Briefings in Bioinformatics, 22(3), 2021, 1–14

doi: 10.1093/bib/bbaa124
Problem Solving Protocol

DeepTorrent: a deep learning-based approach for predicting DNA N4-methylcytosine sites

Quanzhong Liu[†], Jinxiang Chen[†], Yanze Wang, Shuqin Li, Cangzhi Jia, Jiangning Song and Fuyi Li

Corresponding authors: Fuyi Li, Monash Biomedicine Discovery Institute, Department of Biochemistry and Molecular Biology and Monash Centre of Data Science, Monash University, Melbourne, VIC 3800, Australia. E-mail: Fuyi.Li1@monash.edu; Jiangning Song, Monash Biomedicine Discovery Institute, Department of Biochemistry and Molecular Biology and Monash Centre of Data Science, Monash University, Melbourne, VIC 3800, Australia. Tel: +61-3-99029304; E-mail: Jiangning.Song@monash.edu; Cangzhi Jia, School of Science, Dalian Maritime University, Dalian 116026, China. E-mail: cangzhijia@dlmu.edu.cn

[†]The first two authors contributed equally to this work.

Abstract

DNA N4-methylcytosine (4mC) is an important epigenetic modification that plays a vital role in regulating DNA replication and expression. However, it is challenging to detect 4mC sites through experimental methods, which are time-consuming and costly. Thus, computational tools that can identify 4mC sites would be very useful for understanding the mechanism of this important type of DNA modification. Several machine learning-based 4mC predictors have been proposed in the past 3 years, although their performance is unsatisfactory. Deep learning is a promising technique for the development of more accurate 4mC site predictions. In this work, we propose a deep learning-based approach, called DeepTorrent, for improved prediction of 4mC sites from DNA sequences. It combines four different feature encoding schemes to encode raw DNA sequences and employs multi-layer convolutional neural networks with an inception module integrated with bidirectional long short-term memory to effectively learn the higher-order feature representations. Dimension reduction and

Quanzhong Liu is an associate professor in the College of Information Engineering, Northwest A&F University, Yangling, China. He received his PhD in Agricultural Electrisation & Automatization from Northwest A&F University, China. His research interests include data mining, big data and bioinformatics.
Jinxiang Chen is currently a visiting master student at Monash Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, Monash University from the College of Information Engineering, Northwest A&F University, China. His research interests are bioinformatics, big data, deep learning and machine learning.
Yanze Wang is a bachelor student in the College of Information Engineering, Northwest A&F University, China. His research interests are bioinformatics, data mining and software engineering.
Shuqin Li is a professor in the College of Information Engineering, Northwest A&F University, China. Her major research interests include bioinformatics, data mining and big data.
Cangzhi Jia is an associate professor in the School of Science, Dalian Maritime University. She obtained her PhD degree in School of Mathematical Sciences from Dalian University of Technology in 2007. Her major research interests include mathematical modelling in bioinformatics and machine learning.
Jiangning Song is currently an associate professor and group leader in the Monash Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, Monash University, Melbourne, Australia. He is also affiliated with the Monash Centre for Data Science, Faculty of Information Technology, Monash University. His research interests include bioinformatics, computational biology, machine learning, data mining and pattern recognition.
Fuyi Li received his PhD degree in Bioinformatics from Monash University, Australia. He is currently a Research Assistant in the Department of Biochemistry and Molecular Biology and the Infection and Immunity Program, Biomedicine Discovery Institute, Monash University, Australia. His research interests are bioinformatics, computational biology, machine learning, and data mining.

Submitted: 6 March 2020; **Received (in revised form):** 5 May 2020

concatenated feature maps from the filters of different sizes are then applied to the inception module. In addition, an attention mechanism and transfer learning techniques are also employed to train the robust predictor. Extensive benchmarking experiments demonstrate that DeepTorrent significantly improves the performance of 4mC site prediction compared with several state-of-the-art methods.

Key words: DNA N4-methylcytosine sites; bioinformatics; sequence analysis; machine learning; deep learning

Introduction

DNA methylation is an epigenetic modification that plays a significant role in the transmission of non-coding inheritable information into a DNA sequence [1]. DNA methylation is associated with a myriad of biological processes, such as gene expression regulation [2], genomic imprinting [3] and cell differentiation [4]. Moreover, alteration of the DNA methylation pattern is regarded as a mechanism of disease [2], often leading to cancer [5] and other diseases [6].

Common types of DNA methylation in genomes include 5-methylcytosine (5mC), N6-methyladenine (6mA) and N4-methylcytosine (4mC) [7]. These three types of DNA methylation are predominantly found in prokaryotes [8]. In eukaryotic genomes, the dominant type of methylation is 5mC [9, 10]. 6mA is more abundant in prokaryotic genomes than in eukaryotic genomes [11]. 4mC more frequently exists in mesophilic bacteria [12] and is very difficult to detect using traditional technologies in the eukaryotic genomes [10].

Bisulphite treatment based on next-generation sequencing (NGS) is a common technique for DNA methylation site detection from the whole genome [13]. However, this experimental technique is expensive and time-consuming [14], and it is limited to 5mC detection [15]. Single-molecule real-time (SMRT) sequencing can detect various forms of DNA methylation, including 5mC, 4mC and 6mA [14]. However, SMRT sequencing is more expensive than NGS for library preparation [16]. Besides, it remains a significant challenge for conventional experimental techniques to differentiate 4mC from 5mC. To address these problems, 4mC-TAB-seq [16], a 4mC-specific method based on NGS, has been proposed to distinguish 4mC from 5mC accurately. Recently, a 4mC-specific technique has been proposed to differentiate 4mC from 5mC using engineered transcription activator-like effectors [17]. These experimental techniques facilitate DNA methylation site detection; however, they are still labour-intensive and expensive and are not practically suitable for high-throughput DNA methylation site identification. Therefore, computational methods that can predict DNA methylation sites provide a useful and complementary strategy for large-scale identification of DNA methylation sites and can efficiently facilitate experimental studies.

To date, several computational methods have been developed for 5mC and 6mA prediction [18–21]. However, to the best of our knowledge, only a few 4mC site prediction methods and tools are available. Table 1 summarizes the existing methods for 4mC site prediction and covers a wide range of aspects, including the algorithm and features employed, the evaluation strategy and the availability of webserver/software. We briefly categorize these methods into two major groups according to their operating algorithms: (i) the first group of methods is based on conventional machine learning (ML) algorithms, including iDNA4mC [22], 4mCPred [23], 4mcPred-SVM [24], Meta-4mCpred [25] and 4mcPred-IFL [26], and (ii) the second group of methods is based on deep learning (DL) algorithms. To our knowledge, there is only one method belonging to this group, called 4mCCNN [27]. All of the studies listed in Table 1 treat 4mC site prediction as a binary classification problem. Moreover, they were all evaluated using the same datasets, which contain experimentally validated 4mC sites of six species. From Table 1, we can conclude that conventional ML-based methods apply support vector machines (SVMs) or integrated multi-classifiers to build the ensemble prediction models for 4mC site identification from DNA sequences. The methods apply different feature encoding schemes to encode the DNA sequences to feature vectors and then train the predictive models. iDNA4mC uses the nucleotide chemical property (NCP) and nucleotide frequency as input features to construct a feature vector for each sample [22]. 4mCPred applies the position-specific trinucleotide propensity and electron–ion interaction pseudopotentials (EIIPs) to encode DNA sequences [23]. 4mcPred-SVM employs four types of sequence-based feature encodings and a two-step feature optimization strategy to improve the prediction performance [20]. Meta-4mCpred first extracts 14 feature descriptors based on seven different feature encoding schemes and then applies four ML algorithms to generate 56 probabilistic features [25]. Ultimately, these 56 features are used to train the SVM-based prediction models. 4mcPred-IFL [26] first employs eight sequence-based features as the input of the SVM classifier and then generates the probability for each sample as a new feature descriptor. The process is then iterated until the performance reaches convergence. These methods have achieved considerable success at 4mC site prediction, and they have indeed accelerated research on 4mC identification. However, the performance of current methods (i.e. their predictive capability) at distinguishing 4mC sites from non-4mC sites relies considerably on the quality of handcrafted features and the operational algorithm. Thus, for further performance improvement, extensive domain knowledge is needed to design informative, handcrafted features for model training. Due to limited research on 4mC [17], however, it is challenging to extract effective features that have a strong discriminative ability to predict 4mC sites.

