted dimension reduction three times, and the fivefold cross-validation results before and after each dimension reduction are shown in Table 2. After the dimension reduction, enhancer recognition effect is obviously seen to be improved, but as the number of dimension reductions increases, performance is not getting better and better. Instead, the performance is the best when the dimension is reduced to 791 for the first time; therefore, we finally chose 791 dimensional features as the input of the classifier.

After adopting CNN, LSTM, and autoencoder for feature extraction, we failed to achieve ideal results. Since random forest is good at processing high-dimensional data and has strong anti-interference ability, we tried to use it for classification and finally achieved relatively ideal results. In the independent test, the model achieved an accuracy of 77.5% and MCC of 0.552.

3.2. Enhancer Classification. Since the model construction method of identifying enhancers is not ideal when applied to classifying enhancers, we considered introducing a new scheme. In terms of feature representation, *k*-mers are used to segment biological sequences in this paper, and after 3-mers, the 200 long strong and weak enhancer sequences will be converted to 198 words. For example, the sequence "TACATTCA" after 3-mers is divided into 6 words "TAC ACA CAT ATT TTC TCA".

Then, the word2vec model is used to generate words into vectors to represent the relationships between words. word2vec relies on two training modes: continuous bag of words (CBOW) and skip-gram [39]. To achieve better results, we tried to use CBOW and skip-gram models with different parameters and compared their performance. In the experiment, parameters were adjusted from three aspects: the

Table 2: In enhancer identification. The performance comparison of the 5-fold cross-validation before and after each feature dimensionality reduction.

Dimension	ACC (%)	MCC
11025	74.87	0.498
791	75.47	0.510
721	75.37	0.508
699	75.40	0.508

optimization method of the model training mechanism (negative sampling (NS)/hierarchical softmax (HS)), the minimum word frequency of the word vector (Min_count), and the maximum context distance of the word vector (Window). As shown in Table 3, when the CBOW model and HS, Min_count, and Window were set at 5 and 5, respectively, the ACC reached 67.57%, and the MCC was 0.3529, showing the best effect. Then, LSTM, which is a variant of RNN, is used for training. In this paper, the 5-fold cross-validation performance of LSTM, Bi-LSTM, and attention-based Bi-LSTM is compared. As shown in Table 4, the attention-based Bi-LSTM model performs better. An MCC of 0.4881 with an accuracy of 73.5% was achieved in an independent test.

A noteworthy problem is that this model and existing methods such as Enhancer-5Step and iEnhancer-ECNN have higher SN in the enhancer classification results, while SP is lower, at least 20% lower than SN. This shows that the model has a better ability to identify strong enhancers, while the ability to identify weak enhancers is weak. The potential reasons are roughly divided into two aspects: feature extraction and model construction. When the extracted features cannot distinguish weak enhancer samples that are similar to strong enhancer samples, it is identified as a strong enhancer. The second is model building. There are also great differences in the discriminative ability of different computational models for the same dataset. In this regard, we can try more feature extraction algorithms and classification algorithms in the future to improve this problem.

4. Material and Methods

4.1. Benchmark Dataset. In our study, a benchmark dataset was derived from Liu et al. [6]. This dataset is widely used in enhancer classification studies such as EnhancerPred and iEnhancer-EL. The dataset consists of 200 bp DNA sequences, and then in order to avoid redundancy, CD-HIT software [40] was used to delete pairwise sequences (sequences with similarity greater than 20%). Finally, we obtained the training set and independent set used by former researchers, in which the training set included 2,968 samples, and the ratio of nonenhancers, strong enhancers, and weak enhancers was 2 : 1 : 1. The independent test group is composed of 400 samples. Their number ratio is also 2 : 1 : 1.

Table 3: The performance comparison of different parameters in the word2vec model in the enhancer classification in the 5-fold cross-validation.

	HS/NS	Min_count, Window	ACC (%)	MCC
CBOW	HS	3, 3	65.10	0.3123
		3, 5	66.22	0.3412
		5, 3	65.20	0.3119
		5, 5	**67.57**	**0.3529**
	NS	3, 3	61.82	0.2365
		3, 5	66.89	0.3381
		5, 3	63.76	0.2761
		5, 5	63.85	0.2908
Skip-gram	HS	3, 3	66.22	0.3258
		3, 5	65.54	0.3113
		5, 3	66.89	0.3379
		5, 5	65.20	0.3178
	NS	3, 3	66.44	0.3414
		3, 5	63.76	0.2768
		5, 3	64.09	0.2833
		5, 5	63.18	0.2754

Table 4: The performance comparison of LSTM, Bi-LSTM, and attention-based Bi-LSTM in the enhancer classification in the 5-fold cross-validation.

	ACC (%)	MCC
LSTM	64.21	0.2879
Bi-LSTM	61.41	0.2356
Attention-based Bi-LSTM	**67.57**	**0.3529**

4.2. Dinucleotide-Based Auto-Cross Covariance (DACC). our research, we integrate the global sequence-order information into the model by using a feature extraction method based on DACC. It is formed by the combination of dinucleotide-based auto covariance (DAC) and dinucleotide-based cross covariance (DCC). In this combination, the DAC code calculates the correlation of dinucleotides along a lag distance between sequences with the same physical and chemical properties. The calculation form is as follows:

$$\mathrm{DAC}(\varphi, \mathrm{lag}) = \sum_{i=1}^{L-\mathrm{lag}-1} \left(\frac{\left(P_\varphi(R_i R_{i+1}) - \overline{P_\varphi}\right)\left(P_\varphi\left(R_{i+\mathrm{lag}} R_{i+\mathrm{lag}+1}\right) - \overline{P_\varphi}\right)}{L - \mathrm{lag} - 1} \right),$$
$$\overline{P_\varphi} = \sum_{i=1}^{L-1} P_\varphi \frac{R_i R_{i+1}}{L-1}, \tag{1}$$

where L denotes the sequence length; R_i represents the nucleic acid residue located at the i^{th} position; P_φ is a physical and chemical property and φ is a physical and

Table 5: Under the physical and chemical properties of "Shift" and "Slide", the corresponding dinucleotide values are involved in the sequence "TACATTCA".

