



Sequence analysis

MSNet-4mC: Learning effective multi-scale representations for identifying DNA N4-methylcytosine sites

Chunting Liu^{1,2}, Jiangning Song^{3,4}, Hiroyuki Ogata² and Tatsuya Akutsu^{2,1*}

¹Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Kyoto, Kyoto 606-8501, Japan,

²Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji, Kyoto 611-0011, Japan, ³Monash Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, Monash University, Melbourne, VIC 3800, Australia and

⁴Monash Data Futures Institute, Monash University, Melbourne, VIC 3800, Australia

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: N4-methylcytosine (4mC) is an essential kind of epigenetic modification that regulates a wide range of biological processes. However, experimental methods for detecting 4mC sites are time-consuming and labor-intensive. As an alternative, computational methods that are capable of automatically identifying 4mC with data analysis techniques become a reasonable option. A major challenge is how to develop effective methods to fully exploit the complex interactions within the DNA sequences to improve the predictive capability.

Results: In this work, we propose MSNet-4mC, a lightweight neural network building upon convolutional operations with multi-scale receptive fields to perceive cross-element relationships over both short and long ranges of given DNA sequences. With strong imbalances in the number of candidates in different species in mind, we compute and apply class weights in the cross-entropy loss to balance the training process. Extensive benchmarking experiments show that our method achieves a significant performance improvement and outperforms other state-of-the-art methods.

Availability and Implementation: The source code and models are freely available for download at <https://github.com/LIU-CT/MSNet-4mC>, implemented in Python and supported on Linux and Windows.

Contact: liuchunting@kuicr.kyoto-u.ac.jp or takutsu@kuicr.kyoto-u.ac.jp

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

DNA methylation is an important type of epigenetic modification and involved in diverse biological processes, such as regulation of gene expression, regulation of chromatin organization, transposon silencing, or genomic imprinting across all domains of life (Arand *et al.*, 2012; Greenberg and Bourc'his, 2019; Jeudy *et al.*, 2020; Jones, 2012). 5-Methylcytosine (5mC), N6-methyladenine (6mA), and N4-methylcytosine (4mC) are the three prevalent kinds of DNA methylations in genomes (Davis *et al.*, 2013; Roberts *et al.*, 2015). Bisulfite sequencing technique is a commonly used technique for detecting DNA methylation sites across the whole genome. It can detect 5mC modifications but cannot be applied to identify 6mA and

4mC modifications (Barros-Silva *et al.*, 2018). The single-molecule real-time (SMRT) sequencing method is of great practical value as an effective approach for detecting all of these three modifications (Flusberg *et al.*, 2010). However, it is expensive, time-consuming, and labor-intensive, especially for large-scale detection experiments.

Machine learning and deep learning have witnessed remarkable achievements in the fields of computer vision, natural language processing, and textual reasoning in recent years, even surpassing human performance in a variety of scenarios (Cai *et al.*, 2019; Fu *et al.*, 2021; Guo *et al.*, 2022). Currently, they have been applied to extract the information from the biological sequences to explore and understand the underlying correlation. With respect to DNA 4mC site prediction, several computational methods

have been developed. A good performance has been achieved by applying traditional machine learning-based methods and more recent deep learning-based algorithms. Amongst these methods, iDNA4mC (Chen et al., 2017) is the first support vector machine based-predictor proposed by Chen et al. to predict the 4mC sites. The dataset was first built by Chen et al. and further screened against the MethSMRT database (Ye et al., 2016), which is the common dataset in the later studies. Several conventional machine learning-based methods have been proposed since then, including 4mCPred (He et al., 2019), 4mCPred-SVM (Wei et al., 2019a), Meta-4mCpred (Manavalan et al., 2019), and 4mCPred-IFL (Wei et al., 2019b). More recently, deep learning-based methods have been applied to address this task. To the best of our knowledge, 4mCCNN (Khanal et al., 2019) is the first deep learning-based approach for 4mC site prediction, which utilizes the one-hot encoding as the input and two convolutional operations. In another recent work, DeepTorrent (Liu et al., 2021) integrates four key features and stacks multi-layer convolutional neural networks, attention layer, and bidirectional long short-term memory, which leads to state-of-the-art performance. Key characteristics of the existing methods for 4mC site prediction are listed in the Supplementary Table S1 with respect to the computational algorithms and features employed. Overall, the vast majority of the competing solutions rely on multiple encoding representations to effectively exploit the sequence information and physicochemical properties or complex cascading strategies to improve the predictive capability.

Although the DNA 4mC site prediction task has been studied for several years, there exist great challenges for excavating the interactions within the DNA sequences and integrating the information implied in the sequences of other species, which is the focus of this work. Herein, we propose MSNet-4mC, a deep learning-based computational architecture consisting of an opening convolutional layer, two similar blocks, and two fully connected layers. Motivated by the endeavors of the research on the complex context dependencies within the biological sequences, we construct the structure of multi-scale receptive fields based on convolutional operations to perceive both the long- and short-range relationships to improve the predictive capability. Moreover, our method not only extracts useful information from one-hot encoding features without the use of complex encoding schemes as the input but also takes into consideration of the unbalanced samples between different species and adjusts the loss with class weights. We conduct a series of experiments to benchmark the performance of our method over two standard datasets. Experimental results show that the proposed method can efficiently identify the 4mC sites and outperform the state-of-the-art approaches.

