Mathematical Biosciences and Engineering

MBE, 21(1): 253–271.
DOI: 10.3934/mbe.2024012
Received: 12 October 2023
Revised: 30 November 2023
Accepted: 04 December 2023
Published: 12 December 2023

http://www.aimspress.com/journal/MBE

Research article

4mCPred-GSIMP: Predicting DNA N4-methylcytosine sites in the mouse genome with multi-Scale adaptive features extraction and fusion

Jianhua Jia*, Yu Deng*, Mengyue Yi and Yuhui Zhu

School of Information Engineering, Jingdezhen Ceramic University, Jingdezhen 333403, China

*** Correspondence:** Email: jiajianhua@jcu.edu.cn, 2220044006@stu.jcu.edu.cn.

Abstract: The epigenetic modification of DNA N4-methylcytosine (4mC) is vital for controlling DNA replication and expression. It is crucial to pinpoint 4mC's location to comprehend its role in physiological and pathological processes. However, accurate 4mC detection is difficult to achieve due to technical constraints. In this paper, we propose a deep learning-based approach 4mCPred-GSIMP for predicting 4mC sites in the mouse genome. The approach encodes DNA sequences using four feature encoding methods and combines multi-scale convolution and improved selective kernel convolution to adaptively extract and fuse features from different scales, thereby improving feature representation and optimization effect. In addition, we also use convolutional residual connections, global response normalization and pointwise convolution techniques to optimize the model. On the independent test dataset, 4mCPred-GSIMP shows high sensitivity, specificity, accuracy, Matthews correlation coefficient and area under the curve, which are 0.7812, 0.9312, 0.8562, 0.7207 and 0.9233, respectively. Various experiments demonstrate that 4mCPred-GSIMP outperforms existing prediction tools.

Keywords: DNA N4-methylcytosine; multi-scale convolution; selective kernel convolution; global response normalization

1. Introduction

DNA methylation is a process in which methyl groups are added to a DNA molecule. This process can alter the activity of a DNA segment without changing the sequence of the DNA [1,2]. The most commonly occurring form of methylation takes place at the 5-carbon atom of cytosine, which produces 5-methylcytosine (5mC) through the action of enzymes and substrates [3]. 5mC is the primary form of DNA methylation among eukaryotes and is generally present in CpG islands, which are specific

regions of DNA characterized by a high frequency of CpG dinucleotides [4]. CpG islands are frequently located in or near the promoter regions of genes within mammalian genomes. The presence of 5mC on these islands is associated with gene regulation and transcriptional activity. When methylation occurs on the promoter of a gene, it usually inhibits the transcription of the gene. The methylation level of CpG islands can regulate the binding of transcription factors, thereby changing the transcriptional activity of genes [5–7]. It is worth mentioning that some regions in the bacterial genome are different from the surrounding DNA sequence, called genomic islands. Genomic islands are clusters of genes introduced by horizontal gene transfer, which contain some functional genes that are different from the host. Compared with other parts of the genome, they have different G + C content. Genomic islands facilitate gene exchange and evolution among different species, leading to greater genome diversity and adaptability [8–10].

In addition to the common 5mC, there are several other types of DNA methylation, such as N6-methyladenine (6mA), N4-methylcytosine (4mC) and 5-hydroxymethylcytosine (5hmC). These forms of methylation have distinct functions and distributions in different organisms. 5hmC is an oxidized product of 5mC and is involved in active DNA demethylation and gene regulation. It facilitates the promotion of gene expression after DNA demethylation and serves as a marker to recruit proteins to specific DNA sites, altering gene expression and acting as an epigenetic mark [11]. 6mA is the most common form of DNA methylation in prokaryotes, such as bacteria. It is involved in various biological processes in prokaryotic genomes, such as DNA replication, transcription, repair, recombination and others. The distribution and functional implications of 6mA in prokaryotic genomes differ from those in eukaryotic genomes. It is usually not associated with CpG islands but with the coding regions of genes [12,13]. 4mC is another form of DNA methylation, which is naturally present in bacteria and is also related to eukaryotic DNA. Recent studies have found that 4mC can act as an epigenetic mark in eukaryotic genomes, and the underlying enzymatic mechanism has been characterized [14,15].

To identify DNA methylation, traditional experimental techniques like bisulfite sequencing [16] and single-molecule real-time sequencing (SMRT) [17] can detect the modified sites from the DNA signal directly or indirectly. However, these methods are not suitable for large-scale analysis because they are time-consuming and labor-intensive. Thus, developing computational methods is essential to gain insight into the mechanism and function of DNA methylation. Some machine learning-based models have shown progress in predicting methylation sites [18–22], but there is room for improvement. These models typically use artificially designed and selected DNA sequence features and traditional classification algorithms to generate predictions. However, these artificial features require significant domain knowledge and experience. However, the limited research on methylation makes it challenging to identify effective features with a reliable ability to predict sites. Furthermore, conventional classification algorithms fall short in capturing high-order features and semantic information, impeding the accuracy and reliability of predictions.

Deep learning is a cutting-edge technique that surpasses the constraints of conventional computational approaches and enhances the precision of models predicting methylation sites [23–36]. Furthermore, deep learning can combine feature fusion methods to further optimize the prediction effect. Several tools currently utilize deep learning and feature fusion methods to detect methylation sites, including iCpG-Pos [30], iPromoter-5mC [31], iRG-4mC [32] and m6A-NeuralTool [33]. These tools' feature fusion methods are primarily categorized into two groups. The first category is the fusion of various feature encoding methods like One-hot, electron-ion interaction pseudopotential (EIIP), nucleotide chemical property (NCP) and Nucleotide density (ND). These encoding methods extract

diverse information from DNA sequences, including composition, physicochemical properties, periodicity and distribution. Combining encoding methods increases feature diversity and dimensionality, ultimately enhancing feature expressive ability. For instance, the iPromoter-5mC accurately predicts 5mC sites within DNA promoter regions of the genome. The tool employs two feature encoding methods, namely One-hot and DPF, to extract local and global sequence order information from DNA sample sequences. These two features are then fused, and a deep neural network (DNN) is used to construct a prediction model. The second category involves the fusion of various network structures or algorithms, like convolutional neural network (CNN), Long Short-Term Memory (LSTM), DNN, or other machine learning methods. These network structures or algorithms can extract and integrate sequence features from various levels and perspectives, including local or global, spatial or temporal and linear or nonlinear aspects. The fusion of diverse network structures or algorithms can enhance the complexity and adaptability of the model, thereby improving its fitting ability. For instance, m6A-NeuralTool serves as a tool to detect m6A sites in a range of species. The tool employs a one-dimensional convolutional layer and a majority voting strategy, combined with an ensemble model of a fully connected layer, support vector machine and naive Bayes, to extract and integrate sequence features.

To our knowledge, there is currently no feature fusion tool designed specifically for the mouse genome. There are some deep learning-based predictors for predicting 4mC sites in the mouse genome [29,37–40]. 4mCPred-CNN [37] employs one-hot encoding and a CNN to extract features from DNA sequences. On the other hand, Mouse4mC-BGRU [38] and i4mC-GRU [40] utilize bidirectional gated recurrent units (GRU) and sequence embedding features to capture contextual information. These predictors can capture intricate nonlinear relationships, but they fail to fully make use of the various scales of information within DNA sequences, leading to limited model expressiveness. Recently, MultiScale-CNN-4mCPred [39] presented a computational method for predicting 4mC sites in the mouse genome using a multi-scale CNN and adaptive embedding. By employing different sizes of convolutional kernels, the method captures various scale sequence features, thus enhancing the flexibility and precision of feature representation. Nonetheless, this method has some limitations, including a fixed number and size of convolutional kernels which cannot be dynamically adjusted according to the data. Thus, it is essential to devise a new feature fusion tool capable of integrating various features to enhance the prediction accuracy of 4mC sites in the mouse genome.