	TA	AC	CA	AT	TT	TC
Shift	-2.243	0.126	-0.861	-1.019	1.587	0.126
Slide	-1.511	1.289	-0.623	2.513	0.111	-0.394

chemical property index; $P_\varphi(R_iR_{i+1})$ on behalf of the position i dinucleotide R_iR_{i+1} values correspond to the physical and chemical properties P_φ; $\overline{P_\varphi}$ is the numerical mean value of dinucleotides corresponding to physicochemical properties in the whole DNA sequence.

For example, a DNA sequence with a length of 8 is "TACATTCA", and the corresponding dinucleotide value under the "Shift" physicochemical property is shown in the Table 5. Then,

$$\overline{P_\varphi} = \frac{P_\varphi(\mathrm{TA}) + P_\varphi(\mathrm{AC}) + P_\varphi(\mathrm{CA}) + P_\varphi(\mathrm{AT}) + P_\varphi(\mathrm{TT}) + P_\varphi(\mathrm{TC}) + P_\varphi(\mathrm{CA})}{7} = \frac{-2.243 + 0.126 - 0.861 - 1.019 + 1.587 + 0.126 - 0.861}{7} \approx -0.449. \tag{2}$$

When lag is 5 (as shown in Figure 2),

$$\begin{aligned}\mathrm{DAC}(\varphi = \mathrm{Shift}, \mathrm{lag} = 5) &= \frac{\left(P_\varphi(R_1R_2) - \overline{P_\varphi}\right)\left(\left(P_\varphi(R_6R_7) - \overline{P_\varphi}\right) + \left(P_\varphi(R_2R_3) - \overline{P_\varphi}\right)\left(\left(P_\varphi(R_7R_8) - P_\varphi\right)\right.\right.}{L - \mathrm{lag} - 1} \\ &= \frac{\left(P_\varphi(\mathrm{TA}) - \overline{P_\varphi}\right)\left(P_\varphi(\mathrm{TC}) - \overline{P_\varphi}\right) + \left(P_\varphi(\mathrm{AC}) - \overline{P_\varphi}\right)\left(P_\varphi(\mathrm{CA}) - \overline{P_\varphi}\right)}{L - \mathrm{lag} - 1} \\ &= \frac{(-2.243 + 0.449)(0.126 + 0.449) + (0.126 + 0.449)(-0.861 + 0.449)}{2} \approx -0.634.\end{aligned} \tag{3}$$

So, the DAC eigenvalue of the sequence "TACATTCA" is about -0.634 under the physicochemical property of "Shift" and when lag is 5.

The dimension of the feature vector is $N \times \mathrm{LAG}$ after DAC, where N is the number of physicochemical properties and LAG is the maximum of lag (lag = 1, 2, ⋯, LAG). In this paper, LAG is 5.

DCC encoding was used to calculate the correlation of dinucleotides along a lag distance between sequences with different physical and chemical properties, and the calculation form was as follows:

$$\mathrm{DCC}(\varphi_1, \varphi_2, \mathrm{lag}) = \sum_{i=1}^{L-\mathrm{lag}-1} \frac{\left(P_{\varphi_1}(R_iR_{i+1}) - \overline{P_{\varphi_1}}\right)\left(P_{\varphi_2}\left(R_{i+\mathrm{lag}}R_{i+\mathrm{lag}+1}\right) - \overline{P_{\varphi_2}}\right)}{(L - \mathrm{lag} - 1)}, \tag{4}$$

where L denotes the sequence length; P_{φ_1} and P_{φ_2} are the two different physicochemical properties; $P_{\varphi_a}(R_iR_{i+1})$ on behalf of the position i dinucleotide R_iR_{i+1} correspond to the physical and chemical properties of P_{φ_a}, $a = 1, 2$; $\overline{P_{\varphi_a}}$ is the numerical mean value of dinucleotide corresponding to physicochemical properties of P_{φ_a} $(a = 1, 2)$ in the whole DNA sequence.

Similarly, take the sequence "TACATTCA" as an example; φ_1 is the physicochemical property of "Shift" and φ_2 is the physicochemical property of "Slide", and their corresponding dinucleotide values are shown in Table 5. It is known that $\overline{P_{\varphi_1}} = -0.449$; then,

$$\overline{P_{\varphi_2}} = \frac{P_{\varphi_2}(\mathrm{TA}) + P_{\varphi_2}(\mathrm{AC}) + P_{\varphi_2}(\mathrm{CA}) + P_{\varphi_2}(\mathrm{AT}) + P_{\varphi_2}(\mathrm{TT}) + P_{\varphi_2}(\mathrm{TC})}{7} = \frac{-1.511 + 1.289 - 0.623 + 2.513 + 0.111 - 0.394 - 0.623}{7} \approx 0.109. \tag{5}$$

When lag is 5 (as shown in Figure 2),

$$\begin{aligned}\mathrm{DCC}(\varphi_1 = \mathrm{shift}, \varphi_2 = \mathrm{slide}, \mathrm{lag} = 5) &= \frac{\left(P_{\varphi_1}(R_1R_2) - \overline{P_{\varphi_1}}\right)\left(P_{\varphi_2}(R_6R_7) - \overline{P_{\varphi_2}}\right) + \left(P_{\varphi_1}(R_2R_3) - \overline{P_{\varphi_1}}\right)\left(P_{\varphi_2}(R_7R_8) - \overline{P_{\varphi_2}}\right)}{(L - \mathrm{lag} - 1)} \\ &= \frac{\left(P_{\varphi_1}(\mathrm{TA}) - \overline{P_{\varphi_1}}\right)\left(P_{\varphi_2}(\mathrm{TC}) - \overline{P_{\varphi_2}}\right) + \left(P_{\varphi_1}(\mathrm{AC}) - \overline{P_{\varphi_1}}\right)\left(P_{\varphi_2}(\mathrm{CA}) - \overline{P_{\varphi_2}}\right)}{L - \mathrm{lag} - 1} \\ &= \frac{(-2.243 + 0.449)(-0.394 - 0.109) + (0.126 + 0.449)(-0.623 - 0.109)}{2} \approx 0.241.\end{aligned} \tag{6}$$

