Computational and Structural Biotechnology Journal 21 (2023) 3045–3053

Contents lists available at ScienceDirect

Computational and Structural Biotechnology Journal

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

i4mC-GRU: Identifying DNA N^4-Methylcytosine sites in mouse genomes using bidirectional gated recurrent unit and sequence-embedded features

Thanh-Hoang Nguyen-Vo [a,f], Quang H. Trinh [b], Loc Nguyen [a], Phuong-Uyen Nguyen-Hoang [c], Susanto Rahardja [d,e,*], Binh P. Nguyen [a,**]

[a] School of Mathematics and Statistics, Victoria University of Wellington, Wellington 6140, New Zealand
[b] School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi 100000, Vietnam
[c] Computational Biology Center, International University - VNU HCMC, Ho Chi Minh City 700000, Vietnam
[d] School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China
[e] Infocomm Technology Cluster, Singapore Institute of Technology, Singapore 138683, Singapore
[f] School of Innovation, Design and Technology, Wellington Institute of Technology, Wellington 5012, New Zealand

ARTICLE INFO

Article history:
Received 3 August 2022
Received in revised form 12 May 2023
Accepted 12 May 2023
Available online 16 May 2023

Keywords:
DNA
N^4-methylcytosine
Epigenetics
Deep learning
Bidirectional gated recurrent unit
Sequence-embedded features

ABSTRACT

N^4-methylcytosine (4mC) is one of the most common DNA methylation modifications found in both prokaryotic and eukaryotic genomes. Since the 4mC has various essential biological roles, determining its location helps reveal unexplored physiological and pathological pathways. In this study, we propose an effective computational method called i4mC-GRU using a gated recurrent unit and duplet sequence-embedded features to predict potential 4mC sites in mouse (*Mus musculus*) genomes. To fairly assess the performance of the model, we compared our method with several state-of-the-art methods using two different benchmark datasets. Our results showed that i4mC-GRU achieved area under the receiver operating characteristic curve values of 0.97 and 0.89 and area under the precision-recall curve values of 0.98 and 0.90 on the first and second benchmark datasets, respectively. Briefly, our method outperformed existing methods in predicting 4mC sites in mouse genomes. Also, we deployed i4mC-GRU as an online web server, supporting users in genomics studies.

1. Introduction

Epigenetics studies the environmental impacts on gene's functions. Although epigenetic alterations are reversible and do not affect DNA sequences, they may change how the body interprets them [1]. DNA methylation, one of the epigenetic events, is a group of reactions that play vital roles in various biological processes, including chromatin modification, DNA stabilization, gene expression regulation, DNA conformation, X chromosome inactivation, cellular differentiation, cancer progression, imprinting, and epigenetic memory [2–4]. N^4-methylcytosine (4mC) is one of the most common DNA methylation modifications discovered in both prokaryotic and eukaryotic genomes, besides N^6-methyladenine and N^5-methylcytosine [5–7]. 4mC is catalyzed by the N^4 cytosine-specific DNA methyltransferase (DNMT) to create a new bond connecting a methyl group to the amino group at the C^4 position of cytosine. 4mC has also been described as part of the restriction-modification system that protects its DNA sequences from restriction enzyme-mediated degradation [8–10]. Several studies have demonstrated essential biological roles of 4mC in DNA replication and repair, the cell cycle, controlling gene expression levels, epigenetic inheritance, genome stabilization, recombination, and evolution [11–13]. Since the small region of 4mC in the eukaryote genomes prevents it from being well detected, the understanding of its biological functions is still limited [14].

To determine DNA 4mC sites, large-scale experimental studies have been performed on the whole genome using chemical and biomolecular assays, including single-molecule real-time (SMRT) sequencing [15], mass spectrometry [16], 4mC-Tet-assisted bisulfite

* Corresponding author at: School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China.
** Corresponding author.
E-mail addresses: susantorahardja@ieee.org (S. Rahardja), binh.p.nguyen@vuw.ac.nz (B.P. Nguyen).

https://doi.org/10.1016/j.csbj.2023.05.014

sequencing [17], and methylation-precise polymerase chain reaction [18]. Although these experimental approaches are usually costly and time-consuming, their experimentally verified results provide valuable sources for computational biologists to develop *in silico* approaches to assist experimental biologists in determining 4mC sites in DNA sequences. In recent years, multiple *in silico* methods using machine learning and deep learning have been proposed to address various biological issues [19–22,23,24]. Several computational methods have been released to identify 4mC sites in DNA sequences. iDNA4mC [25], proposed by Chen *et al.*, was developed by using support vector machines (SVM) incorporated with chemical characteristics and nucleotide occurrence frequency as input features. 4mCPred [26], introduced by He *et al.*, is a different SVM-based method that extracts the position-specific trinucleotide propensity and electron-ion interaction potentials of sequences as input features. 4mcPred-SVM [27] by Wei *et al.* was another SVM-based model constructed using four feature coding schemes in combination with two-step feature optimizations, showing a slight improvement in performance compared to previous methods. Wei *et al.* also released another model, 4mcPred-IFL [28], to predict 4mC sites by applying an iterative feature representation method to learn probabilistic features from various sequential models and improve feature representation in a supervised and iterative manner. Most recently, Tang *et al.* developed DNA4mC-LIP [29] that linearly integrated all the aforementioned methods for the prediction of 4mC sites. Besides models developed using classic machine learning algorithms, multiple deep learning models were developed to identify 4mC sites [30–33,34,35]. Despite the fact that these approaches yielded satisfactory outcomes, there is still much room for improvement.

In this study, we propose a more effective computational model called i4mC-GRU to predict potential 4mC sites in mouse genomes using a bidirectional gated recurrent unit (Bi-GRU) combined with duplet sequence-embedded features. The utilization of the Bi-GRU architecture on the sequence-embedded features supports the model to efficiently learn important information in both forward and reverse directions with an elevated training speed compared to other classic machine learning algorithms. The idea of sequence embedding evolved from the method of word embedding [36] in order to exploit the serial information of biological sequences defined by the order of appearance of nucleic acids [37,38]. Our modeling experiments were conducted using two independent benchmark datasets. The first benchmark dataset has sequence samples collected from the MethSMRT database [39] and carefully curated. The second benchmark dataset was collected from Manavalan et al.'s study [40]. Both benchmark datasets were confirmed to contain no duplications and highly similar sequences. To fairly examine the performance of i4mC-GRU, our model was compared with five other methods: iDNA4mC [25], 4mCpred-EL [40], 4mCPred-CNN [41], MSNET-4mC [35], and Deep-4mCGP [42].

