Methods xxx (xxxx) xxx

Contents lists available at ScienceDirect

Methods

journal homepage: www.elsevier.com/locate/ymeth

GC6mA-Pred: A deep learning approach to identify DNA N6-methyladenine sites in the rice genome

Jianhua Cai [a,b], Guobao Xiao [a,*], Ran Su [c,*]

[a] Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, College of Computer and Control Engineering, Minjiang University, Fuzhou, China
[b] College of Computer and Data Science, Fuzhou University, Fuzhou, PR China
[c] College of Intelligence and Computing, Tianjin University, Tianjin, China

ARTICLE INFO

Keywords:
Deep learning
DNA N6-methyladenine
Convolution neural network
Graph neural network

ABSTRACT

Motivation: DNA N6-methyladenine (6mA) is a pivotal DNA modification for various biological processes. More accurate prediction of 6mA methylation sites plays an irreplaceable part in grasping the internal rationale of related biological activities. However, the existing prediction methods only extract information from a single dimension, which has some limitations. Therefore, it is very necessary to obtain the information of 6mA sites from different dimensions, so as to establish a reliable prediction method.

Results: In this study, a neural network based bioinformatics model named GC6mA-Pred is proposed to predict N6-methyladenine modifications in DNA sequences. GC6mA-Pred extracts significant information from both sequence level and graph level. In the sequence level, GC6mA-Pred uses a three-layer convolution neural network (CNN) model to represent the sequence. In the graph level, GC6mA-Pred employs graph neural network (GNN) method to integrate various information contained in the chemical molecular formula corresponding to DNA sequence. In our newly built dataset, GC6mA-Pred shows better performance than other existing models. The results of comparative experiments have illustrated that GC6mA-Pred is capable of producing a marked effect in accurately identifying DNA 6mA modifications.

1. Introduction

Among all types of epigenetic modifications, DNA methylation is a common but significant chemical modification [1,2]. The existence of DNA methylation will lead to accidental activation or repression of genes [3]. DNA methylation will affect many DNA properties, such as DNA stability, DNA-protein interaction, DNA conformation and chromatin structure, which will produce an effect on the gene expression [4–10]. This is how DNA methylation controls the gene expression. The process of DNA methylation is to connect methyl or hydroxymethyl to the bases of DNA molecules, which could be adenine bases or cytosine bases. There are four common DNA methylations, including N4-methylcytosine (4mC), N6-methyladenine (6mA), 5-methylcytosine (5mC) and 5-hydroxymethylcytosine (5hmC) [11–14]. N6-methyladenine refers to the adenine nucleotide methylated at the sixth position of the purine ring. 6mA is a low-level and non-canonical DNA based modification, which has been discovered in these major fields: bacteria, archaea and eukaryotes [2,15]. Many biological processes require the 6mA participation. In eukaryotes, it may undertake the task of carrying heritable epigenetic information [5,16–26]. Moreover, the presence of 6mA can also prevent restriction enzymes in bacteria [10]. Therefore, accurately determining the position of 6mA from genes is essential for systematically understanding and interpreting its biological activity process.

To solve this problem, various experimental methods are applied to identify the 6mA distribution such as ultra-high performance liquid chromatography coupled with mass spectrometry (UHPLC-ms/ms) [21], capillary electrophoresis and laser-induced fluorescence (CE-LIF) [27], methylated DNA immunoprecipitation sequencing (MeDIP-seq) [28] and single-molecule real-time sequencing (SMRT-seq) [29]. Through these experimental methods, 6mA modification is identified in 84% of genes in Chlamydomonas [30]. SMRT-seq helped researchers reveal that the content of methylated adenine detected in early-diverging fungi is much higher than that in other eukaryotes [5,22]. Although these experimental techniques can accurately detect 6mA modification in genes, such labor-intensive experimental methods consume a lot of time and energy, and these experiments are very expensive. To this end, some researchers proposed to employ computational methods to identify 6mA

* Corresponding authors.
E-mail addresses: gbx@mju.edu.cn (G. Xiao), ran.su@tju.edu.cn (R. Su).

https://doi.org/10.1016/j.ymeth.2022.02.001
Received 9 January 2022; Received in revised form 31 January 2022; Accepted 5 February 2022
Available online 9 February 2022

modifications [31]. Then, the 6mA modification is predicted in advance by the computational model, which will make the subsequent experiment more efficient and reduce the experimental cost to a great extent. Moreover, the prediction results of the computational model can provide valuable guidance information for laboratory based experiments.

For this purpose, researchers have developed a great quantity of algorithms to deal with the problem of identification of 6mA. Firstly, machine learning algorithms have been proved to be effective means to quickly identify 6mA in genes. For example, i6mA-Pred is the first bioinformatics tool proposed to predict 6mA modification in the rice genome [31]. In model i6mA-Pred, nucleotide chemical properties and nucleotide frequency are applied to describe DNA sequences. Then, the optimized feature vector provides better training data for support vector machine (SVM). Finally, the accuracy of i6mA-Pred for 6mA modification prediction on the rice genome dataset reaches 83.13%. SDM6A is another machine learning model proposed to predict 6mA as well [32]. Different from other models with only one single basic feature and classifier, SDM6A is determined five most appropriate feature encoding methods after exploring a variety of features. Subsequently, the optimal feature vector will be extracted to construct the computational models based on SVM and extremely randomized tree respectively. Finally, SDM6A develops a two-layer ensemble approach to integrate these models to obtain the 6mA prediction and improves the prediction accuracy to 88.2% on the rice genome dataset. 6mA-Finder is an online DNA 6mA modification prediction tool, which greatly improves the sensitivity of the model [33]. 6mA-Finder fuses the information output from seven different sequence encoding mechanisms and three types of physicochemical-based features. The key feature extracted by these methods makes 6mA-Finder show strong feature sensitivity in 6mA prediction. In addition to rice genes, in recent years, researchers also develop a variety of machine learning models to explore 6mA modification in some other different species, such as human, mice, C. elegans and so on [33–36].

Although the prediction models built by machine learning have achieved effective results, these models often rely on manual features, which makes the models lack generalization. Recently, some researchers have found that deep learning methods outperform on this issue. For instance, a neural network model called iIM-CNN was reported to identify 6mA modification in several different species [37]. The model iIM-CNN uses a two-layer CNN and add a max pooling layer after each convolution layer to screen the features. The experimental results indicate that iIM-CNN is capable of predicting 6mA modification in rice genome with 87.5% accuracy and the performance on other species is also better than other methods. Another model i6mA-DNC is invented to further make the performance of the prediction model better [38]. Different from other deep learning methods, i6mA-DNC divides the original DNA sequence into dinucleotide components to represent and learn the information of sequences. The vectorized DNA sequence is introduced into a two-layer CNN network for training and prediction. On the same dataset, i6mA-DNC gains accuracy of 88.6%, which is a little higher than iIM-CNN. Furthermore, a novel deep learning based model, called DNA6mA-MINT, is designed to identify 6mA sites [39]. In DNA6mA-MINT, a three-layer CNN network is employed to describe the input data. Subsequently, a Long Short-Term Memory (LSTM) layer processes these well represented data, and gives an optimal interpretation. Such a tool can predict the 6mA sites in an accuracy of 89.08%. Recently, more and more deep learning methods have emerged in this field [40,41]. Although the methods mentioned above show high performance in identification of 6mA modification, some improvable space and aspects are available. Moreover, most of the existing computational methods pay more attention to the information contained in the DNA sequence itself, but ignore the higher-level information.

