OXFORD

Briefings in Bioinformatics, 2025, **26(1)**, bbae647

https://doi.org/10.1093/bib/bbae647

Problem Solving Protocol

Multi-kernel feature extraction with dynamic fusion and downsampled residual feature embedding for predicting rice RNA N^6-methyladenine sites

Me Za L [1], Z a -L S [2,*], Z a Ze [3], K -Ma La [4]

[1]School of Computer Science and Technology, Anhui University, Hefei 230601, China
[2]School of Electrical Engineering and Automation, Anhui University, Hefei 230601, China
[3]School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China
[4]Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, China

*Corresponding author. School of Electrical Engineering and Automation, Anhui University, 111 Jiulong Road, Economic and Technological Development District, Hefei 230601, China. E-mail: zhlsun2006@126.com

Abstract

RNA N6- e ade e (6A) ac ca e e c d fca c e ea ed ce , de e e, a d e e e. 6A acc a e de f ca , d ec e ea ed ec ce b eed a d e e, f da e a e ea e e e a a d ec a ec a . Faced ce 6A a ab e- e e ce, e de, e a e add a d abe e c d a ada b a e a - e added e e ce f ed c . A ca e a c e e e ce f a , e a e f a a d add , ed c fea e e ac acc a S a e e, e ce- ec f c 6A ed c e d a e a a ea a e. T add e e e, e de e a e e d- -e d dee ea f a e , MFD 6AR ce, f ed c ce 6A e . I a c a, a e a e a e e, e c c a - e e fea e f d e e e e a f a a - e added e e ce - e fea e e ac f c a d effec e a fe f a ba ca a c f f c . C c e c de e c e a d c a a eff c e f -d e a fea e ca ed a d add , e de a d a e d a fea e e bedd d e e fea e ace c e a d ac e e acc a e fea e e e a d eff c e c a a e f a ce. E e e a MFD 6AR ce e f c a e d c - a da , a e- a d c - ec e de e de , e e , de a d b e a d e e a a . T e a ca a e 6A d ca e e MFD 6AR ce' ca ab F e e a a e a c b d ffe e e e fea e , f c ba c a e- ca a a , a d e ea ab e d a a d e d a c ec ca e fea e e e e a a d e ac , e e effec e f a a fe , a d f ca e a ce de e f a ce.

Keywords: RNA N6- e ade e; - e e fea e; ba ca a c f ; d a e d a e bedd ; ce e e

Introduction

There are over 200 post-transcriptional epigenetic modifications of RNA in eukaryotes, such as N^6-methyladenosine (m^6A), N^1-methyladenosine (m^1A), 5-methylcytidine (m^5C), 1-methylguanosine (m^1G), pseudouridine (ψ) [1]. The most prevalent internal modification is m^6A, which occurs widely in mRNA, miRNA, long non-coding RNA, tRNA, and rRNA [2]. Methyltransferase complexes (writers), demethylases (erasers), and m^6A-binding proteins (readers) are the primary components of the m^6A modification system. Writers and erasers add and remove the methyl group ($–CH_3$) to the amino group ($–NH_2$) at the sixth position of adenosines in the RNA, respectively. Readers recognize the m^6A site and play specific regulatory roles [3]. In *Oryza sativa* (rice), m^6A is involved in growth, development [4–6], and response to biotic [7–10] and abiotic stresses [11–14]. For example, m^6A regulates the early degeneration of rice microspores at the vacuolar pollen stage [4]. Dynamic regulation of m^6A occurs during rice-plant virus interactions [7]. Cheng *et al.* found that rice m^6A may be associated with cadmium stress-induced aberrant root development [11].

MeRIP-seq (m^6A-seq) allows the detection of m^6A sites in plants but has a resolution of 100–200 nucleotides, which is too coarse for precise m^6A editing detection [15, 16]. Several improvements with single-base resolution have been developed, including PA-m^6A-seq [17], miCLIP [18], and m^6A-CLIP-seq [19]. However, these techniques are limited by the small sample sizes of high-RNA-metabolism tissues, making biological replication and accurate detection difficult [19]. The DART-seq method [20] requires only 10 ng of RNA, much less than MeRIP-seq. Advances in third-generation sequencing technologies, such as single-molecule real-time sequencing [21] and nanopore sequencing [22], have also enabled better detection of m^6A sites. In 2020, Parker *et al.* successfully used nanopore direct RNA sequencing to map m^6A sites in *Arabidopsis* [23]. This technique is well-suited for small

Received: A 21, 2024. **Revised:** N e be 15, 2024. **Accepted:** N e be 30, 2024

samples and can significantly accelerate m^6A research in plants, aiding the mapping of single-base resolution modifications and editing.

Although bio-experimental m^6A site detection technology is continuously being improved and refined, it still requires substantial human, material, and financial resources. Therefore, there is an urgent need to develop corresponding computational methods. Currently, some m^6A site prediction methods have been proposed for many species, especially *Homo sapiens* [24–28]. In plant research, several computational methods have also been developed to predict m^6A sites across different species. For instance, SMEP [29] is a method designed for rice and *Zea mays* (maize), while m6A-Maize [30] focuses exclusively on maize. PEA-m6A [31] has been applied to various economically important plants, including rice, maize, and *Triticum aestivum L.* (wheat), showing its potential applicability across multiple crops. For *Arabidopsis*, models like RFAthM6A [32] and M6AMRFS [33] have also been proposed.