2. Materials and methods

2.1. Benchmark datasets

To demonstrate the effectiveness of the proposed method, we conducted modeling experiments using two independent benchmark datasets. The pros and cons of these datasets are discussed in *Section 3. Results and Discussion*. Both benchmark datasets are derived from the MethSMRT database [39].

2.1.1. Dataset A

To create the first benchmark dataset (Dataset A), we collected sequence samples from the MethSMRT database [39]. All collected sequences have their lengths fixed at 41 bp, with the Cytosine located at the central position. The CD-HIT software [43] with a sequence identity cut-off of 0.8 was used to exclude similar sequences. The refined dataset included 7576 sequences after removing sequences with high similarities. The training, validation, and test sets account for 70%, 15%, and 15% of the total refined sequences, respectively. It is worth noting that all sequences in the refined dataset are positive sequences, with Cytosine (*C*) positioned at the center of each sequence. To create pseudo-negative sequences for model development and evaluation, we applied truncation and padding.

The truncation process extracts pseudo-negative samples from the original positive sequences. The truncation's sliding window searches for Cytosine, places it in the window's center, and extends equidistantly to both sides. Since the extracted pseudo-negative sequences were shorter than the original positive sequences, the padding process was employed to equalize their lengths. After passing truncation and padding processes, both positive and negative sequences have a central *C* and lengths of 41 bp. Positive samples (sequences) are signified as 'C^*-sequences', while negative samples are signified as '*C*-sequences'. One original positive sample can generate multiple pseudo-negative samples. The processing steps for extracting C^*-sequence and *C*-sequence samples are explained in the next two subsections. Table 1 gives information about the original sequences and the processed samples.

2.1.2. Sequence truncation

This stage is to create pseudo-negative sequences from positive sequences. Since all the collected sequences were positive samples with Cytosine (C) located at the central position, we generated pseudo-negative sequences. A window of 41 bp was applied to these sequences to detect any invalid samples (not having C at the central position). The number of positive samples was equal to the number of valid sequences in each dataset. To obtain pseudo-negative sequences, a 35-bp window was applied to read along all the valid sequences. This window slides in both directions and adopts any C (at its center) until the window arrives at the terminal nucleic acids. After truncation, 35 bp-length pseudo-negative sequences were created. To equalize the sequence lengths between the two classes, the pseudo-negative sequences were padded by three nucleic acids at the ends of both sides. Fig. 1 describes the key steps in extracting pseudo-negative sequences.

2.1.3. Sequence padding

This stage is to equalize the lengths of positive and pseudo-negative sequences. To perform padding, we created a lookup table based on the frequency of appearance of nucleic acid triplet sets (or, more colloquially, "triplet sets") at both terminals of the positive sequences. Only positive sequences in the training set were involved in constructing the lookup table. Initially, the 41 bp-length positive samples were first assumed to be 35 bp-length sequences (yellow-highlighted), so-called "sequences_{temp}". A sequence_{temp} has its outermost nucleic acid duplet sets (or, shortly, "duplet sets") located at positions 4–5 (blue-bounded) and positions 37–38 of the original sequences, while triplet sets located at positions 1–3 (blue-highlighted) and positions 39–41 (purple-highlighted) were assumed to be padding values. For a duplet set, the occurring frequencies of all corresponding triplet sets were computed. The lookup table was

Table 1
Benchmark dataset A used for model training and evaluation.

Dataset	Number of samples			
	Sequence	Positive	Pseudo-negative	Total
Training set	5304	5304	5816	11120
Validation set	1136	1136	1247	2383
Test set	1136	1136	1209	2345
Total	7576	7576	8272	15848

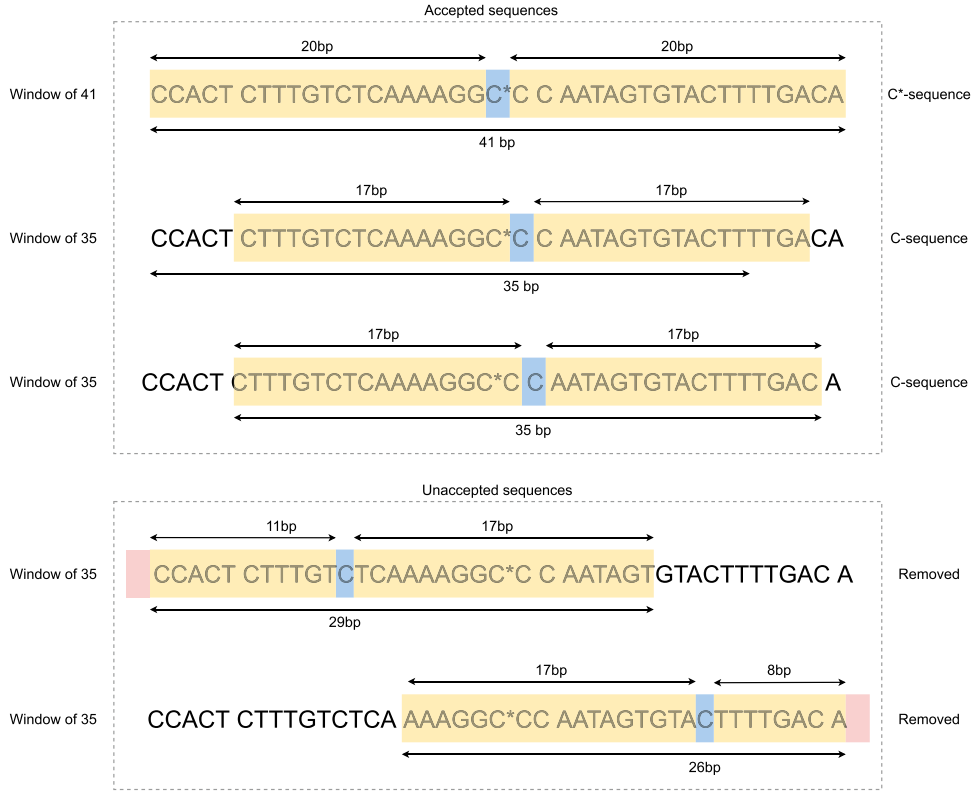


Fig. 1. The truncation process of the original sequences. 41-bp and 35-bp windows were used to truncate the DNA sequences. The 41-bp window fits the original positive sequences, with 4mC sites located at the central position. The 35-bp window slides along the positive sequences to capture other C-positions until it reaches the terminals of the sequences on both sides.

specified by many duplet set - triplet set pairs as '*keys*' for querying and '*values*' for padding, respectively. A single duplet key might have up to 64 corresponding triplet sets, but only five triplet sets with the greatest appearing frequencies were set as padding '*values*'. Consequently, each duplet '*key*' was associated with five triplets '*values*'. Fig. 2 describes the padding steps to rebalance the length of all negative sequences.

