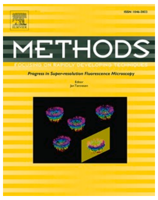
ARTICLE IN PRESS

Methods xxx (xxxx) xxx

Contents lists available at ScienceDirect

Methods

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



Deep transformers and convolutional neural network in identifying DNA N6-methyladenine sites in cross-species genomes

Nguyen Quoc Khanh Le[a,b,c,*], Quang-Thai Ho[d,e]

[a] *Professional Master Program in Artificial Intelligence in Medicine, College of Medicine, Taipei Medical University, Taipei 106, Taiwan*
[b] *Research Center for Artificial Intelligence in Medicine, Taipei Medical University, Taipei 106, Taiwan*
[c] *Translational Imaging Research Center, Taipei Medical University Hospital, Taipei 110, Taiwan*
[d] *College of Information & Communication Technology, Can Tho University, Viet Nam*
[e] *Department of Computer Science and Engineering, Yuan Ze University, Chung-Li, 32003, Taiwan*

ARTICLE INFO

Keywords:
N6-methyladenine site
Post-translational modification
Natural language processing
Deep learning
DNA sequence analysis
Contextualized word embedding

ABSTRACT

As one of the most common post-transcriptional epigenetic modifications, N6-methyladenine (6 mA), plays an essential role in various cellular processes and disease pathogenesis. Therefore, accurately identifying 6 mA modifications is necessary for a deep understanding of cellular processes and other possible functional mechanisms. Although a few computational methods have been proposed, their respective models were developed with small training datasets. Hence, their practical application is quite limited in genome-wide detection. To overcome the existing limitations, we present a novel model based on transformer architecture and deep learning to identify DNA 6 mA sites from the cross-species genome. The model is constructed on a benchmark dataset and explored a feature derived from pre-trained transformer word embedding approaches. Subsequently, a convolutional neural network was employed to learn the generated features and generate the prediction outcomes. As a result, our predictor achieved excellent performance during independent test with the accuracy and Matthews correlation coefficient (MCC) of 79.3% and 0.58, respectively. Overall, its performance achieved better accuracy than the baseline models and significantly outperformed the existing predictors, demonstrating the effectiveness of our proposed hybrid framework. Furthermore, our model is expected to assist biologists in accurately identifying 6mAs and formulate the novel testable biological hypothesis. We also release source codes and datasets freely at https://github.com/khanhlee/bert-dna for front-end users.

1. Introduction

DNA and mRNA modifications play an essential role in all three domains of life (e.g., bacteria, archaea, and eukaryote). To date, roughly 150 modification sites have been discovered, while N6-methyladenine (6 mA) is the most abundant modification type that contributes to various cellular processes [1]. 6 mA influences RNA translation efficiency and structural stability and further researches revealed that it could promote mRNA export, and regulate tissue differentiation. Furthermore, several studies showed that 6 mA was not only found in tRNA, rRNA, and small nuclear RNA (snRNA) but was also found in several long non-coding RNA. In humans, it has been reported that 6 mA is associated with several diseases, such as cancer [1], glioblastomas [2], hormetic heat stress [3], hypertension [4], etc. Therefore, the identification of 6 mA is vital for understanding the DNA modification processes and revealing the epigenetic regulation-related diseases.

Since our knowledge of the DNA 6 mA modification and its dissemination in multicellular eukaryotes remains incomplete, precise identification of its location in the genome is needed. Recently, several wet-lab experiments were carried out to identify DNA 6 mA, including the Nanopore sequencing [5,6], enzyme-assisted sequencing [7] which offers better or even base resolution epitranscriptome examination. Recently, with the development of next generation sequencing (NGS) technologies to contribute to increasing the size of biological databases, it is now timely and imperative to propose efficient computational models to identify modification sites in the DNA levels. Despite the increasing number of experimentally determined 6mAs, the underlying mechanism of DNA 6 mA specificity remains largely unknown. On the other hand, large-scale experimental identification of DNA 6 mA is laborious, time-consuming in common, and some of them have problems

[*] Corresponding author at: Professional Master Program in Artificial Intelligence in Medicine, College of Medicine, Taipei Medical University, Taipei 106, Taiwan.
E-mail address: khanhlee@tmu.edu.tw (N.Q.K. Le).

https://doi.org/10.1016/j.ymeth.2021.12.004
Received 13 November 2021; Received in revised form 30 November 2021; Accepted 9 December 2021
Available online 13 December 2021

with sequencing quality and signal-to-noise ratio. Furthermore, computational methods can be used to compensate for the shortcomings of experimental methods and provide an alternative approach, based on our known experimental data, for the identification of DNA 6 mA.

A variety of computational methods have been developed to identify DNA 6 mA sites on different genomes. Among these methods, i6mA-Pred [8], iN6-methylat (5-step) [9], Zhou et al. [10], iDNA6mA (5-step rule) [11], SNNRice6mA [12], Lv et al. [13], Rahman et al. [14] focused on rice genomes. On the other applications, few predictors have been proposed to predict 6 mA on more genomes i.e., 6 mA-finder [15], SpineNet-6 mA [16], Meta-i6mA [17], DeepM6ASeq-EL [18], TS-m6A-DL [19], and HSM6AP [20]. While all these methods have shown excellent performance, the major drawback was their training dataset size, single machine learning-based model, single genome, and limited feature encodings. Notably, the number of experimental samples is exponentially growing, and the existing computational models did not reach satisfactory results. Therefore, it seems evident that more efforts should be tried to achieve better performance for this AI-based prediction model.

The success of deep transformers natural language processing (NLP) and multi-modal deep learning models in other domains [21] have motivated us to propose a novel method to solve this problem. In this study, we have considered DNA sequences as natural sentences before applying the deep transformers NLP model on them. After getting feature vectors from pre-trained NLP model, we used them as input to the deep learning model predicting DNA 6 mA sites. The results demonstrated the effectiveness of this combination in representing DNA sequences and predicting 6 mA sites with high performance. We also compared the results of our best-performing 6 mA prediction model with the existing state-of-the-art methods to look at the significant differences.