DL has arisen as a powerful form of representation learning and is capable of learning abstract features for multiple layers of representations automatically [28]. In recent years, DL techniques have been successfully applied in many bioinformatics studies with competitive performance [20, 29, 30]. To our knowledge, there is only one DL-based approach for 4mC site prediction, 4mCCNN [27]. 4mCCNN uses convolution neural networks (CNNs) with two one-dimensional convolutional layers and encodes input sequences into a one-hot encoding matrix to feed into the first convolutional layer. Compared to conventional ML algorithms, 4mCCNN achieves better performance with benchmark datasets. However, despite its improved performance over previous methods, 4mCCNN has some limitations in terms of its learning capacity, as the framework employed by 4mCCNN is relatively shallow and the training datasets used are relatively small. With the rapid development of DL methods in recent years, a variety of DL frameworks have been proposed and proven to achieve a better performance. This is the case for hybrid models and deep transfer learning models, even trained with limited number of samples. In this work, it is of our

Table 1. Characteristics of the existing methods and tools for 4mC site prediction

Category	Methods/tools	Year	Model	Features	Evaluation strategy	Webserver/software availability	Species
Conventional ML-based methods	iDNA4mC [22]	2017	SVM	(1) Nucleotide chemical properties; (2) nucleotide frequency.	Jack-knife test	Yes	The dataset contains experimentally validated 4mC sites of six species including (1) *C. elegans*, (2) *D. melanogaster*, (3) *A. thaliana*, (4) *E. coli*, (5) *G. subterraneus* and (6) *G. pickeringii*
	4mCPred [23]	2019		(1) Position-specific trinucleotide propensity (PSTNP); (2) EIIP	(1) Jack-knife test; (2) independent test	Yes	
	4mcPred-SVM [24]	2019		(1) *k*-mer dinucleotide frequency; (2) mono-nucleotide binary encoding; (3) dinucleotide binary encoding; (4) local position-specific dinucleotide frequency	Ten-fold cross-validation	Yes	
	Meta-4mCpred [25]	2019	Ensemble of SVM, Random Forest, Gradient Boosting Decision Tree and extremely randomized tree	(1) *k*-mer composition; (2) binary profile—BPF; (3) dinucleotide binary profile encoding—DPE; (4) local position-specific dinucleotide frequency—LPDF; (5) ring-function-hydrogen-chemical properties—RFHC; (6) dinucleotide physicochemical properties—DPCP; (7) trinucleotide physicochemical properties—TPCP	(1) Ten-fold cross-validation; (2) independent test	Yes	
	4mcPred-IFL [26]	2019	SVM	(1) Binary and *k*-mer frequency—BKF; (2) dinucleotide binary profile and frequency; (3) physical–chemical properties—PCPs; (4) pseudo dinucleotide composition; (5) *k*-nearest neighbour—KNN; (6) EIIPs of trinucleotide; (7) multivariate mutual information—MMI; (8) RFHC	Ten-fold cross-validation	Yes	
DL-based method	4mCCNN [28]	2019	CNN	One-hot encoding	Ten-fold cross-validation	No	

The URL addresses for the listed tools are as follows: iDNA4mC—http://lin.uestc.edu.cn/server/iDNA4mC; 4mCPred—http://server.malab.cn/4mCPred/index.jsp; 4mcPred-SVM—http://server.malab.cn/4mcPred-SVM/; Meta-4mCpred—http://thegleelab.org/Meta-4mCpred/; 4mcPred-IFL—http://server.malab.cn/4mcPred-IFL/

Table 2. Statistical summary of the *Lin_2017* dataset of the six different species

Species	Number of 4mC sites
C. elegans	1554
D. melanogaster	1769
A. thaliana	1978
E. coli	388
G. subterraneus	905
G. pickeringii	569

particular interest to employ DL frameworks and examine the possibility of further improving the performance of 4mC site predictors.

In this work, we propose DeepTorrent (Deep learning predicTor for N4-methylcytosine sites), a DL-based computational framework for 4mC site prediction from DNA sequence data. More specifically, DeepTorrent utilizes four different types of feature encoding schemes to transform the raw DNA sequences as the input to the deep networks, which consist of CNNs with inception, bidirectional long short-term memory (BLSTM) and an attention mechanism. It uses the deep transfer learning strategy to address the small sample size problem. Extensive benchmarking experiments on two different datasets show that DeepTorrent achieves the best performance for 4mC site prediction across all the six tested species compared with state-of-the-art methods. To facilitate high-throughput predictions of 4mC sites, an online webserver for DeepTorrent is implemented and made freely available at http://DeepTorrent.erc.monash.edu/.

Materials and methods

Datasets

All six existing ML-based 4mC site predictors, including iDNA4mC, 4mCPred, 4mcPred-SVM, Meta-4mCpred, 4mcPred-IFL and 4mCCNN, were trained and evaluated using the same dataset previously constructed by Chen *et al.* [22]. The dataset was originally retrieved from the MethSMRT database [9]. We employed this dataset to train the DeepTorrent model and compared its performance to that of the other existing methods. The dataset contained experimentally verified 4mC sites of six different species, including *Caenorhabditis elegans, Drosophila melanogaster, Arabidopsis thaliana, Escherichia coli, Geoalkalibacter subterraneus* and *Geobacter pickeringii*. All sequences of the positive samples in the dataset had a length of 41 bp. Redundant sequences were removed to ensure that the sequence identity of any two sequences in the dataset was less than 80%, which is consistent with previous studies [22–24]. The number of the extracted positive samples of each species is listed in the second column of Table 2. For each species, the 41 bp long sequences with cytosine in the centre that was not detected by the SMRT sequencing technology were regarded as the negative sample candidates. As a result, a large number of negative sample candidates in each species were generated. Subsequently, the same number of negative samples for each species was randomly extracted from the negative sample candidates. In this way, a dataset with both positive and negative samples for each species was constructed.

Moreover, we also collected additional 4mC sites with sequence length of 41 bp in the above six species genomes from the MethSMRT database [9]. As previously described [22], a modQV score of ≥30 indicated a modified position, we therefore retained such sequences and regarded them as positive sample candidates. For each of the six species, we collected a large number of non-4mC sites containing sequences of 41 bp with cytosine in the centre that were not detected by the SMRT sequencing technology. The non-4mC site-containing sequences were considered as the negative sample candidates. As the positive and negative sample candidates contained many redundant samples with high similarity, we used the CD-HIT program [31] to remove redundant samples with a sequence identity cut-off of 0.7. After the above procedure, we obtained 58 396, 57 654, 75 027, 2067, 15 197 and 5724 positive samples in *C. elegans*, *D. melanogaster*, *A. thaliana*, *E. coli*, *G. subterraneus* and *G. Pickeringii*, respectively.

For each species, the procedures of constructing the additional training datasets and additional independent test dataset are as follows: First, we chose the sequences that had modQV scores of ≥50 as the independent test dataset from the extracted positive sequences as described above. The reason why the samples with modQV ≥50 were selected was that a highly reliable independent test dataset was needed to evaluate DeepTorrent and compare it with the other state-of-the-art methods. The remaining positive sequences were then used as the training dataset. Second, to construct a balanced dataset, we randomly selected the same number of negative samples as positive samples. A statistical summary of the constructed training dataset and independent test dataset is provided in Table 3.

For the convenience of description, we renamed the two datasets described above as follows: The dataset constructed by Chen *et al.* [22] was renamed as *Lin_2017*, while the additional dataset was renamed as *Li_2020*. In addition, we also generated the two-sample sequence logos for these two datasets, which are shown in Supplementary Figures S1 andS2.

DeepTorrent framework

Figure 1 illustrates an overview of the DL architecture of DeepTorrent. In this study, the 4mC site prediction task can be regarded as a binary classification problem. To address this, DeepTorrent first learns the features from the sequence using the feature extraction module (e.g. input; convolutional, merge and pooling; attention and recurrent; and merge layers in Figure 1) and then predicts the 4mC site using the prediction module (fully connected and output layers in Figure 1). DeepTorrent first encodes the samples (represented as 41 bp DNA sequences) using four different encoding schemes. The four encoding matrices are input into the input layer of the feature extraction module in parallel, and CNNs with inception, BLSTM and an attention layer transform each encoding to the abstract feature representations. All these abstract features are then merged into a single feature vector. Subsequently, a two-layer fully connected network and the output layer are used to generate the final prediction outcome, i.e. a 4mC site or non-4mC site.

Feature encoding schemes

In this study, we used seven different DNA sequence encoding schemes provided by the *iLearn* package [32] to encode the DNA sequences: one-hot encoding, composition of *k*-spaced nucleic acid pairs (CKSNAP), NCP, EIIPs of nucleotides, enhanced nucleic acid composition (ENAC), accumulated nucleotide frequency (ANF) and position-specific trinucleotide propensity based on single-stranded characteristics (PSTNPss). Here, we considered four types of nucleic acids ('A', 'C', 'G' and 'T')

Table 3. Statistical summary of the *Li_2020* dataset for the six different species

Species	Training datasets		Test datasets	
	Number of 4mC sites	Number of non-4mC sites	Number of 4mC sites	Number of non-4mC sites
C. elegans	55 729	55 729	2667	2667
D. melanogaster	53 970	53 970	3684	3684
A. thaliana	63 720	63 720	11 307	11 307
E. coli	1941	1941	126	126
G. subterraneus	9934	9934	5263	5263
G. pickeringii	4514	4514	1210	1210