Despite rice's economic importance, few methods exist for predicting m^6A sites in rice. In 2021, Wang and colleagues introduced SMEP [29], the first computational method for rice m^6A site prediction, after collecting and processing the first rice m^6A dataset. SMEP used padding and label encoding to handle variable-length m^6A sequences and applied convolutional layers to extract high-level features, followed by a multilayer perceptron for final prediction. While SMEP laid the foundation for rice m^6A site prediction, its use of max-length padding leads to sparse features, affecting extraction, and the multiple convolutional layers focus mainly on local information, missing broader context. Recently, Song *et al.* proposed PEA-m6A [31], an ensemble learning method based on gradient-boosted decision trees, which integrates statistical and deep learning features to improve feature representation and achieve strong performance. However, PEA-m6A is not an end-to-end model and struggles with unstructured data. Additionally, simply stacking features from different sources has limitations for enhancing feature representation.

Considering the advantage of max-length padded sequences that retain complete information, to exploit the information they encode and overcome existing limitations fully, we develop an end-to-end **Rice m^6A** site prediction learning framework called **MFDm^{6}ARice**. This framework combines **M**ulti-kernel feature extraction, dynamic **F**usion of global and local features, and **D**ownsampled residual embedding technology. Since variable-length sequences contain valuable information, we use label encoding and padding to construct max-length padded sequences to facilitate feature learning. However, this padding can lead to sparse and redundant features, reducing the model's ability to capture meaningful sequence-level information. To this end, we develop a multi-kernel feature fusion (MKFF) module that extracts key features across multiple receptive fields to reduce sparsity. These features are then effectively fused and transferred by combining global and local features. To improve computational efficiency, we introduce a downsampling residual feature embedding (DRFE) module, which compresses features efficiently and enhances performance. Experiments show that MFDm^{6}ARice surpasses state-of-the-art methods in performance, robustness, and generalization. Its scalability is demonstrated with maize m^6A data. Importantly, comparative experiments confirm that global–local dynamic fusion (GLDF) of multi-kernel features, appropriate downsampling, and residual connections significantly enhance feature representation and model performance.

Table 1. Details of benchmark dataset and independent test sets

Datasets	Number of positives	Number of negatives	Total number
Benchmark dataset	19 844	39 692	59 536
Same-species independent test set	4963	9923	14 886
Cross-species independent test set	24 532	24 518	49 050

Materials and methods

Datasets

In this study, one benchmark dataset and two independent test sets are used to evaluate the performance of MFDm^{6}ARice. Table 1 tabulates the details of these datasets.

Wang *et al.* [29] proposed the first rice m^6A dataset, with positive samples extracted from m^6A-seq peak sequences of *Japonica* Nipponbare seedling leaves, ranging from 20 to 800 nt in length. Negative samples were selected by extracting equal-length sequences without m^6A sites from the upstream and downstream regions of the positive samples. They used CD-HIT [34] to reduce homology bias and remove redundant sequences. After cleaning and deduplication, 80% of the positive and negative samples were randomly selected as the benchmark dataset for evaluating model performance. The final prediction model was tuned and trained, and its generalizability was tested using independent test sets. The remaining samples were used as a same-species independent test set, and *H. sapiens* m^6A data from DeepM6ASeq [26] were used as a cross-species independent test set. Unlike the peak sequences in the benchmark and same-species sets, the cross-species test set contained precise methylation sites with 101 nt-long positive and negative samples. Notably, these data had been mapped to the reference genome and provided as the corresponding DNA sequences (A, T, G, and C).

Architecture of MFDm^{6}ARice

Figure 1 shows the architecture of MFDm^{6}ARice framework. The framework consists of four main modules: input representation module, MKFF module, DRFE module, and output module.

Input representation

To adapt the input to the model, it is first necessary to convert the m^6A sequence into a numerical representation. The label encoding method is a simple yet effective approach for transforming classified text data into numerical form, assigning each category a unique numerical identifier [35]. Accordingly, we utilize this technique to represent the input m^6A sequences in this study. In particular, m^6A sequences are variable-length data that have rich information. To retain complete sequence information and incorporate it into the model, like SMEP, we pad all sequences to the maximum length of 800 to obtain the max-length padded sequences, of which each sequence has five categories. For example, assuming the maximum sequence length is 10, a sequence 'ATTCG' consisting of four type bases (A, T, G, and C) pads to 'PPPPPATTCG' where 'P' denotes padding. The five categories (P, A, T, G, and C) are assigned numbers by label encoding: 0, 1, 2, 3, and 4, respectively. Consequently, this max-length padded sequence can be represented as $LE\_Fea = [0, 0, 0, 0, 0, 1, 2, 2, 4, 3]$, as shown in Fig. 1A. In addition, we present an experimental analysis of padding length in the Supplementary Section S1 and Supplementary Table S1.

Figure 1. Overall flowchart of MFDm^{6}ARice. (A) Input representation of m^{6}A sequences. Label encoding is performed on sequences after padding to the maximum length of 800 to obtain max-length padded sequences. (B) After the input representation, it is input into the MKFF module. Its multi-kernel feature extraction function extracts multi-kernel features (the upper part of B), then inputs these features into the (C) GLDF function to obtain multi-kernel dynamic fused features. (D) The dynamic fused features are input into the DRFE module to get the corresponding feature embedding. DR_block represents the downsampling residual block. (E) Finally, the output module predicts whether contained m^{6}A sites or not.

Multi-Kernel feature fusion module

Although max-length padded sequences can preserve complete sequence-level information, they also introduce lengthy and ineffective information. Such can lead to sparse and redundant features that hinder the model's ability to capture subsequent meaningful sequence-level information. Therefore, after obtaining the m^{6}A feature representation (*LE_Fea*), we design an MKFF module that combines the multi-kernel feature extraction function and GLDF function to extract and integrate features from different kernels, as shown in Fig. 1B and C.