The remaining parts of this paper are structured as: section 2 discusses the detailed method for collecting data, extracting features, and implementing the learning strategy. Section 3 focuses on the discussion of the outcomes of our method and finally, section 4 gives the conclusion of this paper.

2. Materials & methods

2.1. Benchmark dataset

Most bioinformatics-based prediction models start with a dataset, which comes from benchmark data or manual collection data. In this work, we used the DNA 6 mA dataset from the 6 mA-Finder work [15] to evaluate our representation method. This dataset is a very general one since it was retrieved by combining different cross-species datasets. All the sequences were 41-bp long with the 6 mA site in the center and the CD-HIT software [26] was applied to remove the redundant sequences in the dataset. Moreover, the original papers divided the dataset into training dataset (including 2,500 positive data and 2,500 negative data) and independent dataset (including 268 positive data and 216 negative data). Since we used the benchmark dataset, we did not apply pre-processing steps as well as modify any number of data. Thus, all of these data were used in our further analysis for modeling. We also plotted the two-sample logo (Fig. 1) to see the distributed differences of nucleotides in 6 mA positive and negative data. As shown in this figure, the differences between positive and negative data appeared much at the position of 1 and 2 in the DNA segments. These differences might play an important role to help our model determine the 6 mA positions or not.

2.2. DNA sequence processing

Normally, raw DNA sequences were represented in FASTA format, which is a standard format for all biological sequences. To use them in NLP models, we firstly transformed DNA segments with a specific length into a “DNA sentence” composed of unigram (single nucleotide). Therefore, the single nucleotide could be treated as a word in our NLP model. The length of all these DNA segments was set at a fixed value of sliding window (i.e., 41-bp in this dataset). Normally, pre-trained NLP models suggest adding the special tokens i.e., CLS or SEP, which stand for classing and separating sentences in text classification tasks. In our model, we decided to keep original DNA segments without tokens since these segments are not supposed to form sentence-like information in

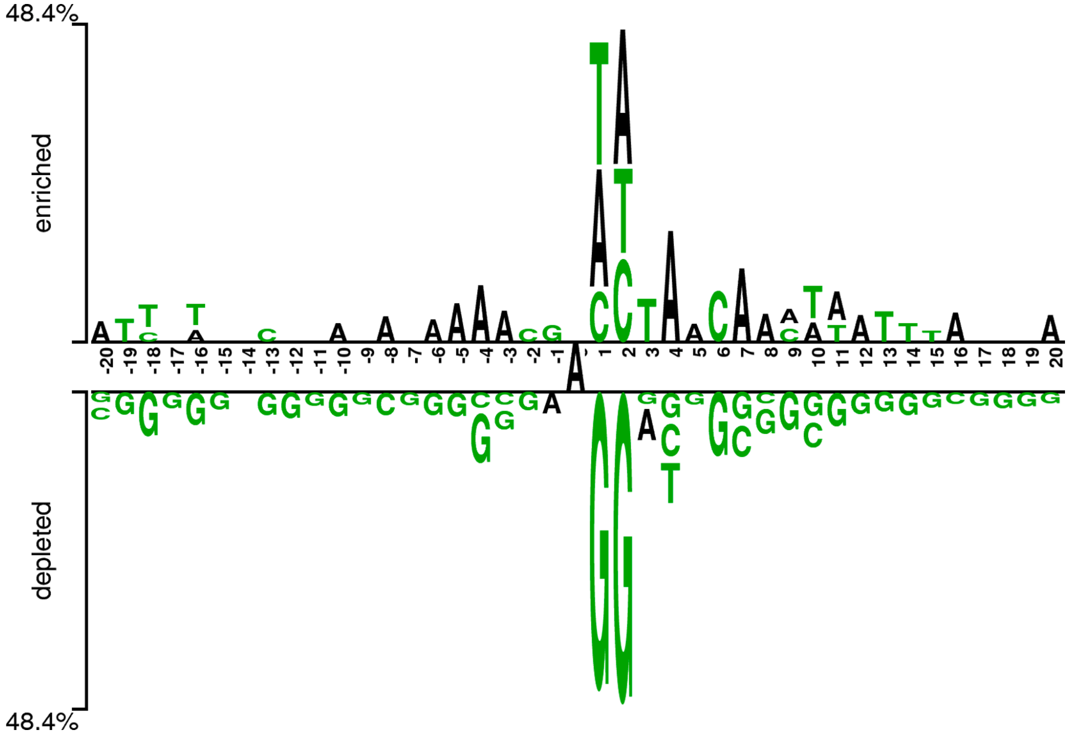


Fig. 1. Two-sample logo of benchmark dataset. There are significantly differences between positive and negative data representation in position 1 and 2.

biology. Thus, our preprocessed sequences contained DNA fragments with length of 41bps (one nucleotide as one word in biological sentence).

$$S = N_1 N_2 \cdots N_{41} (N \in [A, C, G, T]) \tag{1}$$

2.3. Deep transformers NLP pre-trained and fine-tuning

The whole architecture of our study includes two different stages: NLP transformers to extract features and deep convolutional neural network (CNN) to learn features (as shown in Fig. 2). Transformers is a deep learning model that uses the attention mechanism to weigh the importance of each portion of the input data differently. The concept behinds transformer is 'word embedding', which is a learned representation for text. It uses a predefined dense vector representation to represent individual words and has been extensively used in language modeling and NLP application function learning [22]. Word embedding methods study the relationship between sequential elements in a predefined fixed-sized real-valued vector represented for the vocabulary of the corpus. It is mainly used in NLP and especially used to solve many sequence modeling tasks with high performance. With the similar nature concept, it can be also used in representing biological sequences in previous studies [23,24]. Hence, we also assessed the performance of this architecture on our DNA sequence.