To solve these problems, we propose a prediction method based on deep neural networks named 4mCPred-GSIMP. The method combines the two types of fusion methods, feature encoding fusion and network structure fusion, using One-hot, EIIP, NCP and ND four encoding methods to encode the input sequence, to obtain multiple types of features, and then using multi-scale convolution (MSC) [41] and improved selective kernel convolution (SKC) [42] to achieve adaptive extraction and multi-scale fusion of multiple types of sequence features while combining convolutional residual connection, global response normalization (GRN) [43] and pointwise convolution (PWC) [44] techniques to optimize the model. Compared with existing methods, 4mCPred-GSIMP captures DNA sequence features from multiple perspectives, rather than relying on the limitations of one or two encoding schemes, and the unique network structure achieves multi-scale adaptive feature extraction and fusion, than using fixed size and number of convolutional kernels, or only using a single network structure. 4mCPred-GSIMP has better prediction performance and provides a valuable reference for follow-up research.

2. Materials and methods

2.1. Benchmark dataset

To evaluate 4mCPred-GSIMP and compare it with other predictors, we used the benchmark and independent dataset adopted by i4mC-Mouse [20]. This dataset was originally constructed by 4mCpredEL [18] from the MethSMRT database [45], where the DNA sequence window was set to 41 base pairs (bp), with the central position being experimentally validated 4mC sites (positive samples) or unmethylated cytosines (negative samples). To avoid overestimating the prediction model, i4mC-Mouse used a more stringent CD-HIT (70%) [46] to filter the samples. After filtering, they were randomly divided into training and independent datasets at a ratio of 8:2, where the training dataset contained 746 positive (4mC) and 746 negative (non-4mC) samples, and the independent dataset contained 160 positive and 160 negative samples. Using these balanced datasets of positive and negative samples, we can eliminate the impact of class imbalance on model performance, that is, avoid the model bias towards predicting the more abundant class, thereby reducing the prediction accuracy and robustness. In this way, the model will perform more stable and reliably in the test set and the real environment, and also make the evaluation metrics such as accuracy, recall, etc. more meaningful.

2.2. Model construction

Figure 1 illustrates the three major components of 4mCPred-GSIMP: The feature encoding module, the Multi-scale Adaptive Feature Extraction and Fusion module (MSAFEF) and the prediction module. The feature encoding module converts DNA sequences into numerical feature matrices that serve as inputs for the next component. MSAFEF consists of three layers of sub-models that extract and fuse features from multiple scales and dimensions of the input matrix, enhancing their expressiveness and resolution ability. The prediction module uses the final features to perform binary classification, determining whether the DNA sequence contains 4mC sites or not.

2.2.1. Feature encoding module

We use a hybrid encoding scheme that combines One-hot, EIIP, NCP and ND encoding methods to represent DNA sequences. These four encoding mothods produce feature matrices with the same column dimension (41), which allows us to fuse them. As shown in Figure 1(A), for a DNA sequence fragment of length 41 bp, we can obtain feature matrices of sizes 4×41, 1×41, 3×41 and 1×41 using One-hot, EIIP, NCP and ND encoding methods, respectively. Then, we concatenate these four matrices to form a 9×41 feature matrix. Finally, we feed this 9×41 feature matrix into a bias-free linear layer that performs a linear transformation in the first dimension, converting the 9×41 feature matrix into an $N \times 41$ feature matrix. The purpose of the last step is to make the feature matrices generated by different hybrid encoding schemes have the same size for subsequent module processing. It is important to note that this operation requires each encoding method to use a numerical vector to represent the nucleotides at each position in the sequence. Consequently, the encoding can generate a feature matrix of size $L \times 41$, where L is any positive integer.

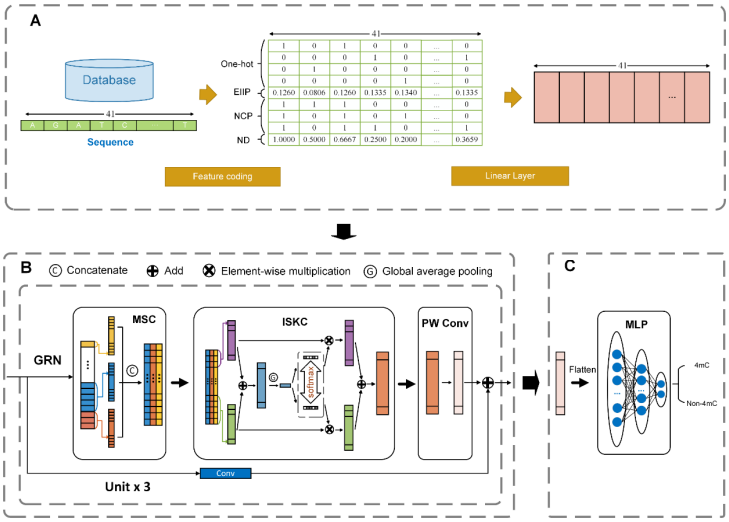

Figure 1. The architecture of 4mCPred-GSIMP. (A) Feature encoding module. The input sequence is encoded by a hybrid of four encoding s: One-hot, EIIP, NCP and ND, resulting in a 9 × 41 feature matrix. Then, a bias-free linear transformation is applied to adjust its dimension to N × 41. (B) Multi-scale adaptive feature extraction and fusion module. It is composed of three stacked multi-scale adaptive convolution units (MSACU). Each unit consists of global GRN, MSC, improved SKC, PWC and residual convolution. (C) Prediction module. The feature matrix is flattened into a feature vector, which is fed into a fully connected neural network to predict whether there is a 4mC site.

1) One-hot encoding

One-hot [36] encoding is considered a feasible encoding method due to its feasibility, efficiency and ability to ensure that each nucleotide letter is encoded independently. This encoding method encodes each nucleotide letter into a four-dimensional vector, where only one dimension is 1 and the rest are 0. For example, given a DNA sequence of length n bp, $S = s_1 s_2 \ldots s_n$, we can construct a function f: $S \rightarrow R^{(4\times n)}$, where $s_i \in \{A, C, G, T\}$. The specific formula is as follows:

$$f(S) = [f(s_1) \quad f(s_2) \quad \cdots \quad f(s_n)], \; f(s_i) = \begin{cases} (1,0,0,0)^T, & s_i = A \\ (0,1,0,0)^T, & s_i = C \\ (0,0,1,0)^T, & s_i = G \\ (0,0,0,1)^T, & s_i = T \end{cases} \tag{1}$$

This way, we get a 4 × n feature matrix, where each column corresponds to the one-hot encoding of a nucleotide letter.

2) NCP encoding

DNA sequences are composed of four nucleotides, which are adenine (A), cytosine (C), guanine (G) and thymine (T). They have different chemical properties, such as ring structure, hydrogen bond strength and chemical function. These properties can affect the interactions between nucleotides, thus affecting the structure and function of DNA. To utilize this information, we can use the NCP [21,47,48] encoding method to represent the chemical properties of each base with a three-dimensional vector, where each dimension uses 0 or 1 to distinguish the category of a certain property of the base. Table 1 shows the chemical properties and NCP encoding of the bases.