In our study, a novel deep learning based model, GC6mA-Pred, is developed to predict the 6mA modification in the rice genome. In this model, we integrate the sequence features extracted by one-dimensional convolutional neural network and the graph structure features extracted by graph neural network. When testing with independent dataset, the prediction accuracy of our model is the highest, reaching 91.34%, among all competing methods. Further experimental analysis indicates that our feature extraction and fusion strategy is beneficial to promote the performance of the model. The information from one-dimensional sequence and two-dimensional graph structure can effectively supplement the unique information, so as to enhance the prediction ability for 6mA methylation sites. Our model effectively enhances the prediction performance of 6mA modification in DNA sequences, and is able to point out the direction in finding new 6mA methylation sites.

2. Materials and methods

Firstly, in this section, we introduce the benchmark datasets adopted in our research, and then provide the network architecture of GC6mA-Pred. After that, we describe the performance evaluation metrics and cross-validation method.

2.1. Benchmark datasets

It is of great significance to establish a more reasonable and efficient supervised learning model by using a higher quality benchmark dataset [42]. Chen et al proposed a benchmark dataset of rice DNA in their work [31], which has become a common dataset that is often used in the research of 6mA methylation identification of DNA sequences. As shown in Table 1, this dataset contains a total of 1760 samples, and the number of positive and negative samples is the same. Since the number of samples in this dataset is too small, we need a new dataset to train and construct our deep learning model. To solve this problem, we construct a new 6mA dataset of rice genome.

First, we search for a new rice genome database that has not been developed and used from the latest research [43]. Zhang et al extracted a new rice gene database and uploaded it to the Gene Expression Omnibus (GEO) database with code GSE108784, which is maintained by National Center for Biotechnology Information (NCBI). There are two kinds of rice genomes in this database, japonica rice and indica rice. The newly constructed data set in our study adopts japonica rice dataset. According to the annotation file in the database, we extract the DNA sequence containing 6mA site from the chromosome genome. To be consistent with other benchmark datasets, the 6mA site is placed in the middle of the sequence. Moreover, the length of each sample sequence remains 41nt. In order to prevent the sequence features on the same chromosome from being too similar, we evenly extract 6mA sequences from 12 chromosomes. Then, 6mA sequences with modification scores lower than 30 are stripped from the dataset. In addition, if there are many redundant data with high similarity in the dataset, it will lead to misjudgment of the model. These similar redundant data will mask the useful data in the dataset to a great extent, making the trained model have virtual high performance. Therefore, high similarity sequences need to be eliminated by CD-HIT software [44]. The filtering threshold is set as 70%, and then 12, 000 positive samples are sorted out. Since 6mA methylation sites appear less in the coding sequence of DNA, a certain number of non 6mA site sequences are extracted from the genome according to the annotation file. It should be noted that these sequences should be evenly distributed on each chromosome to ensure the balance of data sources. Then it also needs to reduce the similarity to less than 70% and finally sort out 12, 000 negative samples.

In this way, we construct a new rice genome dataset, including 12, 000 positive samples (6mA sequences) and 12, 000 negative samples

Table 1
Statistical summary of the datasets.

Dataset	Positive samples	Negative samples	Total
Chen's dataset	880	880	1760
Our dataset	12,000	12,000	24,000

(not 6mA sequences). The statistical summary of these two datasets is recorded in Table 1.

2.2. Feature representation

The proposed model uses DNA sequences as inputs. Examples of DNA sequences are described as follows:

$$D = N_1, N_2, N_3, \cdots, N_L \tag{1}$$

In formula (1), L represents the length of the DNA sequence. $N_i (1 \leq i \leq 41)$ represents the ith base constituting this DNA sequence and $N \in \{A, C, G, T\}$. In our model, the input DNA sequence is described by binary one-hot coding. The specific encoding scheme is described as follows:

$$\begin{cases} A \rightarrow (1,0,0,0) \\ C \rightarrow (0,1,0,0) \\ G \rightarrow (0,0,1,0) \\ T \rightarrow (0,0,0,1) \end{cases} \tag{2}$$

One-hot coding scheme can effectively distinguish and represent the features of a limited number of different elements. It is a common feature representation method in deep learning model. Every DNA sequence with length of 41 can be transformed into a 4×41 matrix after one-hot coding. Since this is an end-to-end network model, we do not require complex manual feature methods.

2.3. Proposed model

In this study, we fuse CNN and GNN methods to build a deep learning network model, GC6mA-Pred, which can identify and predict 6mA methylation sites in DNA sequences. Fig. 1 illustrates the overall network architecture of GC6mA-Pred. The whole network architecture can be divided into three parts: CNN feature representation, GNN feature representation, feature fusion and prediction. In the part of CNN feature representation, the input sample data is transformed into a 4×41 feature matrix through one-hot coding scheme. Then, the feature matrix is fed into a three-layer convolutional neural network for training. After learning, the model will get a 512-dimensional feature vector, which represents the sequence features learned by CNN. In the part of GNN feature representation, we first convert the DNA sequence into the format of chemical molecular formula. Then, the atoms in the molecular formula can be regarded as vertices, and the chemical bonds between atoms can be regarded as edges between vertices, so as to construct a graph. Then, the molecular graph is encoded and learned by a graph neural network, and a 20-dimensional feature vector can be obtained after learning. In the part of feature fusion and prediction, the feature vectors previously trained and learned by CNN and GNN will be fused into a feature vector and feed into the full connection layer. After several dimensionality reduction operations, the Sigmoid function makes the final prediction. When the output of Sigmoid function is 1, it means that the DNA sequence contains the 6mA methylation site. Otherwise, it means that this DNA sequence does not contain the 6mA methylation site. The following section describes the method of our model in detail from the CNN network architecture and the GNN network.

2.3.1. CNN network architecture of proposed model

Three convolution layers constitute the main architecture of our CNN network [45,46]. Each convolution layer is composed of one-dimensional convolution layer, max pooling layer, activation layer and dropout layer. For the DNA sequence data, one-dimensional convolution can better adapt and learn the local information of the sequence. Using the max pooling layer can effectively reduce the dimension of features, reduce the amount of parameters and retain more significant features. The activation function between two convolution layers is able to enhance nonlinear operation and strengthen expression and learning level. Finally, at the end of each layer, the dropout layer is used to discard some features to prevent the model from over fitting.