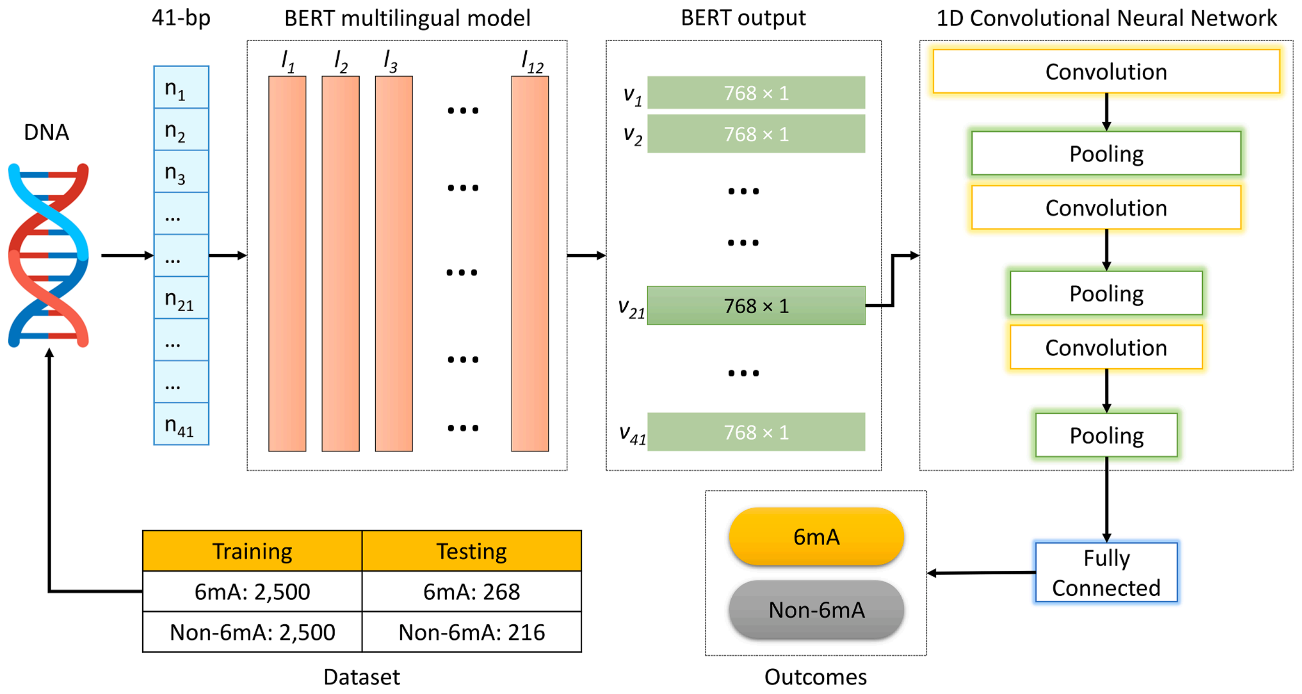
In detail, the idea is to look at the potential of pre-trained transformers models that had been trained on other natural languages and apply it to DNA sequence. Recently, there are an increasing number of pre-trained NLP models released with promising performance i.e., GPT, BERT, CodeBERT, ELMo, or XLNet. This study took BERT [25], one of the popular models among them, as our pre-trained NLP model to extract features from DNA sequences. The basic idea of BERT is to make use of transformers to learn contextual relations between words (or even sub-words) in a text. More detailed information on transformer and BERT model can be referred to in the original technical papers [21,25]. To use BERT model in our specific task, we performed fine-tuning steps to add a small core into pre-trained model. Since our problem is a classification task, we decided to add a classification layer on top of BERT layers to extract the features. There are a lot of BERT pre-trained models released, and we used "*bert-base-multilingual-cased*" which is trained on cased text in the top 104 languages with the largest corpus. In this fine-tuning process, all hyper-parameters of BERT model were kept as default since we did not aim to modify the original pre-trained model. After our DNA sequences went through the BERT model, their corresponding feature vectors were extracted and could be used for further prediction purposes. Because this pre-trained BERT model used 104 languages, 12-layer, and 768-hidden, it generated a vector 1 × 768 for each word. Therefore, the outputs of our DNA sequences (41bps) were matrices of 41 × 768.

2.4. Convolutional neural network architecture

Deep learning algorithms have proven to be more efficient than traditional machine learning algorithms, which automate processes to solve complex tasks. A deep neural network (DNN) is an extension of a multi-hidden network – artificial neural network (ANN) which allows DNN to perform multiple sophisticated tasks, where each layer is connected only from the previous one and connected only to the next layer in the cached portion. The most common types of DNN architectures are recurrent neural networks (RNNs) and CNNs [26,27]. In some cases, RNNs and CNNs can be combined to utilize the benefit of DNN architecture. This study used CNN to learn the features extracted from BERT model.

Since the output of BERT layer is 41 × 786bps, we decided to use the middle position (prediction position) as input to our CNN in this study. CNN, one of the popular deep learning techniques, tries to mimic human brains to learn and interpret the data. It is a biologically inspired neural network, and its neurons can respond to the surrounding coverage units and perform well in feature extraction tasks from input data. CNNs are primarily composed of two layers: pooling and convolutional layers as follows:

- Convolutional layer: In this layer, the convolution or correlation is performed by sliding filters over the input data. Multiple convolution



Training	Testing
6mA: 2,500	6mA: 268
Non-6mA: 2,500	Non-6mA: 216

Fig. 2. Whole architecture of the study. DNA fragments (41-bp) are firstly inserted into deep transformers pre-trained NLP model (12 layers $l_1, l_2, l_3, \ldots, l_{12}$) to extract their features. The generated results are 41 vectors ($v_1, v_2, v_3, \ldots, v_{41}$) with dimension of 768 and then we take the middle one (our target position) as main features. Another CNN model are then employed to learn these features and generate the prediction outcomes (6 mA or non-6 mA).

filters are used to learn multiple features from the input data. In this work, we have used 1D convolution to exploit sequential correlation over DNA sequence vectors. It can be mathematically expressed as:

$$F_i = h\left(\left(F^{\circ}_{i-1} W_i\right) + b_i\right) \tag{2}$$

where F_i is the feature map of the i^{th} layer with W_i as weight matrix of the convolution filter and b_i is the bias vector. h represents the activation function and ° means convolution operation. In CNN, the preferred activation function is ReLU.