Table 1. Chemical properties and NCP encoding of nucleotides.

Nucleotide	Ring structure	Hydrogen bond strength	Chemical function	NCP
A	Purine	Weak	Amino	(1,1,1)
C	Pyrimidine	Strong	Amino	(0,0,1)
G	Purine	Strong	Keto	(1,0,0)
T	Pyrimidine	Weak	Keto	(0,1,0)

3) ND encoding

ND [49] encoding calculates each nucleotide as a scalar based on the cumulative frequency distribution of nucleotides up to that position in the DNA sequence. This encoding method can reflect the density changes and distribution patterns of nucleotides in the sequence, thereby improving the expression ability of features. The calculation formula for the ND value is as follows:

$$d_i = \frac{1}{i}\sum_{j=1}^{i} f(R_j), f(R_j) = \begin{cases} 1, & \text{if } R_j = R_i \\ 0, & \text{otherwise} \end{cases}, i = 1, \ldots, L \quad (2)$$

where L is the sequence length, R_i is the nucleotide at the i-th position, and $\mathrm{f}(\mathrm{R_j})$ is an indicator function, which is 1 when the nucleotide at the j-th position is equal to R_i, and 0 otherwise. For example, in Figure 1(A), for a DNA sequence of 41 bp, the fourth position has a "T" nucleotide, and there are four nucleotides from the beginning to this position, among which only one is "T", so the ND value at this position is 1/4 = 0.2500.

4) EIIP encoding

Similar to ND encoding, EIIP [49] encoding is also a method of representing each nucleotide in a sequence with a single value. EIIP encoding uses the EIIP values directly to represent the nucleotides in the DNA sequence. The EIIP values are calculated from the energy of delocalized electrons in the nucleotides, which are A: 0.1260, C: 0.1340, G: 0.0806 and T: 0.1335, respectively. Therefore, given a DNA sequence of length n bp, we can obtain an n-dimensional numerical vector to represent the sequence, where each element is an EIIP value of a nucleotide.

2.2.2. Multi-scale adaptive feature extraction and fusion module

As shown in Figure 1(B), the multi-scale adaptive feature extraction and fusion module (MSAFEF) consists of three layers of stacked MSACU. Each MSACU can extract local features, and by stacking three layers of units, the MSAFEF can fit the global features of the entire sequence. As mentioned earlier, a 41 bp DNA sequence generates an N × 41-dimensional feature matrix through the feature encoding module. The role of MSAFEF is to extract and capture high-order features and semantic information from the input feature matrix.

1) MSACU

MSACU is a submodule of MSAFEF, which can perform multi-level, multi-scale and multi-dimensional feature extraction and fusion on the input feature matrix and maintain the integrity and consistency of the features through residual connection and pointwise convolution.

As shown in Figure 1(B), for a $C \times L$ feature map $\mathcal{X}$, where C represents the number of channels, L represents the sequence length. First, we apply GRN $GRN(\cdot)$ to $\mathcal{X}$, obtaining $\widehat{\mathcal{X}}$. This technique can enhance the diversity and competitiveness of different channels in the feature map and does not change the size of the feature map.

After that, to achieve multi-scale feature extraction, we perform MSC $F_{msc}(\cdot)$. The idea of this operation is to pass the input feature map $\widehat{\mathcal{X}}$ through three one-dimensional convolution layers with different scales (1,3,5), respectively obtaining three feature sub-maps, and then concatenate them in the channel dimension to form a rich output feature map $\widehat{\mathcal{X}}_{msc}$:

$$\widehat{\mathcal{X}}_{msc} = F_{msc}(\widehat{\mathcal{X}}) = \sigma_h([f_1(\widehat{\mathcal{X}}), f_3(\widehat{\mathcal{X}}), f_5(\widehat{\mathcal{X}})]) \tag{3}$$

where f_1, f_3, f_5 are one-dimensional convolution operations with scales of 1, 3 and 5, respectively, $[\cdot]$ is the concatenation operation, and σ_h is the Hard-Swish activation function [50].

Improved SKC $F_{\text{iskc}}(\cdot)$ is used to further fuse features of different scales. This is an adaptive convolution method that can dynamically adjust the size and shape of the convolution kernel according to the local information of the input feature. Since the stride of the improved SKC is 2, it will halve the sequence length of the input feature map. When the sequence length L of the input feature map is not a multiple of 2, zero padding is also required at the end to ensure that the sequence length of the output feature map $\widehat{\mathcal{X}}_{iskc}$ is an integer. Where $L' = \lceil L/2 \rceil$, $\lceil \cdot \rceil$ represents rounding up.

The pointwise convolution [44] layer is a method used to fuse and reduce features in the channel dimension. It only uses 1 × 1 convolution kernels, which can reduce computation and parameter amounts. Through a pointwise convolution layer, channel dimension changes from $3C$ to $2C$, σ_r represents ReLU activation function:

$$\widehat{\mathcal{X}}_{pw} = F_{pw}(\widehat{\mathcal{X}}_{iskc}) = \sigma_r\left(f_1(\widehat{\mathcal{X}}_{iskc})\right) \tag{4}$$

To retain the original information of the input feature map $\mathcal{X}$, we perform residual connection [51] between it and $\widehat{\mathcal{X}}_{pw}$. This is a common technique that can prevent gradient disappearance and overfitting. To achieve residual connections, we first perform a convolution operation $f_c(\cdot)$ on $\mathcal{X}$ with a kernel size of 3 and a stride of 2. This can make its output feature map have the same dimension as $\widehat{\mathcal{X}}_{pw}$ so that they can be added. In this way, we complete an MSACU, which can extract and fuse features of different scales and shapes from the input feature map, enhancing model performance. After sorting out, MSACU's formula is as follows:

$$U_m(\mathcal{X}) = \sigma_r(f_c(\mathcal{X})) + F_{pw}\left(F_{iskc}\left(F_{msc}(GRN(\mathcal{X}))\right)\right) \tag{5}$$

2) GRN

A novel feature normalization technique called global response normalization (GRN) [43] seeks to enhance feature competition among channels in convolutional neural networks, which enhances the effectiveness and generalizability of the model. It enables contrast and selectivity between different channels by aggregating, normalizing and calibrating the feature maps on each channel with the global L2 norm. Specifically, given an input feature map $X \in R^{H\times W\times C}$, GRN layer first computes the L2 norm on each channel:

$$\mathcal{G}(X) = \{\| X_1 \|, \| X_2 \|, \ldots, \| X_C \|\} \in \mathcal{R}^{\mathcal{C}}, \mathcal{G}(X)_i = \| X_i \| \tag{6}$$

Then, it divides this value by the average norm over all channels, obtaining a relative importance score:

$$\mathcal{N}(|X_i|) \coloneqq |X_i| \in \mathcal{R} \to \frac{|X_i|}{\sum_{j=1,\ldots,C}|X_j|} \in \mathcal{R} \tag{7}$$

This score is used to modulate the original feature map's response:

$$X_i = X_i * \mathcal{N}(\mathcal{G}(X)_i) \in \mathcal{R}^{\mathcal{H}\times\mathcal{W}} \tag{8}$$

In addition, to facilitate optimization, two additional learnable parameters γ, β are added and initialized to zero. Also, a residual connection is added between the input and output of the GRN layer, resulting in the final GRN block:

$$X_i = \gamma * X_i * \mathcal{N}(\mathcal{G}(X)_i) + \beta + X_i \tag{9}$$

This setting allows the GRN layer to initially perform an identity mapping function and gradually adapt during training.