The padding process was performed using the created lookup table. The outermost duplet '*keys*' of all pseudo-negative samples in the three datasets (training, validation, and test sets) were determined. The corresponding triplet '*values*' were selected and padded to both sides afterwards. We examined three padding types: *(i)* randomly padding any possible triplet combinations of A, T, G, and C; *(ii)* randomly padding one in five triplets '*values*' with equal probabilities; and *(iii)* randomly padding one in five triplets '*values*' with probabilities weighted by appearing frequencies. Empirical results reveal that the model trained with samples padded by padding method *(i)* works less effectively compared to those trained with samples padded by padding methods *(ii)* and *(iii)*. Models trained by padding methods *(ii)* and *(iii)* perform almost equally. Therefore, we selected the padding method *(iii)* for model development. After sequence padding was completed, the positive and pseudo-negative sequences had the same length of 41 bp.

2.1.4. Dataset B

The second benchmark dataset (Dataset B) was collected from Manavalan et al.'s study [40]. This dataset was used for developing 4mCpred-EL [40], i4mC-Mouse [44], and 4mCPred-CNN [41]. This dataset was also constructed from sequence samples retrieved from the MethSMRT database [39] like benchmark dataset A. Characteristics of original sequences used for both benchmark datasets are identical with lengths of 41 bp. According to the dataset description in Manavalan et al.'s study, they removed similar positive sequences using the CD-HIT software [43] with a sequence identity cut-off of 0.7 to obtain a refined set of positive sequence samples having 4mC sites located in the central position. To make a balanced dataset, the same number of negative sequences were randomly extracted from non-4mC sequences. The dataset was then divided into a training set of 1492 samples and an independent test set of 320 samples. To develop deep learning models, about 15% of the refined training set was used as a validation set. Table 2 gives information about the benchmark dataset B.

2.2. Sequence-embedded features

An index table containing indices of duplet sets of consecutive nucleic acids was first built. A window of 2 slid along the whole sequence, starting at the first nucleic acid and ending when arriving at the final one. A 41 bp-length sequence was first converted into a list of 40 triplet keys. Each triplet key was then matched with the index table to obtain its corresponding index, and this index replaced the triplet key. Finally, the index vector of size 1×40, defined by a series of distinctly ordered indices, was formed. The index vector is the input of our model. Fig. 3 highlights the main steps of converting sequence samples into index vectors.

2.3. Model architecture

Index vectors of size 1×40 are the input data of the model. The triplet '*value*' for one side is independent of that for the other side. The Adam optimization algorithm [45] was applied to every mini-batch of 64 samples with a learning rate of 1×10^{-4}. Fig. 4 describes the model architecture used in our study. Before being transferred to the batch normalization (BatchNorm) layer, the input data are passed through the embedding layer with an embedding size of 64 to generate embedding matrices of size 40×64. The normalized

Fig. 2. The padding process of truncated sequences. Initially, the 41 bp-length positive samples were assumed to be 35 bp-length sequences (yellow-highlighted) so-called sequences$_{temp}$. All sequences$_{temp}$ have two outermost duplet '*key*' at both sides (blue-bounded and purple-bounded). For each duplet '*key*', occurring frequencies of its corresponding triplet '*values*' (blue-highlighted and purple-highlighted) were computed (on positive samples of the training set only). For a single duplet '*key*', top-5 triplets '*values*' were selected for padding. The lookup table was created by 16 × 5 = 80 duplet '*key*' - triplet '*value*' pairs. To pad a 35 bp-length pseudo-negative sequence, the sequence's outermost duplet '*keys*' is first determined and fetched to the lookup table to retrieve the corresponding triplet '*values*' padded to both sides of the sequence.

Table 2
Benchmark dataset B used for model training and evaluation.

Dataset	Number of samples		
	Positive	Negative	Total
Training set	746	746	1492
Test set	160	160	320
Total	906	906	1812

matrices then enter the bidirectional gated recurrent unit (Bi-GRU) layer, defined with two units and a hidden dimension of 128. The Bi-GRU layer converts normalized matrices of size 40 × 64 into matrices of size 40 × 256 (40 × 128 × 2 directions). After passing through the second Bi-GRU layer, these matrices are then flattened and activated by the Leaky Rectified Linear Unit (LeakyReLU) function before entering the first fully connected (FC1) layer. Getting out of the FC1 layer, vectors of size 1 × 10240 are transformed into vectors of size 1 × 128, and eventually activated by the sigmoid function to return the probabilities. The loss function used is the binary cross-entropy:

$$Loss = \sum_{i=1}^{n} y_i \times \log \hat{y}_i + (1 - y_i) \times \log(1 - \hat{y}_i), \quad (1)$$

where y is the true label and $\hat{y}$ is the predicted probability.

Our models were implemented using PyTorch 1.3.1 and trained on Google Colab equipped with 25 GB of RAM and one NVIDIA Tesla T4 GPU over 25 epochs. It took about 4.5 s and 0.25 s to finish one training epoch and testing a model, respectively. The prediction threshold was set at the default of 0.5. The model at the epoch where the validation loss was minimum was selected as the optimal model. In our experiments, the model converged at epoch 17.