- In contrast, a pooling operation, also known as subsampling, is performed on the output of the convolution layer. The max-pooling and average pooling are the frequently used pooling operations. This layer is used to reduce the feature map's dimension and select the most relevant features. In this work, we have used a max-pooling operation.

In this study, we ran hyperparameter tuning to find the optimal architecture of our CNN model. As a result, our final architecture included three sets of convolutional and max-pooling layers. The outputs from each max-pooling layer were applied to dropout layers, and the ReLU function was used as an activation function of the convolutional layer. After the output from the last max pooling layer was flattened, the final output was obtained by applying the fully connected layer to it. In this layer, a sigmoid function has been used as an activation function. Table 1 shows the architecture with layers and parameters of our CNN model.

2.5. Model setting and evaluation metrics

In this work, we used some common classification indicators to evaluate the effectiveness of the model, including accuracy, sensitivity, specificity, Matthews Correlation Coefficient (MCC), and Area Under Curve (AUC) [28]. Detailed mathematical formulas of these metrics are as follows:

$$Sensitivity = \frac{TP}{TP + FN} \tag{3}$$

$$Specificity = \frac{TN}{TN + FP} \tag{4}$$

$$Accuracy = \frac{TP + TN}{TN + TN + FP + FN} \tag{5}$$

$$MCC = \frac{(TPxTN) - (FPxFN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{6}$$

Finally, we used k-fold cross-validation (k = 5) as a method for evaluating model performance. All NLP and deep learning models were implemented in Python 3 with NVIDIA GeForce GTX 3080.

Table 1
Convolutional neural network architecture.

Layer	Output shape	Parameters
Conv1D	(None, 766, 32)	128
MaxPooling1D	(None, 383, 32)	0
Conv1D	(None, 381, 64)	6208
MaxPooling1D	(None, 190, 64)	0
Conv1D	(None, 188, 128)	24,704
MaxPooling1D	(None, 94, 128)	0
Flatten	(None, 12032)	0
Dense	(None, 128)	1,540,224
Dense	(None, 2)	258

Total parameters: 1,571,522, all of them are trainable parameters.

3. Results & discussions

3.1. Baseline comparison

For the first comparative evaluation, we would like to see the potential in learning BERT features among different baseline classifiers. Hence, we evaluated the efficiency of our DNN compared to conventional machine learning and ensemble learning algorithms such as Random Forest (RF), Support Vector Machine (SVM), Adaptive Boosting (AdaBoost), and eXtreme Gradient Boosting (XGBoost). Hyperparameter tuning processes have been also performed in this step to ensure there was a fair comparison among different representation methods. There existed some methods to optimize such as grid search, direct search, or random search. Grid search will cost a lot of time with trying for every hyper-parameter. Random search shows better efficiency than grid search and is able to find a better solution with a short computation time. In this study, because we did not have many combinations on the hyperparameters, we applied grid search directly to search all possible combinations. Detailed information of final optimal combinations is shown in Table 2.

To see the comparison among different classifiers, we plotted the ROC curves of cross-validation and independent test (Fig. 3A, 3B, respectively). Our choice (CNN) outperformed the other algorithms with the AUC reaching 0.841 in the cross-validation and 0.805 in the independent dataset. The high performance showed that our CNN could learn the BERT features better than other algorithms. It is consistent with the previous bioinformatics works that used CNN to learn the biological features [23,27]. In addition, the less variance between cross-validation and independent convinced that our model did not lie into overfitting. A reason for this non-overfitting problem could be explained as the use of batch normalization and dropout values in our model. These techniques have been proven as useful ones to resolve the overfitting of most DNN architectures. Hence, we can use the trained model to predict the outcomes of any unseen data efficiently.

3.2. Comparison between BERT and well-known feature representations

To convince the effectiveness of any method, an essential step is to compare with previous methods on the same retrieved dataset. In this field, there are many methods to extract the information of DNA sequences and they have been proven the efficiency in reaching good performance. In this study, we, therefore, extracted our DNA sequences using different descriptors to have a comparison with our BERT features. The selected features included kmer, dinucleotide-based auto-covariance (DAC), dinucleotide-based cross-covariance (DCC), dinucleotide-

Table 2
Optimal hyperparameters of all machine learning algorithms.

Algorithm	hyperparameter

RF	max_depth = 10
	max_features = sqrt
	min_samples_leaf = 1
	min_samples_split = 10
	n_estimators = 1500

SVM	C = 0.001
	gamma = 0.1
	kernel = poly

AdaBoost	n_estimators = 600
	learning_rate = 0.1

XGBoost	min_child_weight = 1
	gamma = 2.5
	subsample = 0.6
	colsample_bytree = 0.4
	max_depth = 50
	n_estimators = 200

RF: Random Forest, SVM: support vector machine, AdaBoost: Adaptive Boosting, XGBoost: eXtreme Gradient Boosting.