GRN has some differences and connections with other feature normalization methods. It is similar to Local Response Normalization [52], but GRN does not normalize the responses within a small window of neighboring neurons, but rather normalizes the responses over the entire layer. This can leverage global information to enhance channel-wise competition. Unlike Batch Normalization [53] or Layer Normalization [54], GRN does not perform standardization or scaling operations on each neuron, but rather performs importance evaluation and modulation operations on each channel. This can preserve the distribution and structure of the original feature map.

In order to make the GRN applicable to feature matrices encoded in nucleotide sequences, the input feature $X \in R^{H\times W\times C}$ is adapted to $X \in R^{C\times L}$, and the global feature aggregation, feature normalization and feature calibration are changed. This allows the aggregation and normalization operations to be performed on vectors of length L on each channel, instead of performing these operations on the 2D matrix of $H \times W$. This adaptation maintains the original design intent and function of the GRN layer, which is to improve the quality of the representation by enhancing feature diversity and competitiveness between channels, while adapting to the structure of the feature matrix encoded by the nucleotide sequences.

3) Improved SKC

Motivated by the fact that the receptive field size of human visual neurons can adapt dynamically, Li et al. [42] proposed selective kernel convolution (SKC), which can dynamically select convolution kernels according to the multi-scale information in the features. SKC consists of three steps: Split, Fuse and Select. In the Split phase, convolution kernels of different sizes convolve the input feature maps to generate multiple feature sub-maps. In the Fuse phase, these feature sub-maps are combined and aggregated to obtain a global and comprehensive representation of the weights. In the Select phase, feature sub-maps of different kernel sizes are aggregated based on the selection weights.

We improved the original SKC, and the structure diagram is shown in Figure 2. In the Split phase, we replaced dilated convolution with regular convolution with the same kernel size and used Hard-Swish [50] as the activation function. Although dilated convolution could reduce the model parameters and run time, it caused information loss due to the discontinuity of the convolution kernel and the reduced amount of information. In addition, in the original SKC, in the Fuse phase, the dimension of the feature with channel size after global average pooling was compressed into a compact feature descriptor by a simple fully connected layer. However, while dimensionality reduction could reduce model complexity, the direct correspondence between channel features and attention weights in the Select phase was destroyed by it. This approach of projecting channel features into a low-dimensional space and then mapping them back made the correspondence between channels and their weights indirect, which hurt the acquisition of attention weights, and the acquisition of dependencies was inefficient and unnecessary. Therefore, in the Fuse phase, we did not perform dimensionality reduction on the channel features after global average pooling to maintain a direct correspondence between channel features and attention weights in the selection step. It should be noted that, as with GRN, to

make SKC suitable for feature matrices encoded by nucleotide sequences, we replaced two-dimensional convolution with one-dimensional convolution (PyTorch). Furthermore, we removed all batch normalization operations, as in some cases adding batch normalization would destroy the distribution structure of the feature data, resulting in information loss and thus reducing the predictive performance of the model. In summary, with our improvements, SKC reduced information loss during feature extraction, and enhanced the inter-channel interactions and dependencies, improving feature richness and expressiveness.

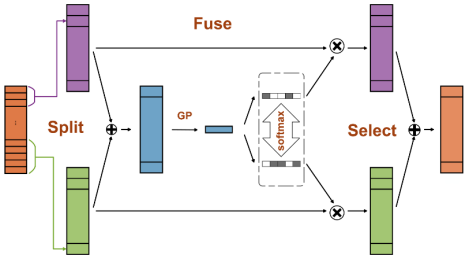

Figure 2. Construction of improved SKC.

2.2.3. Prediction module

Finally, in the prediction module, we used a fully connected neural network as a classifier to generate prediction results. Figure 1(C) shows the structure of this module. After extracting high-level features, we use nn.Flatten (PyTorch) to flatten the feature matrix into a vector, which is then input to the fully connected neural network, where the dropout rate of each layer is 0.5, the first two layers use a Hard-Swish activation function, and the final output layer uses the softmax function to calculate the prediction probability of 4mC sites.

2.3. Performance evaluation

To develop and train our model, we used Python 3.9.16 and torch 2.0.0 + cu118 as tools and evaluated and tested the performance of the model using 10-fold cross-validation. Our classifier was trained for 200 epochs with a batch size of 28 in each fold, fitting on the training set and tuning on the validation set. We adopted cross-entropy as the loss function, used Adam as the optimizer and set the learning rate to 8×10^{-5}. To avoid overfitting, we terminated the training process when the maximum Matthews correlation coefficient (MCC) value on the validation set did not improve for 40 consecutive epochs. To measure our model performance, we chose the following five evaluation metrics: Sensitivity (Sn), specificity (Sp), accuracy (Acc), MCC and area under the curve (AUC) [55]. These metrics reflect the model's performance on the classification problem, and their formulas are as follows:

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

where TP, FP, TN and FN denote the number of true positives, false positives, true negatives and false negatives, respectively. AUC is calculated by plotting Sn versus (1-Sp) for different threshold settings and computing the area under the receiver operating characteristic curve. The higher the metric values, the better the model performance.

3. Result and Discussion

3.1. Comparison of different features coding schemes

Feature engineering is one of the critical steps in building an adequate model, so it is essential to choose an appropriate encoding method to represent feature data. In this study, four encoding techniques are involved, namely, One-hot encoding, NCP encoding, EIIP encoding and ND encoding. We designed nine encoding schemes. Specifically, the first encoding scheme uses all four encoding techniques, the second to fifth encoding schemes remove one encoding technique each time and use the remaining three, and the sixth to ninth encoding schemes use each encoding technique separately. We feed the feature matrices generated by the nine encoding schemes into our 4mCPred-GSIMP network framework, conduct experiments on both training and independent test datasets and measure the model performance using various metrics, where we repeated ten-fold cross-validation ten times on the training dataset and take the average of the results. The experimental results are shown in Figure 3, where we use the four letters O, P, E and D to represent One-hot, NCP, EIIP and ND, respectively, for simplicity. The results show that the first encoding scheme performs best on most performance metrics, indicating that each encoding method has its contribution and that the combination of the four encoding techniques can achieve the best results. Therefore, we adopted the first encoding scheme as the final encoding method in this study.

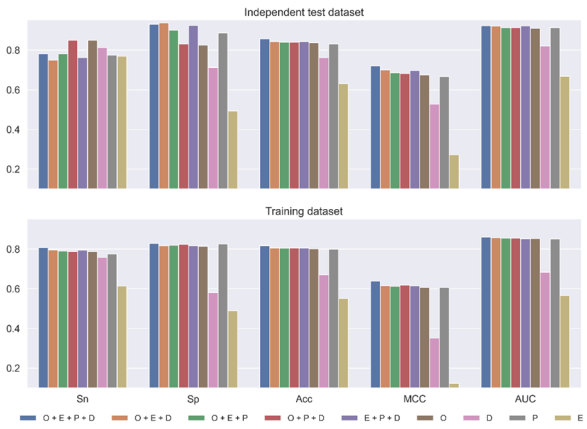

Figure 3. Performance comparison of different feature encoding schemes on training and independent test datasets.