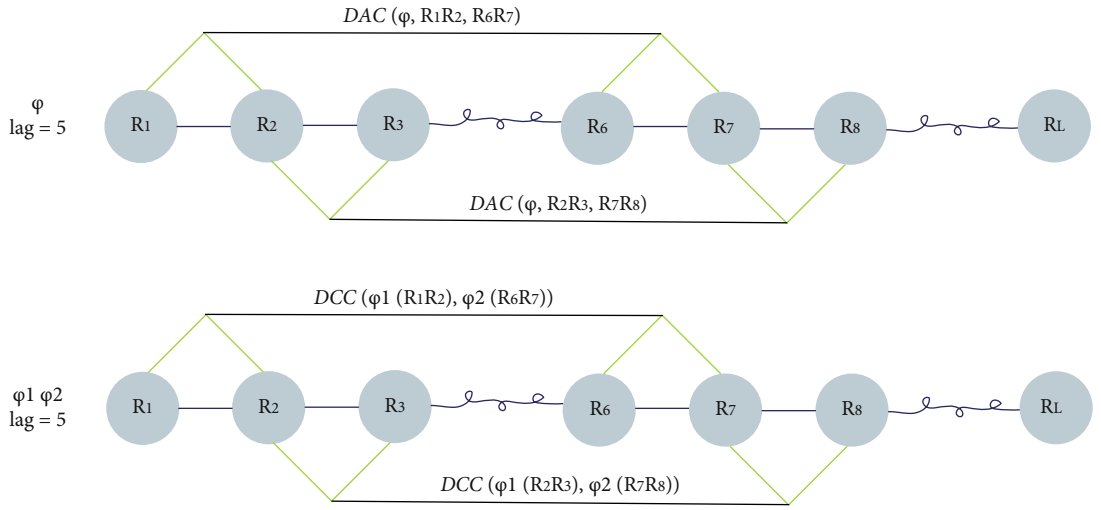

Figure 2: The process of generating DAC and DCC feature vectors of sequence "$R_1R_2, \cdots, R_L$".

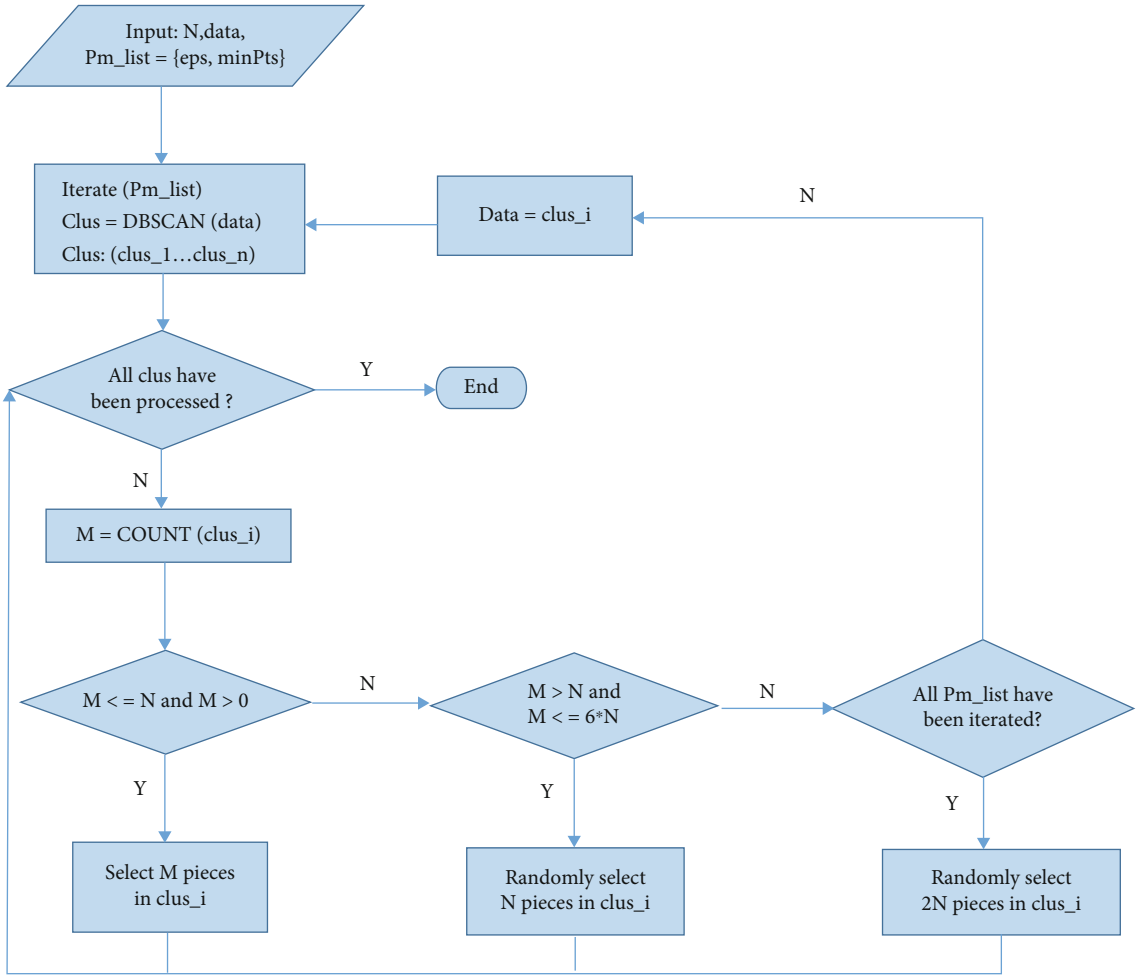

Figure 3: The process of screening physical and chemical properties by DBSCAN clustering.