2 Materials and Methods

2.1 Datasets

In this study, we adopted two datasets to assess the capability of MSNet-4mC for identifying DNA 4mC sites. The first dataset was constructed by Chen et al. (Chen et al., 2017), which was originally derived from the MethSMRT database. It consisted of six species, i.e., *C.elegans*, *D.melanogaster*, *A.thaliana*, *E.coli*, *G.subterraneus*, and *G.pickeringii*. The positive samples were obtained by applying the Modification QV (modQV) score larger than 30 and the cutoff threshold of CD-HIT at 0.8. The default threshold of modQV score is 30 for calling a position as modified (Chen et al., 2017). The CD-HIT software was used to remove the redundant sequences with high similarity (Fu et al., 2012). All the sequences of the positive samples had the length of 41 bp with the 4mC sites located in the center. The corresponding negative samples had the same length with the cytosine in the center but were not detected by the SMRT sequencing technology. Therefore, there existed a large number of samples that could be negative samples. To generate a balanced dataset, Chen et al.

randomly picked out the same number of negative samples as the positive samples for each species. In this paper, the samples were randomly divided into the training dataset and the test dataset according to the ratio of 14:1 for each species. For each training or test dataset, it has the same numbers of positive and negative samples. A statistical summary of the dataset is provided in Table 1. For the convenience of description and comparison, we used the same name Lin_2017 for the first dataset as Liu et al. (Liu et al., 2021) did.

The second dataset was first used by Liu et al. (Liu et al., 2021), namely Li_2020. Similar to the Lin_2017 dataset, it was also derived from the MethSMRT database and consisted of six species. In this dataset, the positive samples were selected with the modQV score larger than 30 and the cutoff threshold of CD-HIT at 0.7. Different from the first dataset randomly split as the training and test dataset, the test dataset of the second dataset was filtered with the modQV value larger than 50. A statistical summary of the Li_2020 dataset is provided in Table 2. Each species has the same numbers of positive and negative samples, except *D.melanogaster*. The number difference between the positive and the negative samples of *D.melanogaster* is not large. Accordingly, for a fair and objective comparison, the training and test datasets in this work were still kept the same as those of DeepTorrent. Note that there is no overlap between the training and test sets.

2.2 MSNet-4mC architecture

Figure 1 illustrates an overview of the architecture of MSNet-4mC. MSNet-4mC employs a novel convolutional neural network framework to predict the 4mC sites given the feature matrix. It consists of an opening convolutional layer, two similar blocks, and two fully connected layers. The two blocks are composed of convolutional layers with the shortcut identity connections inspired by ResNet (He et al., 2016) but with a different structure of multi-scale receptive fields to perceive both the long- and short-range relationships within the DNA sequences. Besides, MSNet-4mC applies the class weights in the cross-entropy loss to adjust the training on different species with unbalanced numbers of samples. This is achieved by the following steps: A species sign is attached to each sequence of the training datasets to indicate the species class when encoding the sequence to obtain the input feature. The class weights for different species are calculated based on the frequencies of the corresponding species. When the sequence is fed into the network, the class weight is applied with its species sign (shown in Figure 1) to the CE-loss, where the species sign is used as an index to select the class weight. Note that the sign is not used in the test phase.

2.2.1 Input feature matrix

Different from previous methods that integrate various feature encoding schemes to represent the sequence as the input to train the model, MSNet-4mC only utilizes the simplest and most common encoding scheme, i.e. one-hot encoding. From a theoretical perspective, effective neural networks can extract more powerful and generic features and perceive the inner mathematical relations from the sequence without the need for manually encoding the sequence repeatedly. However, generally speaking, neural networks can only learn from numerical data rather than categorical data. As a consequence, one-hot encoding is used to encode the DNA sequences in this paper. In this encoding scheme, each nucleotide is encoded by a four-bit binary vector. Specifically, 'A', 'C', 'G', and 'T' are encoded by (1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0), and (0, 0, 0, 1), respectively. Each DNA sequence with n nucleotides is represented by a $4 \times n$ dimensional binary vector. Additionally, there is a sign indicating the species class for each DNA sequence to introduce the class weights.

Table 1. Statistical summary of the Lin_2017 dataset

Species	No. 4mC	No. non-4mC	Training dataset		Test dataset	
			No. 4mC	No. non-4mC	No. 4mC	No. non-4mC
<i>C.elegans</i>	1554	1554	1450	1450	104	104
<i>D.melanogaster</i>	1769	1769	1651	1651	118	118
<i>A.thaliana</i>	1978	1978	1846	1846	132	132
<i>E.coli</i>	388	388	362	362	26	26
<i>G.subterraneus</i>	905	905	845	845	60	60
<i>G.pickeringii</i>	569	569	531	531	38	38

Table 2. Statistical summary of the Li_2020 dataset

Species	No. 4mC	No. non-4mC	Training dataset		Test dataset	
			No. 4mC	No. non-4mC	No. 4mC	No. non-4mC
<i>C.elegans</i>	58396	58396	55729	55729	2667	2667
<i>D.melanogaster</i>	57654	57504	53970	53970	3684	3534
<i>A.thaliana</i>	75027	75027	63720	63720	11307	11307
<i>E.coli</i>	2067	2067	1941	1941	126	126
<i>G.subterraneus</i>	15197	15197	9934	9934	5263	5263
<i>G.pickeringii</i>	5724	5724	4514	4514	1210	1210