3.2. Comparison with or without GRN model

To improve the prediction performance of the model, we introduce GRN in 4mCPred-GSIMP, which is a technique that can enhance the contrast and selectivity between feature map channels. We evaluate the effectiveness of GRN by using repeated ten-fold cross-validation ten times and independent test and analyze its impact on model prediction performance. Figure 4 shows the results

of the two validation methods. Compared with the model without GRN, the model with GRN on the training dataset increased Sn, Acc, MCC, and AUC by 2.67, 0.88, 1.47, 0.53%, respectively, while decreasing Sp by 0.93%. The model with GRN on the independent test dataset improved Sp, Acc, MCC and AUC by 4.37, 0.31, 1.28 and 0.21%, respectively, while Sn decreased (3.76%). Overall, the models with GRN achieve better prediction results in both cross-validation and independent test.

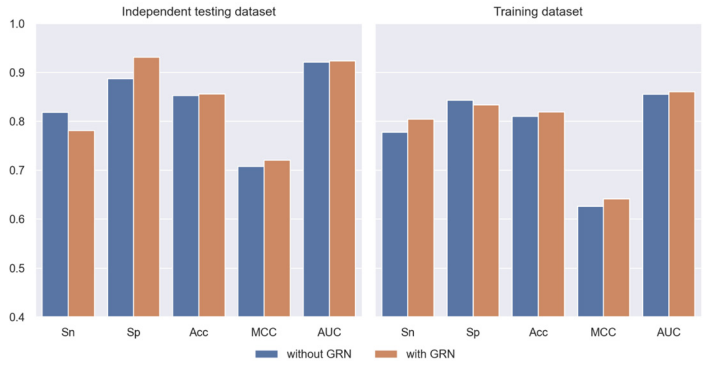

Figure 4. Performance comparison of models with and without GRN on training and independent test datasets.

3.3. Effectiveness of improved selective kernel convolution

To verify the effectiveness of our improvement, we conducted comparative experiments between the Improved SKC and the original SKC. We used two evaluation methods: repeated ten-fold cross-validation ten times and independent testing and compared the two schemes from the perspectives of Sn, Sp, Acc, MCC and AUC. The results are shown in Figure 5. On the training dataset, the Improved SKC improved Sp, Acc, MCC and AUC by 5.38, 1.92, 3.44 and 0.33%, respectively, compared to the Original SKC, while Sn decreased by 1.37%. On the independent test dataset, the Improved SKC improved Sn, Acc, MCC and AUC by 11.87, 4.68, 7.34 and 3.17%, respectively, compared to the Original SKC, while Sp slightly decreased by 2.51%. Based on the data and figure, we can see that the Improved SKC can predict the positive and negative samples more balanced and has higher Acc, MCC and AUC, demonstrating the effectiveness of our improvement.

Figure 5. Performance comparison of the improved SKC and the original SKC training and independent test datasets.

3.4. Comparison of 4mCPred-GSIMP with existing predictors

To demonstrate the effectiveness of 4mCPred-GSIMP, we compared it with several existing predictors, including 4mCpred-EL, i4mC-Mouse, Mouse4mC-BGRU and MultiScale-CNN-4mCPred. As mentioned earlier, different computational techniques are used for these predictors and will not be described here. We performed ten-fold cross-validation on the training dataset and performance evaluation on the independent test dataset, where cross-validation was repeated ten times. Tables 2 and 3 show the specific values of different evaluation metrics for the two validation methods. As can be seen from the table, 4mCPred-GSIMP shows some advantages in cross-validation, especially in Acc, which achieves the highest score of 0.8178, indicating that it can predict DNA 4mC sites accurately. Moreover, it also achieves high scores on Sn and Sp, which are 0.8080 and 0.8292, respectively, surpassing the threshold of 0.8. In the independent test, 4mCPred-GSIMP performed best in all metrics except Sn. Compared with the current optimal predictor MultiScale-CNN-4mCPred, 4mCPred-GSIMP has 9.37, 0.93 and 1.55% higher Sp, Acc and MCC, respectively. In addition, we also compared 4mCPred-GSIMP with 4mCPred-CNN, which is implemented on a user-friendly web server: http://nsclbio.jbnu.ac.kr/tools/4mCPred-CNN/. We uploaded the independent test dataset to the web server of 4mCPred-CNN for prediction and counted the prediction performance at different thresholds, and also listed the performance of 4mCPred-GSIMP at different thresholds, as shown in Table 4. It can be seen that 4mCPred-GSIMP is better than 4mCPred-CNN in terms of Acc and MCC values. Moreover, Figure 6 shows the ROC curve of 4mCPred-GSIMP on the independent test dataset, with an AUC value of 0.9233. These results confirm that 4mCPred-GSIMP is an effective and advanced tool for predicting DNA 4mC sites.

Table 2. The performance over the 10-fold cross-validation.

Predictor	Sn	Sp	Acc	MCC
4mCpred-EL	0.8040	0.7870	0.7950	0.5910
i4mC-Mouse	0.6831	**0.9020**	0.7930	**0.6510**
Mouse4mC-BGRU	0.7940	0.8400	0.8100	0.6200
MultiScale-CNN-4mCPred	0.8008	0.8294	0.8166	0.6335
4mCPred-GSIMP	**0.8080**	0.8292	**0.8178**	0.6389

Table 3. The performance over the independent test.

Predictor	Sn	Sp	Acc	MCC
4mCpred-EL	0.7572	0.8251	0.7910	0.5840
i4mC-Mouse	0.8071	0.8252	0.8161	0.6330
Mouse4mC-BGRU	0.8000	0.8500	0.8250	0.6510
MultiScale-CNN-4mCPred	**0.8563**	0.8375	0.8469	0.6939
4mCPred-GSIMP	0.7812	**0.9312**	**0.8562**	**0.7207**

Table 4. Comparison with 4mCPred-CNN over the independent test for various thresholds.

Predictor	4mCPred-CNN				4mCPred-GSIMP			
	Sn	Sp	Acc	MCC	Sn	Sp	Acc	MCC
0.1	0.9250	0.3438	0.6344	0.3302	0.9875	0.3125	0.6500	0.4066
0.2	0.8500	0.6063	0.7281	0.4704	0.9563	0.6187	0.7875	0.6108
0.3	0.7875	0.7375	0.7625	0.5257	0.8813	0.7688	0.8250	0.6542
0.4	0.6813	0.8000	0.7406	0.4847	0.8313	0.8625	0.8469	0.6941
0.5	0.6125	0.8625	0.7375	0.4906	0.7812	0.9312	0.8562	0.7207
0.6	0.4938	0.9125	0.7031	0.4474	0.6625	0.9563	0.8094	0.6473
0.7	0.3813	0.9438	0.6625	0.3931	0.5188	0.9688	0.7437	0.5459
0.8	0.2688	0.9750	0.6219	0.3443	0.4125	0.9937	0.7031	0.4992
0.9	0.1563	0.9875	0.5719	0.2586	0.2875	1.0000	0.6438	0.4097

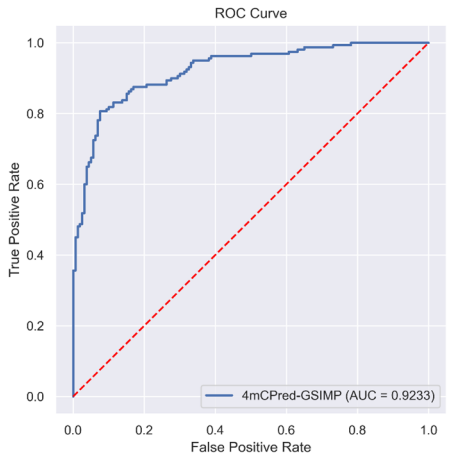

Figure 6. ROC curves for 4mCPred-GSIMP on independent test dataset.