So, the DCC eigenvalue of the sequence "TACATTCA" is about 0.241 under the physicochemical property of "Shift" and "Slide" and when lag is 5.

The dimension of the feature vector is $N \times (N-1) \times \mathrm{LAG}$ after DCC, where N is the number of physicochemical properties and LAG is the maximum of lag (lag = 1, 2, ⋯, LAG). In

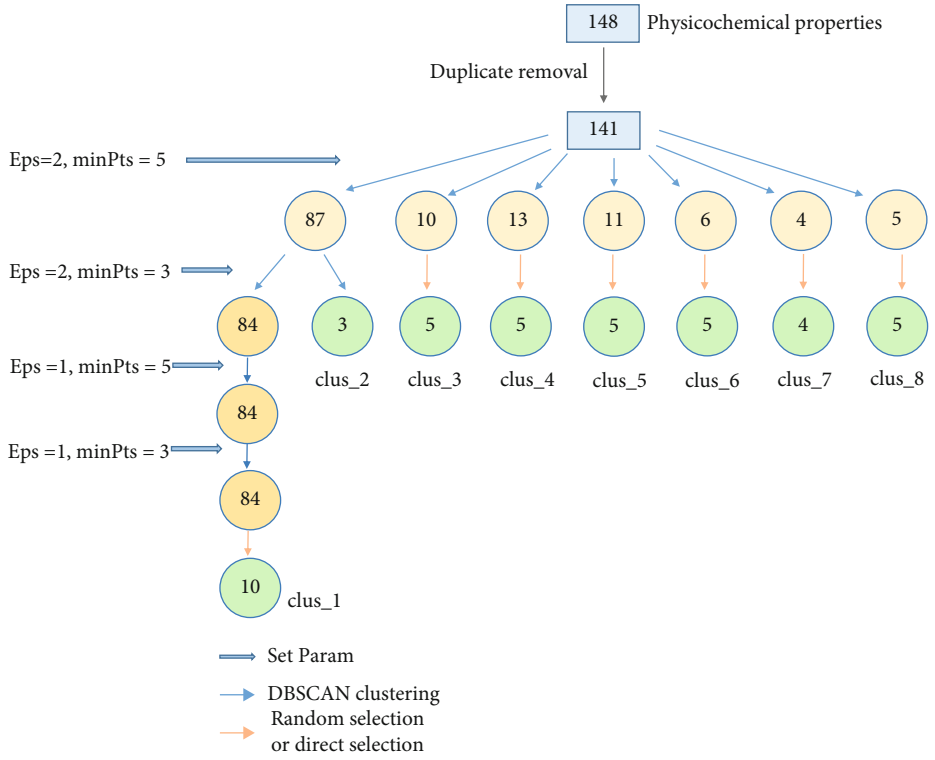

Figure 4: The specific situation after each clustering and the screening process of physical and chemical properties.

this paper, LAG is 5. Therefore, the final dimension of the eigenvector of DACC is $N \times N \times \text{LAG}$.

4.3. Density-Based Spatial Clustering of Applications with Noise (DBSCAN). DBSCAN can find clusters of any shape and can identify noise, which can achieve a better clustering effect for physical and chemical property data [12, 41]. The clusters are customized according to the parameters, respectively, "eps" (*e*-neighborhood with data point as center and *eps* as radius) and "minPts" (minimum number of data points in *e*-neighborhood). The steps of the DBSCAN algorithm are listed in Supplementary Materials.

In this paper, DBSCAN was used to screen the physicochemical properties of dinucleotides. Our DBSCAN clustering process of data points is shown in Figure 3. First, to avoid overfitting, one of the equivalent physicochemical property indexes was randomly selected, and 141 kinds were obtained. Then, we input four sets of parameter values, which make the clustering algorithm increasingly strict. By observing the results of the first round of clustering in Figure 4, it can be found that except for the large number of data in the first cluster, the number of data in other clusters is between 4 and 13. In order to select an appropriate amount of physical and chemical properties from the clusters, we set the data threshold N as 5. According to the rules we made, we filter the clusters after each DBSCAN until all the clusters are processed. The number of data points obtained by each clustering is counted as M. The screening rule is that when M is between 0 and N, all physicochemical properties are selected. When M is between N and $6 * N$, N pieces of data are randomly selected. When M is greater than $6 * N$, the next set of parameters is used to recluster the cluster. If all parameters have been tried and M are still greater than $6 * N$, $2 * N$ data will be randomly selected.

Figure 4 shows the number of clusters and the physicochemical properties number in each cluster after each cluster. Then, select the data in the cluster and a total of 47 dinucleotide physicochemical properties of 9 cluster types were finally obtained. The most representative physicochemical dinucleotide indexes were selected as much as possible, as detailed in Supplementary Materials.

4.4. Attention-Based Bi-LSTM. LSTM is a kind of time recurrent neural network that solves the long-term dependence problem of RNNs [42–45]. We can see the principal structure of LSTM in Figure 5(b), and its important components are the input gate, forgetting gate, and output gate. C_t is the cell state, which carries the memorized information and stores the information obtained through varied "gate" processing. C_t is similar to a kind of "long-term memory", and C_{t-1} is the cell state of the previous stage. h_t is similar to a kind of "short-term memory".