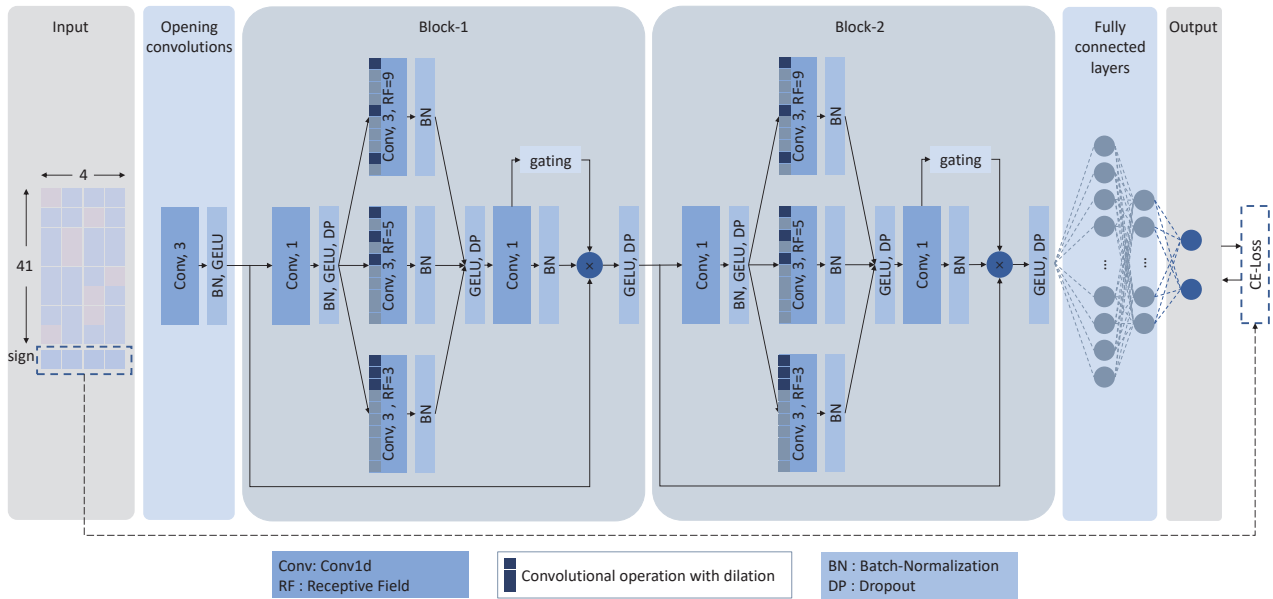


Fig. 1. An overview of the architecture of MSNet-4mC. The model encodes multi-scale features via a series of dilation convolutions, where convolutions with receptive fields of 3, 5, 9 correspond to dilations of 1, 2, 4, respectively. The class weight is applied with its species sign to the CE-loss, where the species sign serves as an index to enable the correct weight assignment.

2.2.2 Convolutional neural network

Recent years have seen the outstanding success of convolutional neural networks, with various methods proposed and applied in real-world applications. Herein, we design MSNet-4mC by leveraging CNN techniques. The input matrices are fed into the first convolutional layer with 16 output channels, and 3 kernel sizes followed by batch normalization and GELU activation. The output of this layer is then fed to the subsequent block with a shortcut identity connection similar to the ResNet structure. These two blocks consist of a convolution layer with one kernel followed by batch normalization, GELU activation, and dropout layer, three parallel

convolution layers with the same output channels and kernel sizes but different receptive fields followed by batch normalization, GELU activation and dropout layer for the merged result of the three anterior outputs, and a convolution layer with one kernel followed by batch normalization. In addition, there is a gating for the output from the convolution layer before merging with the residual. Between these shortcut blocks, we use GELU activation and dropout layer. The output of the second block is fed to two fully connected layers. These two layers contain 192×41 units and 192 units, respectively. The final output layer is equipped with the weighted

cross-entropy loss to balance the number of candidates in different species and the softmax classifier to predict the 4mC sites.

The opening convolutional layer is conducive to handle the issue of vanishing/exploding gradients which obstruct convergence from the start (Glorot and Bengio, 2010; He *et al.*, 2015; Saxe *et al.*, 2013). Each convolutional layer followed by a batch normalization also serves for the same purpose. The residual is designed as described in ResNet to address the degradation problem, which is realized by the shortcut connections without introducing extra parameters and computational complexity. The three parallel convolutional layers are one of the key components of our framework. Different sites in the biological sequences can have contextual dependencies with each other, where the ranges of these dependencies are difficult to confirm (Wong *et al.*, 2016). Several previous studies illustrate that there are also strong correlations within the flanking bases in DNA sequences (Arenas, 2015; Bird, 1980; Lim and Blanchette, 2020; Makova and Hardison, 2015). Inspired by these insights, we propose to excavate the long- and short-range dependencies of the sequences. Specifically, we realize it through employing convolutional operations with a series of dilations to obtain convolutions with different receptive fields, which can effectively explore the inner relationships.

2.2.3 Class weights for loss

The number of samples for each species is small. An important step of our pipeline is to integrate the training dataset of six different species into one training dataset and train a base model to avoid overfitting and fully exploit the cross-species relationships. The major problem is that a species with far more samples can have a far stronger influence on the model optimization than a species with far fewer samples and as such, biased training for the base model can be triggered. A possible rectification is to assign larger weights to the species with fewer samples to alleviate the imbalanced training. As a result, we calculated class weights based on the frequencies of corresponding species and applied these weights to the cross-entropy loss (CE-loss) during the training process. Generally, there are different variations of formula that can be used to calculate the class weights. The essential property is to meet the requirement that a smaller size dataset has a greater weight. One optimization goal of the weighted loss is to pay more attention to the data from the small size species. Nevertheless, inappropriate weights may cause over-bias for the small size dataset, or cannot have an obvious effect.