Multi-Kernel feature extraction function. To help mitigate the sparsity problem associated with max-length padded sequence encoding, we develop the multi-kernel feature extraction function, which extracts crucial features across multiple receptive fields. This function guarantees efficacious feature capture from different scales and regions, providing a more comprehensive feature representation and contextual information of the sequence. Specifically, firstly, the *LE_Fea* is passed through the embedding layer (*Embedding*) to obtain embedding vectors. Besides, for embedding vectors, a point-wise convolution (i.e. a kernel size of 1, $Conv_1$) is used to transform dimension and feature to obtain O_{embed}. As follows:

$$O_{embed} = Conv_1(Embedding(LE\_Fea)) \tag{1}$$

Then, four parallel paths are operated on O_{embed} to extract information at different kernel receptive fields:

$$k1 = Conv_1(O_{embed}) \tag{2}$$

$$k2 = Conv_3(Conv_1(O_{embed})) \tag{3}$$

$$k3 = Conv_5(Conv_1(O_{embed})) \tag{4}$$

$$k4 = Conv_1(MaxPool_3(O_{embed})) \tag{5}$$

where (2), (3), and (4) use convolution layers with kernel sizes of 1, 3, and 5 ($Conv_1$, $Conv_3$, $Conv_5$), respectively, to extract information of different spatial dimensions. The rationale for choosing these kernel sizes is provided in Supplementary Section S2. In addition, we also perform experimental analysis on other kernel sizes, such as 7 and 9, see Supplementary Section S2 and Table S2. Equations (3) and (4) initially perform $Conv_1$ on the input to reduce the number of channels, thereby decreasing the number of parameters and the complexity of the model. Equation (5) uses a max pooling layer with a kernel size of 3 ($MaxPool_3$) and then employs $Conv_1$ to change the number of channels. Appropriate padding is applied to these paths to maintain consistent input and output heights and widths. $k1$, $k2$, $k3$, and $k4$ are the multi-kernel features obtained.

Global-local dynamic fusion function. Having obtained multi-kernel features, if merely concatenating them is simple and practical, it does not consider the differences between the features at various kernels. Therefore, it is pivotal to integrate the multi-kernel features and efficiently transfer them. For this reason, we propose a GLDF function as an alternative to the simple concatenation process, ensuring the preservation of broad contextual information and fine-grained details.

This function contains two main parts: global channel dynamic fusion and local spatial dynamic fusion, as shown in Fig. 1. The lower part is global channel dynamic fusion, the upper part is local spatial dynamic fusion. Details are as follows.

(i) Feature concatenation

The first step is to concatenate the extracted multi-kernel features. After, $Conv_1$ is applied to perform point-wise channel information interaction and context aggregation at each spatial position.

$$MK\_Fea = Conv_1(cat(k1, k2, k3, k4)) \tag{6}$$

(ii) Global channel dynamic fusion

In global channel dynamic fusion, channel aggregation features are first generated by compressing the global spatial information to the channel using global average pooling *AvgPool*. The global channel attention weights $Global_{weight}$ are then dynamically generated by convolution with kernel size $Conv_{k_{adp}}$ and the *Sigmoid* nonlinearity function.

$$Global_{weight} = Sigmoid(Conv_{k_{adp}}(AvgPool(MK\_Fea))) \tag{7}$$

where k_{adp} represents the adaptive convolution kernel size, which is used to flexibly capture the dependencies between different channels [36]. Given the channel dimension C, it is calculated as follows:

$$k_{adp} = \left| \frac{_2(C)}{\gamma} + \frac{b}{\gamma} \right|_{odd} \tag{8}$$

where $||_{odd}$ means rounding down the absolute value to the nearest odd number. γ and b are two hyperparameters. As with ECA [36], they are set to 2 and 1, respectively.

Therefore, the global dynamic fusion features can be obtained by the following equation:

$$Global\_Fea = MK\_Fea \otimes Global_{weight} \tag{9}$$

where $\otimes$ is the element-wise multiplication.

(iii) Local spatial dynamic fusion

For local spatial dynamic fusion, the spatial attention weights $Local_{weight}$ are calculated through a bottleneck structure as follows:

$$squeeze = Conv_{k_{adp}}^{squeeze}(MK\_Fea) \tag{10}$$

$$Local_{weight} = Softmax(Conv_{k_{adp}}^{excitation}(squeeze)) \tag{11}$$

Given the input channel dimension C, the output channel of $Conv_{k_{adp}}^{squeeze}$ and the input channel of $Conv_{k_{adp}}^{excitation}$ are $C//r$, and the output channel of $Conv_{k_{adp}}^{excitation}$ is C. Here, r is the reduction ratio [37], a key hyperparameter. *Softmax* is the softmax function. Specifically, $Conv_{k_{adp}}^{squeeze}$ is applied to the *MK_Fea*, compressing the spatial features into a compact representation while retaining essential information. This squeezing operation reduces capacity and computational cost. Subsequently, $Conv_{k_{adp}}^{excitation}$ and a nonlinear transformation softmax function are employed to activate the information aggregated during the squeeze operation. These operations allow dynamic weighting of spatial features, facilitating the capture of dependencies between spatial features and enhancing critical features.

After obtaining the local spatial attention weights, $Local_{weight}$, the local dynamic fusion features are derived as follows:

$$Local\_Fea = MK\_Fea \otimes Local_{weight} \tag{12}$$

(iv) Combining global and local features

By concatenating *Global_Fea* and *Local_Fea* and feeding into $Conv_1$, the GLDF features (*GLDF_Fea*) are generated by the following:

$$GLDF\_Fea = Conv_1(cat(Global\_Fea, Local\_Fea)) \tag{13}$$

Ultimately, O_{embed} is transferred to *GLDF_Fea* through a residual connection operation [38] to obtain the final multi-kernel dynamic fusion features O_{MKFF}, which transfers more information and reduces distortion.

$$O_{MKFF} = O_{embed} \oplus GLDF\_Fea \tag{14}$$

where $\oplus$ is the broadcasting addition.