In order to describe the CNN network architecture in more detail, its parameter configuration is showed in Table 2. The specific operating parameters of CNN module can be clearly learned from Table 2. $Conv1D(f, s, d)$ means a one-dimensional convolution operation in which

Table 2
Detailed configuration of the proposed CNN

Layer	Output shape
Input	(4, 41)
Conv1D (8, 7, 2)	(8, 18)
ReLU	(8, 18)
Dropout (0.2)	(8, 18)
Conv1D (32, 3, 1)	(32, 18)
ReLU	(32, 18)
MaxPool1D (2, 2)	(32, 9)
Dropout (0.2)	(32, 9)
Conv1D (128, 3, 1)	(128, 9)
ReLU	(128, 9)
MaxPool1D (2, 2)	(128, 4)
Flatten	512

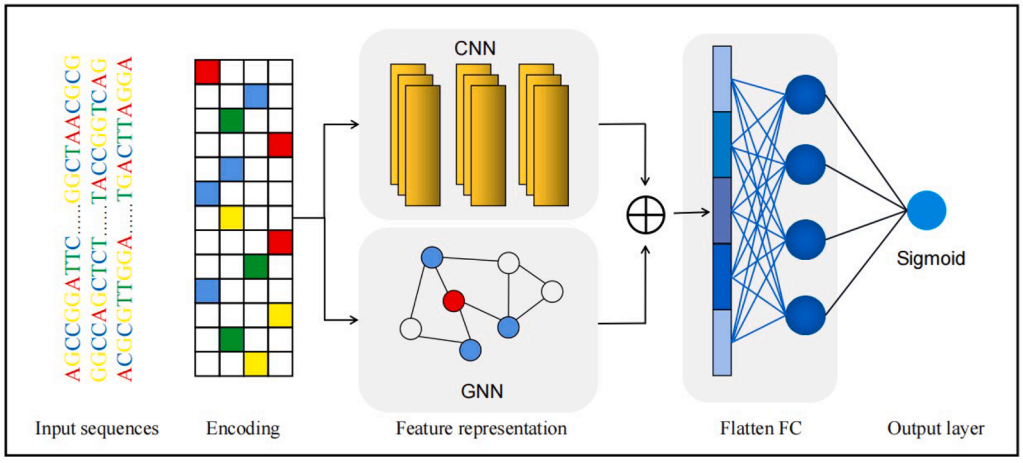

Fig. 1. Overall network architecture of GC6mA-Pred. Firstly, the input sequence data is encoded into a matrix. Then, the linear features are captured by one-dimensional convolutional neural network and two-dimensional structural features are extracted by graph neural network. The third step is to integrate the features from two levels and feed them into the full connection layer. Finally, the prediction is made by Sigmoid function in the output layer.

f indicates the number of filters, s indicates the size of the filter and d indicates the stride. The matrix after convolution will immediately pass through a nonlinear activation function ReLU. The $MaxPool1D(p, e)$ means a max-pooling layer in which p represents the pool-size and e represents the stride. $Dropout(r)$ means a dropout layer with a probability of r.

In the process of training CNN network, we mainly set two hyperparameters: learning rate and batch size. After screening and learning optimization, the learning rate is finally set to 0.0001. Too high learning rate may cause the network to miss the optimal solution. After testing, when the batch size is set to 8, the performance of the CNN network model is the best.

2.3.2. Graph neural network for molecular graph

Here, we employ a graph neural network to process the molecular graph transformed from DNA sequence. This GNN method is based on the article of Masashi Tsubaki et al [47].

First, we transform the DNA sequence into a molecular expression. This process mainly relies on the Rdkit toolkit. With this toolkit, we can easily transform DNA sequences into corresponding molecular expression forms. Then, the transition function in the GNN method can update the information of vertices (atoms in molecules) and edges (chemical bonds in molecules), and finally output it as a low dimensional vector through the output function. The whole GNN method mainly includes three modules:

2.3.2.1. Embedding based on molecular subgraph. We set $G = (D, E)$ to represent a molecular graph, where D represents the set of vertexes and E represents the set of edges. In the molecular graph, $d_x \in D$ means the *xth* vertex (atom) and $e_{xy} \in E$ represents the edge (chemical bond) between atoms x and y. In k-range neighbors, we define a k-neighbor subgraph as a set of all adjacent vertexes and edges within the k-hop range of vertexes. For more detailed point description, in the graph $G = (D, E)$, all adjacent vertex indexes in the k-hop range from the *xth* atom are expressed as $R(x, k)$. Then, the k-neighbor subgraph of vertex d_x can be defined as:

$$d_x^{(k)} = \left(\mathrm{D}_x^{(k)}, \mathrm{E}_x^{(k)}\right) \tag{3}$$

In formula (3) above, $\mathrm{D}_x^{(k)}$ and $\mathrm{E}_x^{(k)}$ can be calculated by the following formula.

$$\mathrm{D}_x^{(k)} = \left\{d_y | \mathrm{y} \in R(x, k)\right\} \tag{4}$$

$$\mathrm{E}_x^{(k)} = \left\{e_{mn} \in E | (m, n) \in R(x, k) \times R(x, k-1)\right\} \tag{5}$$

What's more, the k-neighbor subgraph of edge e_{xy} is defined as follows:

$$\mathrm{e}_{xy}^{(k)} = \left(\mathrm{D}_x^{(k-1)} \cup \mathrm{D}_y^{(k-1)}, \mathrm{E}_x^{(k)} \cap \mathrm{E}_y^{(k)}\right) \tag{6}$$

2.3.2.2. Transition functions. The transition function of vertexes in GNN is defined as follow. In the graph G, the embedding of the *xth* vertex in the time step s is expressed as $\mathbf{d}_x^{(s)}$. Then, the embedding update formula of this vertex at the next time step is:

$$\mathbf{d}_x^{(s+1)} = \sigma\left(\mathbf{d}_x^{(s)} + \sum_{\mathrm{y} \in R(x)} \mathbf{h}_{xy}^{(s)}\right) \tag{7}$$

In formula (7), σ is a Sigmoid function and $R(x)$ is the set of neighbor indexes of vertex x. $\mathbf{h}_{xy}^{(s)}$ represents the feature vector aggregated from the surrounding neighbors, and it is calculated by the following formula:

$$\mathbf{h}_{xy}^{(s)} = f\left(\boldsymbol{W}_{neighbor}\begin{bmatrix}\boldsymbol{d}_y^{(s)} \\ \mathbf{e}_{xy}^{(s)}\end{bmatrix} + \boldsymbol{b}_{neighbor}\right) \tag{8}$$

In above formula, f represents a nonlinear activation function like *ReLU*, $\boldsymbol{W}_{neighbor}$ represents weight matrix, $\boldsymbol{b}_{neighbor}$ represents bias vector, and $\mathbf{e}_{xy}^{(s)}$ represents the embedding of the edge between the *xth* vertex and the *yth* vertex when the time step is s. It can be seen from the formula that the update of vertexes mainly depends on the information of neighbor vertexes and the edges between them. As shown in Fig. 2(a), each vertex updates its own information by aggregating the information of surrounding vertexes and edges. Therefore, with the increase of time step, each vertex can gradually obtain the information of vertexes farther away, and even the global information.

The update iteration process of vertexes is also effective for edges. The update formula of $\mathbf{e}_{xy}^{(s)}$ can be expressed as:

$$\mathbf{e}_{xy}^{(s+1)} = \sigma\left(\mathbf{e}_{xy}^{(s)} + \mathbf{g}_{xy}^{(s)}\right) \tag{9}$$

$\mathbf{g}_{xy}^{(s)}$ in formula (9) can be calculated by the following formula.

$$\mathbf{g}_{xy}^{(s)} = f\left(\boldsymbol{W}_{side}\left(\mathbf{d}_x^{(s)} + \mathbf{d}_y^{(s)}\right) + \boldsymbol{b}_{side}\right) \tag{10}$$

In above formula, $\boldsymbol{W}_{side}$ represents weight matrix, $\boldsymbol{b}_{side}$ represents bias vector. Through above two formulas, we learn that the update of edge information mainly comes from the information aggregation of the two vertexes connected by this edge. As described in Fig. 2(b), the information update of edge $\mathbf{e}_{xy}^{(s)}$ is to aggregate its own information and the information of the two vertexes connected by this edge.