The overall CE-loss can be described as

$$L = \frac{\sum_{n=1}^N l_n}{N}, \quad (1)$$

where N spans the minibatch dimension, l_n is the CE-loss for the n -th sequence. Constructed on the vanilla CE-loss, the weighted CE-loss for the n -th sequence can be defined as

$$l_n = - \sum_{i=1}^2 \tilde{\omega}_c p_i \ln q_i, \quad (2)$$

where p_i is the ground truth, *i.e.*, represented as a binary indicator (0 or 1), q_i is the probability which is calculated by the softmax function, $\tilde{\omega}_c$ is the normalized weight of the c -th class, respectively. Note that the species sign is used for selecting $\tilde{\omega}_c$. $\tilde{\omega}_c$ is calculated as

$$\tilde{\omega}_c = \frac{C}{\sum_{c'} \omega_{c'}} \omega_c, \quad (3)$$

where ω_c is the unnormalized weight of the c -th class. Directly applying the reciprocal of the frequency as the weight for the species to adjust the loss may introduce a concomitant problem of excessive rectification, since our model uses a Softmax-based classifier to perform probability mapping.

Considering the sizes of the two datasets used in this work and the difference of the numbers of samples among species, especially for the second dataset where there is a huge imbalance, we adopt a relatively smooth formula. To this end, the class weights are calculated by the logarithm of the reciprocal of the frequencies to ensure relatively gentle rectifications. The formula used for the class weights is as follows:

$$\omega_c = \ln \frac{\sum_{c'} f_{c'}}{f_c} = \ln \frac{1}{f_c}, c \in (1, 2, \dots, C), \quad (4)$$

where f_c is the frequency of the c -th species, that is, the number of the c -th species's training dataset divided by the sum of the numbers of the six training datasets, C indicates the number of the classes which is 6 in this paper. The sign mentioned in Figure 1 is used to indicate the class for the sequence.

2.2.4 Parameters

In this section, we present the hyperparameters for the network and training. Supplementary Table S2 shows the fixed hyperparameters for the CNN. Besides, for the first dataset, we trained the network using a batch size of 256 with the SGD optimizer and the cosine learning rate schedule. The number of warm-up epochs was 20, and the sum of training epochs was 300. The initial learning rate was 0.05 for the base model, and 0.01 for the fine-tuning process. In general, for training deep learning models, small datasets can lead to the problem of overfitting. Applying dropout layers is a common method to avoid overfitting. Since the number of samples for each species is different, the dropout rate should be assigned accordingly. That is, a small-size dataset (of a species) requires to be trained with relatively higher dropout rates. With dropouts, some of the feature units are randomly and temporarily deactivated according to the probabilities (*i.e.* the assigned rates of the dropout layers) and will not be optimized during the iterations. There are several dropout layers in the network. More specifically, three dropout layers in block 1 and the first two dropout layers in block 2 were assigned the same dropout rates, respectively. The last one was the last dropout layer in block 2. Three dropout rates were 0.2, 0.5, and 0.8 in turn when training the base model. When it came to performing fine-tuning on the training dataset of each specific species, if the number of the samples for the training set was larger than 2900, three dropout rates were set as 0.2, 0.5, and 0.8 in turn. Otherwise, the three dropout rates were set as 0.2, 0.8, and 0.8 in turn. When training the species-specific models from scratch, the dropout rates were assigned the same values as those used for fine-tuning.

For the second dataset, we trained the base model with a batch size of 512 and an initial learning rate of 0.025, respectively. The initial learning rate was 0.01 for the fine-tuning phase. The optimizer, learning strategy, and training epochs were the same as those for the first dataset. Because the number of samples is substantially increased compared to the first dataset, the dropout rates have been adjusted to a certain extent. The dropout layers in block 1 were no longer used. While training the base model, the first two dropout layers in block 2 were not used. Only the last dropout layer in block 2 was valid with the value of 0.5. When conducting the fine-tuning step, the first two dropout layers and the last dropout layer in block 2 were assigned the dropout rates of 0.25 and 0.5, respectively. Supplementary Table S3 summarizes the training protocols with the corresponding hyperparameters. MSNet-4mC was trained on an Intel(R) Core(TM) i7-11800H @ 2.30 GHz and NVIDIA GeForce RTX 3060 GPU. Its deep learning framework was implemented using PyTorch 1.11.0 for training/testing.

2.3 Performance evaluation metrics

To evaluate the predictive capability of MSNet-4mC, we utilized six common performance evaluation metrics, including sensitivity (SN), specificity (SP), precision, accuracy (ACC), Matthew's correlation coefficient

(MCC), and F1-score. The mathematical expressions of these evaluation metrics are provided as follows:

$$SN = \frac{TP}{TP + FN}, \quad (5)$$

$$SP = \frac{TN}{TN + FP}, \quad (6)$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (7)$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \quad (8)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \quad (9)$$

$$F1_{score} = \frac{2 \times TP}{2 \times TP + FP + FN}, \quad (10)$$

where TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative, respectively. In addition, the receiver-operating characteristic (ROC) curve is also plotted to intuitively evaluate the overall performance. Another important performance measure is the area-under-the-curve (AUC), which is utilized to quantitatively assess the overall performance of the model.

3 Results

In this section, we discuss the performance evaluation results of MSNet-4mC in detail on both the Lin_2017 dataset (Chen *et al.*, 2017) and Li_2020 dataset (Liu *et al.*, 2021).

3.1 Performance evaluation on the Lin_2017 dataset

3.1.1 Ablation study

To investigate the contributions of the class weight module in the proposed method, an ablation study was performed on the Lin_2017 dataset (Chen *et al.*, 2017). The experiments were implemented on the merged training dataset by combining six species with or without the class weight module. We conducted each kind of experiment 10 times. The average results on all the performance metrics are presented in the Supplementary Table S4. The results show that the component of class weights resulted in a modest decline of accuracy around of 0.1% on three large size species but a distinct boost on the other three small size species, which is consistent with our expectation. The minor change in the loss resulted in a more than 3% gain on *G.subterraneus* and *G.pickeringii*. Table 3 shows the results of the per-class mean and variance on the metrics. Compared with the variance results in Table 3, the models with the class weight module achieved a more balanced performance on different species. These results indicate that such component is helpful for the effectiveness of the pre-training model to extract the information derived from the small size species. The cross-species features utilized by the pre-training model can play an important role in the subsequent fine-tuning step on specific-species models.