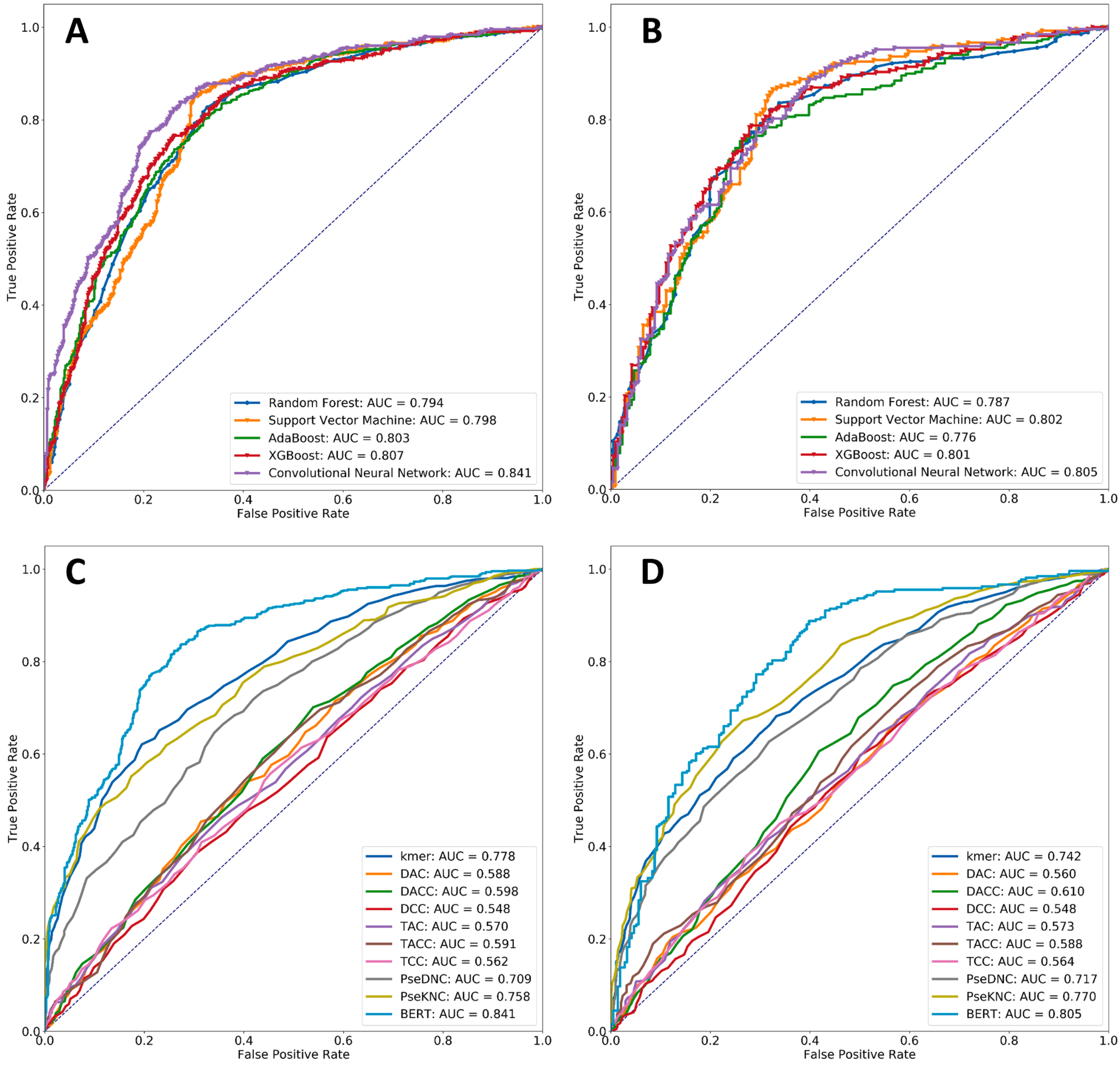


Fig. 3. ROC curves for comparing the performance among different algorithms: (A) cross-validation, (B) independent test; different features: (C) cross-validation and (D) independent test. The performance of the model using CNN and BERT features outperformed the others.

based auto-cross covariance (DACC), trinucleotide-based auto-covariance (TAC), trinucleotide-based cross-covariance (TCC), trinucleotide-based auto-cross covariance (TACC), pseudo dinucleotide composition (PseDNC), and pseudo k-tupler nucleotide composition (PseKNC). The aforementioned features are common and widely used in previous studies, even 6 mA predictors such as 6 mA-Finder [15], iDNA6mA-PseKNC [29], MM-6mAPred [30]. Therefore, such comparison can help to evaluate our features and relatively compare to other predictors on the same dataset. As shown in their algorithms, the aforementioned features depend on biological insights such as nucleotide-based correlation or composition, etc. They could be used to represent the DNA sequences, however, NLP models that rely on the biological language information hold potential in learning the sequences. Especially, in this study, we used the BERT multilingual algorithm which is the state-of-the-art NLP model and had been proven its efficiency in different fields. It also makes a significant contribution in NLP-based bioinformatics field since most previous similar studies focused on other pre-trained NLP models such as fastText [9,31] or ELMo [32]. Moreover, the use of only-middle position helped us focus on specific position which is more efficient in site-level prediction, different from previous BERT use case [23].

As shown in Fig. 3C and 3D, our model constructed from BERT features have outperformed the other features with AUC of 0.841 and 0.805 in cross-validation and independent test, respectively. To see more information on the comparison, we also reported all the measurement metrics in Table 3. It is easy to see that in addition to AUC, the sensitivity, specificity, and accuracy also improved at about 5–10% compared to the other extracted features.

3.3. Comparison with the existing predictors

In this section, we compared the predictive performance of our model with previously published predictors, such as iDNA6mA-PseKNC [29], iDNA6mA-Rice [33], i6mA-Pred [8], MM-6mAPred [30], SDM6A [34], and 6 mA-Finder [15] on the independent set. For a fair comparison, we used their tools/web servers to predict our independent dataset

Table 3
Comparative performance among different representation features using CNN.

	Cross-validation				Independent			
Features	Sens	Spec	Acc	MCC	Sens	Spec	Acc	MCC
kmer	64.2	75.2	69.7	0.396	66.8	61.6	64.5	0.283
DAC	55.4	54.8	55.1	0.102	56.3	54.6	55.6	0.109
DCC	58.3	56.4	57.3	0.147	55.2	56.0	55.6	0.112
DACC	51.6	54.6	53.1	0.062	50.7	56.0	53.1	0.067
TAC	53.6	56.1	54.9	0.097	54.1	53.7	53.9	0.078
TCC	55.4	57.5	56.4	0.129	51.1	53.7	52.3	0.048
TACC	53.3	56.4	54.8	0.097	53.0	55.6	54.1	0.085
PseDNC	62.1	67.5	64.8	0.296	65.3	65.3	65.3	0.304
PseKNC	64.3	74.6	69.4	0.390	77.0	66.7	71.9	0.440
BERT	86.4	68.8	77.6	0.651	84.3	73.1	79.3	0.580

Sens: Sensitivity, Spec: Specificity, Acc: Accuracy, MCC: Matthews Correlation Coefficient.

and compute the corresponding performance measurements. Among them, MM-6mAPred and SDM6A returned AUC values as their metrics and they showed AUC of 0.6284 and 0.6970, respectively. Compared to both of them on the same independent dataset, our model was improved at more than 10% in AUC. Moreover, iDNA6mA-PseKNC, iDNA6mA-Rice, and i6mA-Pred returned the confusion matrix values in their model, thus we took the sensitivity and specificity values to compare to our model. The detailed comparison is shown in Fig. 4. From this figure, we have seen that our model performance was slightly lower than 6 mA-Finder and iDNA6mA-PseKNC. However, we were superior to all previous predictors in specificity. It strongly indicates that our model was efficient and could be comparable to the other methods or predictors. Therefore, this combination between deep transformers BERT and 1D CNN would be a potential fit for solving this problem in particular and other sequence-based problems in general.