The first step in LSTM is to remove some information by working with the forgetting gate. The forgetting gate reads h_{t-1}, x_t and passes through the sigmoid neural layer. The element value range of the output vector is 0~1, which represents the probability of information retention. The point-by-point multiplication operation updates the information to the cell state.

$$f_t = \text{sigmod}\left(W_f \bullet [h_{t-1}, x_t] + b_f\right). \tag{7}$$

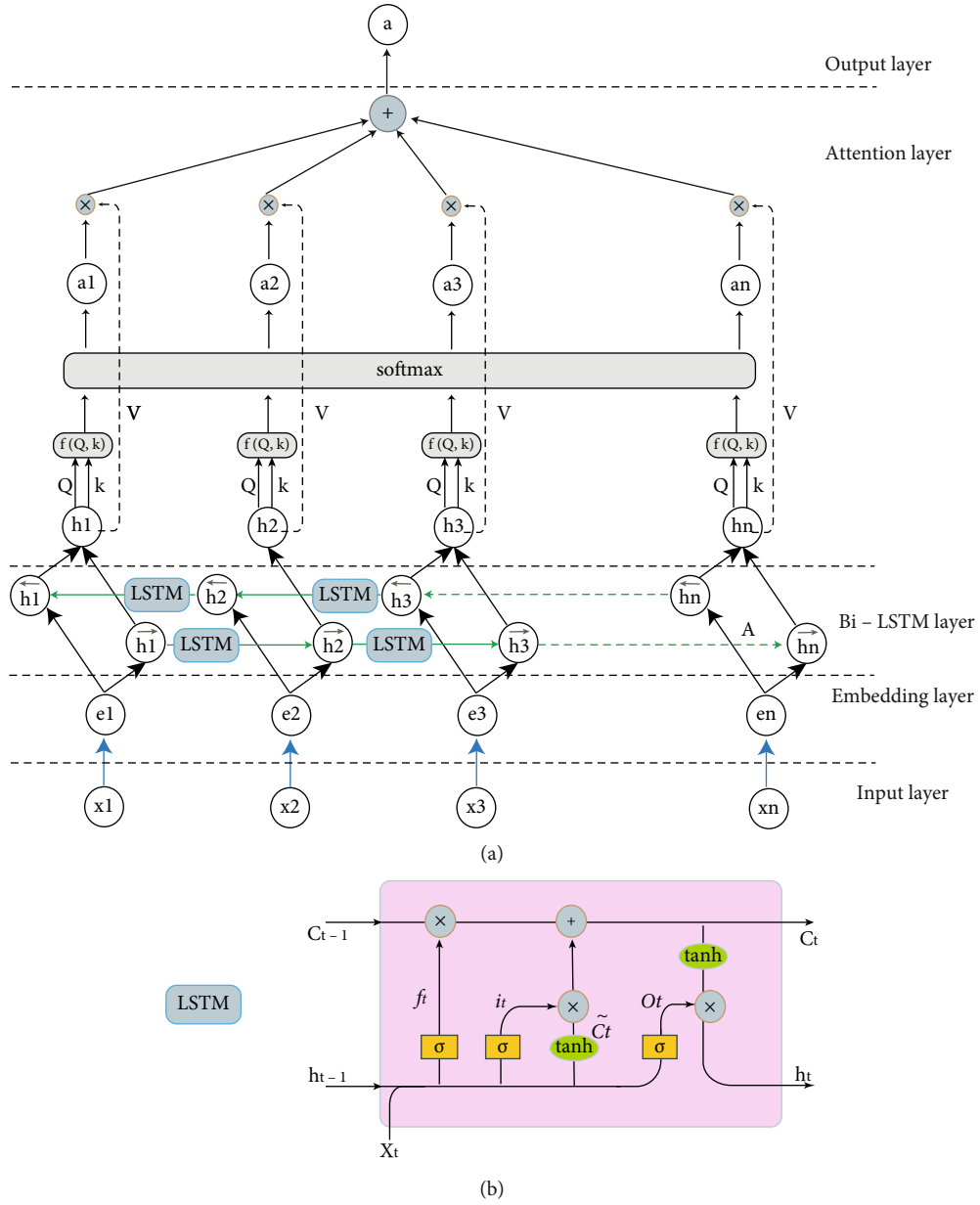

FIGURE 5: (a) The structure of attention-based Bi-LSTM. (b) The structure of LSTM in (a).

The second step is to add new information from the input gate. The second step is divided into three steps: first, let the sigmoid layer of the input gate determine which parts of the information need to be updated, then let the tanh layer generate alternative updates, and finally, combine the two parts to add the information to the cell state.

$$\begin{aligned} i_t &= \text{sigmoid}(W_i \bullet [h_{t-1}, x_t] + b_i), \\ \widetilde{C_t} &= \tanh\,(W_C \bullet [h_{t-1}, x_t] + b_C). \end{aligned} \tag{8}$$

The last step is to calculate and output the "short-term memory" state h_t by the output gate. First, let the sigmoid layer of the output gate decide the information part that needs to be updated; then, the tanh layer processes the cell state that has been updated and finally multiplies the two parts together to obtain h_t.

$$\begin{aligned} o_t &= \text{sigmoid}(W_o \bullet [h_{t-1}, x_t] + b_o), \\ h_t &= o_t \bullet \tanh\,(C_t). \end{aligned} \tag{9}$$

Therefore, the most special feature of LSTM is that it can forget unwanted information, add needed information, and obtain "short-term memory" according to "long-term memory" processing.

Bi-LSTM can better capture bidirectional semantic dependencies. Figure 5(a)shows the Bi-LSTM structure in

this article. After mapping, each word x_i obtains the word vector e_i. After LSTM, the forward output is $\overrightarrow{h_i}$, while the backward output is $\overleftarrow{h_i}$. After Bi-LSTM, the vector obtained is $h_i = [\overrightarrow{h_i} + \overleftarrow{h_i}]$, where "+" represents the sum of corresponding elements. We can see in Figure 5(a) the Bi-LSTM layer.