In addition, to examine the effectiveness of the convolutional operations with multi-scale receptive fields, an ablation study was further conducted on the Lin_2017 dataset. In particular, we removed the two convolutions with receptive fields of 5 and 9 in two blocks for performance comparison. The experiments were implemented on the merged training dataset of six species with or without parallel convolutional layers. Accordingly, we also conducted experiments 10 times for each model. The average performance results in terms of all the performance metrics are provided in the Supplementary Table S5. As can be seen, the models trained with the convolutional operations with multi-scale receptive fields outperformed the models trained only with a single-scale receptive field in the majority of the cases. This clearly demonstrates the effectiveness of the proposed modules.

3.1.2 Performance comparison with the existing methods on the Lin_2017 dataset

In order to comprehensively evaluate MSNet-4mC, we implemented two sets of experiments: The first one is to train the model from scratch on each species training dataset and test its performance on the corresponding test dataset. The second one consists of two steps, which involve firstly training a base model with the merged training dataset and then fine-tuning the hyperparameters to retrain the species-specific model on each species training dataset. Pre-trained models can serve as the initialization parameters for subsequent training tasks, which help to avoid overfitting. Fine-tuning the learnable parameters to retrain the models can improve classification accuracy. On the other hand, training a base model at first with the merged dataset of six species can be useful for extracting potential cross-species features. Thereby, we conducted the experiments with the two-step training strategy.

The performance results of MSNet-4mC and DeepTorrent (Liu *et al.*, 2021) trained from scratch for six species in terms of all evaluation metrics are presented in the Supplementary Table S6. The best result in the table is highlighted in bold. Our models achieved higher MCC, ACC, SN, and AUC values across all species. It achieved the best performance in terms of all six metrics on five species. Particularly, the results show that our method gained around 6% improvement in accuracy and around 8% improvement in AUC. Our method clearly outperformed DeepTorrent on the Lin_2017 dataset by a larger margin when training the species-specific models from scratch, which demonstrates that our proposed framework could effectively extract and leverage the implicit information from biological sequences.

Furthermore, we performed the two-step experiments. The performance of MSNet-4mC is shown in Figure 2(a), with the ROC curves of MSNet-4mC on six species plotted in Figure 2(b). The results show that MSNet-4mC achieved the AUC value of higher than 0.86 and the ACC value of higher than 0.84 across all the six different species, with an average AUC value of 0.92 and ACC value of 0.89, respectively. Supplementary Table S7 provides the performance of MSNet-4mC in terms of all metrics. The detailed performance results of MSNet-4mC and the other three state-of-the-art methods for the species-specific models are provided in Supplementary Table S8. The results of other methods are quoted from the Supplementary materials of DeepTorrent (Liu *et al.*, 2021). As can be seen, our method achieved the best performance in terms of all performance metrics (*i.e.* SN, SP, ACC, and MCC) for three species (*C.elegans*, *A.thaliana* and *G.pickeringii*). In addition, our method achieved the best performance in terms of accuracy and MCC on all six species. The results show the identification capability of our method is better than other methods. The results of the species-specific models both trained from scratch and trained after fine-tuning show that MSNet-4mC outperformed DeepTorrent, thereby suggesting the effectiveness of MSNet-4mC.

To better understand how MSNet-4mC learns effective representations and contributes to the improved performance, we further calculated the spatial distributions of the 4mC and non-4mC samples for each species. Specifically, we utilized a visualization tool t-SNE (Van der Maaten and Hinton, 2008) to intuitively visualize and compare the input features and those extracted from MSNet-4mC. Figure 3 shows the distributions of the 4mC and non-4mC samples of the *E.coli* dataset in the two-dimensional space. Figure 3(a), (b), and (c) are the t-SNE plots of the input features, the output features of block 2, and the output features of second fully connected layer, respectively. It can be seen from Figure 3(a) that the original distribution of the positive and negative samples is entirely mixed together whose boundary is difficult to distinguish. In Figure 3(b), the distribution of the features learned after block 2 shows two relatively clear clusters despite a few overlapping samples. Furthermore, as shown in Figure 3(c), the distribution of the features extracted from the second fully connected layer between the positive and negative samples in the feature

Table 3. Performance results on per-class mean and variance of the metrics

Module	Metrics	MCC	ACC	F1_score	SN	SP	Precision	AUC
No class weight	Mean	0.5925	0.7944	0.7892	0.7767	0.8122	0.8072	0.8583
	Variance	0.0121	0.0031	0.0036	0.0062	0.0031	0.0029	0.0021
Class weight	Mean	0.6138	0.8052	0.8004	0.7881	0.8224	0.8178	0.8676
	Variance	0.0081	0.0021	0.0027	0.0056	0.0024	0.0018	0.0011