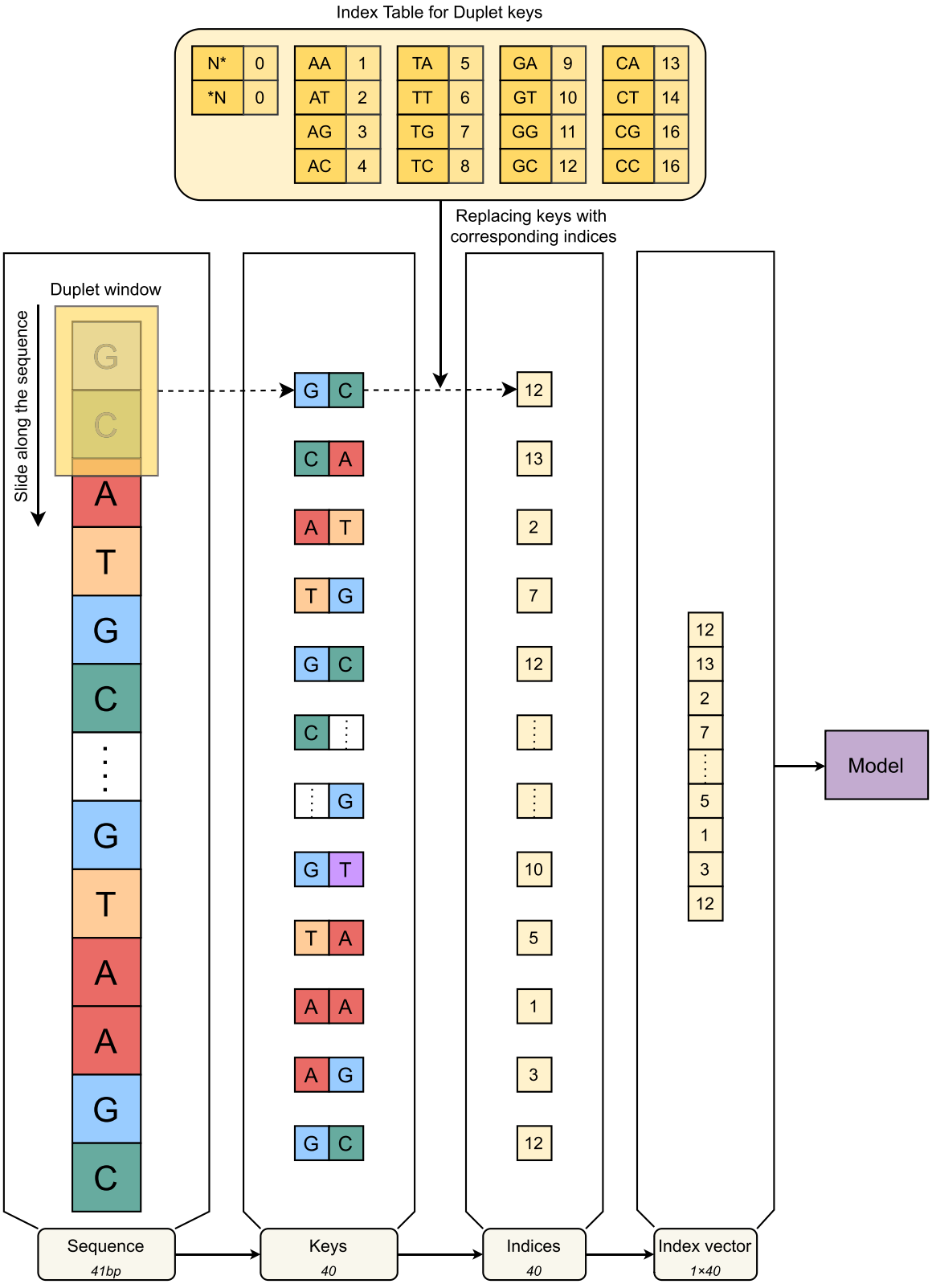


Fig. 3. Processing steps for converting sequence samples into index vectors. First, a duplet window slides along the sequences to extract all duplet keys (in order) to create key vectors. Each duplet key is assigned a specific index. From the key vectors, index vectors comprising all converted indices are created. The index vectors of size 1 × 40 are the model input.

2.4. Evaluation metrics

To assess the performance of the model, several metrics, including the balanced accuracy (BA), sensitivity (SN), specificity (SP), precision (PR), Matthews's correlation coefficient (MCC), the area under the receiver operating characteristic (ROC) curve (AUCROC), and the area under the precision-recall (PR) curve (AUCPR), were measured. TP, FP, TN, and FN are abbreviations for True Positives,

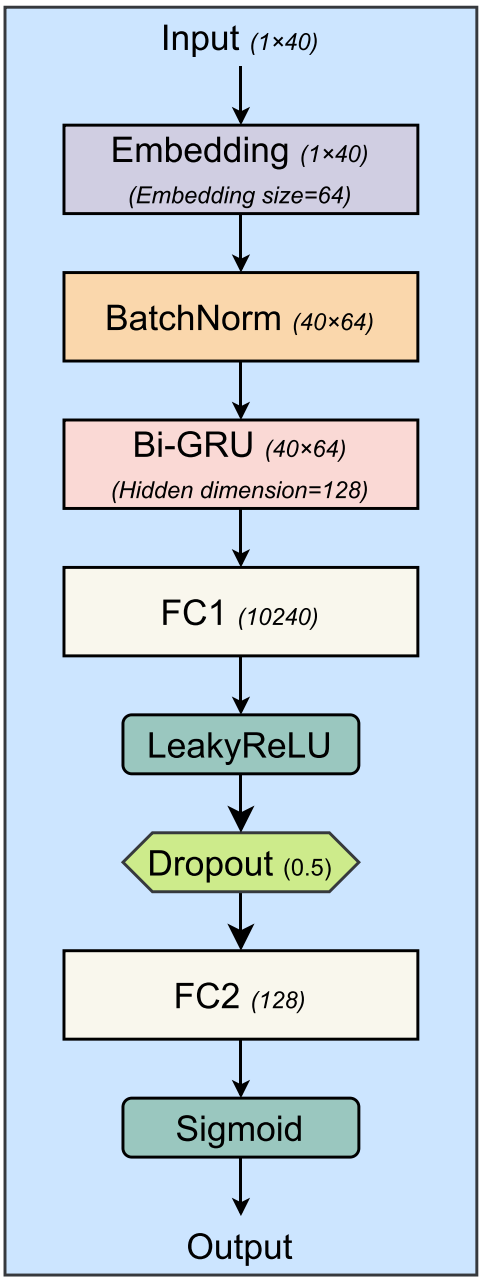


Fig. 4. Model architecture.

False Positives, True Negatives, and False Negatives, respectively. The mathematical formulas for these metrics are expressed below.

$$SN = \frac{TP}{TP + FN}, \tag{2}$$

$$SP = \frac{TN}{TN + FP}, \tag{3}$$

$$PR = \frac{TP}{TP + FP}, \tag{4}$$

$$BA = \frac{SN + SP}{2}, \tag{5}$$

$$F1 = 2 \times \frac{PR \times SN}{PR + SN}, \tag{6}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{7}$$

3. Results and discussion

3.1. Comparative analysis

Our method was compared to five other methods, including iDNA4mC [25], 4mCpred-EL [40], 4mCPred-CNN [41], MSNET-4mC [35], and Deep-4mCGP [42]. The AUCROC and AUCPR are two decisive metrics used for model assessment. iDNA4mC and 4mCpred-EL are machine learning-based methods while 4mCPred-CNN, MSNET-4mC, and Deep-4mCGP are deep learning-based methods. These methods were reimplemented based on their model descriptions and training conditions. These five methods and ours were trained, validated, and tested using the same training, validation, and test sets, respectively. Table 3 provides the evaluated model performances of all methods on the independent test sets (of benchmark datasets A and B).