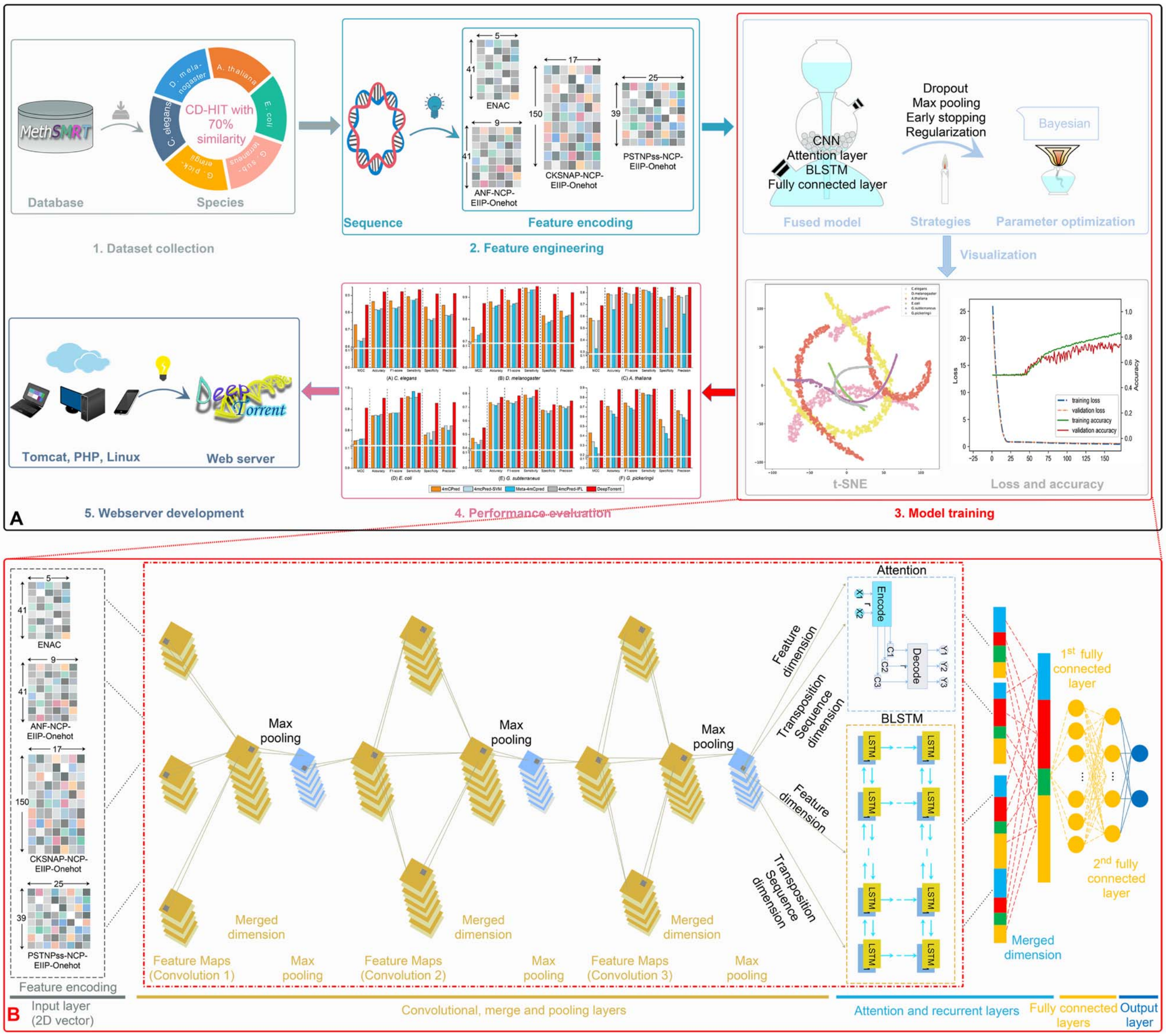


Figure 1. The overall framework of DeepTorrent. (**A**) The workflow of the development and assessment process of DeepTorrent. (**B**) The structure of the DeepTorrent framework, including the input layer, convolutional layers, merger layers, inception module, attention layers, fully connected layers and output layer.

and the unknown character '-' in DNA sequences. These seven encoding schemes are introduced in the following subsections.

One-hot encoding. One-hot encoding is used to describe the nucleotide acid composition along the DNA sequence. It has been shown that one-hot encoding is an effective encoding scheme for predicting 4mC [24] and 6mA sites from genomic sequences [18]. For this encoding scheme, 'A', 'C', 'G', 'T' and '-' are represented by a binary vector of (1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0), (0, 0, 0, 1) and (0, 0, 0, 0), respectively. Accordingly, a DNA sequence with n nucleotides is encoded into a 4 ×n dimensional binary vector.

Composition of k-spaced nucleic acid pairs. CKSNAP, also known as nucleotide pair spectrum encoding, have been previously used for 6mA site prediction [18]. CKSNAP transform a DNA sequence into a numerical vector by calculating the occurrence frequency of all possible *k*-spaced nucleotide pairs along the DNA sequence. A *k*-spaced nucleotide pair denotes that there are *k* spaces between these two nucleotides. For example, in the sequence 'AXXGXXXT', 'AG' is a two-spaced nucleotide pair, and 'GT' is a three-spaced nucleotide pair. Let *knp* denote the *k*-spaced nucleotide pair. The frequency of *knp* can be defined as:

$$f\left(\mathrm{knp}\right)=\frac{\mathrm{Count}\left(\mathrm{knp}\right)}{l-k-1} \tag{1}$$

where Count(knp) represents the count of knp along the DNA sequence, l is the length of the DNA sequence and $k \in \left[0, k_{\max}\right]$ is the space between nucleotide pairs. Thus, $(l-k-1)$ denotes the number of all k-spaced nucleotide pairs along a DNA sequence with length l. Therefore, the DNA sequence was encoded into a $5\times\ 5\ \times(k_{\max}+1)$-dimensional vector. In this work, we set $k_{\max}=5$.

Nucleotide chemical property. Each nucleotide in a DNA sequence has three types of chemical properties: the ring structure, functionality and hydrogen bond [33]. The ring structure contains purine with two rings and pyrimidines with one ring. 'A' and 'G' are purine and encoded as '1', whereas 'C' and 'T' are pyrimidines and encoded as '0' (see the second column of Table 4). According to the chemical functionality, 'A' and 'C' belong to the amino group and are encoded with 1; 'G' and 'T' belong to the keto group and are encoded with 0 (see the third column of Table 4). 'A' and 'T' form the weak hydrogen bond and are encoded as '1'; and 'C' and 'G' form the strong hydrogen bond and are encoded as '0' (see the fourth column of Table 4). As a result, 'A', 'C', 'G', 'T' and '-' are encoded as (1,1,1), (0, 1, 0), (1, 0, 0), (0, 0, 1) and (0, 0, 0), respectively. Accordingly, a DNA sequence can be encoded as a binary vector. For example, 'GTTGACT' can be encoded as (1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1).

Electron–ion interaction pseudopotential. The EIIP [34] of nucleotides describes the distribution of free-electron energies along the DNA sequence. It has been previously shown that the EIIP is an effective feature for 4mC site prediction [23]. The EIIP values of nucleotides 'A', 'G', 'C', 'T' and '-' are 0.1260, 0.0806, 0.1340, 0.1335 and 0, respectively. Accordingly, a DNA sequence can be encoded as a numerical vector using the EIIP encoding scheme. For example, 'GTTGACT' is encoded as a numerical vector composed of (0.0806, 0.1335, 0.1335, 0.0806, 0.1260, 0.1340, 0.1335).

Enhanced nucleic acid composition. The ENAC encodes each nucleotide into a five-dimensional vector by calculating the density information of each nucleotide in a sequence window of a DNA sequence. It describes the local sequence-order information in a DNA sequence. Suppose $\mathrm{W}=\mathrm{s}_{\mathrm{i}}, \mathrm{s}_{\mathrm{i}+1}, \ldots, \mathrm{s}_{\mathrm{i}+w-1}$, which represents a nucleotide sequence window in a DNA sequence, wheres_{i} denotes the ith nucleotide in the DNA sequence and w is the window size. Here, s_{i} is encoded by the vector $(f(\mathrm{A}), f(\mathrm{C}), f(\mathrm{G}), f(\mathrm{T}), f(-))$, and the function f represents the frequency of the corresponding nucleotide in W. Each nucleotide can be encoded by sliding the window along the DNA sequence. In this study, we set $w=2$.

Accumulated nucleotide frequency. The ANF describes the density information of each nucleotide in a DNA sequence [33]. It is an effective feature encoding scheme for 4mC site prediction [22]. We used the ANF feature descriptors for DNA sequences. Let $\mathrm{S}=\mathrm{s}_1\mathrm{s}_2\ldots\mathrm{s}_{\mathrm{i}}\ldots\mathrm{s}_l$ denote a DNA sequence, and let $\mathrm{S}_j=\mathrm{s}_1\mathrm{s}_2\ldots\mathrm{s}_j$ represent the jth prefix sequence of S. Here, l is the length of S and s_j is the jth nucleotide of S. The ANF value of s_j is defined as follows:

$$\mathrm{ANFs}_j=\frac{f\left(\mathrm{s}_j\right)}{\mid \mathrm{S}_j \mid} \tag{2}$$

$$f\left(\mathrm{s}_j\right)=\sum_{t=1}^{j} T\left(\mathrm{s}_t\right),\quad T\left(\mathrm{s}_t\right)=\begin{cases} 1, & \mathrm{s}_t=\mathrm{s}_j \\ 0, & \mathrm{s}_t\neq \mathrm{s}_j \end{cases} \tag{3}$$

where $\mid \mathrm{S}_j \mid$ is the length of the subsequence S_j. For example, the sequence 'GTTGACT' is encoded as (1, 0.5, 0.67, 0.5, 0.2, 0.17, 0.43).