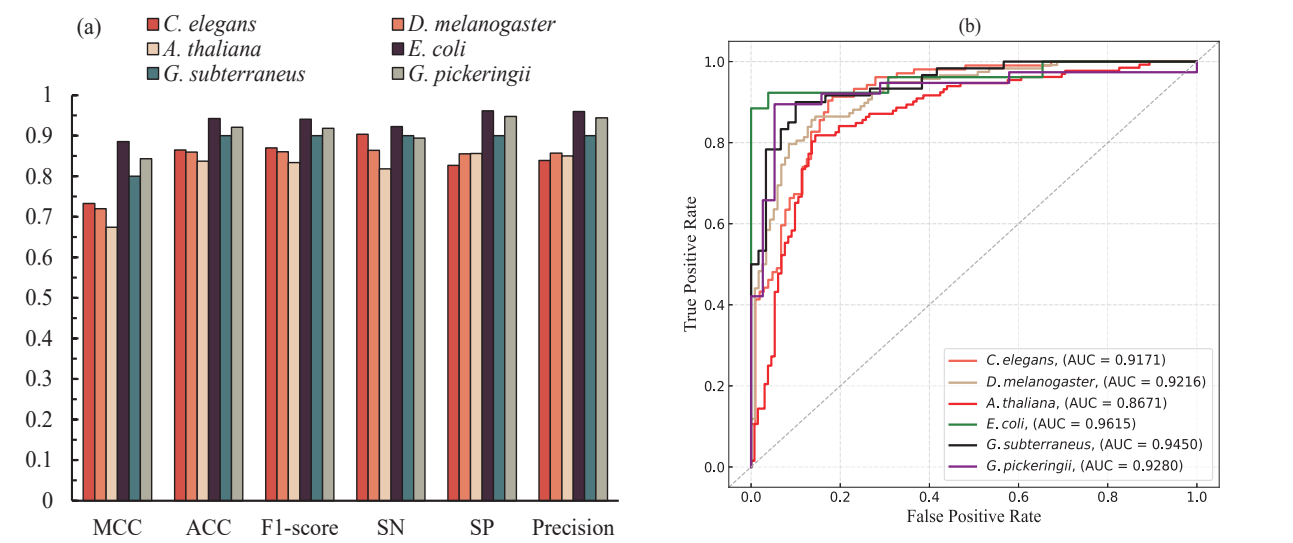


Fig. 2. Species-specific performance of MSNet-4mC on the Lin_2017 dataset.

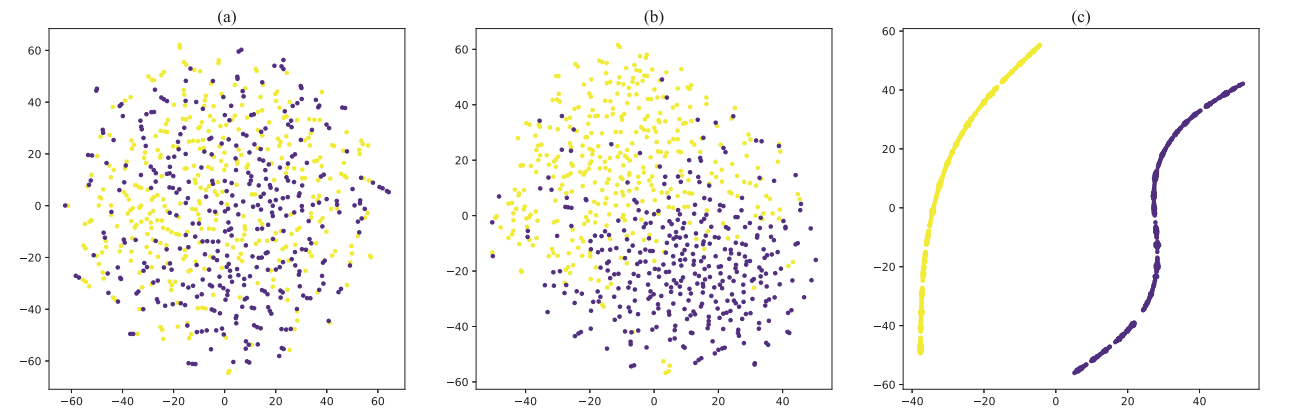


Fig. 3. t-SNE visualization of the *E. coli* dataset in a two-dimensional feature space: (a) the input features, (b) the features after block 2, and (c) the features of the second fully connected layer

space is further clear and separated from each other. In addition, the t-SNE plots for the other five species are provided in the Supplementary Figures S1-S5. Altogether, these results suggest that MSNet-4mC is indeed capable of learning effective representations for 4mC site identification.

3.2 Performance evaluation on the Li_2020 dataset

To demonstrate the generalization ability and effectiveness of MSNet-4mC, we further conducted experiments on the Li_2020 dataset (Liu et al., 2021). Similar to the evaluation on the first dataset, we implemented two

sets of experiments. We conducted the experiments by training the models from scratch using the second dataset. The performance results of MSNet-4mC trained from scratch for the six species in terms of all evaluation metrics are provided in the Supplementary Table S9. Furthermore, we trained a base model on the training dataset by combining the training datasets of the six species at first. Then we retrained the model on each species-specific training dataset to obtain species-specific models after fine-tuning the unified training framework.