3.4. Model interpretation

We integrated three representative feature techniques namely t-SNE [35], UMAP [36], and SHAP [37] analyses to understand the robustness behind our deep transformers model. Mathematical formulation and detailed description of them are given in previous studies [35–37]. Three of them are useful in machine learning interpretation and they have been widely applied in computational biology also. We used the same procedure here and computed SHAP calculation. As shown in Fig. 5A and 5B, the deep transformers features could help to separate 6 mA and non-6 mA samples clearly in t-SNE and UMAP analyses, respectively. Moreover, Fig. 5C shows 20 top features ranking based on SHAP values and their directionality for each model. Thus, the positive values and negative values respectively influence the prediction towards 6mAs and non-6mAs. The result shows that our deep transformers and CNN model generated predicted probability values employed here, ranked from the top 1 to 5 positions, played a significant role, and the remaining other features act as the complementary role in 6 mA prediction. Thus, overall, all encoding employed in this study contributed to the final prediction, but the probability features of each classifier are varied. Finally, the SHAP results show that including more feature encodings in the model framework may be in future studies will improve the prediction performance.

4. Conclusion

This work presents a novel method to predict 6 mA sites based on deep transformers architecture with DNA sequence as input. As for feature encoding, we selected the BERT pre-trained model on multilingual language to apply it to our DNA sequence. The features were then learned by a 1D CNN to extract the hidden information among features, as well as sequences. It is amongst the challenging tasks in bioinformatics and we achieved consistent cross-validation and independent evaluation performance, indicating that our model did not suffer from over-fitting and working well on unseen data. Furthermore, our model was consistently superior to the baseline models and previously published predictors on the same problem or data.

Funding

This work was supported by the Ministry of Science and Technology, Taiwan [grant number MOST110-2221-E-038-001-MY2].

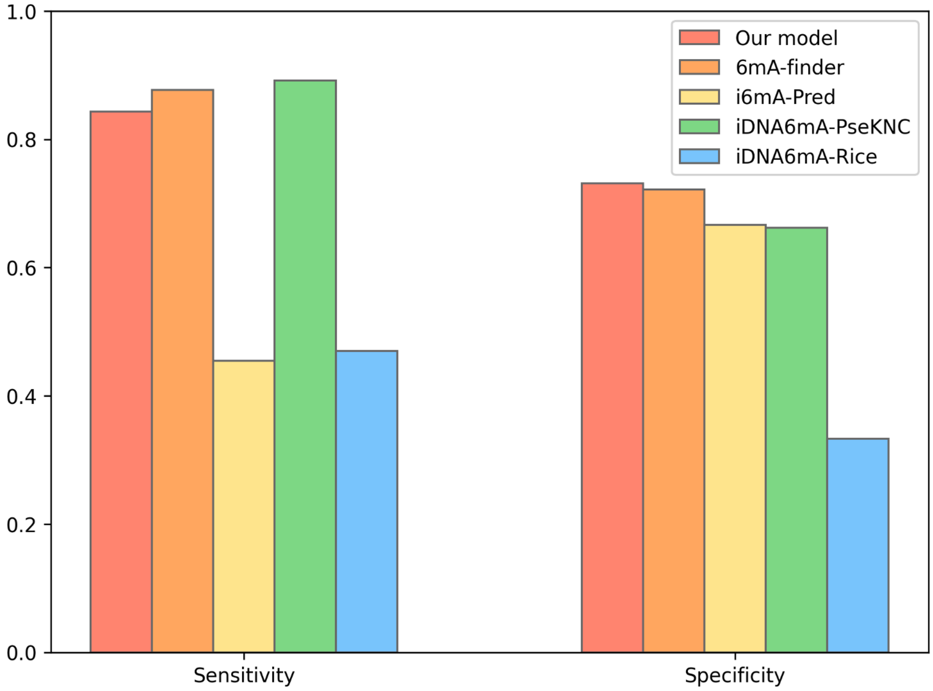


Fig. 4. Performance comparison among different 6 mA predictors in independent dataset. The proposed model had a slightly lower performance than 6 mA-finder and iDNA6mA-PseKNC, but it was superior to all predictors in terms of specificity.