Comparative results indicate that i4mC-GRU obtains better performance compared to other methods with AUCROC values of about 0.97 and 0.89 and AUCPR values of about 0.98 and 0.90 on the test sets A and B, respectively. The performance of 4mCPred-CNN, a simple deep learning model, is only slightly lower than that of i4mC-GRU, but it still works better than these two other machine learning-based methods. While MSNET-4mC and Deep-4mCGP, two other deep learning models, work less effectively than other methods. For the benchmark dataset A, the balanced accuracy, precision, MCC value, and F1 score of i4mC-GRU are also higher than those of other methods, followed by 4mCPred-CNN, 4mCpred-EL, iDNA4mC, MSNET-4mC, and Deep-4mCGP. The sensitivities of i4mC-GRU and 4mCPred-CNN are equivalent and both higher than those of the

Table 3
The performance of i4mC-GRU and other methods on the independent test set.

Dataset	Method	AUCROC	AUCPR	BA	SN	SP	PR	MCC	Fl
A	iDNA4mC	0.9151	0.9188	0.8296	0.8089	0.8504	0.807	0.659 1	0.8281
	4mCpred-EL	0.9368	0.9418	0.8369	0.8147	0.8592	0.8133	0.6737	0.8356
	4mCPred-CNN	0.9418	0.9507	0.8710	**0.8908**	0.8512	0.8799	0.7432	0.8653
	MSNET-4mC	0.8115	0.8222	0.728	0.7298	0.7262	0.7147	0.4558	0.7222
	Deep-4mCGP	0.7650	0.7624	0.7008	0.6250	0.7767	0.7245	0.4070	0.6711
	i4mC-GRU (ours)	**0.9732**	**0.9788**	**0.9289**	**0.8908**	**0.9669**	**0.9620**	**0.8619**	**0.9250**
B	iDNA4mC	0.8179	0.8164	0.7188	0.6688	0.7688	0.7431	0.4397	0.7039
	4mCpred-EL	0.8578	0.8648	**0.7875**	0.7312	**0.8438**	**0.8239**	**0.5787**	0.7748
	4mCPred-CNN	0.7953	0.8049	0.7188	0.6313	0.8063	0.7652	0.4444	0.6918
	MSNET-4mC	0.8322	0.8339	0.7656	0.7688	0.7241	0.7640	0.5313	0.7664
	Deep-4mCGP	0.7181	0.7303	0.6375	0.6125	0.6625	0.6447	0.2753	0.6282
	i4mC-GRU (ours)	**0.8877**	**0.8993**	0.7812	**0.7812**	0.7812	0.7812	0.5625	**0.7812**

other methods. Our method achieves higher specificity compared to the others. For the benchmark dataset B, the performances of all models, including ours, are decreased, our model is still dominant over other methods. The difference in performance between these two benchmark datasets may be explained by the different data volume. With the achieved performances on the benchmark datasets A and B, i4mC-GRU proves its robustness, stability, and effectiveness in identifying 4mC sites.

The combination of an effective architecture design, suitable featurization, and a well-designed modeling strategy may partially explain the efficiency of the proposed method. iDNA4mC and 4mCpred-EL models were developed using conventional machine learning methods. While iDNA4mC was simply designed with Support Vector Machines, 4mCpred-EL was developed using an ensemble learning strategy with multiple learning algorithms and feature sets. Based on their design, 4mCpred-EL requires more effort in featurization, hyperparameter tuning, training, and evaluation compared to iDNA4mC. Both methods, however, become computationally demanding once the data volume expands. The 4mCPred-CNN model was constructed with a convolutional neural network (CNN), one of the most frequently used deep learning architectures to address many problems. The results suggest that the CNN-based deep learning model is also highly competitive. Nevertheless, the CNN-based models work more effectively in processing image-like data by extracting features with a focus on regional significance rather than serial data characterized by orders of occurrence. MSNET-4mC is a more advanced CNN-based model adapting with multi-scale receptive fields to capture more information but the data volume is a concern issue. The application of deeper layers with multiple receptive fields on the small set of data ineffectively leverages the power of those receptive fields. Deep-4mCGP is a hybrid deep learning model consisting of long short-term memory (LSTM), a recurrent neural network, and CNN. The LSTM layer is responsible for dealing with ordered biological sequences and the CNN layer is organized to extract the sequence's features. Our model architecture is designed with a gate recurrent unit (GRU) network, a lightweight recurrent neural network, to capture serial features of biological sequences Figs. 5, 6.