2.3.2.3. Molecular vector representation. After the transition function, we achieve the final vector embedding of a series of vertexes. In order to obtain the final vector output of a molecule, we calculate the average of all vertex vectors to represent the molecule. The formula is described as follows:

$$\mathrm{y}_{molecule} = \frac{1}{|D|}\sum_{x=1}^{|D|} \mathbf{d}_x^{(s)} \tag{11}$$

2.4. Performance evaluation

In this part, the cross-validation method and several common

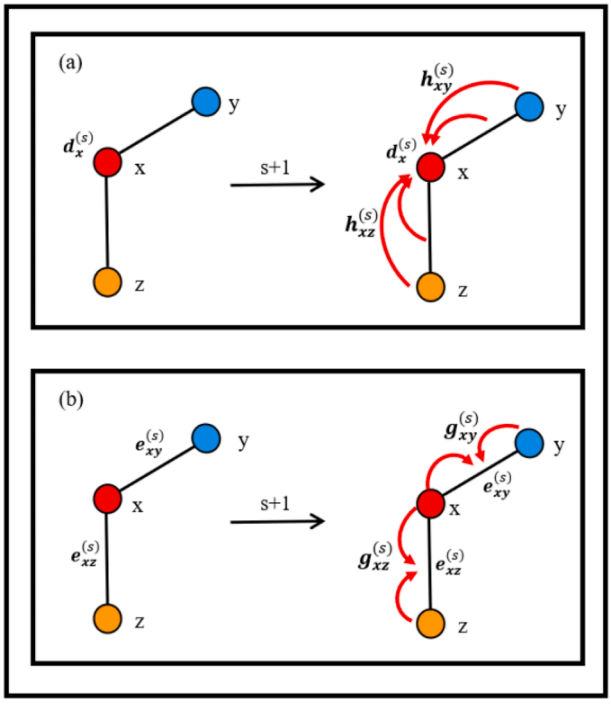

Fig. 2. Two transition functions in GNN. (a) depicts the transition function on the vertex. (b) describes the transition function on the edge.

evaluation metrics are introduced. K-fold cross validation, jackknife test and independent dataset test are several cross-validation methods often used [48,49]. When using k-fold cross validation, we need to divide the dataset into k subsets evenly. Then, k-1 of these subsets will be selected as the training set and the remaining one as the test set. The test set is used to test the model trained by other k-1 subsets. After testing the model k times, all the subsets have been tested once in turn. In these experiments, we obtain k groups of evaluation metrics. Finally, the average values of these k groups of evaluation metrics are calculated as the result to estimate the performance of predictor. Jackknife test is an extreme case of k-fold cross validation. The idea of jackknife test is to retain only one sample at a time as the test set. Then use the same method as k-fold cross validation to test k times, where k is the total number of samples. In the independent dataset test, some samples need to be segmented from the dataset as a test set before model training, and this test set does not participate in any training and validating process of the model but only for the final test. This dataset division method can effectively prevent the data in the test set from being learned by the model in advance. In the experiments of our study, the independent dataset test method is applied to evaluate the performance of the model.

Among so many evaluation metrics, accuracy (ACC), sensitivity (Sn), specificity (Sp) and Matthew's correlation coefficient (MCC) are certain effective representations often used to assess the quality of models [50–60].

The metric accuracy is the proportion of the correctly predicted samples in the total samples. Accuracy is the most commonly used and intuitive prediction metric. The accuracy formula is defined as follows:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{12}$$

In the above formula (12), TP means true positive, which represents the number of correct classification conclusions made by the model in positive samples. TN means true negative, which represents the number of correct classification conclusions made by the model in negative samples. FP means false positive, which represents the number of false classification conclusions made by the model in negative samples. FN means false negative, which represents the number of false classification conclusions made by the model in positive samples. In this study, a positive sample means that this DNA sequence contain 6mA methylation sites in the middle, and a negative sample means that this DNA sequence does not contain 6mA methylation sites.

The sensitivity indicates the proportion of correct classification made by the model in the positive samples. It is usually used to express the performance of the classifier on positive samples. The formula for sensitivity is defined as follows:

$$Sn = \frac{TP}{TP + FN} \tag{13}$$

The metric specificity indicates the proportion of correct classification made by the model in the negative samples. It shows the classification ability of the model for negative samples. The formula for specificity is defined as follows:

$$Sp = \frac{TN}{TN + FP} \tag{14}$$

The evaluation metric MCC reflects the overall quality of the predictor. The value of MCC ranges from −1 to 1. The higher the score of MCC, the stronger the classification ability of the classifier. While the MCC score is 0, it indicates that the prediction at this time is random. On the contrary, the lower the score of MCC, the weaker the classification ability of classifier. The MCC formula is defined as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TN + FN) \times (TN + FP) \times (TP + FN) \times (TP + FP)}} \tag{15}$$

In addition, the Receiver Operating Characteristic (ROC) curve is also used to visualize the prediction ability of the predictor. The predictor corresponding to the curve closer to the upper left corner of the coordinate axis has better prediction performance. At the same time, we calculate the area under the ROC curve, called AUC, to further measure the performance of the classifier. The value of AUC ranges from 0 to 1. Having a higher AUC value can prove that the prediction performance of the model is better. If the AUC value is 0.5, it means the classification result made by the predictor is random. AUC value below 0.5 means poor predictor performance.

3. Results and discussion

3.1. Performance on independent dataset

Firstly, the independent dataset is applied to validate our proposed model GC6mA-Pred. The evaluation metrics obtained from the validation are presented in the third row of Table 3. As recorded in Table 3, the test results of GC6mA-Pred on ACC, Sn, Sp, MCC and AUC are 91.34%, 95%, 90.52%, 0.86 and 0.928 respectively. We draw the performance radar chart in Fig. 3 to reflect the quality of GC6mA-Pred more intuitively. Each vertex in the pentagonal radar chart of GC6mA-Pred is close to the edge. It indicates that all metrics of the model perform well.

In addition, the visualization of confusion matrix performance of GC6mA-Pred is also provided in Fig. 4. The darker the box color in the visual confusion matrix, the higher the score of this part. From the confusion matrix, we can observe the prediction of GC6mA-Pred for each class. For example, among the positive samples, GC6mA-Pred can correctly predict 93% of the samples and mispredict the remaining 7%. While in the negative samples, only 89% of the samples are correctly predicted by GC6mA-Pred. From the comparison results of confusion matrix, we learn that GC6mA-Pred shows better predictive performance in positive samples.

3.2. Comparison results of different features

In this part, we mainly discuss the influence of single feature and fused features on model performance. The proposed model GC6mA-Pred is composed of two parts: one-dimensional convolutional neural network for processing linear sequence and graph neural network for processing graph. The experimental results of using one feature alone and fusing two features are recorded in Table 3. Obviously, the prediction accuracy obtained by using CNN method alone is significantly higher than that by using GNN method alone. This result shows the advantage of one-dimensional convolutional neural network in processing linear samples. Although the prediction performance obtained by using GNN features alone is poor, we find that the fusion of these two features can slightly ameliorate the prediction accuracy and other indicators of the classifier. Hence, it illustrates that the method of capturing and fusing features from one-dimensional linear level and two-dimensional graph level is feasible.