Position-specific trinucleotide propensity based on single-stranded characteristics. The PSTNPss depicts the position-specific trinucleotide statistical propensity based on the single-stranded characteristics of DNA [35]. There are $4^3=64$ trinucleotides derived from the alphabet [A, C, G, T], such as 'AAA', 'AAC', ... 'TTT', etc. Supposing that $\mathrm{A}\prec\mathrm{C}\prec\mathrm{G}\prec\mathrm{T}$, the 64 trinucleotides strings can be sorted by the defined partial order $\prec$ .Let L be the sorted trinucleotide list. For a 41 bp long sequence, the trinucleotide positional specificity can be represented by a 64×39 matrix F. f_{ij}. The value of the ith row and jth column in F is defined as follows:

$$f_{ij}=\mathrm{F}^{+}\left(\mathrm{L}_{i}\middle|\ j\right)-\mathrm{F}^{-}\left(\mathrm{L}_{i}\middle|\ j\right), i\in[1,64], j\in[1,39] \tag{4}$$

where $\mathrm{F}^{+}(\mathrm{L}_i|\ j)$ and $\mathrm{F}^{-}(\mathrm{L}_i|\ j)$ denote the frequency of the ith trinucleotide in L at the jth position appearing in the positive and negative samples, respectively. If a trinucleotide contains an unknown nucleotide '-', the value of the trinucleotide will be set to zero. Thus, let $\mathrm{S}=\mathrm{s}_1\mathrm{s}_2\ldots\ \mathrm{s}_j\ \mathrm{s}_{j+1}\ \mathrm{s}_{j+2}\ \ldots\mathrm{s}_{41}$ denote a 41 bp long DNA sequence, such that S can be encoded as a 39-dimensional numerical vector $P=(p_1, p_2, \ldots, p_j, \ldots, p_{39})$, where p_j is calculated as follows:

$$p_j=f_{ij},\quad \text{if } \mathrm{s}_j\mathrm{s}_{j+1}\mathrm{s}_{j+2}=\mathrm{L}_i, i\in[1,64] \tag{5}$$

Four different feature encoding combinations. We further grouped the seven encoding schemes described above into four feature encoding combinations. These four combinations and their dimensions are shown in Table 5. For example, 'ANF + NCP + EIIP + One-hot' combines ANF, NCP, EIIP and one-hot encoding schemes and encodes a 41 bp long DNA sequence as a (41×9)-dimensional vector.

Table 4. The NCP encoding scheme

Nucleic acid	Ring structure	Functionality	Hydrogen bond
A	1	1	1
C	0	1	0
G	1	0	0
T	0	0	1
–	0	0	0

Table 5. Four combinations of encoding schemes

Combination ID	Encoding scheme	Dimension
1	ENAC	41×5
2	ANF + NCP + EIIP + One-hot	41 ×9
3	CKSNAP+NCP + EIIP + One-hot	150×17
4	PSTNPss + NCP + EIIP + One-hot	39×25

Convolution neural network

There are three convolutional layers in the DeepTorrent framework. In the first convolutional layer, three convolution blocks with different convolution kernel sizes are used to extract the features from the encoding matrix in parallel. The kernel sizes of these three convolutional blocks are 1, 3 and 5, respectively. There are 32 filters in these three convolutional blocks, the L2 regularization value is 0.002 and ReLU is used as the activation function.

Using the first convolutional layer's three convolution blocks, we obtain three different kinds of feature representations. For an input $m \times n$ encoding matrix, each type of feature is represented by an $m \times 32$ matrix following the first convolutional layer. Subsequently, a merging layer with a dropout value of 0.5 is used to merge the three $m \times 32$ matrices into an $m \times 96$ matrix, which had a higher dimension and more meaningful feature representation. Visualization of different kernel sizes is provided as an example in Supplementary Figure S1.

The second convolutional layer, used as an inception module in [36], also contains three convolutional blocks. The first convolutional block with the kernel size 1 was first used to extract higher abstraction feature representations from the concatenated feature maps extracted by the first convolutional layer. Then, the output feature representations are used as the input to the second convolutional block with a kernel size of three and as the input to the third convolutional block with a kernel size of five. The number of filters in these three convolutional blocks is 136, the L2 regularization value is 0.002 and ReLU is used as the activation function. Thus, three $m \times 136$ matrices are generated by the second layer of convolutional operations. Subsequently, a merging layer is used to merge three $m \times 136$ matrices into an $m \times 408$ matrix. The dropout value is set as 0.5 in the merging layer.

Similarly, the third convolutional layer also applies the inception module, following the same operation as that of the second convolutional layer to obtain higher-dimensional abstract features. The kernel size of the three convolutional blocks in the third convolutional layer is 1, 3 and 5, respectively. The number of filters in these three convolutional blocks is 48, the L2 regularization value is 0.002 and ReLU is used as the activation function. Like the first two convolutional layers, the third convolutional layer is also followed by a merged layer with a dropout value of 0.5.

Attention layer

The attention mechanism adaptively focuses on important positions and relevant parts and ignores irrelevant parts [37, 38]. It has been widely used in a variety of DL applications, including visual processing [38], kinase-specific phosphorylation site prediction [37], speech recognition [39] and natural language processing [40]. In this study, we were motivated by the prospect of implementing the attention layer in DeepTorrent. The attention layer selects important features from two dimensions: the feature dimension and the sequence dimension. This means that the output matrix from the CNN layer and its transposed matrix ('transposition' in Figure 1) are fed into the attention layer. In this manner, the two different feature representations can be selected by the attention layer.

Bidirectional long short-term memory

BLSTM is a special type of recurrent neural network (RNN), consisting of two reversed unidirectional LSTM networks [41]. BLSTM can capture the interdependencies across the sequence and integrate both forward and backward information in the sequence [42]. According to a recent study, a good strategy for utilizing the merits of both CNN and RNN is to use the CNN as the pre-processing step for the RNN [43]. In addition, the combination of LSTM with CNN has been used to predict the subcellular localization of proteins from sequence information [44] and to quantify the function of DNA sequences [45]. In DeepTorrent, the CNN layer is connected to two additional BLSTMs, each of which processes the sequence dimension and feature dimension outputs of the CNN. In this way, we obtain four feature representations, among the two feature representations from the attention layer and the two feature representations from BLSTM. Subsequently, four feature representations are combined into a more powerful feature representation by the merging layer.

Fully connected layers and output layer

We used multiple combinations of the four encoding schemes as the input to train the DeepTorrent models in parallel. With three convolutional layers, an attention layer and two BLSTM networks, each feature matrix generated by each encoding combination is converted into four feature vectors. As a result, 16 feature vectors are generated in total. Then, a merging layer is used to merge these 16 feature vectors into a combined feature vector.

Subsequently, the combined feature vector is fed into a fully connected network. The first layer of the fully connected network contains 64 units. Again, the activation function is ReLU, and the dropout value is 0.5. The second fully connected layer has eight units and uses ReLU activation. The final output layer is equipped with softmax loss as the classifier to generate the prediction results.

Parameter optimization

We employed Bayesian optimization [46] to optimize the hyperparameters of the convolutional networks of DeepTorrent. The Bayesian optimization method first models a learning algorithm's generalization performance as a sample from a Gaussian process and then automatically finds better hyperparameters to optimize the performance of the learning algorithm. Bayesian optimization has been shown to select better hyperparameters than human expert-level optimization for CNNs [46]. In DeepTorrent, the predefined value range for each hyperparameter before optimization and its optimal value after optimization are provided in Supplementary Table S1.

Evaluation metrics

To quantify the performance of DeepTorrent and compare with other methods, we used six common performance evaluation metrics [47–55]: sensitivity (Sn), specificity (Sp), precision, accuracy (Acc), Matthew's correlation coefficient (MCC) and F1-score, respectively, defined as follows:

$$\mathrm{Sn} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$$

$$\mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$$

$$\mathrm{Acc} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$$

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

$$\mathrm{F1}-\mathrm{score} = 1-\frac{\mathrm{TP}+\mathrm{TN}}{2*\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}$$

where TP, TN, FP and FN denote the numbers of true positives, true negatives, false positives and false negatives, respectively. In addition, we plotted the receiver-operating characteristic (ROC) curves based on the output of DeepTorrent and accordingly calculated the area-under-the-curve (AUC) values.

Results and discussion

In this section, we discuss the performance evaluation results of DeepTorrent in detail. In particular, we conducted performance evaluation tests on both the *Lin_2017* dataset and *Li_2020* dataset.

Performance evaluation on the *Lin_2017* dataset

Performance evaluation on the Lin_2017 training dataset

To benchmark the performance of DeepTorrent, we performed a test using all encoding inputs derived from all possible combinations of the four encoding schemes listed in Table 5. Let *m*, *n*, *p* and *t* (i.e. 1, 2, 3 and 4, respectively) represent the feature combination IDs (as shown in the first column of Table 5). Accordingly, *m*–*n* denotes a di-encoding composition containing the encoding scheme combinations *m* and *n*; *m*–*n*–*p* denotes a tri-encoding composition that consists of the encoding scheme combinations *m, n* and p and *m*–*n*–*p*–*t* denotes a tetra-encoding composition. There were 15 possible encoding compositions: four individual encoding compositions, six di-encoding compositions, four tri-encoding compositions and one tetra-encoding composition. Taking *m*–*n*–*p*–*t* as an example, DeepTorrent worked as follows: first, the encoding scheme combinations *m*, *n*, *p* and *t* were input into DeepTorrent in parallel; second, for each encoding scheme combination, the corresponding abstract feature representation was extracted by DeepTorrent; finally, four kinds of abstract feature were concatenated into the feature vector (the merged layer in Figure 1) as the feature descriptors of the predictor.