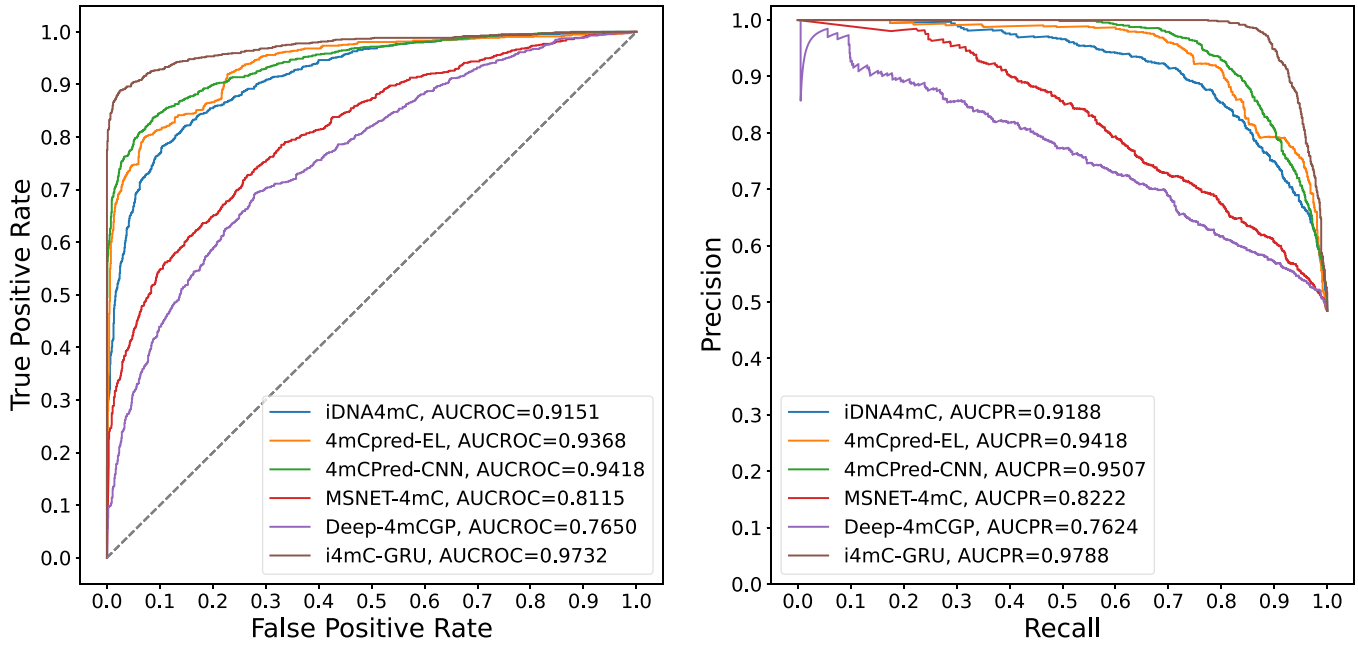
3.2. Limitations and future work

Not only is it challenge for our work, but also for other studies, to develop a quality negative set. There are a variety of techniques for generating decoy sequences in computational genomics research (i.e., pseudo-negative samples), for example, randomized matching generation [46], machine learning-aided selection [47], background sequence selection [48], and sequence recombination [48,49]. Each technique has its own characteristics, and the optimal strategy will depend on the specifics of the prediction task and the research objectives. In order to determine the most effective method for a particular situation, researchers may need to experiment with a variety of approaches. In the future, additional decoy generation techniques can be utilized to improve the prediction accuracy under a variety of conditions.

In our study, we used two benchmark datasets. The first benchmark dataset was specified with positive samples (which were collected and refined from the database) and pseudo-negative samples (which were created based on positive samples). The model developed on a dataset with the pseudo-negative samples may need to manage the false positive rate. While the second benchmark dataset contains positive samples (which were collected from the same sources as the first benchmark dataset) and negative samples (which were collected from other sources of verified non-4mC sequences). Since the verified non-4mC sequences are far different from 4mC sequences, the model developed on a dataset with highly distinct negative samples are not forced to learn important feature to distinguish between 4mC and non-4mC sites.

3.3. Software availability

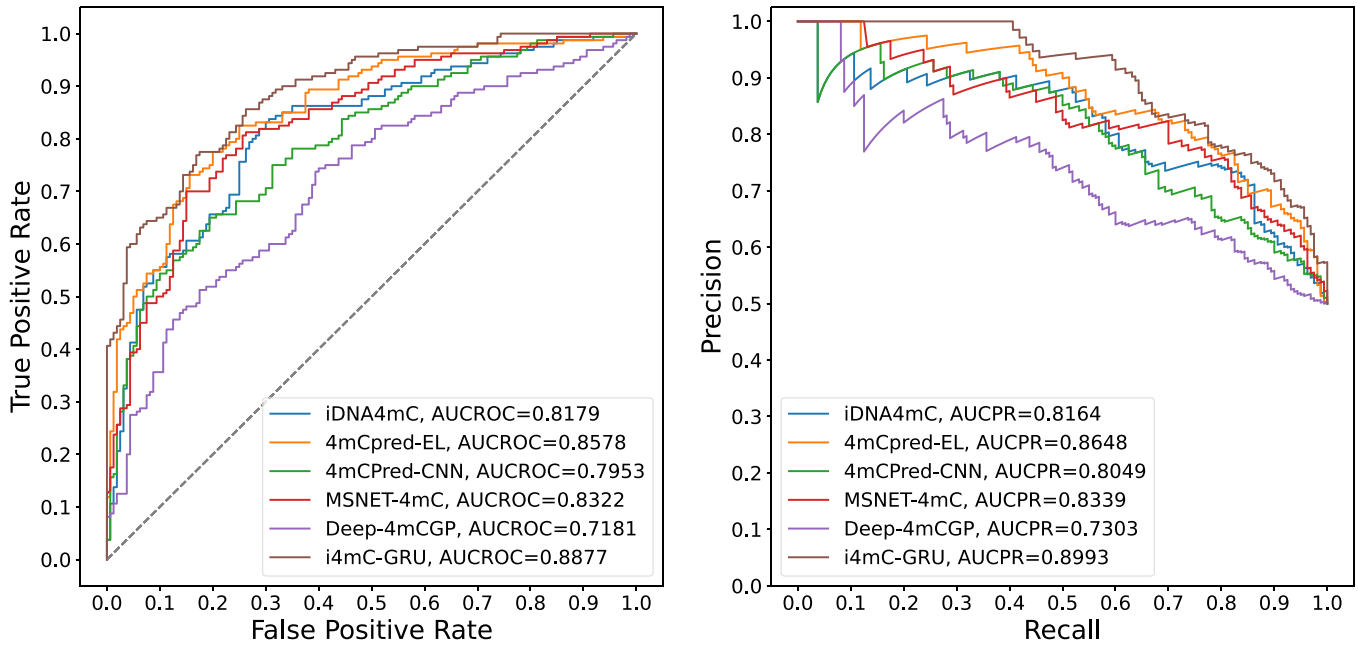
To support researchers in determining 4mC sites, a web server implementing i4mC-GRU was developed, and it can be accessed at https://github.com/mldlproject/2022-i4mC-GRU. The web server was designed with a user-friendly interface. i4mC-GRU assists users in locating potential 4mC sites in mouse genomes without the need for manual computation. Users can easily start their prediction tasks with i4mC-GRU by following the guidelines on the web server.



(a) ROC curves

(b) PR curves

Fig. 5. ROC and PR curves of i4mC-GRU and other methods on benchmark dataset A.



(a) ROC curves (b) PR curves

Fig. 6. ROC and PR curves of i4mC-GRU and other methods on benchmark dataset B.

4. Conclusions

In this study, we proposed i4mC-GRU, an efficient computational model using gated recurrent unit networks in combination with duplet sequence-embedded features, for predicting possible 4mC sites in mouse genomes. Our method achieved high AUCROC and AUCPR values in both benchmark datasets. Furthermore, i4mC-GRU was demonstrated to work better than other state-of-the-art methods based on a fair evaluation of the same independent test sets. Users can easily use our online web server to predict 4mC sites in mouse genomes.