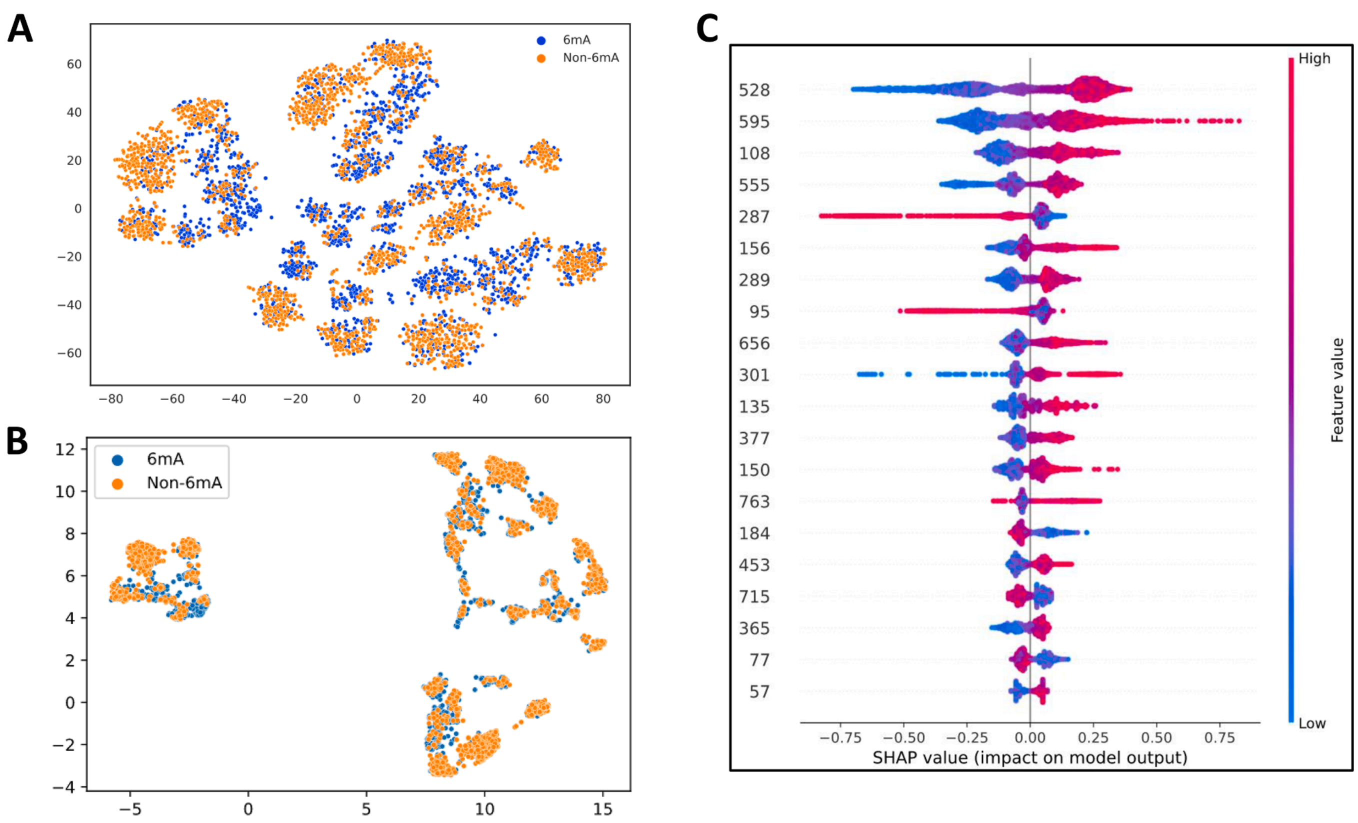


Fig. 5. BERT feature representation. (A) t-SNE analysis, (B) UMAP analysis, (C) SHAP analysis.

CRediT authorship contribution statement

Nguyen Quoc Khanh Le: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Funding acquisition. **Quang-Thai Ho:** Conceptualization, Methodology, Software, Formal analysis, Data curation, Visualization.

Declaration of Competing Interest

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

References

[1] L. He, H. Li, A. Wu, Y. Peng, G. Shu, G. Yin, Functions of N6-methyladenosine and its role in cancer, Mol. Cancer 18 (1) (2019), https://doi.org/10.1186/s12943-019-1109-9.

[2] Q. Xie, et al., N6-methyladenine DNA modification in glioblastoma, Cell 175 (5) (2018) 1228–1243.e20.

[3] Q.-L. Wan, X. Meng, W. Dai, Z. Luo, C. Wang, X. Fu, J. Yang, Q. Ye, Q. Zhou, N6-methyldeoxyadenine and histone methylation mediate transgenerational survival advantages induced by hormetic heat stress, Sci. Adv. 7 (1) (2021), https://doi.org/10.1126/sciadv.abc3026.

[4] Y.e. Guo, Y. Pei, K. Li, W. Cui, D. Zhang, DNA N6-methyladenine modification in hypertension, Aging 12 (7) (2020) 6276–6291.

[5] A.B.R. McIntyre, N. Alexander, K. Grigorev, D. Bezdan, H. Sichtig, C.Y. Chiu, C. E. Mason, Single-molecule sequencing detection of N6-methyladenine in microbial reference materials, Nat. Commun. 10 (1) (2019), https://doi.org/10.1038/s41467-019-08289-9.

[6] A. Tourancheau, E.A. Mead, X.-S. Zhang, G. Fang, Discovering multiple types of DNA methylation from bacteria and microbiome using nanopore sequencing, Nat. Methods 18 (5) (2021) 491–498.

[7] G.-Z. Luo, F. Wang, X. Weng, K. Chen, Z. Hao, M. Yu, X. Deng, J. Liu, C. He, Characterization of eukaryotic DNA N6-methyladenine by a highly sensitive restriction enzyme-assisted sequencing, Nat. Commun. 7 (1) (2016), https://doi.org/10.1038/ncomms11301.

[8] W. Chen, H. Lv, F. Nie, H. Lin, J. Hancock, i6mA-Pred: identifying DNA N6-methyladenine sites in the rice genome, Bioinformatics 35 (16) (2019) 2796–2800.

[9] N.Q.K. Le, iN6-methylat (5-step): identifying DNA N6-methyladenine sites in rice genome using continuous bag of nucleobases via Chou's 5-step rule, Mol. Genet. Genomics 294 (5) (2019) 1173–1182.

[10] C. Zhou, C. Wang, H. Liu, Q. Zhou, Q. Liu, Y. Guo, T. Peng, J. Song, J. Zhang, L. Chen, Y.u. Zhao, Z. Zeng, D.-X. Zhou, Identification and analysis of adenine N6-methylation sites in the rice genome, Nat. Plants 4 (8) (2018) 554–563.

[11] M. Tahir, H. Tayara, K.T. Chong, iDNA6mA (5-step rule): Identification of DNA N6-methyladenine sites in the rice genome by intelligent computational model via Chou's 5-step rule, Chemomet. Intell. Lab. Syst. 189 (2019) 96–101.