Attention-based Bi-LSTM was first proposed by Zhou et al. in 2016 [46, 47]. Bi-LSTM with an attention mechanism avoids complicated feature engineering in traditional work. The attention mechanism allocates attention to each word in the process of learning the current information to make the model more focused on learning and thus improve learning efficiency [48]. The model has various variants, and self-attention [49] is adopted in this paper. Attention values can be calculated in three steps. Above all, we calculate the similarity between query (Q) and each key (K) by $f(Q, K)$ to obtain weights. Then, the softmax function is used to normalize these weights. Finally, the weighted sum of the weights and the corresponding key value (V) is carried out to obtain the final attention value. In the self-attention model, query, key, and value are the same, that is, the input sentence sequence information h_i shown in Figure 5(a) which is the attention layer.

4.5. Model Evaluation. For evaluating and optimizing model performance, four evaluation indexes were used in this paper: ACC, SN, SP, representing accuracy, sensitivity, specificity, respectively, and MCC [38, 50–61]. Their mathematical formula is as follows:

$$\begin{aligned}
\text{Accuracy (ACC)} &= \frac{\text{TN} + \text{TP}}{\text{TN} + \text{TP} + \text{FN} + \text{FP}},\\
\text{Sensitivity (SN)} &= \frac{\text{TP}}{\text{TP} + \text{FN}},\\
\text{Specificity (SP)} &= \frac{\text{TN}}{\text{TN} + \text{FP}},\\
\text{MCC} &= \frac{\text{TN} \times \text{TP} - \text{FN} \times \text{FP}}{\sqrt{(\text{TP} + \text{FP})(\text{TN} + \text{FN})(\text{TP} + \text{FN})(\text{TN} + \text{FP})}},
\end{aligned} \tag{10}$$

where TP, TN, FP, and FN represent the true positive, true negative, false positive, and false negative values, respectively.

Data Availability

The data covered in this article can be found in Supplementary Materials.

Consent

Consent is not applicable.

Conflicts of Interest

No potential conflict of interest was reported by the authors.

Authors' Contributions

S.Z. designed the experiments and participated in coding the experiments. Q.Z. and L.S. conceived the study and participated in designing the study. Y.J., X.S., and Q.P. participated in performing the statistical analysis and coding the experiments and drafting the manuscript. All authors read and approved the final manuscript. Shulin Zhao and Qingfeng Pan contributed equally to this work.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (No. 61922020 and No. 62072385), the Special Science Foundation of Quzhou (2021D004), and the Sichuan Provincial Science Fund for Distinguished Young Scholars (2021JDJQ0025).

Supplementary Materials

The "S_47 physicochemical properties.txt" file is 47 of the 148 dinucleotides extracted after DBSCAN clustering of the physical and chemical properties. The "S_DBSCAN algorithm.docx" file is the DBSCAN algorithm framework. The "Training set.txt" file is the sequence samples in the training set. The "Independent testing set.txt" file is the sequence samples in the independent test set. *(Supplementary Materials)*

References

[1] L. Liu, L. R. Zhang, F. Y. Dao, Y. C. Yang, and H. Lin, "A computational framework for identifying the transcription factors involved in enhancer-promoter loop formation," *Molecular Therapy–Nucleic Acids*, vol. 23, pp. 347–354, 2021.

[2] L. W. K. Lim, H. H. Chung, Y. L. Chong, and N. K. Lee, "A survey of recently emerged genome-wide computational enhancer predictor tools," *Computational Biology and Chemistry*, vol. 74, pp. 132–141, 2018.

[3] T. Zhang, R. Wang, Q. Jiang, and Y. Wang, "An information gain-based method for evaluating the classification power of features towards identifying enhancers," *Current Bioinformatics*, vol. 15, no. 6, pp. 574–580, 2020.

[4] X. Yu, J. Zhou, M. Zhao et al., "Exploiting XG boost for predicting enhancer-promoter interactions," *Current Bioinformatics*, vol. 15, no. 9, pp. 1036–1045, 2021.

[5] C. Jia and W. He, "EnhancerPred: a predictor for discovering enhancers based on the combination and selection of multiple features," *Scientific Reports*, vol. 6, no. 1, 2016.

[6] B. Liu, L. Fang, R. Long, X. Lan, and K. C. Chou, "iEnhancer-2L: a two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition," *Bioinformatics*, vol. 32, no. 3, pp. 362–369, 2016.

[7] B. Liu, K. Li, D. S. Huang, and K. C. Chou, "iEnhancer-EL: identifying enhancers and their strength with ensemble learning approach," *Bioinformatics*, vol. 34, no. 22, pp. 3835–3842, 2018.

[8] Y. E. K. Le NQ, Q. T. Ho, N. Nagasundaram, Y. Y. Ou, and H. Y. Yeh, "iEnhancer-5Step: identifying enhancers using hidden information of DNA sequences via Chou's 5-step rule and

word embedding," *Analytical Biochemistry*, vol. 571, pp. 53–61, 2019.

[9] K. K. Tan, N. Q. K. Le, H. Y. Yeh, and M. C. H. Chua, "Ensemble of deep recurrent neural networks for identifying enhancers via dinucleotide physicochemical properties," *Cell*, vol. 8, no. 7, p. 767, 2019.

[10] Q. H. Nguyen, T. H. Nguyen-Vo, N. Q. K. le, T. T. T. Do, S. Rahardja, and B. P. Nguyen, "iEnhancer-ECNN: identifying enhancers and their strength using ensembles of convolutional neural networks," *BMC Genomics*, vol. 20, no. S9, p. 951, 2019.