In order to further observe the performance differences between different features, we draw the ROC curve of the models using different features in Fig. 5. It is obvious that the ROC curve of the fusion feature (the black line) is obviously located at the top of the three curves. The ROC curve of using CNN feature (the yellow line) takes the second place, and the ROC curve of using GNN feature (the pink line) is at the bottom. This means that the fusion of these two different features can promote the performance of the model. Moreover, the AUC values of the three curves can also confirm this view.

Table 3
Validation result of single feature and fused feature

Feature	ACC(%)	Sn(%)	Sp(%)	MCC	AUC
CNN	90.56	94.09	85.86	0.77	0.908
GNN	64.15	55.91	69.39	0.61	0.697
CNN + GNN	**91.34**	**95.00**	**90.52**	**0.86**	**0.928**

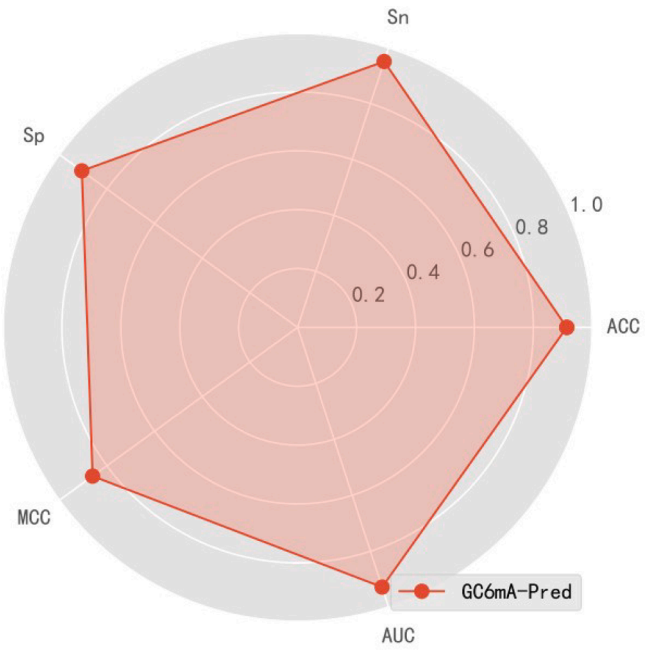

Fig. 3. Performance radar chart of GC6mA-Pred.

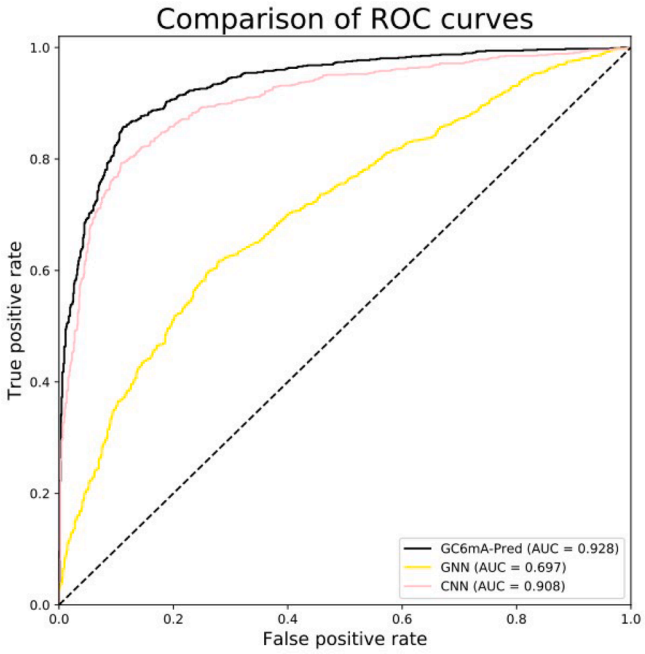

Fig. 5. ROC curves of different features.

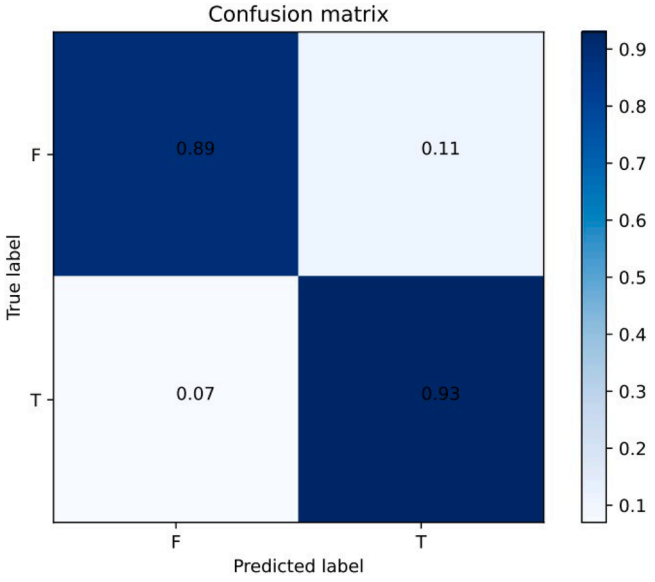

Fig. 4. The visualization of confusion matrix of GC6mA-Pred.

3.3. *Performance on Chen's dataset*

To explore the generalization ability of GC6mA-Pred, we apply the model to the dataset provided by Chen et al [31]. In this experiment, GC6mA-Pred is trained by our train dataset and tested on Chen's dataset, and the experimental results are listed in Table 4. For comparison, it also lists the prediction results of the model i6mA-Pred proposed by Chen et al in Table 4. It can be clearly seen from Table 4 that GC6mA-Pred performs slightly better than i6mA-Pred on Chen's dataset. Specifically, the prediction accuracy of our model is 0.39% higher than that of i6mA-Pred. Furthermore, the metric Sn of GC6mA-Pred is slightly lower than that of i6mA-Pred, but the Sp value is about 3% higher. This illustrates that GC6mA-Pred and i6ma-Pred have their own advantages in the processing of positive and negative samples. Model i6mA-Pred is better at predicting positive samples, while our model is more specific

Table 4
Comparison on Chen's dataset

Method	ACC(%)	Sn(%)	Sp(%)	MCC	AUC
i6mA-Pred	83.13	**82.95**	83.30	0.66	0.886
GC6mA-Pred	**83.52**	81.02	**86.02**	**0.67**	**0.889**

for negative samples. AUC value and other evaluation indicators suggest that the prediction ability of GC6mA-Pred is slightly better than i6mA-Pred.