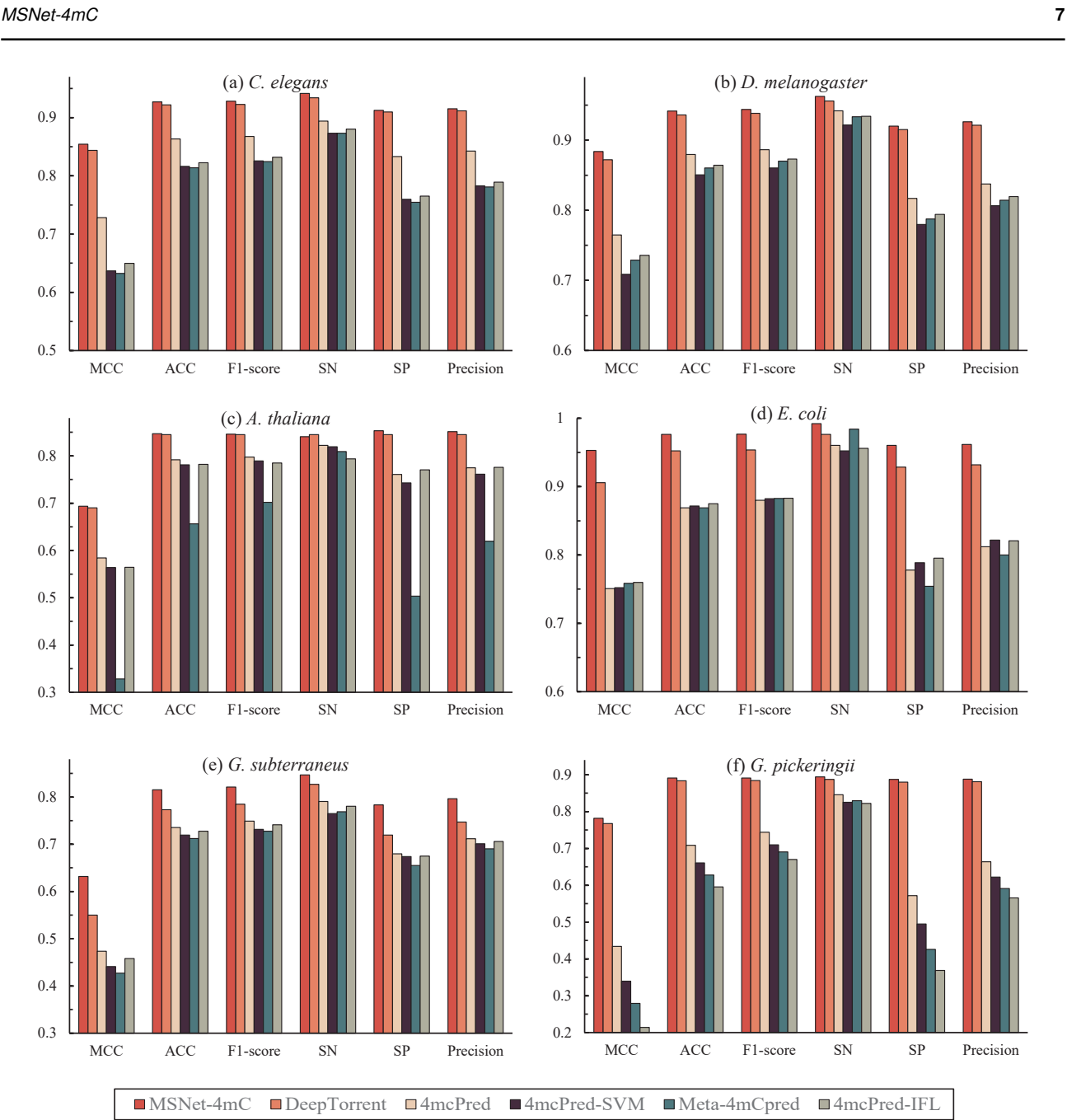


Fig. 4. Performance comparison between MSNet-4mC and other state-of-the-art methods on the Li_2020 dataset.

As a result, the base model achieved a prediction accuracy of 81.09% on the merged test dataset. The performance of the models after fine-tuning is shown in Supplementary Figure S6. Supplementary Figure S6(a) shows the evaluation metrics, and the ROC curves of MSNet-4mC on six species are plotted in Supplementary Figure S6(b). Our method achieved an average AUC value of 0.95 and an average ACC value of 0.90, respectively. The AUC for predicting 4mC sites in *C.elegans*, *D.melanogaster*, *A.thaliana*, *E.coli*, *G.subterraneus* and *G.pickeringii* were 0.976, 0.982, 0.916, 0.996, 0.898 and 0.954, respectively.

Supplementary Table S10 presents the performance comparison of MSNet-4mC and other state-of-the-art methods in terms of all evaluation metrics. Figure 4 shows the performance comparison results for the species-specific models. The results of other methods are quoted directly

from the Supplementary materials of DeepTorrent (Liu *et al.*, 2021). As can be seen, our method outperformed the other methods across all species in terms of six performance metrics (including AUC, MCC, ACC, F1-score, SP, and Precision). It achieved the best performance in terms of all seven metrics with five species, with the only exception of *A.thaliana*. Specifically, our method gained around 1.3% performance improvement of accuracy over SOTA. With regard to *G. subterraneus*, it resulted in 4.2% improvement on accuracy and 4.4% improvement on AUC. Additionally, some performances in the specific dataset were saturated, making the gain less pronounced. Overall, the performance of MSNet-4mC is superior to others.

4 Conclusions

In this study, we proposed MSNet-4mC, a novel deep learning-based approach for predicting DNA 4mC sites. MSNet-4mC is developed based on the CNN, consisting of an opening convolutional layer, two similar blocks based on convolutional layers, and two fully connected layers. The main innovation of our work is the construction of a built-in scale-aware learning structure based on convolutional operations with multi-scale receptive fields to perceive the long- and short-range dependencies implied in the biological sequences and incorporation of the species frequency to rectify the training on different species with unbalanced numbers of samples. We performed the experiments on two benchmark datasets. Extensive experiments consistently show that our method outperforms others despite not using complex encoding representations as the input, thereby highlighting the effectiveness of MSNet-4mC to extract the important and relevant features for identification of 4mC sites. In addition, we have also provided an optimization perspective for exploiting the long- and short-range relationships to better address the biological sequence classification problems.