We integrated six species-specific datasets into a large dataset and used approximately 90% of the entries in the datasets as the training dataset and the remaining entries as the validation dataset. The performance results of the DeepTorrent models trained using each encoding composition are provided in Supplementary Table S2. In addition, we plotted the ROC curves of DeepTorrent trained using each encoding, as shown in Supplementary Figure S2. We then identified the encoding compositions with the best performance among the individual encoding compositions, di-encoding compositions, tri-encoding compositions and tetra-encoding compositions (Supplementary Table S2 and Figure S2). The results indicate that the best encoding compositions were 4, 3-4, 2-3-4 and 1-2-3-4. Performance comparison of the four selected encoding compositions is shown in Figure 2A. The results show that DeepTorrent achieved the best overall performance based on the model trained using the 1-2-3-4 tetra-encoding composition.

In terms of species-specific 4mC prediction, the dataset for each species was relatively small. As is well known, the use of small datasets for training DL models can cause overfitting [56]. Therefore, we used a larger dataset by combining all six species to train a base network to avoid overfitting. For each of the six species, we used the corresponding training dataset to retrain the species-specific model. The procedure was as follows: we first copied *n* layers of the base network as the first *n* layers of the species-specific model and then froze the first *n* layers of the species-specific model and merely fine-tuned the hyperparameters of other layers to retrain the species-specific model using species-specific training datasets. This strategy has been shown to be effective at improving the performance of classification problems with limited sample datasets in recent studies on image classification [57] and kinase-specific phosphorylation site prediction [37]. Figure 2B shows the training loss and accuracy of the base model with respect to the training epochs on both the training and validation data. During the process of training, the early stopping strategy was adopted, and we monitor accuracy changes when testing for when to stop training. The training process would be interrupted when the validation accuracy was no longer improved after 20 epochs. It is clear that the base model displayed similar training loss and validation loss for training epochs and achieved higher training accuracy (0.8241) and validation accuracy (0.7406) after about 170 epochs. This indicates that the base model provides a strong foundation for tuning the species-specific model based on transfer learning.

To illustrate how DeepTorrent learned effective feature representations, we used the t-SNE plot [58] to visualize feature representations of two dimensions that were automatically learned by DeepTorrent. The original feature representation is shown in Figure 3A. As can be seen, it is difficult to visually distinguish each species with the original feature representations. Based on the feature representations (Figure 3B) learned after the attention layer of the model, we could identify the principal components. Moreover, the feature representations (Figure 3C) after the second fully connected layer could be used to better identify and separate each species. These results suggest that DeepTorrent is able to learn good feature representations effectively.

Performance comparison with the existing methods on the Lin_2017 independent test dataset

To compare the performance of DeepTorrent with existing 4mC prediction methods, we performed comparisons with existing methods using independent dataset. For the independent test of 4mCPred [23], the positive and negative training datasets of each species' benchmark dataset were randomly divided into 15 subsets of approximately equal size, of which 14 subsets were selected as the training datasets and the remaining one was used to test the model. For a fair and objective comparison, we used the same training datasets and the independent test datasets for each of the six species. Supplementary Figure S3 shows the ROC curves of each species-specific model with deep transfer learning. In addition, we performed an independent test of the species-specific models without transfer learning using the same training and independent test datasets and plotted the ROC curves in Supplementary Figure S4, for a comparison with models trained with transfer learning. As can be seen from Supplementary Figure S4, each species-specific model trained with transfer learning consistently achieved higher AUC values than its counterpart without transfer learning, e.g. 0.893 versus 0.832 on *C. elegans*, 0.911 versus 0.847 on *D. melanogaster*, 0.815 versus 0.783 on *A. thaliana*, 0.935 versus 0.796 on *E. coli*, 0.939 versus 0.856 on *G. subterraneus* and 0.929 versus 0.875 on *G. pickeringii*. The performance comparison results of species-specific models with and without transfer learning in terms of all evaluation metrics are provided in Supplementary Table S3. As can be seen, the species-specific models trained with transfer learning outperformed those trained without transfer learning in terms of all major evaluation metrics.

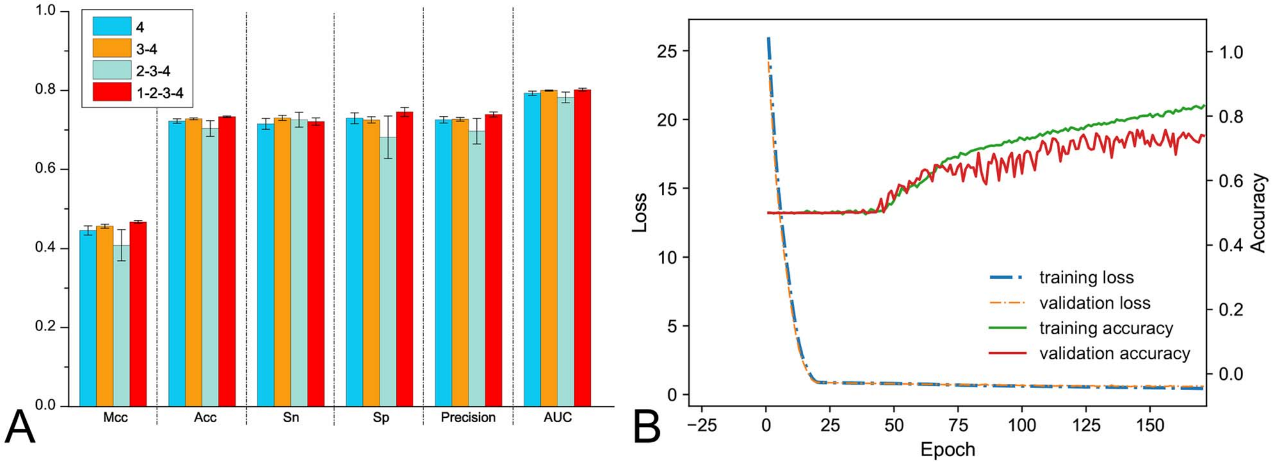


Figure 2. Performance evaluations of DeepTorrent. (**A**) Performance comparisons of different encoding compositions. Here, IDs 1, 2, 3 and 4 denote 'ENAC', 'ANF + NCP + EIIP + One-hot', 'CKSNAP + NCP+EIIP + One-hot' and 'PSTNPss + NCP + EIIP + One-hot' encoding schemes, respectively. Among these, 1-2-3-4 denotes encoding compositions 1, 2, 3 and 4. Other encoding compositions have similar implications. (**B**) Loss and accuracy plot of the model.

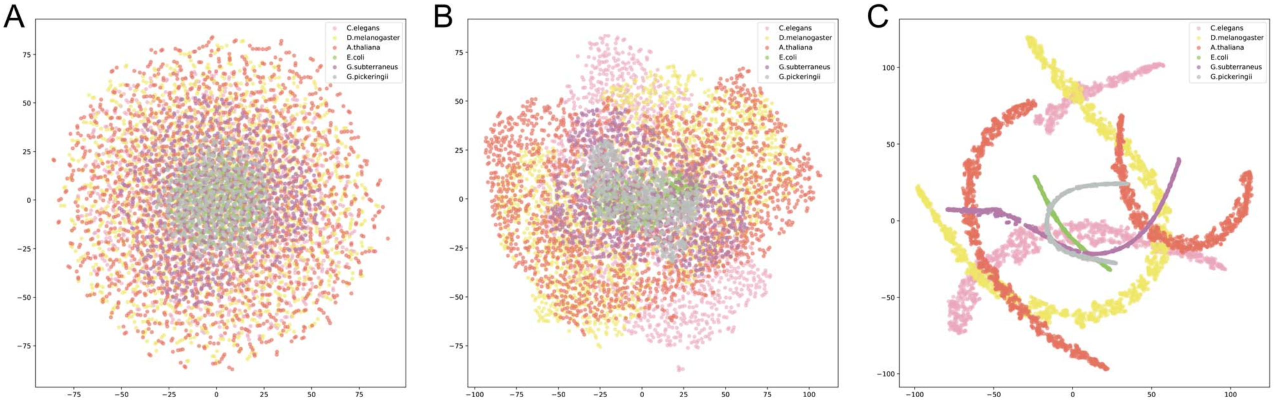


Figure 3. t-SNE plots of input encoding (**A**), feature representation after the attention layer (**B**) and feature representation of the second fully connected layer (**C**) for the base models using positive samples of six different species.

We performed an independent test to compare the performance of DeepTorrent to that of two state-of-the-art methods: 4mCPred [23] and iDNA4mC [22]. The performance results for the six species are provided in Supplementary Table S4. As can be seen, compared to 4mCPred and iDNA4mC, DeepTorrent achieved the best performance in terms of all performance metrics (viz. Sn, Sp, Acc and MCC) for four species (viz. *C. elegans*, *D. melanogaster*, *A. thaliana* and *G. pickeringii*). For *E. coli* and *G. subterraneus*, DeepTorrent achieved the best performance in terms of Sp, Acc and MCC. In terms of the two major comprehensive metrics (Acc and MCC), DeepTorrent achieved the best performance on all six species. These results demonstrate that DeepTorrent is a powerful 4mC site predictor capable of accurately identifying potential m4C sites.

Ten-fold cross-validation test on the Lin_2017 training dataset

To evaluate the performance of DeepTorrent, we performed 10-fold cross-validation tests to compare DeepTorrent with several existing methods, including iDNA4mC [22], 4mCPred [23], 4mcPred-SVM [24], Meta-4mCPred [25] and 4mCCNN [27]. The performance results of these methods for the six species are provided in Supplementary Table S5. As shown in Supplementary Table S5, compared with these five methods, DeepTorrent achieved the best performance in terms of Acc and Sp across all the six species. We have also plotted the MCC values for the six methods with the six species, with the results shown in Figure 4. As can be seen, DeepTorrent achieved the highest MCC values for five out of the six species (with the exception of *A. thaliana*).