To compare the indicators of these two methods more directly and clearly, their radar chart is shown in Fig. 6. From the radar chart, it can be seen that the metrics of the two models are very close. Significantly,

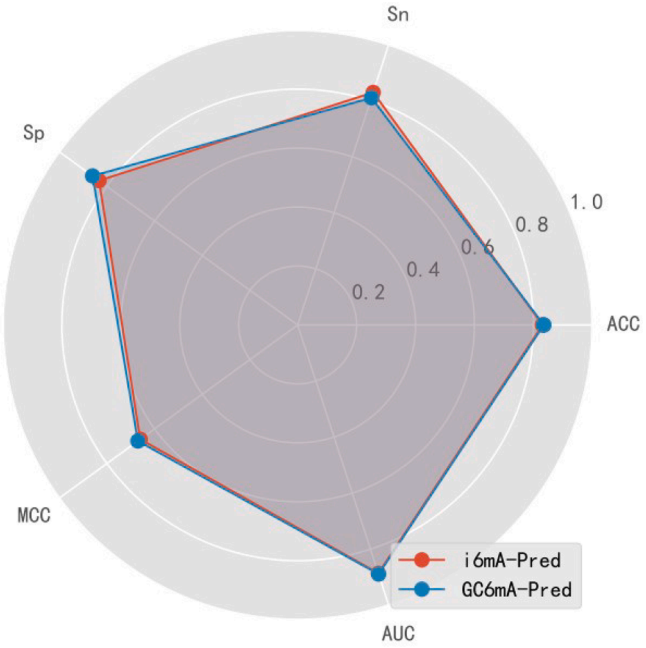

Fig. 6. Performance comparison on Chen's dataset.

GC6mA-Pred was not trained with Chen's dataset before testing. And then, it can slightly outperform i6mA-Pred, which shows good generalization ability.

3.4. Performance comparison with the existing methods

In this section, we validate GC6ma-Pred with several existing models on our new proposed dataset. These models include i6mA-Pred [31], SDM6A [32], iIM-CNN [37], 6mA-Finder [33], i6mA-DNC [38], DNA6mA-MINT [39] and i6mA-stack [61]. In this comparative experiment, the independent dataset test validation method is applied to our dataset. The specific experimental results of every model are presented in Table 5. It is obvious from Table 5 that the prediction accuracy of GC6mA-Pred is the highest among these models, reaching 91.34%. For the evaluation metric Sn, the performance of model i6mA-DNC is 0.94% higher than that of our proposed GC6mA-Pred, which indicates that i6mA-DNC is more prominent in the prediction of positive samples. In the remaining metrics, GC6mA-Pred outperforms other models. In particular, the Sp value of GC6mA-Pred is significantly higher than that of other predictor, which shows that our GC6mA-Pred has higher performance on negative samples. These experimental results indicate that GC6mA-Pred is capable of capturing the characteristic specificity in the DNA sequences effectively, especially in negative samples.

For intuitive comparison, we further draw the radar chart of the model performance as illustrated in Fig. 7. As seen, the pentagonal graph of GC6mA-Pred almost contains the radar graphs of other models. This phenomenon can intuitively indicate that the overall quality of our provided GC6mA-Pred outperforms the other existing models in all aspects. It also shows that the idea of using two levels to extract sequence and graph level features is correct. By effectively combining one-dimensional sequence information and two-dimensional graph information, the model can better understand the sample data and make more accurate judgment.

4. Conclusions

In this study, a deep learning based bioinformatics predictor, called GC6mA-Pred, have been proposed to identify 6mA methylation in the rice genome. To sufficiently extract the characteristic information of 6mA site, we adopt a two-level feature extraction method. The features captured by one-dimensional convolutional neural network are one-dimensional sequence level features. The features extracted by graph neural network are the two-dimensional structure features. Through the feature analysis experiment, it can be concluded that the fused features have more representation ability. This also demonstrates that different levels of feature information are complementary. The fusion of one-dimensional sequence features and two-dimensional graph structure features can enhance the overall performance of the model. In addition, we also compare our proposed GC6mA-Pred with other models on the same independent dataset. The experimental results indicate that our model achieves higher scores in all the metrics except Sn. In addition to experimenting on the new constructed dataset, we also test our GC6mA-Pred on the old dataset. The results of comparison illustrate that our proposed model still achieves a slightly higher prediction accuracy even without using this dataset for training. Overall, our model provides a more exact and effective avenue to identify 6 mA methylation in rice genome. In subsequent studies, we will concentrate more on the accurate prediction of the model on cross datasets or cross species.

CRediT authorship contribution statement

Jianhua Cai: Conceptualization, Methodology, Software, Writing – original draft. **Guobao Xiao:** Methodology, Writing – review & editing, Supervision. **Ran Su:** Visualization, Investigation, Writing – review & editing.

Table 5
Validation results of GC6mA-Pred and existing models

Model	ACC(%)	Sn(%)	Sp(%)	MCC	AUC
i6mA-Pred	82.65	87.39	77.92	0.66	0.860
SDM6A	88.38	91.04	78.85	0.70	0.895
iIM-CNN	85.62	94.48	76.77	0.72	0.901
6 mA-Finder	83.38	91.56	75.21	0.68	0.885
i6mA-DNC	87.03	**95.94**	78.13	0.75	0.901
DNA6mA-MINT	78.85	80.00	77.71	0.58	0.722
i6mA-stack	85.73	90.21	81.25	0.72	–
GC6mA-Pred	**91.34**	95.00	**90.52**	**0.86**	**0.928**

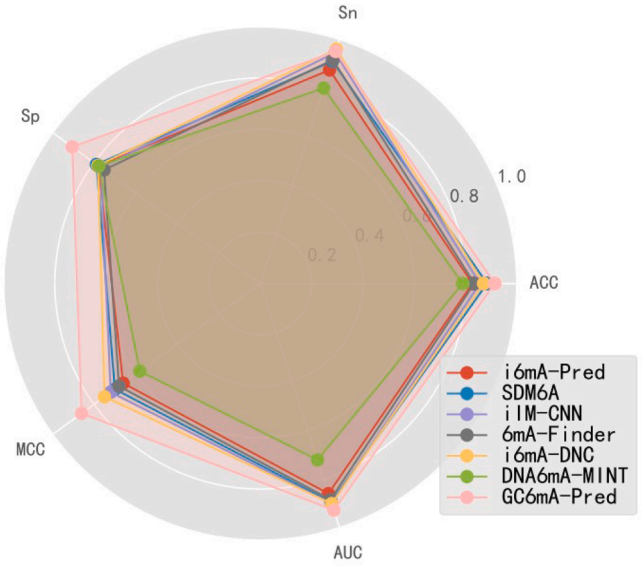

Fig. 7. Performance comparison of different models.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grants 62072223, and supported by the Natural Science Foundation of Fujian Province under Grant 2020J01131199.

