


DeepLncPro: an interpretable convolutional neural network model for identifying long non-coding RNA promoters

Tianyang Zhang, Qiang Tang, Fulei Nie, Qi Zhao  and Wei Chen

Corresponding authors: Qi Zhao, School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China. E-mail: zhaoshi@lnu.edu.cn; Wei Chen, School of Life Sciences, North China University of Science and Technology, Tangshan 063210, China; Innovative Institute of Chinese Medicine and Pharmacy, Chengdu University of Traditional Chinese Medicine, Chnegdu 611137, China. E-mail: greatchen@ncst.edu.cn

Abstract

Long non-coding RNA (lncRNA) plays important roles in a series of biological processes. The transcription of lncRNA is regulated by its promoter. Hence, accurate identification of lncRNA promoter will be helpful to understand its regulatory mechanisms. Since experimental techniques remain time consuming for genome-wide promoter identification, developing computational tools to identify promoters are necessary. However, only few computational methods have been proposed for lncRNA promoter prediction and their performances still have room to be improved. In the present work, a convolutional neural network based model, called DeepLncPro, was proposed to identify lncRNA promoters in human and mouse. Comparative results demonstrated that DeepLncPro was superior to both state-of-the-art machine learning methods and existing models for identifying lncRNA promoters. Furthermore, DeepLncPro has the ability to extract and analyze transcription factor binding motifs from lncRNAs, which made it become an interpretable model. These results indicate that the DeepLncPro can server as a powerful tool for identifying lncRNA promoters. An open-source tool for DeepLncPro was provided at <https://github.com/zhangtian-yang/DeepLncPro>.

Keywords: long non-coding RNA, promoter, convolution neural network, model interpretability, transcription factors

Introduction

Long non-coding RNA (lncRNA) is a kind of non-coding RNAs with the length greater than 200 nucleotides [1]. Although they lack the protein-coding potential, lncRNAs play important roles in various of biological processes [2, 3], such as the regulation of cell cycle, apoptosis, transcription, splicing, translation, genomic rearrangement and genetic imprinting [4–9], etc. Furthermore, a growing number of evidences demonstrated that lncRNAs also associated with human diseases and even cancer development [10, 11]. For example, the abnormal expression of lncRNA is associated with the development of cardiovascular diseases and Huntington's disease [12, 13]. Hence, in order to reveal their biological functions, more researches on lncRNAs are needed and necessary [14]. Knowing about the origins of lncRNA is the first step to illustrate their regulatory roles. A promoter is a regulatory element located upstream of the transcription start site (TSS) [15], which initiates and regulates the transcription of RNA through the binding of transcription factors [16, 17]. Therefore, accurately identifying the promoter of lncRNA will be not only helpful to determine its origins, but also to understand its regulatory mechanisms.

Experimental methods for identifying promoters are mainly mutation analysis and immunoprecipitation analysis [18, 19]. Although these methods are gold standard for determining promoters, they are still time consuming and cost-ineffective for genome wide analysis [20–22]. Fortunately, a large amount of data was generated from these experiments, especially for *Homo sapiens* and *Mus musculus*, which are valuable resources for developing in computational methods for identifying lncRNA promoters. In 2019, Alam *et al.* proposed a deep learning based method, called DeepCNPP [23], for identifying human lncRNA promoters. However, neither the web-server nor source code was provided for DeepCNPP, which hindered its applications in lncRNA promoter identification. Later on, Tang *et al.* proposed a freely accessible web-server ncPro-ML for identifying lncRNA promoters in human and mouse [24]. Unfortunately, ncPro-ML is only based on hand-crafted features and is lack of biological interpretability. In conclusion, both DeepCNPP and ncPro-ML fall short in interpreting the model from biological perspectives, and their predictive accuracies for identifying lncRNA promoters still have room for improvement. Therefore, there is a need

Tian-Yang Zhang is a graduate student at the School of Life Sciences, North China University of Science and Technology. His research interests focus on bioinformatics.

Qiang Tang is a Ph.D. candidate at School of Basic Medical Sciences, Chengdu University of Traditional Chinese Medicine. His research interests include bioinformatics and machine learning.

Fulei Nie is a Ph.D. candidate at School of Life Sciences, North China University of Science and Technology. Her research interests include bioinformatics and machine learning.

Qi Zhao is a professor at School of Computer Science and Software Engineering, University of Science and Technology Liaoning. His research interests include bioinformatics, complex network and machine learning.

Wei Chen is a professor at Innovative Institute of Chinese Medicine and Pharmacy, Chengdu University of Traditional Chinese Medicine. His research interests include bioinformatics and machine learning.