3.5. Generalization ability of 4mCPred-GSIMP

To assess the predictive accuracy of 4mCPred-GSIMP for different DNA methylation types and species, we utilized 17 datasets sourced from the web application of Lv et al. [56]. The datasets encompass three methylation types across various species: 4mC, 6mA and 5hmC. Specifically, the species of 4mC include *C. equisetifolia* (732), *F. vesca* (31579), *S. cerevisiae* (3958) and *Tolypocladium* (30654); the species of 6mA include *A. thaliana* (63746), *C. elegans* (15922), *C. equisetifolia* (12132), *D. melanogaster* (22382), *F. vesca* (6204), *H. sapiens* (36670), *R. chinensis* (1200), *S. cerevisiae* (7572), *T. thermophile* (215200), *Tolypocladium* (6759) and *Xoc BLS256* (34430); the species of 5hmC include *H. sapiens* (4688) and *M. musculus* (7358). The numbers in parentheses indicate the total number of samples in each species' corresponding dataset.

In these datasets, the sample number of the training dataset and the independent test dataset for each dataset is 1:1, and the number of positive and negative samples is also 1:1. Each sample of each dataset is a 41 bp long DNA fragment, and the target site is located at the center position. We trained 4mCPred-GSIMP using the training dataset and evaluated its performance using the independent test dataset. To further evaluate its performance, we compared it with two other transformer-based [57] methods, iDNA-ABF [27] and MuLan-Methyl [28]. Both of these methods use transfer learning and

adversarial training to improve generalization ability and accuracy, but MuLan-Methyl uses the average probability of five language models as the final result, and combines species classification information as an additional feature.

Figure 7 illustrates the Acc and AUC of three methods across 17 datasets. The prediction performance of 4mCPred-GSIMP varies based on different methylation types. Regarding 4mC, 4mCPred-GSIMP yields higher Acc and AUC than iDNA-ABF and MuLan-Methyl on most datasets. For 6mA, 4mCPred-GSIMP also exhibits the optimal or nearly optimal level in various datasets except for *C. equisetifolia*, *R. chinensis* and *Tolypocladium*. However, the prediction performance of 4mCPred-GSIMP for 5hmC still falls short when compared to iDNA-ABF and MuLan-Methyl. At the species level, 4mCPred-GSIMP holds greater predictive advantages for plants and fungi over animals, and only in three animal datasets (5hmC_M.musculus, 5hmC_H.sapiens, 6mA_H.sapiens), both AUC and Acc are lower than the other two methods.

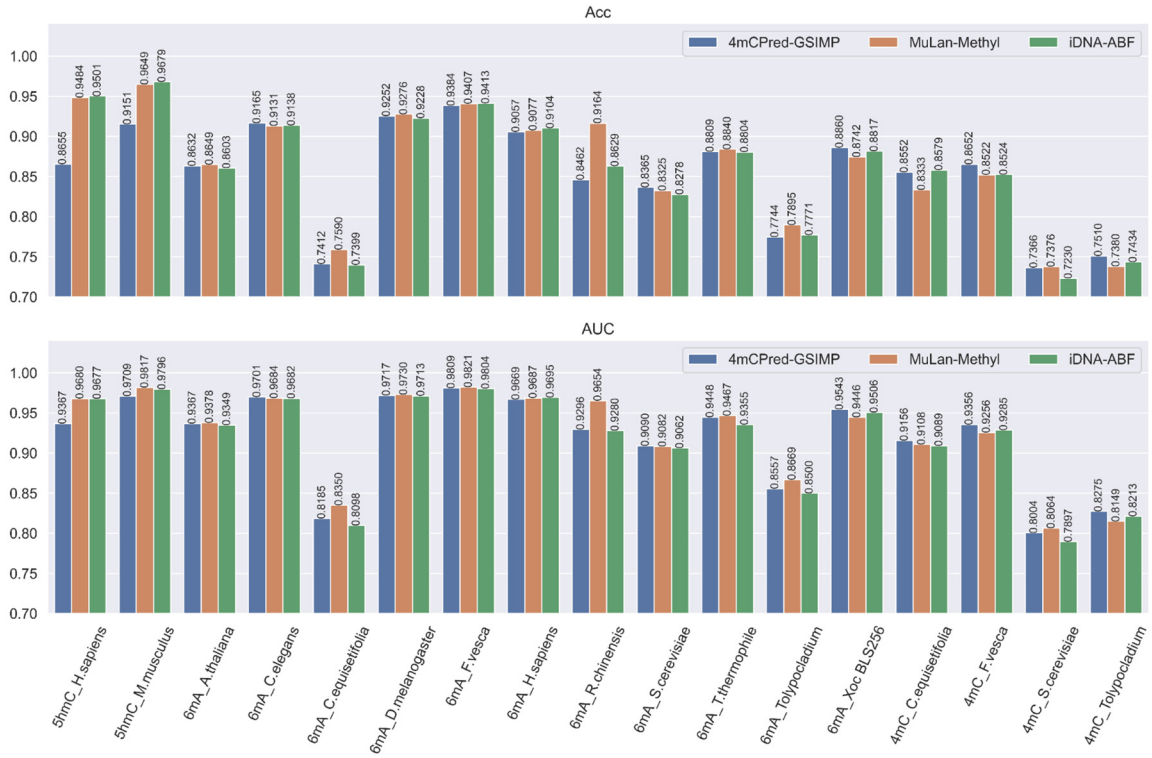

Figure 7. Comparison with iDNA-ABF and MuLan-Methyl on 17 datasets.

4. Conclusions

In this paper, we address the problem of predicting DNA 4mC sites in mouse genomes and proposes a new method based on deep learning, named 4mCPred-GSIMP, which can effectively improve prediction accuracy. The method uses four feature encoding methods, namely One-hot, EIIP, NCP and ND, to obtain various types of sequence features, and then adopts a combination of MSC and improved SKC, to achieve adaptive feature extraction and fusion from different scales, thereby enhancing the feature representation and optimization capabilities. In addition, the method also introduces convolutional residual connections, GRN and pointwise convolution techniques, to further

optimize the model structure. The experimental results on the mouse genome dataset show that 4mCPred-GSIMP outperforms the existing methods in terms of prediction performance. To verify the generalization ability of 4mCPred-GSIMP on different species and different DNA methylation types of modification sites, we also tested it on 17 datasets involving multiple species and three methylation types (4mC, 6mA and 5hmC). The results show that 4mCPred-GSIMP has good generalization performance, and can achieve a high level of prediction on different species and different methylation types of sites. Compared with existing prediction tools, 4mCPred-GSIMP can capture the features of DNA sequences from multiple perspectives, rather than relying solely on the limitations of one or two encoding schemes, and its unique network structure can achieve multi-scale adaptive feature extraction and fusion, rather than using fixed size and number of convolutional kernels, or only using a single network structure. The limitation of 4mCPred-GSIMP is that due to the over-parameterization of the deep learning network, it is prone to overfitting on the datasets with fewer samples, which limits the generalization performance on the independent test set, and it also lacks sufficient interpretability for the prediction results. In our future work, we plan to improve our model, explore the combination of transfer learning or meta-learning techniques, optimize the model performance on small samples and enhance its interpretability.

Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Data availability

The dataset and source code used in this study can be easily derived from https://github.com/DellCode233/4mCPred-GSIMP.

Acknowledgements

This work was sponsored in part by the National Science Foundation of China (Nos. 61761023, 62162032, and 31760315), the Natural Science Foundation of Jiangxi Province, China (Nos. 20202BABL202004 and 20202BAB202007), the Scientific Research Plan of the Department of Education of Jiangxi Province, China (GJJ190695 and GJJ2202814). These funders had no role in the study design, data collection and analysis, decision to publish or preparation of manuscript.

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. L. Zhao, J. Song, Y. Liu, C. Song, C. Yi, Mapping the epigenetic modifications of DNA and RNA, *Protein Cell*, **11** (2020), 792–808. https://doi.org/10.1007/s13238-020-00733-7
2. L. D. Moore, T. Le, G. Fan, DNA methylation and its basic function, *Neuropsychopharmacology*, **38** (2013), 23–38. https://doi.org/10.1038/npp.2012.112

3. N. Zhang, C. Lin, X. Huang, A. Kolbanovskiy, B. E. Hingerty, S. Amin, et al., Methylation of cytosine at C5 in a CpG sequence context causes a conformational switch of a benzo[a]pyrene diol epoxide-N^2-guanine adduct in DNA from a minor groove alignment to intercalation with base displacement, *J. Mol. Biol.*, **346** (2005), 951–965. https://doi.org/10.1016/j.jmb.2004.12.027
4. A. Breiling, F. Lyko, Epigenetic regulatory functions of DNA modifications: 5-methylcytosine and beyond, *Epigenet. Chromatin*, **8** (2015), 24. https://doi.org/10.1186/s13072-015-0016-6
5. A. Jeltsch, R. Z. Jurkowska, New concepts in DNA methylation, *Trends Biochem. Sci.*, **39** (2014), 310–318. https://doi.org/10.1016/j.tibs.2014.05.002
6. D. Schübeler, Function and information content of DNA methylation, *Nature*, **517** (2015), 321–326. https://doi.org/10.1038/nature14192
7. N. P. Blackledge, R. Klose, CpG island chromatin, *Epigenetics*, **6** (2011), 147–152. https://doi.org/10.4161/epi.6.2.13640
8. F. J. Clasen, R. E. Pierneef, B. Slippers, O. Reva, EuGI: a novel resource for studying genomic islands to facilitate horizontal gene transfer detection in eukaryotes, *BMC Genomics*, **19** (2018). https://doi.org/10.1186/s12864-018-4724-8
9. X. Guo, Y. Guo, H. Chen, X. Liu, P. He, W. Li, et al., Systematic comparison of genome information processing and boundary recognition tools used for genomic island detection, *Comput. Biol. Med.*, **166** (2023), 107550. https://doi.org/10.1016/j.compbiomed.2023.107550
10. Q. Dai, C. Bao, Y. Hai, S. Ma, T. Zhou, C. Wang, et al., MTGIpick allows robust identification of genomic islands from a single genome, *Briefings Bioinf.*, **19** (2018), 361–373. https://doi.org/10.1093/bib/bbw118
11. A. Chialastri, S. Sarkar, E. E. Schauer, S. Lamba, S. S. Dey, Combinatorial quantification of 5mC and 5hmC at individual CpG dyads and the transcriptome in single cells reveals modulators of DNA methylation maintenance fidelity, preprint.
12. G. Luo, M. A. Blanco, E. L. Greer, C. He, Y. Shi, DNA N^6-methyladenine: a new epigenetic mark in eukaryotes?, *Nat. Rev. Mol. Cell Biol.*, **16** (2015), 705–710. https://doi.org/10.1038/nrm4076
13. J. Beaulaurier, E. E. Schadt, G. Fang, Deciphering bacterial epigenomes using modern sequencing technologies, *Nat. Rev. Genet.*, **20** (2019), 157–172. https://doi.org/10.1038/s41576-018-0081-3
14. M. Ehrlich, G. G. Wilson, K. C. Kuo, C. W. Gehrke, N4-methylcytosine as a minor base in bacterial DNA, *Journal of Bacteriology*, **169** (1987), 939–943. https://doi.org/10.1128/jb.169.3.939-943.1987
15. F. Rodriguez, I. A. Yushenova, D. DiCorpo, I. R. Arkhipova, Bacterial N4-methylcytosine as an epigenetic mark in eukaryotic DNA, *Nat. Commun.*, **13** (2022), 1072. https://doi.org/10.1038/s41467-022-28471-w
16. M. Yu, L. Ji, D. A. Neumann, D. Chung, J. Groom, J. Westpheling, et al., Base-resolution detection of N 4-methylcytosine in genomic DNA using 4mC-Tet-assisted-bisulfite- sequencing, *Nucleic Acids Res.*, **43** (2015), 148. https://doi.org/10.1093/nar/gkv738
17. S. Ardui, A. Ameur, J. R. Vermeesch, M. S. Hestand, Single molecule real-time (SMRT) sequencing comes of age: applications and utilities for medical diagnostics, *Nucleic Acids Res.*, **46** (2018), 2159–2168. https://doi.org/10.1093/nar/gky066
18. B. Manavalan, S. Basith, T. H. Shin, D. Y. Lee, L. Wei, G. Lee, 4mCpred-EL: An ensemble learning framework for identification of DNA N4-methylcytosine sites in the mouse genome, *Cells*, **8** (2019), 1332. https://doi.org/10.3390/cells8111332

19. W. He, C. Jia, Q. Zou, 4mCPred: machine learning methods for DNA N^4-methylcytosine sites prediction, *Bioinformatics*, **35** (2019), 593–601. https://doi.org/10.1093/bioinformatics/bty668
20. M. M. Hasan, B. Manavalan, W. Shoombuatong, M. S. Khatun, H. Kurata, i4mC-Mouse: Improved identification of DNA N4-methylcytosine sites in the mouse genome using multiple encoding schemes, *Comput. Struct. Biotechnol. J.*, **18** (2020), 906–912. https://doi.org/10.1016/j.csbj.2020.04.001
21. W. Chen, H. Yang, P. Feng, H. Ding, H. Lin, iDNA4mC: identifying DNA N^4-methylcytosine sites based on nucleotide chemical properties, *Bioinformatics*, **33** (2017), 3518–3523. https://doi.org/10.1093/bioinformatics/btx479
22. 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. Ther.-Nucleic Acids*, **16** (2019), 733–744. https://doi.org/10.1016/j.omtn.2019.04.019
23. H. Xu, P. Jia, Z. Zhao, Deep4mC: systematic assessment and computational prediction for DNA N4-methylcytosine sites by deep learning, *Briefings Bioinf.*, **22** (2021). https://doi.org/10.1093/bib/bbaa099
24. Q. Liu, J. Chen, Y. Wang, S. Li, C. Jia, J. Song, et al., DeepTorrent: a deep learning-based approach for predicting DNA N4-methylcytosine sites, *Briefings Bioinf.*, **22** (2021). https://doi.org/10.1093/bib/bbaa124
25. X. Yu, J. Ren, Y. Cui, R. Zeng, H. Long, C. Ma, DRSN4mCPred: accurately predicting sites of DNA N4-methylcytosine using deep residual shrinkage network for diagnosis and treatment of gastrointestinal cancer in the precision medicine era, *Front. Med.*, **10** (2023). https://doi.org/10.3389/fmed.2023.1187430
26. Y. Yu, W. He, J. Jin, G. Xiao, L. Cui, R. Zeng, et al., iDNA-ABT: advanced deep learning model for detecting DNA methylation with adaptive features and transductive information maximization, *Bioinformatics*, **37** (2021), 4603–4610. https://doi.org/10.1093/bioinformatics/btab677
27. J. Jin, Y. Yu, R. Wang, X. Zeng, C. Pang, Y. Jiang, et al., iDNA-ABF: multi-scale deep biological language learning model for the interpretable prediction of DNA methylations, *Genome Biol.*, **23** (2022). https://doi.org/10.1186/s13059-022-02780-1
28. W. Zeng, A. Gautam, D. H. Huson, MuLan-Methyl-multiple transformer-based language models for accurate DNA methylation prediction, *GigaScience*, **12** (2023). https://doi.org/10.1093/gigascience/giad054
29. M. U. Rehman, H. Tayara, K. T. Chong, DCNN-4mC: Densely connected neural network based N4-methylcytosine site prediction in multiple species, *Comput. Struct. Biotechnol. J.*, **19** (2021), 6009–6019. https://doi.org/10.1016/j.csbj.2021.10.034
30. S. Park, M. U. Rehman, F. Ullah, H. Tayara, K. T. Chong, I. Birol, iCpG-Pos: an accurate computational approach for identification of CpG sites using positional features on single-cell whole genome sequence data, *Bioinformatics*, **39** (2023). https://doi.org/10.1093/bioinformatics/btad474
31. L. Zhang, X. Xiao, Z. Xu, iPromoter-5mC: A novel fusion decision predictor for the identification of 5-Methylcytosine sites in genome-Wide DNA promoters, *Front. Cell Dev. Biol.*, **8** (2020). https://doi.org/10.3389/fcell.2020.00614
32. D. Y. Lim, M. U. Rehman, K. T. Chong, iRG-4mC: Neural network based tool for identification of DNA 4mC sites in rosaceae genome, *Symmetry*, **13** (2021), 899. https://doi.org/10.3390/sym13050899