CRediT authorship contribution statement

Thanh-Hoang Nguyen-Vo: conceptualization, methodology, data curation, investigation, formal analysis, validation, visualization, writing original draft, writing review & editing. **Quang H. Trinh**: investigation, software. **Loc Nguyen**: investigation, data curation. **Phuong-Uyen Nguyen-Hoang**: writing original draft, writing review & editing. **Susanto Rahardja**: formal analysis, writing review & editing, funding acquisition. **Binh P. Nguyen**: conceptualization, methodology, formal analysis, visualization, writing review & editing, supervision.

Declaration of Competing Interests

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 manuscript.

Acknowledgements

The work of T.-H. N.-V. was partly supported by the Whitireia and WelTec Contestable Fund.

References

[1] Delcuve GP, Rastegar M, Davie JR. Epigenetic control. J Cell Physiol 2009;219(2):243–50.
[2] He X-J, Chen T, Zhu J-K. Regulation and function of DNA methylation in plants and animals. Cell Res 2011;21(3):442–65. https://doi.org/10.1038/cr.2011.23
[3] Moore LD, Le T, Fan G. DNA methylation and its basic function. Neuropsychopharmacology 2013;38(1):23–38. https://doi.org/10.1038/npp.2012.112
[4] Schübeler D. Function and information content of DNA methylation. Nature 2015;517(7534):321–6. https://doi.org/10.1038/nature14192
[5] Korlach J, Turner SW. Going beyond five bases in DNA sequencing. Curr Opin Struct Biol 2012;22(3):251–61. https://doi.org/10.1016/j.sbi.2012.04.002
[6] Davis BM, Chao MC, Waldor MK. Entering the era of bacterial epigenomics with single molecule real time DNA sequencing. Curr Opin Microbiol 2013;16(2):192–8. https://doi.org/10.1016/j.mib.2013.01.011
[7] Roberts RJ, Vincze T, Posfai J, Macelis D. REBASE-a database for DNA restriction and modification: enzymes, genes and genomes. Nucleic Acids Res 2015;43(D1):D298–9. https://doi.org/10.1093/nar/gku1046
[8] Ehrlich M, Wilson GG, Kuo KC, Gehrke CW. N4-methylcytosine as a minor base in bacterial DNA. J Bacteriol 1987;169(3):939–43. https://doi.org/10.1128/jb.169.3.939-943.1987
[9] Morgan RD, Luyten YA, Johnson SA, Clough EM, Clark TA, Roberts RJ. Novel m4C modification in type I restriction-modification systems. Nucleic Acids Res 2016;44(19):9413–25. https://doi.org/10.1093/nar/gkw743
[10] Murray IA, Luyten YA, Fomenkov A, Dai N, Jr IRC, Farmerie WG, et al. Structural and functional diversity among Type III restriction-modification systems that confer host DNA protection via methylation of the N4 atom of cytosine. Plos One 2021;16(7):e0253267. https://doi.org/10.1371/journal.pone.0253267
[11] Glickman BW, Radman M. Escherichia coli mutator mutants deficient in methylation-instructed DNA mismatch correction. Proc Natl Acad Sci 1980;77(2):1063–7. https://doi.org/10.1073/pnas.77.2.1063
[12] Sánchez-Romero MA, Cota I, Casadesús J. DNA methylation in bacteria: from the methyl group to the methylome. Curr Opin Microbiol 2015;25:9–16. https://doi.org/10.1016/j.mib.2015.03.004
[13] Kumar S, Karmakar BC, Nagarajan D, Mukhopadhyay AK, Morgan RD, Rao DN. N4-cytoeisine dna methylation regulates transcription and pathogenesis in Helicobacter pylori. Nucleic Acids Res 2018;46(7):3429–45. https://doi.org/10.1093/nar/gky126
[14] Rathi P, Maurer S, Summerer D. Selective recognition of N4-methylcytosine in DNA by engineered transcription-activator-like effectors. Philos Trans R Soc B Biol Sci 2018;373(1748):20170078. https://doi.org/10.1098/rstb.2017.0078
[15] Flusberg BA, Webster DR, Lee JH, Travers KJ, Olivares EC, Clark TA, et al. Direct detection of DNA methylation during single-molecule real-time sequencing. Nat Methods 2010;7(6):461–5. https://doi.org/10.1038/nmeth.1459
[16] Boulias K, Greer EL. Detection of DNA Methylation in Genomic DNA by UHPLC-MS/MS. Springer,; 2021. https://doi.org/10.1007/978-1-0716-0876-0_7