Cross-species validation

To examine the potential relationships between two different species, we further conducted cross-species validation using data from one species to train the DeepTorrent model and then applied the trained model to predict 4mC sites in other species. The rationale for this is that transfer learning can transfer the source domain knowledge (species-specific training data) to the target domain (another species) [23]. The cross-species performance results of DeepTorrent, iDNA4mC [22] and 4mCPred [23] are listed in Supplementary Tables S6–S8, respectively. Note that the results in Tables S7 and S8 were originally collected from [23]. In addition, we also conducted cross-species validation using the 4mcPred-SVM webserver [24]. The corresponding results of 4mcPred-SVM

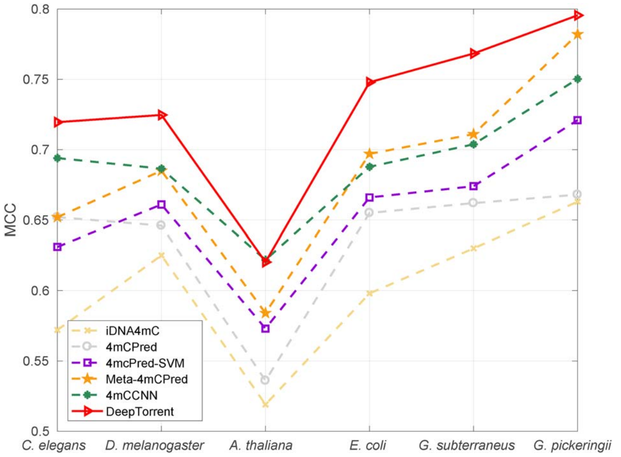


Figure 4. Species-specific performance comparison of DeepTorrent and other predictors in terms of MCC values on the existing datasets of six species.

are listed in Supplementary Table S9. As can be seen from Supplementary Tables S6–S9, in cases where two species differ in terms of the source domain and target domain, 4mCPred achieved the strongest transfer learning capability from *A. thaliana* to *D. melanogaster* and *G. subterraneus* to *G. pickeringii,* whereas DeepTorrent showed the strongest ability to transfer knowledge in all other cases.

Performance evaluation on the *Li_2020* dataset

Ten-fold cross-validation test on the Li_2020 training dataset

Similar to the performance test described in Performance evaluation on the *Lin_2017* dataset, we first trained a base network on a large number of datasets integrated with the training datasets of the six species shown in Table 3. We then retrained the species-specific model for each of the six species using the corresponding species-specific training dataset. After that, we performed 10-fold cross-validation test and evaluated the species-specific performance of DeepTorrent model. The results are shown in Figure 5A and ROC curves are plotted in Figure 5B. The detailed performance results are provided in Supplementary Table S10. As can be seen, DeepTorrent achieved an AUC value of higher than 0.86 across all the six species and an average AUC value of 0.94 and Acc value of 0.87, respectively. These results show that DeepTorrent provides a reasonable predictive performance for species-specific 4mC prediction.

Performance comparison with other existing methods on the Li_2020 independent test dataset

To further evaluate the predictive capability of DeepTorrent, we performed the independent test and compared its performance with other existing methods using the additional independent test datasets. The performance results of all the compared methods are visualized in Figure 6, and ROC curves plotted in Figure 7. The detailed results are provided in Supplementary Table S11. The performance results of the other predictors were calculated using the corresponding web servers of these methods. As the web servers of iDNA4mC [22] and 4mCCNN [27] were unavailable, the performance results of these two methods were not included in the performance comparison. As can be seen from Figures 6 and 7 and Supplementary Table S11, DeepTorrent outperformed the other methods on all seven metrics with five species, except for *E. coli*, for which DeepTorrent outperformed the other methods in terms of six (of seven) performance metrics.

Webserver implementation

As an implementation of the proposed DeepTorrent method, we developed an online webserver based on PHP, which is freely available at http://DeepTorrent.erc.monash.edu/. The webserver is managed by Tomcat 7 and configured in a Linux environment on an 8-core server machine with 32 GB of memory and a 500 GB hard disk. The webserver also provides access to the trained models and to the source code for DeepTorrent. To utilize the webserver, users need to upload DNA sequences or paste them in the 'TEXT AREA' in FASTA format (Submission of up to 100 sequences are permitted). Step-by-step guidelines for the

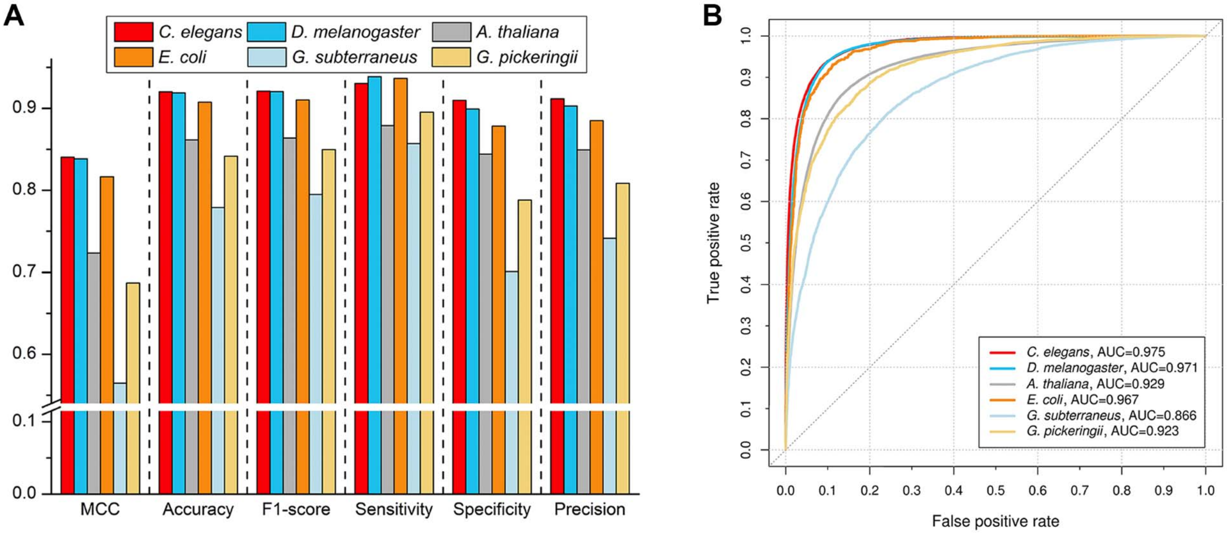
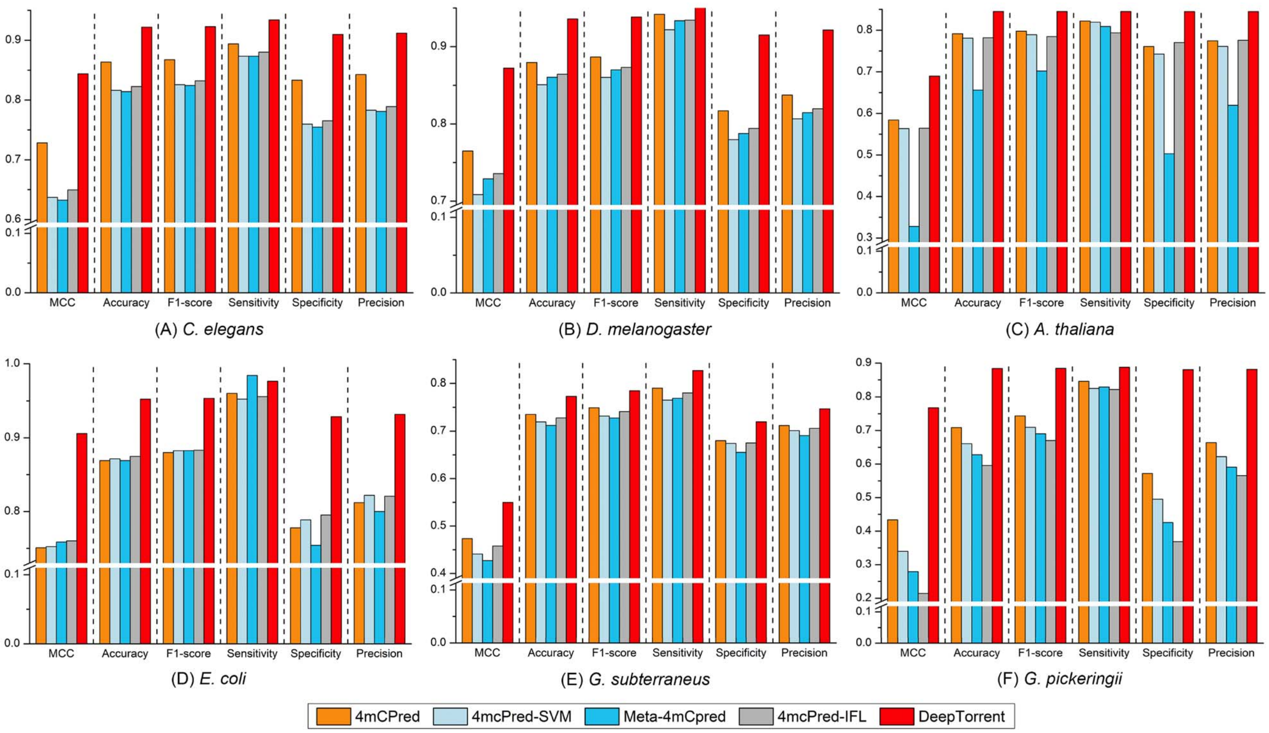