References

[1] H. Lv, F.-Y. Dao, D. Zhang, H. Yang, H. Lin, Advances in mapping the epigenetic modifications of 5-methylcytosine (5mC), N6-methyladenine (6mA), and N4-methylcytosine (4mC), Biotechnol Bioeng. 118 (11) (2021) 4204–4216.
[2] Z.K. O'Brown, E.L. Greer, N6-methyladenine: a conserved and dynamic DNA Mark, Adv. Exp. Med. Biol. 945 (2016) 213–246.
[3] B. Jin, Y. Li, K.D. Robertson, DNA methylation: superior or subordinate in the epigenetic hierarchy? Genes Cancer 2 (6) (2011) 607–617.
[4] D. Yalcin, H.H. Otu, An unbiased predictive model to detect DNA methylation propensity of CpG Islands in the human genome, Curr. Bioinf. 16 (2) (2021) 179–196.
[5] Y. Zuo, M. Song, H. Li, X. Chen, P. Cao, L. Zheng, G. Cao, Analysis of the epigenetic signature of cell reprogramming by computational DNA methylation profiles, Curr. Bioinf. 15 (6) (2020) 589–599.
[6] Z. Du, X. Xiao, V.N. Uversky, Classification of chromosomal DNA sequences using hybrid deep learning architectures, Curr. Bioinf. 15 (10) (2021) 1130–1136.
[7] O. Koukoura, D.A. Spandidos, A. Daponte, S. Sifakis, DNA methylation profiles in ovarian cancer: implication in diagnosis and therapy (Review), Mol. Med. Rep. 10 (1) (2014) 3–9.
[8] C.-L. Xiao, S. Zhu, M. He, D.e. Chen, Q. Zhang, Y. Chen, G. Yu, J. Liu, S.-Q. Xie, F. Luo, Z. Liang, D.-P. Wang, X.-C. Bo, X.-F. Gu, K. Wang, G.-R. Yan, N(6)-