Downsampling residual feature embedding module

In Fig. 1D, to tackle the issues of high-dimensional features and low computational efficiency resulting from ineffective padding, we introduce a DRFE module, combining layer-by-layer downsampling, equal-length convolution, and residual connections. This module aims to efficiently compress the features, reducing dimensionality while retaining critical information.

First, we extract initial features on O_{MKFF} through two convolutions. This operation captures the fundamental patterns and features present in O_{MKFF}, thereby facilitating the generation of more expressive feature representations. As follows:

$$stem = Conv_5(Pad(Conv_5(Pad(O_{MKFF})))) \tag{15}$$

where a padding layer (*Pad*) ensures equal-length convolution that expands the receptive field, effectively captures long-range dependencies, and reduces information loss without increasing the parameters and computational complexity [39]. It also maintains the dimensions of the feature map, aiding subsequent layer-by-layer downsampling.

Then, we build the downsampling residual block (DR block). Details are as follows:

$$X^{down} = MaxPool_3(Pad(stem)) \tag{16}$$

$$X^{DRFE} = Conv_5(Pad(Conv_5(Pad(X^{down})))) \oplus X^{down} \tag{17}$$

By alternating between maximum pooling ($MaxPool_3$) and equal-length convolution, the spatial size of the feature map gradually reduces. Additionally, residual connections enable the network to retain important high-frequency information while reducing feature map size and maintaining the spatial consistency of feature maps [38]. Accordingly, this module employs residual connections to ensure the preservation of the original information integrity, thereby preventing the loss of significant features during the downsampling process. Three DR blocks are used in the DRFE module to obtain the required embedded features.

Furthermore, to learn a feature space where similar samples are closer and dissimilar samples are further apart, we introduce a contrastive learning loss function [40] ($\mathcal{L}_{cl}$) for representation optimization about DRFE features during training. For feature representations of two sequence X_i^{DRFE} and X_j^{DRFE}:

$$\mathcal{L}_{cl} = \frac{1}{2}((1-K)D(X_i^{DRFE}, X_j^{DRFE})^2 + K\{max(0, M - D(X_i^{DRFE}, X_j^{DRFE}))^2\}) \tag{18}$$

where $K = 0$ if two sequences belong to the same class, otherwise $K = 1$. D represents the Euclidean distance. M is the margin. If $K = 1$ and $D(X_i^{DRFE}, X_j^{DRFE}) < M$, we optimize by moving them away from each other. To facilitate understanding, we draw a schematic diagram of the contrastive learning process, see Supplementary Figure S1.

Output module

Finally, the probability of a sample containing a m^{6}A site ($\hat{y}$) is calculated as follows (Fig. 1E):

$$\hat{y} = FC_{Sigmoid}(Flatten(X^{DRFE})) \tag{19}$$

where $FC_{Sigmoid}$ denotes a fully connected layer with a *Sigmoid* activation function. If $\hat{y}$ is less than 0.5, the sequence is classified as a negative sample (not containing m^{6}A site). Otherwise, it is classified as a positive sample (contains a m^{6}A site).

Here, we utilize the binary cross-entropy loss function as the objective function to minimize:

$$\mathcal{L}_{bce} = -\frac{1}{N}\sum_{i=1}^{N}(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)) \tag{20}$$

where N is the batch size, y_i is the true label, and $\hat{y}_i$ is the predicted probability. Hence, the total loss function of the MFDm^{6}ARice is:

$$\mathcal{L}_{total} = \mathcal{L}_{cl} + \mathcal{L}_{bce} \tag{21}$$

Evaluation metrics

In this study, 5-fold cross-validation (5-CV) is adopted to evaluate the performance of MFDm^{6}ARice and state-of-the-art methods. For the benchmark dataset, we randomly select 80% as the training set and use the remaining 20% as the validation set to optimize model parameters. The final 5-CV result is the average of the results of five validation sets. Traditional evaluation metrics [41, 42], including accuracy (ACC), Matthew's correlation coefficient (MCC), the area under the receiver-operating characteristic curve (AUC), and the area under the precision-recall curve (AUPR) are used to assess the performance of our proposed method and other methods.

Results and discussion

This section introduces the experimental results, parameter analysis, and the effectiveness of each component of the MFDm^{6}ARice, accompanied by relevant visualizations and research.

Comparison with state-of-the-art methods

To evaluate the performance of MFDm^{6}ARice, we compare it with SMEP and PEA-m6A, the leading methods for predicting m^{6}A sites in rice, using two types of datasets mentioned previously. Simultaneously, we choose DeepM6ASeq, a classic and well-established method for predicting m^{6}A in other species, as a comparison.

Cross-validation performance on benchmark dataset

The predictive performance of MFDm^{6}ARice is assessed on the benchmark dataset via 5-CV. As demonstrated in Table 2, MFDm^{6}ARice exhibits superior overall performance compared to the other methods. In particular, the model achieves an ACC of 0.8321, an MCC of 0.6225, an AUC of 0.9038, and an AUPR of 0.8201, exhibiting an 11.81% improvement in overall performance compared to the second-best method, DeepM6ASeq.

Although the standard deviation (std) of MCC is not as optimal as PEA-m6A, the std of other metrics is superior to that of the comparative methods. These findings indicate that MFDm^{6}ARice demonstrates both notable performance and robustness.

Performance on independent test sets

To further illustrate the superiority and generalizability of MFDm^{6}ARice, we compare it with existing prediction methods on independent test sets, as presented in Table 3.