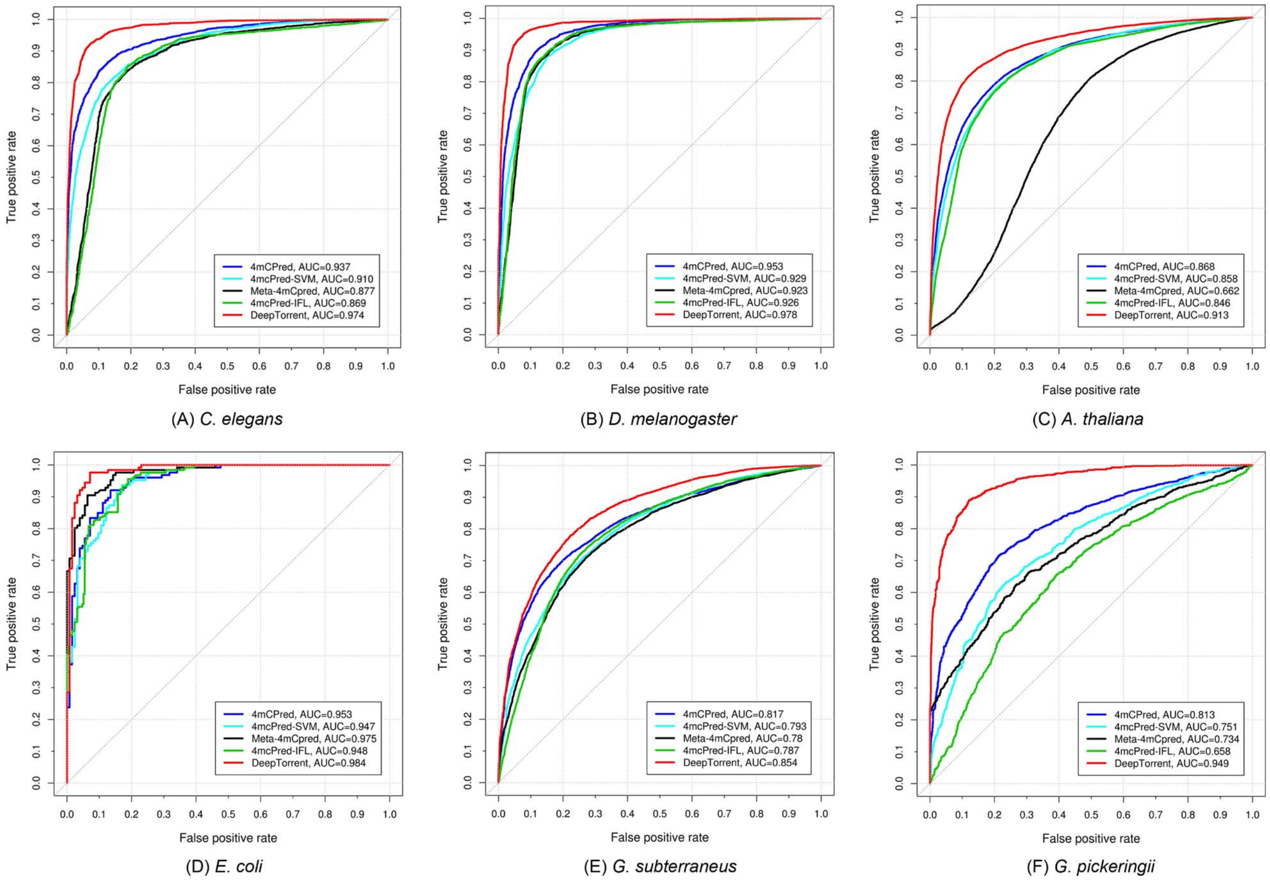
Figure 5. Ten-fold cross-validation performance comparison of DeepTorrent on the additional independent test datasets.



(A) *C. elegans* (B) *D. melanogaster* (C) *A. thaliana*

(D) *E. coli* (E) *G. subterraneus* (F) *G. pickeringii*

Figure 6. Performance comparison between DeepTorrent and four other existing methods on the additional independent test datasets.



(A) *C. elegans* (B) *D. melanogaster* (C) *A. thaliana*

(D) *E. coli* (E) *G. subterraneus* (F) *G. pickeringii*

Figure 7. Performance comparison of DeepTorrent and other state-of-the-art methods in terms of the AUC value on the additional independent test datasets.

DeepTorrent webserver can be found at http://deeptorrent.erc.monash.edu.au/help.html.

Challenges and future work

Despite the competitive predictive performance of DeepTorrent in 4mC site prediction, we believe there is room for further improvement, especially in terms of algorithm learning techniques. Herein, we discuss the challenges and potentially useful strategies for improving 4mC site prediction methods. The first challenge is about how to choose the appropriate ML/DL framework for model training. The majority of existing methods are primarily based on manual trial-and-error selection of the optimal ML/DL framework to build the prediction model, which is a time-consuming and laborious process as there are a variety of ML/DL frameworks and algorithms available. Moreover, DL frameworks often require substantial computational resources and time to train and optimize the model. In this regard, automated machine learning (AutoML) packages, such as Auto-PyTorch (https://github.com/automl/Auto-PyTorch) and AutoKeras (https://autokeras.com/), are suggested to apply to identify the well-performing architecture of deep neural networks. In addition, such tools can also help simplify the model optimization process in DL model training, which can greatly facilitate the model training and improve the robustness of the trained models.

Conclusion

In this study, we proposed a novel DL-based approach, called DeepTorrent, for 4mC site prediction. DeepTorrent is based on a CNN framework with inception modules and BLSTM, and it is integrated with an attention mechanism on both the sequence and feature dimensions for identifying more important and relevant features. Moreover, DeepTorrent combines multi-encoding schemes to find optimal encoding inputs. As a result, four encodings are input into the DL network in a parallel manner. The model uses these encoding inputs to derive complex features, which are concatenated into a single feature vector as the input of fully connected layers for predicting 4mC sites. This unique architecture has been shown to be effective through the visualization of the feature representations.

To address the potential problem of overfitting from the use of small datasets, we introduced an effective transfer learning strategy using the datasets of six species to first learn a base model and then transfer the base model to train species-specific models. Compared to existing methods, the species-specific models trained by transfer learning achieved better performance with four species (viz. *C. elegans*, *D. melanogaster*, *A. thaliana* and *G. pickeringii*) and a better predictive performance according to three major metrics for the other two species (viz. *E. coli* and *G. subterraneus*). Further, our models achieved the best performance in terms of both accuracy and MCC for all six species.

To validate DeepTorrent, we performed cross-species validation and evaluated the performance of the different methods. The results indicate that DeepTorrent provides a competitive performance and knowledge transfer capability compared with several state-of-the-art methods.

Moreover, we constructed an additional dataset and further assessed the performance of DeepTorrent relative to the other methods on this dataset. The results demonstrate that DeepTorrent offers an improved predictive performance.

A user-friendly webserver and source code for DeepTorrent are freely available at http://DeepTorrent.erc.monash.edu/. Overall, DeepTorrent is poised to be a powerful tool for accurate and high-throughput 4mC site prediction from DNA sequences.

Key Points

- We reviewed existing methods for DNA N4-methylcytosine site prediction and categorized these methods into two major groups according to the operating algorithms.
- We proposed DeepTorrent, a novel deep learning-based method, for DNA N4-methylcytosine site prediction.
- Experimental results on two datasets demonstrate the superior performance of DeepTorrent compared to existing machine learning-based methods.
- A webserver (http://DeepTorrent.erc.monash.edu/) was developed to facilitate high-throughput prediction of DNA N4-methylcytosine sites.

Supplementary Data

Supplementary data are available online at https://academic.oup.com/bib.

Funding

National Health and Medical Research Council of Australia (NHMRC) (1092262); Australian Research Council (ARC) (LP110200333 and DP120104460); National Institute of Allergy and Infectious Diseases of the National Institutes of Health (R01 AI111965); Major Inter-Disciplinary Research (IDR) project awarded by Monash University; Collaborative Research Program of Institute for Chemical Research, Kyoto University (2019-32).

References

1. Jeltsch A. Beyond Watson and Crick: DNA methylation and molecular enzymology of DNA methyltransferases. *Chembiochem* 2002;**3**(4):275–93.
2. Santos KF, Mazzola TN, Carvalho HF. The prima donna of epigenetics: the regulation of gene expression by DNA methylation. *Braz J Med Biol Res* 2005;**38**(10):1531–41.
3. Tycko B. DNA methylation in genomic imprinting. *Mutat Res Rev Mutat Res* 1997;**386**(2):131–40.
4. Wu H, Sun YE. Epigenetic regulation of stem cell differentiation. *Pediatr Res* 2006;**59**(4):21R–5.
5. Wang XK. *Next-Generation Sequencing Data Analysis*. CRC Press, Inc, 2016.
6. Robertson KD. DNA methylation and human disease. *Nat Rev Genet* 2005;**6**(8):597–610.
7. Cheng XD. DNA modification by methyltransferases. *Curr Opin Struct Biol* 1995;**5**(1):4–10.
8. Hattman S. DNA- adenine methylation in lower eukaryotes. *Biochemistry (Moscow)* 2005;**70**(5):550–8.
9. Ye P, Luan Y, Chen K, *et al.* MethSMRT: an integrative database for DNA N6-methyladenine and N4-methylcytosine generated by single-molecular real-time sequencing. *Nucleic Acids Res* 2017;**45**(D1):D85–9.
10. Lyko F. The DNA methyltransferase family: a versatile toolkit for epigenetic regulation. *Nat Rev Genet* 2018;**19**(2):81–92.