[17] Doherty R, Couldrey C. Exploring genome wide bisulfite sequencing for DNA methylation analysis in livestock: a technical assessment. Front Genet 2014;5:126. https://doi.org/10.3389/fgene.2014.00126
[18] Buryanov YI, Shevchuk TV. DNA methyltransferases and structural-functional specificity of eukaryotic DNA modification. Biochemistry 2005;70(7):730–42. https://doi.org/10.1007/s10541-005-0178-0
[19] Chen W, Feng P, Ding H, Lin H, Chou K-C. iRNA-Methyl: Identifying N6-methyladenosine sites using pseudo nucleotide composition. Anal Biochem 2015;490:26–33. https://doi.org/10.1016/j.ab.2015.08.021
[20] Cheng X, Zhao S-G, Xiao X, Chou K-C. iATC-mHyb: a hybrid multi-label classifier for predicting the classification of anatomical therapeutic chemicals. Oncotarget 2017;8(35):58494. https://doi.org/10.18632/oncotarget.17028
[21] Nguyen QH, Nguyen-Vo T-H, Le NQK, Do TTT, Rahardja S, Nguyen BP. iEnhancer-ECNN: identifying enhancers and their strength using ensembles of convolutional neural networks. BMC Genom 2019;20(9):1–10. https://doi.org/10.1186/s12864-019-6336-3
[22] Nguyen BP, Nguyen QH, Doan-Ngoc G-N, Nguyen-Vo T-H, Rahardja S. iProDNA-CapsNet: identifying protein-DNA binding residues using capsule. Neural Netw BMC Bioinforma 2019;20(23):1–12. https://doi.org/10.1186/s12859-019-3295-2
[23] Feng P, Yang H, Ding H, Lin H, Chen W, Chou K-C. iDNA6mA-PseKNC: Identifying DNA N6-methyladenosine sites by incorporating nucleotide physicochemical properties into PseKNC. Genomics 2019;111(1):96–102. https://doi.org/10.1016/j.ygeno.2018.01.005
[24] Nguyen-Vo T-H, Nguyen QH, Do TTT, Nguyen T-N, Rahardja S, Nguyen BP. iPseU-NCP: Identifying RNA pseudouridine sites using random forest and NCP-encoded features. BMC Genom 2019;20(10):1–11. https://doi.org/10.1186/s12864-019-6357-y
[25] Chen W, Yang H, Feng P, Ding H, Lin H. iDNA4mC: identifying DNA N4-methylcytosine sites based on nucleotide chemical properties. Bioinformatics 2017;33(22):3518–23. https://doi.org/10.1093/bioinformatics/btx479
[26] He W, Jia C, Zou Q. 4mCPred: machine learning methods for DNA N4-methylcytosine sites prediction. Bioinformatics 2019;35(4):593–601. https://doi.org/10.1093/bioinformatics/bty668
[27] Wei L, Luan S, Nagai LAE, Su R, Zou Q. Exploring sequence-based features for the improved prediction of DNA N4-methylcytosine sites in multiple species. Bioinformatics 2019;35(8):1326–33. https://doi.org/10.1093/bioinformatics/bty824
[28] Wei L, Su R, Luan S, Liao Z, Manavalan B, Zou Q, et al. Iterative feature representations improve N4-methylcytosine site prediction. Bioinformatics 2019;35(23):4930–7. https://doi.org/10.1093/bioinformatics/btz408
[29] Tang Q, Kang J, Yuan J, Tang H, Li X, Lin H, et al. DNA4mC-LIP: a linear integration method to identify N4-methylcytosine site in multiple species. Bioinformatics 2020;36(11):3327–35. https://doi.org/10.1093/bioinformatics/btaa143
[30] Zeng R, Cheng S, Liao M. 4mCPred-MTL: accurate identification of DNA 4mC sites in multiple species using multi-task deep learning based on multi-head attention mechanism. Front Cell Dev Biol 2021;9:819. https://doi.org/10.3389/fcell.2021.664669
[31] Alam W, Tayara H, Chong KT. i4mC-Deep: an intelligent predictor of N4-methylcytosine sites using a deep learning approach with chemical properties. Genes 2021;12(8):1117. https://doi.org/10.3390/genes12081117
[32] Wahab A, Mahmoudi O, Kim J, Chong KT. DNC4mC-Deep: identification and analysis of DNA N4-methylcytosine sites based on different encoding schemes by using deep learning. Cells 2020;9(8):1756. https://doi.org/10.3390/cells9081756
[33] Xu H, Jia P, Zhao Z. Deep4mC: systematic assessment and computational prediction for DNA N4-methylcytosine sites by deep learning. Brief Bioinforma 2021;22(3):bbaa099. https://doi.org/10.1093/bib/bbaa099
[34] Liu Q, Chen J, Wang Y, Li S, Jia C, Song J, et al. DeepTorrent: a deep learning-based approach for predicting DNA N4-methylcytosine sites. Brief Bioinforma 2021;22(3):bbaa124. https://doi.org/10.1093/bib/bbaa124
[35] Liu C, Song J, Ogata H, Akutsu T. MSNet-4mC: learning effective multi-scale representations for identifying DNA N4-methylcytosine sites. Bioinformatics 2022;38(23):5160–7.
[36] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space (2013). 10.48550/ARXIV.1301.3781.
[37] Nguyen-Vo T-H, Nguyen L, Do N, Le PH, Nguyen T-N, Nguyen BP, et al. Predicting drug-induced liver injury using convolutional neural network and molecular fingerprint-embedded features. ACS Omega 2020;5(39):25432–9. https://doi.org/10.1021/acsomega.0c03866
[38] Nguyen-Vo T-H, Trinh QH, Nguyen L, Nguyen-Hoang P-U, Nguyen T-N, Nguyen DT, et al. iCYP-MFE: Identifying human cytochrome P450 inhibitors using multitask learning and molecular fingerprint-embedded encoding. J Chem Inf Model 2021. https://doi.org/10.1021/acs.jcim.1c00628
[39] Ye P, Luan Y, Chen K, Liu Y, Xiao C, Xie Z. MethSMRT: an integrative database for DNA N6-methyladenine and N4-methylcytosine generated by single-molecular real-time sequencing. Nucleic Acids Res 2016:gkw950. https://doi.org/10.1093/nar/gkw950
[40] Manavalan B, Basith S, Shin TH, Lee DY, Wei L, Lee G. 4mCpred-EL: an ensemble learning framework for identification of DNA N4-methylcytosine sites in the mouse genome. Cells 2019;8(11):1332. https://doi.org/10.3390/cells8111332
[41] Abbas Z, Tayara H, Chong KT. 4mCPred-CNN-prediction of DNA N4-methylcytosine in the mouse genome using a convolutional neural network. Genes 2021;12(2):296. https://doi.org/10.3390/genes12020296
[42] Zulfiqar H, Huang Q-L, Lv H, Sun Z-J, Dao F-Y, Lin H. Deep-4mCGP: a deep learning approach to predict 4mC sites in geobacter pickeringii by using correlation-based feature selection technique. Int J Mol Sci 2022;23(3):1251.
[43] Huang Y, Niu B, Gao Y, Fu L, Li W. CD-HIT Suite: a web server for clustering and comparing biological sequences. Bioinformatics 2010;26(5):680–2. https://doi.org/10.1093/bioinformatics/btq003
[44] Hasan MM, Manavalan B, Shoombuatong W, Khatun MS, Kurata H. i4mC-Mouse: improved identification of DNA N4-methylcytosine sites in the mouse genome using multiple encoding schemes. Comput Struct Biotechnol J 2020;18:906–12.
[45] D.P. Kingma, A method for stochastic optimization (2014). 10.48550/ARXIV.1412.6980.
[46] Caballero J, Smit AFA, Hood L, Glusman G. Realistic artificial DNA sequences as negative controls for computational genomics. Nucleic Acids Res 2014;42(12):e99. https://doi.org/10.1093/nar/gku356
[47] Akhter N, Chennupati G, Djidjev H, Shehu A. Decoy selection for protein structure prediction via extreme gradient boosting and ranking. BMC Bioinforma 2020;21(1):1–21. https://doi.org/10.1186/s12859-020-3523-9
[48] Krützfeldt L-M, Schubach M, Kircher M. The impact of different negative training data on regulatory sequence predictions. PloS One 2020;15(12):e0237412. https://doi.org/10.1371/journal.pone.0237412
[49] Nguyen-Vo T-H, Trinh QH, Nguyen L, Nguyen-Hoang P-U, Rahardja S, Nguyen BP. iPromoter-Seqvec: identifying promoters using bidirectional long short-term memory and sequence-embedded features. BMC Genom 2022;23(5):1–12. https://doi.org/10.1186/s12864-022-08829-6