[11] J. Khanal, H. Tayara, and K. T. Chong, "Identifying enhancers and their strength by the integration of word embedding and convolution neural network," *IEEE Access*, vol. 8, pp. 58369–58376, 2020.

[12] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited," *ACM Transactions on Database Systems*, vol. 42, no. 3, pp. 1–21, 2017.

[13] B. Liu, Y. Liu, X. Jin, X. Wang, and B. Liu, "iRSpot-DACC: a computational predictor for recombination hot/cold spots identification based on dinucleotide-based auto-cross covariance," *Scientific Reports*, vol. 6, no. 1, 2016.

[14] S. D. He, F. Guo, Q. Zou, and HuiDing, "MRMD2.0: a python tool for machine learning with feature ranking and reduction," *Current Bioinformatics*, vol. 15, no. 10, pp. 1213–1221, 2021.

[15] Q. He, W. Liu, and Z. Cai, "B&Anet: combining bidirectional LSTM and self-attention for end-to-end learning of task-oriented dialogue system," *Speech Communication*, vol. 125, pp. 15–23, 2020.

[16] R. Su, J. Hu, Q. Zou, B. Manavalan, and L. Wei, "Empirical comparison and analysis of web-based cell-penetrating peptide prediction tools," *Briefings in Bioinformatics*, vol. 21, no. 2, pp. 408–420, 2020.

[17] R. Su, X. Liu, L. Wei, and Q. Zou, "Deep-Resp-Forest: a deep forest model to predict anti-cancer drug response," *Methods*, vol. 166, pp. 91–102, 2019.

[18] R. Su, X. Liu, G. Xiao, and L. Wei, "Meta-GDBP: a high-level stacked regression model to improve anticancer drug response prediction," *Briefings in Bioinformatics*, vol. 21, no. 3, pp. 996–1005, 2020.

[19] R. Su, H. Wu, B. Xu, X. Liu, and L. Wei, "Developing a multi-dose computational model for drug-induced hepatotoxicity prediction based on toxicogenomics data," *IEEE-ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 4, pp. 1231–1239, 2019.

[20] J. Li, Y. Pu, J. Tang, Q. Zou, and F. Guo, "DeepATT: a hybrid category attention neural network for identifying functional effects of DNA sequences," *Briefings in Bioinformatics*, vol. 22, no. 3, pp. 1–1, 2021.

[21] J. Li, Y. Pu, J. Tang, Q. Zou, and F. Guo, "DeepAVP: a dual-channel deep neural network for identifying variable-length antiviral peptides," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 10, pp. 3012–3019, 2020.

[22] Y. Shang, L. Gao, Q. Zou, and L. Yu, "Prediction of drug-target interactions based on multi-layer network representation learning," *Neurocomputing*, vol. 434, pp. 80–89, 2021.

[23] Z. Hong, X. Zeng, L. Wei, and X. Liu, "Identifying enhancer–promoter interactions with neural network based on pre-trained DNA vectors and attention mechanism," *Bioinformatics*, vol. 36, no. 4, pp. 1037–1043, 2020.

[24] X. Min, C. Ye, X. Liu, and X. Zeng, "Predicting enhancer-promoter interactions by deep learning and matching heuristic," *Briefings in Bioinformatics*, vol. 22, no. 4, 2021.

[25] X. Zeng, Y. Zhong, W. Lin, and Q. Zou, "Predicting disease-associated circular RNAs using deep forests combined with positive-unlabeled learning methods," *Briefings in Bioinformatics*, vol. 21, no. 4, pp. 1425–1436, 2020.

[26] X. Zhang, Q. Zou, A. Rodriguez-Paton, and X. Zeng, "Meta-path methods for prioritizing candidate disease miRNAs," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 1, pp. 283–291, 2019.

[27] Y. Liu, X. Zeng, Z. He, and Q. Zou, "Inferring microRNA-disease associations by random walk on a heterogeneous network with multiple data sources," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 4, pp. 905–915, 2017.

[28] X. Zeng, Y. Liao, Y. Liu, and Q. Zou, "Prediction and validation of disease genes using HeteSim scores," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 3, pp. 687–695, 2017.

[29] Y. Hu, J. Y. Sun, Y. Zhang et al., "rs1990622 variant associates with Alzheimer's disease and regulates TMEM106B expression in human brain tissues," *BMC Medicine*, vol. 19, no. 1, p. 11, 2021.

[30] Y. Hu, H. Zhang, B. Liu et al., "rs34331204 regulates TSPAN13 expression and contributes to Alzheimer's disease with sex differences," *Brain*, vol. 143, no. 11, article e95, 2020.

[31] Y. Hu, S. Qiu, and L. Cheng, "Integration of multiple-omics data to analyze the population-specific differences for coronary artery disease," *Computational and Mathematical Methods in Medicine*, vol. 2021, Article ID 7036592, 2021.

[32] X. Lin, Z. Quan, Z. J. Wang, H. Huang, and X. Zeng, "A novel molecular representation with BiGRU neural networks for learning atom," *Briefings in Bioinformatics*, vol. 21, no. 6, pp. 2099–2111, 2020.

[33] X. Fu, L. Cai, X. Zeng, and Q. Zou, "StackCPPred: a stacking and pairwise energy content-based prediction of cell-penetrating peptides and their uptake efficiency," *Bioinformatics*, vol. 36, no. 10, pp. 3028–3034, 2020.

[34] L. Yu, M. Wang, Y. Yang et al., "Predicting therapeutic drugs for hepatocellular carcinoma based on tissue-specific pathways," *PLoS Computational Biology*, vol. 17, no. 2, article e1008696, 2021.

[35] Z. Chen, P. Zhao, F. Li et al., "iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data," *Briefings in Bioinformatics*, vol. 21, no. 3, pp. 1047–1057, 2020.