11. Liu J, Zhu Y, Luo G-Z, *et al.* Abundant DNA 6mA methylation during early embryogenesis of zebrafish and pig. *Nat Commun* 2016;**7**:1–7.
12. Ehrlich M, Wilson GG, Kuo KC, *et al.* N4-methylcytosine as a minor base in bacterial DNA. *J Bacteriol* 1987;**169**(3):939–43.
13. Lister R, Ecker JR. Finding the fifth base: genome-wide sequencing of cytosine methylation. *Genome Res* 2009;**19**(6):959–66.
14. Flusberg BA, Webster DR, Lee JH, *et al.* Direct detection of DNA methylation during single-molecule, real-time sequencing. *Nat Methods* 2010;**7**(6):461–U472.
15. Feng Z, Li J, Zhang J-R, *et al.* qDNAmod: a statistical model-based tool to reveal intercellular heterogeneity of DNA modification from SMRT sequencing data. *Nucleic Acids Res* 2014;**42**(22):13488–99.
16. Yu M, Ji L, Neumann DA, *et al.* Base-resolution detection of N-4-methylcytosine in genomic DNA using 4mC-Tet-assisted-bisulfite-sequencing. *Nucleic Acids Res* 2015; **43**(21):e148–e148.
17. 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.
18. Zhou Y, Zeng P, Li Y-H, *et al.* SRAMP: prediction of mammalian N-6-methyladenosine (m(6)a) sites based on sequence-derived features. *Nucleic Acids Res* 2016;**44**(10):e91–e91.
19. Feng P, Ding H, Chen W, *et al.* Identifying RNA 5-methylcytosine sites via pseudo nucleotide compositions. *Mol Biosyst* 2016;**12**:3307–11.
20. Jin Q, Meng Z, Pham TD, *et al.* DUNet: a deformable network for retinal vessel segmentation. *Knowl Based Syst* 2019;**178**:149–62.
21. Feng P, Yang H, Ding H, *et al.* iDNA6mA-PseKNC: identifying DNA N-6-methyladenosine sites by incorporating nucleotide physicochemical properties into PseKNC. *Genomics* 2019;**111**(1):96–102.
22. Chen W, Yang H, Feng P, *et al.* iDNA4mC: identifying DNA N-4-methylcytosine sites based on nucleotide chemical properties. *Bioinformatics* 2017;**33**(22):3518–23.
23. He W, Jia C, Zou Q. 4mCPred: machine learning methods for DNA N-4-methylcytosine sites prediction. *Bioinformatics* 2019;**35**(4):593–601.
24. Wei L, Luan S, Nagai LAE, *et al.* Exploring sequence-based features for the improved prediction of DNA N4-methylcytosine sites in multiple species. *Bioinformatics* 2019; **35**(8):1326–33.
25. Manavalan B, Basith S, Shin TH, *et al.* Meta-4mCpred: a sequence-based meta-predictor for accurate DNA 4mC site prediction using effective feature representation. *Mol Ther Nucleic Acids* 2019;**16**:733–44.
26. Wei L, Su R, Luan S, *et al.* Iterative feature representations improve N4-methylcytosine site prediction. *Bioinformatics* 2019;**35**(23):4930–4937.
27. Khanal J, Nazari I, Tayara H, *et al.* 4mCCNN: identification of N4-Methylcytosine sites in prokaryotes using convolutional neural network. *IEEE Access* 2019;**7**:145455–145461.
28. Esteva A, Robicquet A, Ramsundar B, *et al.* A guide to deep learning in healthcare. *Nat Med* 2019;**25**(1):24–9.
29. Li F, Chen J, Leier A, *et al.* DeepCleave: a deep learning predictor for caspase and matrix metalloprotease substrates and cleavage sites. *Bioinformatics* 2019;**36**(4):1057–1065.
30. Hong J, Luo Y, Mou M, *et al.* Convolutional neural network-based annotation of bacterial type IV secretion system effectors with enhanced accuracy and reduced false discovery. *Brief Bioinform* 2019. doi:10.1093/bib/bbz120
31. Fu L, Niu B, Zhu Z, *et al.* CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* 2012;**28**(23):3150–2.
32. Chen Z, Zhao P, Li F, *et al.* iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data. *Brief Bioinform* 2020;**21**(3):1047–1057.
33. Bari A, Reaz MR, Jeong BS. Effective DNA encoding for splice site prediction using SVM. *Match Commun Math Comput Chem* 2014;**71**(1):241–58.
34. Nair AS, Sreenadhan SP. A coding measure scheme employing electron-ion interaction pseudopotential (EIIP). *Bioinformation* 2006;**1**(6):197–202.
35. He W, Jia C, Duan Y, *et al.* 70ProPred: a predictor for discovering sigma70 promoters based on combining multiple features. *BMC Syst Biol* 2018;**12**:44.
36. Szegedy C, Liu W, Jia Y, *et al.* Going deeper with convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 2015*, pp. 1–9, IEEE, New Jersey, USA.
37. Wang D, Zeng S, Xu C, *et al.* MusiteDeep: a deep-learning framework for general and kinase-specific phosphorylation site prediction. *Bioinformatics* 2017;**33**(24):3909–16.
38. Mnih V, Heess N, Graves A, *et al.*: Recurrent Models of Visual Attention. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, Quebec, Canada, 2014*, **3**, pp. 2204–12. Curran Associates, NY, USA.
39. Fan R, Zhou P, Chen W, *et al.* An online attention-based model for speech recognition. arXiv preprint 2018;1811.05247.
40. Yin W, Schutze H, Xiang B, *et al.* ABCNN: attention-based convolutional neural network for Modeling sentence pairs. *Trans Assoc Comput Linguist* 2016;**4**:259–72.
41. Li S, Chen J, Liu B. Protein remote homology detection based on bidirectional long short-term memory. *BMC Bioinformatics* 2017;**18**:443.
42. Jurtz VI, Johansen AR, Nielsen M, *et al.* An introduction to deep learning on biological sequence data: examples and solutions. *Bioinformatics* 2017;**33**(22):3685–90.
43. Chollet F. Deep Learning with Python. New York, NY: Manning 2018; 229–9.
44. Sønderby SK, Sønderby CK, Nielsen H, *et al.* Convolutional LSTM networks for subcellular localization of proteins. In: *Algorithms for Computational Biology*, Vol. **9199**. New York, NY: Springer International Publishing, 2015, 68–80.
45. Quang D, Xie X. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Res* 2016;**44**(11).
46. Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. In: *Advances in Neural Information Processing Systems*, Curran Associates, NY, USA, Vol. **25**, 2012, 2960–8.
47. Li F, Li C, Marquez-Lago TT, *et al.* Quokka: a comprehensive tool for rapid and accurate prediction of kinase family-specific phosphorylation sites in the human proteome. *Bioinformatics* 2018;**34**(24):4223–31.
48. Li F, Wang Y, Li C, *et al.* Twenty years of bioinformatics research for protease-specific substrate and cleavage site prediction: a comprehensive revisit and benchmarking of existing methods. *Brief Bioinform* 2019; **20**(6):2150–2166.

49. Li F, Zhang Y, Purcell AW, *et al*. Positive-unlabelled learning of glycosylation sites in the human proteome. *BMC Bioinformatics* 2019;**20**(1):112.
50. Zhang M, Li F, Marquez-Lago TT, *et al*. MULTiPly: a novel multi-layer predictor for discovering general and specific types of promoters. *Bioinformatics* 2019;**35**(17):2957–65.
51. Rao B, Zhou C, Zhang G, *et al*. ACPred-fuse: fusing multi-view information improves the prediction of anticancer peptides. *Brief Bioinform* 2019;doi:10.1093/bib/bbz088.
52. Su R, Zhang J, Liu X, *et al*. Identification of expression signatures for non-small-cell lung carcinoma subtype classification. *Bioinformatics* 2020;**36**(2):339–346.
53. Wei L, Hu J, Li F, *et al*. Comparative analysis and prediction of quorum-sensing peptides using feature representation learning and machine learning algorithms. *Brief Bioinform* 2020;**21**(1):106–119.
54. Mei S, Li F, Leier A, *et al*. A comprehensive review and performance evaluation of bioinformatics tools for HLA class I peptide-binding prediction. *Brief Bioinform* 2019;doi:10.1093/bib/bbz051.
55. Li F, Li C, Wang M, *et al*. GlycoMine: a machine learning-based approach for predicting N-, C- and O-linked glycosylation in the human proteome. *Bioinformatics* 2015;**31**(9):1411–9.
56. Yosinski J, Clune J, Bengio Y, *et al*.: How transferable are features in deep neural networks? In: *Advances in Neural Information Processing Systems* 2014, Curran Associates, NY, USA, **27**:3320–8.
57. Esteva A, Kuprel B, Novoa RA, *et al*. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 2017;**542**(7639):115–118.
58. Maaten LVD, Hinton G. Visualizing data using t-SNE. *J Mach Learn Res* 2008;**9**(Nov):2579–605.