In Table 3, our proposed method exhibits greater effectiveness than the comparative method concerning the same-species independent test set. Specifically, MFDm^{6}ARice improves ACC, MCC, AUC, and AUPR by 2.12, 5.59, 1.86, and 3.68%, respectively, exhibiting a 13.25% improvement in overall performance over the suboptimal method. Meanwhile, our proposed method is comparable to the cross-species independent test set. MFDm^{6}ARice has an overall performance slightly higher than the second-best method.

Notably, the results from the same-species independent test set in rice are comparable to those of the 5-CV results, indicating that our method exhibits robust and transferable performance on the same species. However, as expected, the performance of all methods diminishes markedly on the cross-species independent test set due to species and data type discrepancies. These findings highlight the challenges of developing multi-species m^{6}A prediction methods and identifying unseen data. Despite these difficulties, MFDm^{6}ARice outperforms the comparison methods in overall performance, suggesting its resilience to challenges and potential.

Extended application of MFDm^{6}ARice framework on maize

In addition, to evaluate the model's utility and scalability on other species from the *Poaceae* family, we collect maize m^{6}A dataset from SMEP [29]. This dataset contains 11 150 positive samples and 22 300 negative samples. Like rice, the data in this dataset also have variable-length peak sequences. Then, we use the MFDm^{6}ARice framework tuned on the rice dataset to extend to this dataset. The processing and training of the MFDm^{6}ARice framework on this dataset are consistent with those on the rice dataset.

Supplementary Table S3 shows all results from the compared methods. As anticipated, the performance metrics of the models trained by directly applying the frameworks to the maize m^{6}A dataset are lower than those trained on the rice dataset. However, our method remains competitive. These findings indicate that MFDm^{6}ARice exhibits utility and potential for extension to other *Poaceae* plants.

Parameter analysis

To evaluate the influence of main hyperparameter settings on model performance, we conduct a hyperparameter sensitivity analysis [43, 44] encompassing batch size, output channels, reduction ratio (in GLDF's Local dynamic fusion), DR block (in DRFE), and margin through 5-CV. The hyperparameters ultimately used in this study are summarized in Table 4. At the same time, we further explore the impact of these hyperparameter changes on the validation set and the same-species independent test set. The results are shown in Fig. 2. Supplementary Section S3, Supplementary Tables S4 and S5 provide results and analyses of additional hyperparameters, such as the learning rate and the number of layers in the convolutional blocks. We also offer a hyperparameter selection process in Supplementary Section S4.

The impact of various hyperparameter settings on validation set

Examining the validation set results (solid line) in Fig. 2, we observe the following:

Table 4. The optimal hyperparameter settings of MFDm^{6}ARice

Hyperparameter	Setting
Batch size	128
Out channel	128
Ratio	4
DR block	3
Margin	2

N e: T e a e abbe a f ed c a .DR b c e d a e d a b c .

network structure by controlling feature map resolution, computational complexity, and information flow through downsampling and residual connections. Moderate information compression enhances performance, but excessive compression (e.g. DR_block=4) limits feature information, ultimately reducing performance.

(iii) Batch size: Batch size has a noticeable, though smaller, effect. As shown in the first column of Fig. 2, changing the batch size from 32 to 128 results in performance differences of 0.97, 2.78, 1.18, and 1.99% in ACC, MCC, AUC, and AUPR, respectively. Larger batch sizes offer more accurate gradient estimates, stabilizing convergence. However, very large batch sizes (e.g. batch size = 256) may cause the model to converge to suboptimal solutions, leading to a slight performance decrease.

(iv) Reduction ratio and margin: These two hyperparameters have minimal impact on performance, suggesting that the model is relatively insensitive to them compared to the more influential settings above.

Comparing results for different hyperparameter settings on validation set and independent test set

By comparing the results of the validation set and independent test set (dashed line) based on 5-CV, we observe that the general trends and impacts of different hyperparameter settings are closely consistent between the validation set and the independent test set. It shows that the hyperparameters selected by cross-validation in this work are effective. What's more, there is little difference between the validation and independent test set results. This highlights the model's strong generalization ability. These consistencies show that our model learns to generalize effectively and to remember the training data, ensuring its practical application in real-world scenarios.

Effectiveness of the MKFF module's functions

MFDm^{6}ARice, proposed in this work, designs a multi-kernel feature extraction with a GLDF module, MKFF. The module begins with extracting multi-kernel features and then fuses them through a global–local dynamic attention mechanism. To assess the efficacy of the MKFF module, we conduct an ablation study and effectiveness evaluation on two key functions: the multi-kernel feature extraction and the GLDF. Details are as follows.

Validity of multi-kernel feature extraction function

In the multi-kernel feature extraction function, we use (2), (3), (4), and (5) to obtain the multi-kernel features (k1234) of kernel1 ($k1$), kernel2 ($k2$), kernel3 ($k3$), and kernel4 ($k4$). To verify the effectiveness of k1234, we set up a comparative experiment with different kernel feature combinations, as shown in Fig. 3. In this experiment, k1 means only using kernel1 features. k12 represents the features combination of kernel1 and kernel2, k123 denotes the features combination of kernel1, kernel2, and kernel3, and so forth. Note that, except for the different feature combinations, the remainder of the model framework remains unaltered in this comparative experiment.

As evidenced in Fig. 3, the overall performance of the combined features exceeds that of a single kernel feature. The k1234 used in this work is optimal, i.e. four-kernel features combination. The performance of the three-kernel feature combination is superior to that of the two-kernel feature combination. It shows that, with the addition of different kernel features, the model can capture feature information of varying receptive fields, obtain more diverse feature maps, and provide more comprehensive information input for subsequent layers. It enhances model performance and demonstrates the efficacy of multi-kernel features.