methyladenine DNA modification in the human genome, Mol. Cell 71 (2) (2018) 306–318.e7.
[9] D. Wion, J. Casadesús, N6-methyl-adenine: an epigenetic signal for DNA-protein interactions, Nat. Rev. Microbiol. 4 (3) (2006) 183–192.
[10] H. Heyn, M. Esteller, An adenine code for DNA: a second life for N6-methyladenine, Cell 161 (4) (2015) 710–713.
[11] Z. Lv, D. Wang, H. Ding, B. Zhong, L. Xu, Escherichia Coli DNA N-4-methycytosine site prediction accuracy improved by light gradient boosting machine feature selection technology, IEEE Access 8 (2020) 14851–14859.
[12] H. Zulfiqar, Z.-J. Sun, Q.-L. Huang, S.-S. Yuan, H. Lv, F.-Y. Dao, H. Lin, Y.-W. Li, Deep-4mCW2V: A sequence-based predictor to identify N4-methylcytosine sites in Escherichia coli, Methods (2021), https://doi.org/10.1016/j.ymeth.2021.07.011.
[13] J. Cheng, P. Baldi, A machine learning information retrieval approach to protein fold recognition, Bioinformatics 22 (12) (2006) 1456–1463.
[14] X. Luo, F. Wang, G. Wang, Y. Zhao, Identification of methylation states of DNA regions for Illumina methylation BeadChip, BMC Genomics 21 (Suppl 1) (2020) 672.
[15] G.-Z. Luo, C. He, DNA N(6)-methyladenine in metazoans: functional epigenetic mark or bystander? Nat. Struct. Mol. Biol. 24 (6) (2017) 503–506.
[16] S. Pei, J. Guan, Classifying cognitive normal and early mild cognitive impairment of Alzheimer's disease by applying restricted Boltzmann machine to fMRI Data, Curr. Bioinf. 16 (2) (2021) 252–260.
[17] L. Luo, L. Zhang, Quantum patterns of genome size variation in angiosperms, Curr. Bioinf. 16 (1) (2021) 80–89.
[18] Y. Zhang, J. Yan, S. Chen, M. Gong, D. Gao, M. Zhu, W. Gan, Review of the applications of deep learning in bioinformatics, Curr. Bioinf. 15 (8) (2021) 898–911.
[19] Y. Zhang, Artificial intelligence for bioinformatics and biomedicine, Current Bioinformatics 15 (8) (2021) 801–802.
[20] Z. Lv, H. Ding, L. Wang, Q. Zou, A convolutional neural network using dinucleotide one-hot encoder for identifying DNA N6-methyladenine sites in the rice genome, Neurocomputing 422 (2021) 214–221.
[21] E.L. Greer, M.A. Blanco, L. Gu, E. Sendinc, J. Liu, D. Aristizabal-Corrales, C.H. Hsu, L. Aravind, C. He, Y. Shi, DNA methylation on N6-adenine in C. elegans, Cell 161 (4) (2015) 868–878.
[22] S.J. Mondo, R.O. Dannebaum, R.C. Kuo, K.B. Louie, A.J. Bewick, K. LaButti, S. Haridas, A. Kuo, A. Salamov, S.R. Ahrendt, R. Lau, B.P. Bowen, A. Lipzen, W. Sullivan, B.B. Andreopoulos, A. Clum, E. Lindquist, C. Daum, T.R. Northen, G. Kunde-Ramamoorthy, R.J. Schmitz, A. Gryganskyi, D. Culley, J. Magnuson, T. Y. James, M.A. O'Malley, J.E. Stajich, J.W. Spatafora, A. Visel, I.V. Grigoriev, Widespread adenine N6-methylation of active genes in fungi, Nat Genet 49 (6) (2017) 964–968.
[23] L. Wei, S. Luan, L.A.E. Nagai, R. Su, Q. Zou, Exploring sequence-based features for the improved prediction of DNA N4-methylcytosine sites in multiple species, Bioinformatics 35 (8) (2019) 1326–1333.
[24] L. Wei, R. Su, B. Wang, X. Li, Q. Zou, X. Gao, Integration of deep feature representations and handcrafted features to improve the prediction of N-6-methyladenosine sites, Neurocomputing 324 (2019) 3–9.
[25] L. Wei, H. Chen, R. Su, M6APred-EL: a sequence-based predictor for identifying N6-methyladenosine sites using ensemble learning, Mol. Therapy-Nucleic Acids 12 (2018) 635–644.
[26] L. Wei, W. He, A. Malik, R. Su, L. Cui, B. Manavalan, Computational prediction and interpretation of cell-specific replication origin sites from multiple eukaryotes by exploiting stacking framework, Brief Bioinf. (2020).
[27] A.M. Krais, M.G. Cornelius, H.H. Schmeiser, Genomic N(6)-methyladenine determination by MEKC with LIF, Electrophoresis 31 (21) (2010) 3548–3551.
[28] K.R. Pomraning, K.M. Smith, M. Freitag, Genome-wide high throughput analysis of DNA methylation in eukaryotes, Methods 47 (3) (2009) 142–150.
[29] B.A. Flusberg, D.R. Webster, J.H. Lee, K.J. Travers, E.C. Olivares, T.A. Clark, J. Korlach, S.W. Turner, Direct detection of DNA methylation during single-molecule, real-time sequencing, Nat. Methods 7 (6) (2010) 461–465.
[30] Y.e. Fu, G.-Z. Luo, K. Chen, X. Deng, M. Yu, D. Han, Z. Hao, J. Liu, X. Lu, L. Doré, X. Weng, Q. Ji, L. Mets, C. He, N6-methyldeoxyadenosine marks active transcription start sites in Chlamydomonas, Cell 161 (4) (2015) 879–892.
[31] Chen W, Lv H, Nie F, Lin H: i6mA-Pred: identifying DNA N6-methyladenine sites in the rice genome. *Bioinformatics* 2019, 35(16):2796–2800.
[32] S. Basith, B. Manavalan, T.H. Shin, G. Lee, SDM6A: a web-based integrative machine-learning framework for predicting 6mA Sites in the rice genome, Mol. Ther. Nucleic Acids 18 (2019) 131–141.
[33] H. Xu R. Hu P. Jia Z. Zhao P. Luigi Martelli 6mA-Finder: a novel online tool for predicting DNA N6-methyladenine sites in genomes 36 10 2020 2020 3257 3259.
[34] C. Pian, G. Zhang, F. Li, X. Fan, MM-6mAPred: identifying DNA N6-methyladenine sites based on Markov model, Bioinformatics 36 (2) (2020) 388–392.
[35] H. Lv, F.Y. Dao, Z.X. Guan, D. Zhang, J.X. Tan, Y. Zhang, W. Chen, H. Lin, iDNA6mA-Rice: a computational tool for detecting N6-methyladenine sites in rice, Front. Genet. 10 (2019) 793.
[36] H.T. Wang, F.H. Xiao, G.H. Li, Q.P. Kong, Identification of DNA N(6)-methyladenine sites by integration of sequence features, Epigenetics Chromatin 13 (1) (2020) 8.
[37] A. Wahab, S.D. Ali, H. Tayara, K. To Chong, iIM-CNN: intelligent identifier of 6ma sites on different species by using convolution neural network, IEEE Access 7 (2019) 178577–178583.
[38] S. Park, A. Wahab, I. Nazari, J.H. Ryu, K.T. Chong, i6mA-DNC: Prediction of DNA N6-Methyladenosine sites in rice genome based on dinucleotide representation using deep learning, Chemomet. Intelligent Lab. Syst. 204 (2020) 104102, https://doi.org/10.1016/j.chemolab.2020.104102.
[39] M.U. Rehman, K.T. Chong, DNA6mA-MINT: DNA-6mA modification identification neural tool, Genes (Basel) 11 (8) (2020) 898, https://doi.org/10.3390/genes11080898.
[40] P. Ni, N. Huang, Z. Zhang, D.P. Wang, F. Liang, Y. Miao, C.L. Xiao, F. Luo, J. Wang, DeepSignal: detecting DNA methylation state from Nanopore sequencing reads using deep-learning, Bioinformatics 35 (22) (2019) 4586–4595.
[41] H. Yu, Z. Dai, SNNRice6mA: a deep learning method for predicting DNA N6-methyladenine sites in rice genome, Front. Genet. 10 (2019) 1071.
[42] W. Su, M.-L. Liu, Y.-H. Yang, J.-S. Wang, S.-H. Li, H. Lv, F.-Y. Dao, H. Yang, H. Lin, PPD: a manually curated database for experimentally verified prokaryotic promoters, J. Mol. Biol. 433 (11) (2021) 166860.
[43] Q. Zhang, Z. Liang, X. Cui, C. Ji, Y. Li, P. Zhang, J. Liu, A. Riaz, P.u. Yao, M. Liu, Y. Wang, T. Lu, H. Yu, D. Yang, H. Zheng, X. Gu, N(6)-methyladenine DNA methylation in Japonica and Indica rice genomes and its association with gene expression, plant development, and stress responses, Mol Plant 11 (12) (2018) 1492–1508.
[44] L. Fu, B. Niu, Z. Zhu, S. Wu, W. Li, CD-HIT: accelerated for clustering the next-generation sequencing data, Bioinformatics 28 (23) (2012) 3150–3152.
[45] H. Lv, F.Y. Dao, H. Zulfiqar, H. Lin, DeepIPs: comprehensive assessment and computational identification of phosphorylation sites of SARS-CoV-2 infection using a deep learning-based approach, Brief. Bioinf. (2021).
[46] H. Lv, F.Y. Dao, Z.X. Guan, H. Yang, Y.W. Li, H. Lin, Deep-Kcr: accurate detection of lysine crotonylation sites using deep learning method, Brief. Bioinf. 22 (4) (2021).
[47] M. Tsubaki K. Tomii J. Sese J. Wren Compound-protein interaction prediction with end-to-end learning of neural networks for graphs and sequences 35 2 2019 2019 309 318.
[48] D. Zhang, H.-D. Chen, H. Zulfiqar, S.-S. Yuan, Q.-L. Huang, Z.-Y. Zhang, K.-J. Deng, W. Shoombuatong, iBLP: An XGBoost-based predictor for identifying bioluminescent proteins, Comput. Mathem. Methods Med. 2021 (2021) 1–15.
[49] F.Y. Dao, H. Lv, D. Zhang, Z.M. Zhang, L. Liu, H. Lin, DeepYY1: a deep learning approach to identify YY1-mediated chromatin loops, Brief. Bioinf. 22 (4) (2021).
[50] Z. Lv, F. Cui, Q. Zou, L. Zhang, L. Xu, Anticancer peptides prediction with deep representation learning features, Brief. Bioinf. (2021), https://doi.org/10.1093/bib/bbab1008.
[51] Z. Lv, J. Zhang, H. Ding, Q. Zou, RF-PseU: a random forest predictor for RNA pseudouridine sites, Front. Bioeng. Biotechnol. 8 (2020) 134.
[52] Z. Lv, P. Wang, Q. Zou, Q. Jiang, Identification of Sub-Golgi protein localization by use of deep representation learning features, Bioinformatics 36 (24) (2020) 5600–5609.
[53] H. Yang, Y. Luo, X. Ren, M. Wu, X. He, B. Peng, K. Deng, D. Yan, H. Tang, H. Lin, Risk Prediction of Diabetes: Big data mining with fusion of multifarious physical examination indicators, Inf. Fusion 75 (2021) 140–149.
[54] Q. Jin, Z. Meng, T.D. Pham, Q.i. Chen, L. Wei, R. Su, DUNet: A deformable network for retinal vessel segmentation, Knowledge-Based Syst. 178 (2019) 149–162.
[55] B. Manavalan, S. Basith, T.H. Shin, L. Wei, G. Lee, Meta-4mCpred: A sequence-based meta-predictor for accurate DNA 4mC site prediction using effective feature representation, Mol. Therapy-Nucleic Acids 16 (2019) 733–744.
[56] Manayalan B, Basith S, Shin TH, Wei L, Lee G: mAHTPred: a sequence-based meta-predictor for improving the prediction of anti-hypertensive peptides using effective feature representation. *Bioinformatics* 2019, **35**(16):2757–2765.
[57] L. Wei, J. Tang, Q. Zou, Local-DPP: An improved DNA-binding protein prediction method by exploring local evolutionary information, Inf. Sci. 384 (2017) 135–144.
[58] L. Wei, P. Xing, J. Zeng, J. Chen, R. Su, F. Guo, Improved prediction of protein-protein interactions using novel negative samples, features, and an ensemble classifier, Artif. Intelligence Med. 83 (2017) 67–74.
[59] L. Wei, C. Zhou, H. Chen, J. Song, R. Su, ACPred-FL: a sequence-based predictor using effective feature representation to improve the prediction of anti-cancer peptides, Bioinformatics 34 (23) (2018) 4007–4016.
[60] Q. Zou, P. Xing, L. Wei, B. Liu, Gene2vec: gene subsequence embedding for prediction of mammalian N-6-methyladenosine sites from mRNA, Rna 25 (2) (2019) 205–218.
[61] J. Khanal, D.Y. Lim, H. Tayara, K.T. Chong, i6mA-stack: A stacking ensemble-based computational prediction of DNA N6-methyladenine (6mA) sites in the Rosaceae genome, Genomics 113 (1) (2021) 582–592.