Despite the outstanding performance of MSNet-4mC, we speculate that there is still much room to further improve its potential. Comparing the results of the trained models from scratch on each species dataset and the base model on the merged six species datasets, the average performance of the models trained from scratch was better than that of the base model. The decrease in the classification performance of the base model might be due to the noisy/redundant information between different species whose removal might help to further improve the performance. Accordingly, effective strategies to improve this aspect may help to achieve a substantial improvement in the predictive performance of identifying 4mC sites from genomic sequences. In addition, there might also exist several potential useful strategies that can be further employed to improve our framework. To enhance the perceptive capability of the framework by leveraging the long- and short-range relationships, one could design new strategies to identify more important representations by taking into consideration varying distances from the central 4mC sites rather than equal distances applied in this study. Moreover, the attention mechanism can be incorporated into the framework to adaptively pay attention to more important information and effectively extract features with strong correlations. Finally, the transfer learning strategy can also be leveraged to obtain a pre-trained model on a larger dataset and retrain the species-specific model to improve its performance.

Acknowledgements

The authors would like to thank the anonymous reviewers for their constructive comments.

Funding

This work was supported by JST, the establishment of university fellowships towards the creation of science technology innovation [JPMJFS2123]. This work was also supported by the Collaborative Research Program of Institute for Chemical Research, Kyoto University. JS was supported by the JSPS Invitational Fellowship [ID L20503]. TA was partially supported by JSPS KAKENHI [#22H00532].

References

Arand, J. et al. (2012). In vivo control of CpG and non-CpG DNA methylation by DNA methyltransferases. *PLoS Genetics*, **8**(6), e1002750.

Arenas, M. (2015). Trends in substitution models of molecular evolution. *Frontiers in Genetics*, **6**, 319.

Barros-Silva, D. et al. (2018). Profiling DNA methylation based on next-generation sequencing approaches: new insights and clinical applications. *Genes*, **9**(9), 429.

Bird, A. P. (1980). DNA methylation and the frequency of CpG in animal DNA. *Nucleic Acids Research*, **8**(7), 1499–1504.

Cai, S. et al. (2019). Ground-to-aerial image geo-localization with a hard exemplar reweighting triplet loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8391–8400.

Chen, W. et al. (2017). iDNA4mC: identifying DNA N4-methylcytosine sites based on nucleotide chemical properties. *Bioinformatics*, **33**(22), 3518–3523.

Davis, B. M., Chao, M. C., and Waldor, M. K. (2013). Entering the era of bacterial epigenomics with single molecule real time DNA sequencing. *Current Opinion in Microbiology*, **16**(2), 192–198.

Flusberg, B. A. et al. (2010). Direct detection of DNA methylation during single-molecule, real-time sequencing. *Nature Methods*, **7**(6), 461–465.

Fu, L. et al. (2012). CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, **28**(23), 3150–3152.

Fu, Y. et al. (2021). CONSK-GCN: conversational semantic-and knowledge-oriented graph convolutional network for multimodal emotion recognition. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Greenberg, M. V. and Bourc'his, D. (2019). The diverse roles of DNA methylation in mammalian development and disease. *Nature Reviews Molecular Cell Biology*, **20**(10), 590–607.

Guo, Y. et al. (2022). Soft Exemplar Highlighting for Cross-View Image-Based Geo-Localization. *IEEE Transactions on Image Processing*, **31**, 2094–2105.

He, K. et al. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034.

He, K. et al. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

He, W., Jia, C., and Zou, Q. (2019). 4mCPred: machine learning methods for DNA N4-methylcytosine sites prediction. *Bioinformatics*, **35**(4), 593–601.

Jeady, S. et al. (2020). The DNA methylation landscape of giant viruses. *Nature Communications*, **11**(1), 1–12.

Jones, P. A. (2012). Functions of DNA methylation: islands, start sites, gene bodies and beyond. *Nature Reviews Genetics*, **13**(7), 484–492.

Khanal, J. et al. (2019). 4mCCNN: Identification of N4-methylcytosine sites in prokaryotes using convolutional neural network. *IEEE Access*, **7**, 145455–145461.

Lim, D. and Blanchette, M. (2020). EvoLSTM: Context-dependent models of sequence evolution using a sequence-to-sequence LSTM. *Bioinformatics*, **36**(Supplement_1), i353–i361.

Liu, Q. et al. (2021). DeepTorrent: a deep learning-based approach for predicting DNA N4-methylcytosine sites. *Briefings in Bioinformatics*, **22**(3), bbaa124.

Makova, K. D. and Hardison, R. C. (2015). The effects of chromatin organization on variation in mutation rates in the genome. *Nature Reviews Genetics*, **16**(4), 213–223.

Manavalan, B. et al. (2019). Meta-4mCpred: a sequence-based meta-predictor for accurate DNA 4mC site prediction using effective feature representation. *Molecular Therapy-Nucleic Acids*, **16**, 733–744.

Roberts, R. J. et al. (2015). REBASE—a database for DNA restriction and modification: enzymes, genes and genomes. *Nucleic Acids Research*, **43**(D1), D298–D299.

Saxe, A. M., McClelland, J. L., and Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, **9**(11).

Wei, L. et al. (2019a). Exploring sequence-based features for the improved prediction of DNA N4-methylcytosine sites in multiple species. *Bioinformatics*, **35**(8), 1326–1333.

Wei, L. et al. (2019b). Iterative feature representations improve N4-methylcytosine site prediction. *Bioinformatics*, **35**(23), 4930–4937.

Wong, K.-C., Li, Y., and Peng, C. (2016). Identification of coupling DNA motif pairs on long-range chromatin interactions in human K562 cells. *Bioinformatics*, **32**(3), 321–324.

Ye, P. et al. (2016). MethSMRT: an integrative database for DNA N6-methyladenine and N4-methylcytosine generated by single-molecular real-time sequencing. *Nucleic Acids Research*, page gkw950.