33. M. U. Rehman, K. J. Hong, H. Tayara, K. T. Chong, m6A-NeuralTool: Convolution neural tool for RNA N6-Methyladenosine site identification in different species, *IEEE Access*, **9** (2021), 17779–17786. https://doi.org/10.1109/access.2021.3054361
34. Q. H. Nguyen, H. V. Tran, B. P. Nguyen, T. T. T. Do, Identifying transcription factors that prefer binding to methylated DNA using reduced G-Gap dipeptide composition, *ACS Omega*, **7** (2022), 32322–32330. https://doi.org/10.1021/acsomega.2c03696
35. Z. Li, H. Jiang, L. Kong, Y. Chen, K. Lang, X. Fan, et al., Deep6mA: A deep learning framework for exploring similar patterns in DNA N6-methyladenine sites across different species, *PLOS Comput. Biol.*, **17** (2021), 1008767. https://doi.org/10.1371/journal.pcbi.1008767
36. X. Cheng, J. Wang, Q. Li, T. Liu, BiLSTM-5mC: A bidirectional long short-term memory-based approach for predicting 5-Methylcytosine sites in genome-wide DNA promoters, *Molecules*, **26** (2021), 7414 . https://doi.org/10.3390/molecules26247414
37. Z. Abbas, H. Tayara, K. T. Chong, 4mCPred-CNN-Prediction of DNA N4-Methylcytosine in the mouse genome using a convolutional neural network, *Genes*, **12** (2021), 296. https://doi.org/10.3390/genes12020296
38. J. Jin, Y. Yu, L. Wei, Mouse4mC-BGRU: Deep learning for predicting DNA N4-methylcytosine sites in mouse genome, *Methods*, **204** (2022), 258–262. https://doi.org/10.1016/j.ymeth.2022.01.009
39. P. Zheng, G. Zhang, Y. Liu, G. Huang, MultiScale-CNN-4mCPred: a multi-scale CNN and adaptive embedding-based method for mouse genome DNA N4-methylcytosine prediction, *BMC Bioinformatics*, **24** (2023). https://doi.org/10.1186/s12859-023-05135-0
40. T. Nguyen-Vo, Q. H. Trinh, L. Nguyen, P. Nguyen-Hoang, S. Rahardja, B. P. Nguyen, i4mC-GRU: Identifying DNA N4-Methylcytosine sites in mouse genomes using bidirectional gated recurrent unit and sequence-embedded features, *Comput. Struct. Biotechnol. J.*, **21** (2023), 3045–3053. https://doi.org/10.1016/j.csbj.2023.05.014
41. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., Going deeper with convolutions, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* IEEE, (2015), 1–9. https://doi.org/10.1109/CVPR.2015.7298594
42. X. Li, W. Wang, X. Hu, J. Yang, Selective kernel networks, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 510–519. https://doi.org/10.1109/CVPR.2019.00060
43. S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, et al., ConvNeXt V2: Co-Designing and scaling ConvNets with masked autoencoders, in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2023), 16133–16142. https://doi.org/10.1109/CVPR52729.2023.01548
44. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al., MobileNets: efficient convolutional neural networks for mobile vision applications, preprint, arXiv:1704.04861.
45. P. Ye, Y. Luan, K. Chen, Y. Liu, C. Xiao, Z. Xie, MethSMRT: an integrative database for DNA N6-methyladenine and N4-methylcytosine generated by single-molecular real-time sequencing, *Nucleic Acids Res.*, **45** (2017), 85–89. https://doi.org/10.1093/nar/gkw950
46. L. Fu, B. Niu, Z. Zhu, S. Wu, W. Li, CD-HIT: accelerated for clustering the next-generation sequencing data, *Bioinformatics*, **28** (2012), 3150–3152. https://doi.org/10.1093/bioinformatics/bts565

47. W. Chen, H. Tang, J. Ye, H. Lin, K. Chou, iRNA-PseU: Identifying RNA pseudouridine sites, *Mol. Ther.-Nucleic Acids*, **5** (2016), 332.
48. T. Nguyen-Vo, Q. H. Nguyen, T. T. T. Do, T. Nguyen, S. Rahardja, B. P. Nguyen, iPseU-NCP: Identifying RNA pseudouridine sites using random forest and NCP-encoded features, *BMC Genomics*, **20** (2019). https://doi.org/10.1186/s12864-019-6357-y
49. A. S. Nair, S. P. Sreenadhan, A coding measure scheme employing electron-ion interaction pseudopotential (EIIP), *Bioinformation*, **1** (2006), 197–202.
50. R. Avenash, P. Viswanath, Semantic segmentation of satellite images using a modified CNN with hard-swish activation function, in *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, (2019), 413–420.
51. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 770–778. https://doi.org/10.1109/CVPR.2016.90
52. A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Commun. ACM*, **60** (2017), 84–90. https://doi.org/10.1145/3065386
53. S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, preprint, arXiv:1502.03167.
54. J. L. Ba, J. R. Kiros, G. E. Hinton, Layer normalization, preprint, arXiv:1607.06450.
55. M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, *Inf. Process. Manage.*, **45** (2009), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002
56. H. Lv, F. Dao, D. Zhang, Z. Guan, H. Yang, W. Su, et al., iDNA-MS: An integrated computational tool for detecting DNA modification sites in multiple genomes, *iScience*, **23** (2020), 100991. https://doi.org/10.1016/j.isci.2020.100991
57. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., Attention is all you need, preprint, arXiv:1706.03762v5.

AIMS Press