Validity of global–local dynamic fusion function

To assess the effectiveness of the GLDF function, we design two types of comparative experiments: internal and external comparisons.

For the internal comparisons, we construct four GLDF variants:

- Replacing the GLDF module with concatenation (cat).
- Replacing the GLDF module with addition (add).
- Removing local dynamic fusion (global).
- Removing global dynamic fusion (local).

For external comparisons, we use other widely adopted feature fusion techniques: SE [37], ECA [36], and CBAM [45]. These methods are often employed to strengthen feature representation by focusing on specific feature channels or spatial locations. By replacing GLDF with these techniques, we can evaluate GLDF's relative effectiveness.

The results in Fig. 4 demonstrate that the GLDF module generally outperforms both internal and external comparison models, underscoring its effectiveness in dynamically fusing multi-kernel features. Concurrently, we observe the following:

(i) Superiority over concatenation and addition: The GLDF, SE, ECA, and CBAM modules perform better than simple concatenation or addition. Even the isolated components of GLDF (global and local dynamic fusion) outperform concatenation and addition. It suggests that simple fusion techniques like concatenation or addition fail to account for the distinct importance of each kernel feature map, which can impair model performance.

(ii) Addition versus concatenation: Adding feature maps yields worse results than concatenation. Element-wise addition of feature maps from different kernels can lead to information loss and reduce the complementarity between feature maps, resulting in less effective feature interaction and weaker feature representation.

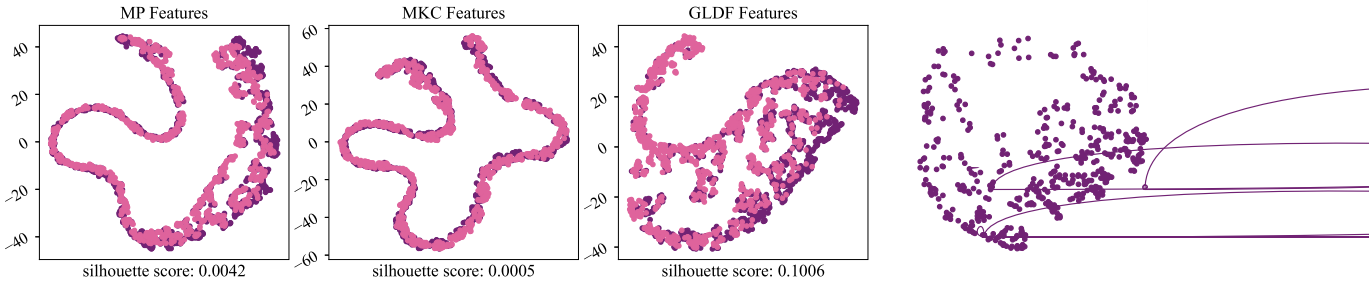
(iii) Importance of Global Features: Global fusion contributes more to performance than local fusion. In classification tasks, global features often carry more importance than local details. Global channel dynamic fusion captures overall statistical information through global average pooling, enabling a more adaptive and comprehensive adjustment of each channel's importance. In contrast, local spatial dynamic fusion focuses on adjusting weights within smaller areas, constrained by the receptive field, limiting its ability to capture global context.

(iv) Effective of Combined Global and Local Fusion: Despite the global fusion being more effective than the local fusion on its own, combining both global and local dynamic fusion

Table 5. The performance of MFDm^{6}ARice and its variants using 5-CV

Variants	ACC	MCC	AUC	AUPR
MFDm^{6}ARice	**0.8321**	**0.6225**	**0.9038**	**0.8201**
MFDm^{6}ARice w/o CL	0.8317	0.6186	0.9017	0.8169
MFDm^{6}ARice w/o MKFF	0.7305	0.4358	0.8079	0.6496
MFDm^{6}ARice w/o DRFE	0.7873	0.5174	0.8535	0.7339

N e: e a a e feac e c b ded.



size and complexity. Results from the benchmark and independent datasets show that MFDm^{6}ARice outperforms existing methods in accuracy, robustness, and generalization. It is also scalable to maize. Ablation studies and t-SNE visualizations confirm the effectiveness of dynamic fusion and downsampling in improving feature representation and m^{6}A site prediction accuracy.

Despite its promising performance, MFDm^{6}ARice presents several limitations that warrant attention. First, considering the variable-length sequences, simplicity, and computational efficiency, we use label encoding with maximum length padding to process RNA sequences. However, we recognize that this encoding method may overlook the chemical properties of nucleotides and potential sequence dependencies within RNA sequences. Therefore, in future work, we aim to explore alternative encoding methods based on biochemical properties and embedding-based representations to enhance the feature representation of variable-length sequences. Additionally, we discuss the potential impacts of these alternative encoding methods on model interpretability and performance in Supplementary Section S6. These could provide a more comprehensive encoding method, thereby addressing current limitations.

While the DRFE module effectively optimizes feature space compression, reducing model parameter count and computational consumption, it still presents relatively high computational costs and model complexity compared to simpler architectures such as those without DRFE, MLP, or stacked convolutional layers. These differences in efficiency and complexity are detailed in Supplementary Section S7 and Supplementary Table S8. As a result, the high computational demands may limit the model deployment in resource-constrained environments, such as large-scale agricultural applications. Therefore, in future work, we aim to address these challenges, including reducing storage and memory consumption and exploring techniques such as model pruning, quantization, and lightweight deployment frameworks. Detailed discussions can be found in Supplementary Section S8.