[36] X. Gu, Z. Chen, and D. Wang, "Prediction of G protein-coupled receptors with CTDC extraction and MRMD2.0 dimension-reduction methods," *Frontiers in Bioengineering and Biotechnology*, vol. 8, 2020.

[37] Z. Tao, Y. Li, Z. Teng, and Y. Zhao, "A method for identifying vesicle transport proteins based on LibSVM and MRMD," *Computational and Mathematical Methods in Medicine*, vol. 2020, Article ID 8926750, 2020.

[38] Y. Zhai, Y. Chen, Z. Teng, and Y. Zhao, "Identifying antioxidant proteins by using amino acid composition and protein-protein interactions," *Frontiers in Cell and Development Biology*, vol. 8, article 591487, 2020.

[39] A. Khatua, A. Khatua, and E. Cambria, "A tale of two epidemics: contextual Word2Vec for classifying twitter streams

during outbreaks," *Information Processing & Management*, vol. 56, no. 1, pp. 247–257, 2019.

[40] W. Z. Li and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658-1659, 2006.

[41] K. Indira and M. K. K. Devi, "Multi cloud based service recommendation system using DBSCAN algorithm," *Wireless Personal Communications*, vol. 115, no. 2, pp. 1019–1034, 2020.

[42] P. Ma, B. Jiang, Z. Lu, N. Li, and Z. Jiang, "Cybersecurity named entity recognition using bidirectional long short-term memory with conditional random fields," *Tsinghua Science and Technology*, vol. 26, no. 3, pp. 259–265, 2021.

[43] H. Lv, F. Y. Dao, Z. X. Guan, H. Yang, Y. W. Li, and H. Lin, "Deep-Kcr: accurate detection of lysine crotonylation sites using deep learning method," *Briefings in Bioinformatics*, vol. 22, no. 4, 2021.

[44] W. Ying, L. Zhang, and H. Deng, "Sichuan dialect speech recognition with deep LSTM network," *Frontiers of Computer Science*, vol. 14, no. 2, pp. 378–387, 2020.

[45] J. Chen, J. Li, and Q. Zou, "DeepM6ASeq-EL: prediction of human N6-methyladenosine (m6A) sites with LSTM and ensemble learning," *Frontiers of Computer Science*, vol. 16, no. 2, pp. 1–7, 2022.

[46] P. Zhou, W. Shi, J. Tian et al., "Attention-based bidirectional long short-term memory networks for relation classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 207–212, Berlin, Germany, 2016.

[47] X. Zheng and W. Chen, "An attention-based Bi-LSTM method for visual object classification via EEG," *Biomedical Signal Processing and Control*, vol. 63, p. 102174, 2021.

[48] D. Wang, Z. Zhang, Y. Jiang et al., "DM3Loc: multi-label mRNA subcellular localization prediction and analysis based on multi-head self-attention mechanism," *Nucleic Acids Research*, vol. 49, no. 8, p. e46, 2021.

[49] P. Bhuvaneshwari, A. N. Rao, and Y. H. Robinson, "Spam review detection using self attention based CNN and bidirectional LSTM," *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 18107–18124, 2021.

[50] D. Chicco, N. Totsch, and G. Jurman, "The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation," *Biodata Mining*, vol. 14, no. 1, p. 13, 2021.

[51] L. Wei, H. Chen, and R. Su, "M6APred-EL: a sequence-based predictor for identifying N6-methyladenosine sites using ensemble learning," *Molecular Therapy–Nucleic Acids*, vol. 12, pp. 635–644, 2018.

[52] L. Wei, Y. Ding, R. Su, J. Tang, and Q. Zou, "Prediction of human protein subcellular localization using deep learning," *Journal of Parallel and Distributed Computing*, vol. 117, pp. 212–217, 2018.

[53] L. Wei, M. Liao, Y. Gao, R. Ji, Z. He, and Q. Zou, "Improved and promising identification of human microRNAs by incorporating a high-quality negative set," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 1, pp. 192–201, 2014.

[54] L. Wei, C. Zhou, H. Chen, J. Song, and R. Su, "ACPred-FL: a sequence-based predictor using effective feature representation to improve the prediction of anti-cancer peptides," *Bioinformatics*, vol. 34, no. 23, pp. 4007–4016, 2018.

[55] H. Wang, Y. Ding, J. Tang, and F. Guo, "Identification of membrane protein types via multivariate information fusion with Hilbert-Schmidt Independence Criterion," *Neurocomputing*, vol. 383, pp. 257–269, 2020.

[56] Y. T. Ding, J. Tang, and F. Guo, "Identification of drug-target interactions via dual Laplacian regularized least squares with multiple kernel fusion," *Knowledge-Based Systems*, vol. 204, p. 106254, 2020.

[57] Y. Ding, J. Tang, and F. Guo, "Identification of drug-target interactions via fuzzy bipartite local model," *Neural Computing & Applications*, vol. 32, no. 14, pp. 10303–10319, 2020.

[58] L. Yu, D. Zhou, L. Gao, and Y. Zha, "Prediction of drug response in multilayer networks based on fusion of multiomics data," *Methods*, vol. 192, pp. 85–92, 2021.

[59] B. Małysiak-Mrozek, T. Baron, and D. Mrozek, "Spark-IDPP: high-throughput and scalable prediction of intrinsically disordered protein regions with Spark clusters on the Cloud," *Cluster Computing*, vol. 22, no. 2, pp. 487–508, 2019.

[60] X. Wang, Y. Yang, J. Liu, and G. Wang, "The stacking strategy-based hybrid framework for identifying non-coding RNAs," *Briefings in Bioinformatics*, vol. 22, no. 5, 2021.

[61] X. Zhao, Q. Jiao, H. Li et al., "ECFS-DEA: an ensemble classifier-based feature selection for differential expression analysis on expression profiles," *BMC Bioinformatics*, vol. 21, no. 1, p. 43, 2020.