[12] H. Yu, Z. Dai, SNNRice6mA: A deep learning method for predicting DNA N6-methyladenine sites in rice genome, Front. Genet. 10 (2019) 1071.

[13] Z. Lv, H. Ding, L. Wang, Q. Zou, A convolutional neural network using dinucleotide one-hot encoder for identifying DNA N6-methyladenine sites in the rice genome, Neurocomputing 422 (2021) 214–221.

[14] C.R. Rahman, R. Amin, S. Shatabda, M.S.I. Toaha, A convolution based computational approach towards DNA N6-methyladenine site identification and motif extraction in rice genome, Sci. Rep. 11 (1) (2021), https://doi.org/10.1038/s41598-021-89850-9.

[15] H. Xu, R. Hu, P. Jia, Z. Zhao, P. Luigi Martelli, 6mA-Finder: a novel online tool for predicting DNA N6-methyladenine sites in genomes, Bioinformatics 36 (10) (2020) 3257–3259.

[16] Z. Abbas, H. Tayara, K.t. Chong, SpineNet-6mA: A novel deep learning tool for predicting DNA N6-methyladenine sites in genomes, IEEE Access 8 (2020) 201450–201457.

[17] M.M. Hasan, S. Basith, M.S. Khatun, G. Lee, B. Manavalan, H. Kurata, Meta-i6mA: an interspecies predictor for identifying DNA N6-methyladenine sites of plant genomes by exploiting informative features in an integrative machine-learning framework, Briefings Bioinf. 22 (3) (2021), https://doi.org/10.1093/bib/bbaa202.

[18] J. Chen, Q. Zou, J. Li, DeepM6ASeq-EL: prediction of human N6-methyladenosine (m6A) sites with LSTM and ensemble learning, Front. Comput. Sci. 16 (2) (2021), 162302.

[19] Z. Abbas, H. Tayara, Q. Zou, K.T. Chong, TS-m6A-DL: Tissue-specific identification of N6-methyladenosine sites using a universal deep learning model, Comput. Struct. Biotechnol. J. 19 (2021) 4619–4625.

[20] J. Li, S. He, F. Guo, Q. Zou, HSM6AP: a high-precision predictor for the Homo sapiens N6-methyladenosine (mˆ6 A) based on multiple weights and feature stitching, RNA Biol. 18 (11) (2021) 1882–1892.

[21] Vaswani, A., et al. *Attention is all you need.* in *Advances in neural information processing systems*. 2017.

[22] O. Levy, Y. Goldberg, Neural word embedding as implicit matrix factorization, Adv. Neural Inform. Process. Syst. 27 (2014) 2177–2185.

[23] N.Q.K. Le, Q.-T. Ho, T.-T.-D. Nguyen, Y.-Y. Ou, A transformer architecture based on BERT and 2D convolutional neural network to identify DNA enhancers from sequence information, Briefings Bioinf. 22 (5) (2021), https://doi.org/10.1093/bib/bbab005.

[24] H. Zulfiqar, Z.-J. Sun, Q.-L. Huang, S.-S. Yuan, H. Lv, F.-Y. Dao, H. Lin, Y.-W. Li, Deep-4mCW2V: A sequence-based predictor to identify N4-methylcytosine sites in Escherichia coli, Methods (2021), https://doi.org/10.1016/j.ymeth.2021.07.011.

[25] Devlin, J., et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* in *NAACL-HLT (1)*. 2019.

[26] Y. Sha, C. Ma, X. Wei, Y. Liu, Y.u. Chen, L. Li, DeepSADPr: A hybrid-learning architecture for serine ADP-ribosylation site prediction, Methods (2021), https://doi.org/10.1016/j.ymeth.2021.09.008.

[27] N.Q.K. Le, Q.-T. Ho, E.K.Y. Yapp, Y.-Y. Ou, H.-Y. Yeh, DeepETC: A deep convolutional neural network architecture for investigating and classifying electron transport chain's complexes, Neurocomputing 375 (2020) 71–79.

[28] Q. Zou, Latest machine learning techniques for biomedicine and bioinformatics, Curr. Bioinform. 14 (3) (2019) 176–177.

[29] P. Feng, H. Yang, H. Ding, H. Lin, W. Chen, K.-C. Chou, iDNA6mA-PseKNC: Identifying DNA N6-methyladenosine sites by incorporating nucleotide physicochemical properties into PseKNC, Genomics 111 (1) (2019) 96–102.

[30] C. Pian, et al., MM-6mAPred: identifying DNA N6-methyladenine sites based on Markov model, Bioinformatics 36 (2) (2020) 388–392.

[31] D.T. Do, T.Q.T. Le, N.Q.K. Le, Using deep neural networks and biological subwords to detect protein S-sulfenylation sites, Briefings Bioinf. 22 (3) (2021).

[32] M. Heinzinger, A. Elnaggar, Y.u. Wang, C. Dallago, D. Nechaev, F. Matthes, B. Rost, Modeling aspects of the language of life through transfer-learning protein sequences, BMC Bioinf. 20 (1) (2019), https://doi.org/10.1186/s12859-019-3220-8.

[33] H. Lv, F.-Y. Dao, Z.-X. Guan, D. Zhang, J.-X. Tan, Y. Zhang, W. Chen, H. Lin, iDNA6mA-Rice: A computational tool for detecting N6-methyladenine sites in rice, Front. Genet. 10 (2019), https://doi.org/10.3389/fgene.2019.00793.

[34] S. Basith, B. Manavalan, T.H. Shin, G. Lee, SDM6A: A web-based integrative machine-learning framework for predicting 6mA sites in the rice genome, Mol. Ther. Nucleic Acids 18 (2019) 131–141.

[35] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Machine Learn. Res. 9 (11) (2008).

[36] E. Becht, L. McInnes, J. Healy, C.-A. Dutertre, I.W.H. Kwok, L.G. Ng, F. Ginhoux, E. W. Newell, Dimensionality reduction for visualizing single-cell data using UMAP, Nat. Biotechnol. 37 (1) (2019) 38–44.

[37] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions. Proceedings of the 31st International Conference On Neural Information Processing Systems, 2017.