The dataset used in this study is specific to rice, which may introduce biases and limit the model generalizability. Although MFDm^{6}ARice demonstrates scalability with maize, it is not designed for multi- or cross-crop applications, limiting its broader utility. In contrast, methods like those proposed by [31, 53, 54] aim to predict methylation sites across multiple species. Future research could focus on developing a broad-spectrum m^{6}A prediction model that accommodates various crops or species, enhancing its applicability. Additionally, inherent biases in the dataset, such as the under-representation of specific sequence types or environmental factors influencing m^{6}A modifications, should be addressed in future studies. Incorporating more diverse datasets or integrating omics data from different conditions could help provide a more comprehensive biological context and improve model performance. For a more concrete plan, see Supplementary Section S9.

In summary, future efforts will focus on expanding the model's applicability to multi-crops prediction, integrating more detailed molecular features, and refining the rice m^{6}A dataset, which collectively will enhance both the accuracy and generalizability of m^{6}A site predictions.

Key Points

- We propose an end-to-end rice m^{6}A site prediction learning network, MFDm^{6}ARice, which can effectively learn the complete information of max-length padded sequences.
- To reduce feature sparsity caused by max-length padded sequences and effectively transfer the features extracted by multi-kernel, we design the MKFF module with a multi-kernel feature extraction and a GLDF mechanism that enriches and enhances feature representation and suppresses useless information.
- To solve the high-dimensional features and low computational efficiency caused by invalid padding, we introduce the DRFE module to efficiently compress features through layer-by-layer downsampling and residual connections, ensure valid information transfer, and improve computational efficiency.
- Extensive comparative, ablation experiments, and visualization studies demonstrate the superior performance of MFDm^{6}ARice and the rationality and effectiveness of the MKFF and DRFE modules.

Acknowledgments

We thank anonymous reviewers for their valuable suggestions.

Supplementary data

Supplementary data are available at *Briefings in Bioinformatics* online.

Conflict of interest: The authors declare no conflict of interest.

Funding

This work was supported by the National Natural Science Foundation of China (No. 61972002).

Data Availability

The data set and source code can be downloaded from https://github.com/zhlSunLab/MFDm6ARice.

References

1. Boccaletto P, Machnicka MA, Purta E. *et al*. MODOMICS: a database of RNA modification pathways. 2017 update. *Nucleic Acids Res* 2018;**46**:D303–7. https://doi.org/10.1093/nar/gkx1030.
2. Cantara WA, Crain PF, Rozenski J. *et al*. The RNA modification database, RNAMDB: 2011 update. *Nucleic Acids Res* 2010;**39**:D195–201. https://doi.org/10.1093/nar/gkq1028.
3. Zheng HX, Sun X, Xs Z. *et al*. m6A editing: new tool to improve crop quality? *Trends Plant Sci* 2020;**25**:859–67. https://doi.org/10.1016/j.tplants.2020.04.005.
4. Zhang F, Zhang YC, Liao JY. *et al*. The subunit of RNA N6-methyladenosine methyltransferase OsFIP regulates early degeneration of microspores in rice. *PLoS Genet* 2019;**15**:e1008120. https://doi.org/10.1371/journal.pgen.1008120.
5. Ma K, Han J, Zhang Z. *et al*. OsEDM2L mediates m6A of EAT1 transcript for proper alternative splicing and polyadenylation regulating rice tapetal degradation. *J Integr Plant Biol* 2021;**63**:1982–94. https://doi.org/10.1111/jipb.13167.
6. Huang Y, Zheng P, Liu X. *et al*. OseIF3h regulates plant growth and pollen development at translational level presumably

through interaction with OsMTA2. *Plants* 2021;**10**:1101. https://doi.org/10.3390/plants10061101.
7. Zhang K, Zhuang X, Dong Z. *et al*. The dynamics of N6-methyladenine RNA modification in interactions between rice and plant viruses. *Genome Biol* 2021;**22**:1–36. https://doi.org/10.1186/s13059-021-02410-2.
8. Shi Y, Wang H, Wang J. *et al*. N6-methyladenosine RNA methylation is involved in virulence of the rice blast fungus Pyricularia oryzae (syn. Magnaporthe oryzae). *FEMS Microbiol Lett* 2019;**366**:fny286. https://doi.org/10.1093/femsle/fny286.
9. Tian S, Wu N, Zhang L. *et al*. RNA N6-methyladenosine modification suppresses replication of rice black streaked dwarf virus and is associated with virus persistence in its insect vector. *Mol Plant Pathol* 2021;**22**:1070–81. https://doi.org/10.1111/mpp.13097.
10. Ren Z, Tang B, Xing J. *et al*. MTA1-mediated RNA m6A modification regulates autophagy and is required for infection of the rice blast fungus. *New Phytol* 2022;**235**:247–62. https://doi.org/10.1111/nph.18117.
11. Cheng Q, Wang P, Wu G. *et al*. Coordination of m6A mRNA methylation and gene transcriptome in rice response to cadmium stress. *Rice* 2021;**14**:62–15. https://doi.org/10.1186/s12284-021-00502-y.
12. Wang Y, Du F, Li Y. *et al*. Global N6-methyladenosine profiling revealed the tissue-specific epitranscriptomic regulation of rice responses to salt stress. *Int J Mol Sci* 2022;**23**:2091. https://doi.org/10.3390/ijms23042091.
13. Chen J, Cao H, Chen D. *et al*. Transcriptome-wide analysis of m6A methylation reveals genetic responses to cadmium stress at germination stage in rice. *Environ Exp Bot* 2023;**205**:105130. https://doi.org/10.1016/j.envexpbot.2022.105130.
14. Chen D, Fu L, Su T. *et al*. N6-methyladenosine methylation analysis reveals transcriptome-wide expression response to salt stress in rice roots. *Environ Exp Bot* 2022;**201**:104945. https://doi.org/10.1016/j.envexpbot.2022.104945.
15. Dominissini D, Moshitch-Moshkovitz S, Schwartz S. *et al*. Topology of the human and mouse m6A RNA methylomes revealed by m6A-seq. *Nature* 2012;**485**:201–6. https://doi.org/10.1038/nature11112.
16. Meyer KD, Saletore Y, Zumbo P. *et al*. Comprehensive analysis of mRNA methylation reveals enrichment in 3' UTRs and near stop codons. *Cell* 2012;**149**:1635–46. https://doi.org/10.1016/j.cell.2012.05.003.
17. Chen K, Lu Z, Wang X. *et al*. High-resolution N6-methyladenosine (m6A) map using photo-crosslinking-assisted m6A sequencing. *Angew Chem* 2015;**127**:1607–10. https://doi.org/10.1002/ange.201410647.
18. Linder B, Grozhik AV, Olarerin-George AO. *et al*. Single-nucleotide-resolution mapping of m6A and m6Am throughout the transcriptome. *Nat Methods* 2015;**12**:767–72. https://doi.org/10.1038/nmeth.3453.
19. Ke S, Alemu EA, Mertens C. *et al*. A majority of m6A residues are in the last exons, allowing the potential for 3' UTR regulation. *Genes Dev* 2015;**29**:2037–53. https://doi.org/10.1101/gad.269415.115.
20. Meyer KD. DART-seq: an antibody-free method for global m6A detection. *Nat Methods* 2019;**16**:1275–80. https://doi.org/10.1038/s41592-019-0570-0.
21. Ayub M, Bayley H. Individual RNA base recognition in immobilized oligonucleotides using a protein nanopore. *Nano Lett* 2012;**12**:5637–43. https://doi.org/10.1021/nl3027873.
22. Garalde DR, Snell EA, Jachimowicz D. *et al*. Highly parallel direct RNA sequencing on an array of nanopores. *Nat Methods* 2018;**15**:201–6. https://doi.org/10.1038/nmeth.4577.
23. Parker MT, Knop K, Sherwood AV. *et al*. Nanopore direct RNA sequencing maps the complexity of Arabidopsis mRNA processing and m6A modification. *Elife* 2020;**9**:e49658. https://doi.org/10.7554/eLife.49658.
24. Zhou Y, Zeng P, Li YH. *et al*. SRAMP: prediction of mammalian N6-methyladenosine (m6A) sites based on sequence-derived features. *Nucleic Acids Res* 2016;**44**:e91–1. https://doi.org/10.1093/nar/gkw104.
25. Liu L, Lei X, Meng J. *et al*. WITMSG: large-scale prediction of human intronic m6A RNA methylation sites from sequence and genomic features. *Curr Genomics* 2020;**21**:67–76. https://doi.org/10.2174/1389202921666200211104140.
26. Zhang Y, Hamada M. DeepM6ASeq: prediction and characterization of m6A-containing sequences using deep learning. *BMC Bioinformatics* 2018;**19**:524–11. https://doi.org/10.1186/s12859-018-2516-4.
27. Chen J, Zou Q, Li J. DeepM6ASeq-EL: prediction of human N6-methyladenosine (m6A) sites with LSTM and ensemble learning. *Front Comp Sci* 2022;**16**:1–7. https://doi.org/10.1007/s11704-020-0180-0.
28. Rehman MU, Tayara H, Chong KT. DL-m6A: identification of N6-methyladenosine sites in mammals using deep learning based on different encoding schemes. *IEEE/ACM Trans Comput Biol Bioinform* 2022;**20**:904–11. https://doi.org/10.1109/TCBB.2022.3192572.
29. Wang Y, Zhang P, Guo W. *et al*. A deep learning approach to automate whole-genome prediction of diverse epigenomic modifications in plants. *New Phytol* 2021;**232**:880–97. https://doi.org/10.1111/nph.17630.
30. Liang Z, Zhang L, Chen H. *et al*. m6A-maize: weakly supervised prediction of m6A-carrying transcripts and m6A-affecting mutations in maize (Zea mays). *Methods* 2022;**203**:226–32. https://doi.org/10.1016/j.ymeth.2021.11.010.
31. Song M, Zhao J, Zhang C. *et al*. PEA-m6A: an ensemble learning framework for accurately predicting N 6-methyladenosine modifications in plants. *Plant Physiol* 2024;**195**:1200–13. https://doi.org/10.1093/plphys/kiae120.
32. Wang X, Yan R. RFAthM6A: a new tool for predicting m6A sites in Arabidopsis thaliana. *Plant Mol Biol* 2018;**96**:327–37. https://doi.org/10.1007/s11103-018-0698-9.
33. Qiang X, Chen H, Ye X. *et al*. M6AMRFS: robust prediction of N6-methyladenosine sites with sequence-based features in multiple species. *Front Genet* 2018;**9**:495. https://doi.org/10.3389/fgene.2018.00495.
34. Fu L, Niu B, Zhu Z. *et al*. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* 2012;**28**:3150–2. https://doi.org/10.1093/bioinformatics/bts565.
35. Pedregosa F, Varoquaux G, Gramfort A. *et al*. Scikit-learn: machine learning in Python. *J Mach Learn Res* 2011;**12**:2825–30.
36. Wang Q, Wu B, Zhu P. *et al*. ECA-net: efficient channel attention for deep convolutional neural networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11534–42. Seattle, Washington, USA: IEEE, 2020.
37. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–41. Salt Lake City, Utah, USA: IEEE, 2018.
38. He K, Zhang X, Ren S. *et al*. Deep residual learning for image recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–8. Las Vegas, Nevada, USA: IEEE, 2016.
39. Johnson R, Zhang T. Deep pyramid convolutional neural networks for text categorization. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 562–70. Vancouver, Canada: ACL, 2017.