Received: July 19, 2022. Revised: September 14, 2022. Accepted: September 17, 2022

© The Author(s) 2022. Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oup.com

Table 1. Detail information of the datasets used in this study

Name	Training dataset		Validation dataset		Testing dataset	
	Positive	Negative	Positive	Negative	Positive	Negative
Human	1403	1403	468	468	468	468
Mouse	1846	1846	616	616	615	615

to develop interpretable models to accurately identify lncRNA promoters.

We therefore proposed a convolutional neural network (CNN)-based method, called DeepLncPro, to identify lncRNA promoters in human and mouse. In DeepLncPro, the sequences were encoded by using one-hot, nucleotide chemical properties and dinucleotide physical-chemical properties. In order to obtain a robust model, the hyperparameter optimization process was performed to obtain optimal hyper-parameters of CNN. The evaluations based on independent test dataset showed that DeepLncPro outperformed state-of-the-art machine learning methods. In addition, comparative results demonstrated that DeepLncPro is superior to existing methods for predicting lncRNA promoters. DeepLncPro also has the benefits in the biological interpretation and is capable of capturing sequence motifs, which can be matched to transcription factor binding motifs. For facilitating researchers to implement DeepLncPro, the command-line version of DeepLncPro was available at <https://github.com/zhangtian-yang/DeepLncPro>. We expect that DeepLncPro will be helpful for the identification of lncRNA promoters.

Materials and methods

Dataset

In this study, we constructed the benchmark dataset in a similar way to our previous work [24]. The promoter sequences of lncRNA from human and mouse were obtained from the Eukaryotic Promoter Database (EPD) [25]. Considering that RNA polymerases usually bind in the upstream regions of the TSS [26], positive samples were taken around the TSS and contained more upstream regions. Negative samples were taken from the downstream regions away from the TSS. Considering that core promoter elements usually locate in the upstream region of the TSS and the length of the upstream region may have an impact on the model performance, we constructed seven datasets based on sequence lengths from 61 to 301 bp with a step of 40 bp. For a dataset with the sequences of n bp length, positive samples were extracted from $(n-20)$ bp upstream of the TSS to 20 bp downstream of the TSS. Negative samples were extracted in the same way, but 1000 bp downstream of the TSS. The ratio of positive to negative samples was kept at 1:1. For each dataset, 60% of the samples were randomly selected out and used as training data to train the model, 20% were used as validation data to tune the model parameters, and the remaining 20% were used as test data to evaluate the final model (Figure 1A). The details of the datasets were shown in Table 1.

Feature representation algorithms

For the convenience of feature description, a DNA sequence were denoted as $S=D_1D_2 \dots D_L$, where L is the length of the sequence and $D_i \in \{A, T, G, C\}$ represents the deoxynucleotide at the i -th position in the sequence. The one-hot, nucleotide chemical

properties (NCP) and dinucleotide physical-chemical properties were used to encode the samples in the dataset, Figure 1B.

One-hot

One-hot encoding method can effectively represent DNA sequences [27, 28] and encode deoxynucleotides into binary vectors. On the basis of this method, 'A' was encoded as (1, 0, 0, 0), 'T' as (0, 1, 0, 0), 'G' as (0, 0, 1, 0) and 'C' as (0, 0, 0, 1). Hence, a DNA sequence of length L can be transformed into a $4 \times L$ matrix A_1 .

$$A_1 = \begin{bmatrix} \text{one-hot}_1(1) & \dots & \text{one-hot}_1(L) \\ \vdots & \ddots & \vdots \\ \text{one-hot}_4(1) & \dots & \text{one-hot}_4(L) \end{bmatrix} \quad (1)$$

NCP

The four deoxyribonucleotides carry different bases that differ in ring structure, hydrogen bond strength and chemical function [29]. In terms of the number of rings, 'A' and 'G' contain two rings, while 'C' and 'T' contain one ring. In terms of hydrogen bond strength, a weak hydrogen bond is formed between 'A' and 'T', while a strong hydrogen bond is formed between 'C' and 'G'. In terms of chemical composition, 'A' and 'C' belong to the amino group, while 'G' and 'T' belong to the ketone group. Accordingly, 'A' was encoded as (1, 1, 1), 'T' as (0, 1, 0), 'G' as (1, 0, 0) and 'C' as (0, 0, 1):

$$\text{NCP}_1(i) = \begin{cases} 1 & \text{if } D_i \in \{A, G\} \\ 0 & \text{if } D_i \in \{C, T\} \end{cases}, \quad \text{NCP}_2(i) = \begin{cases} 1 & \text{if } D_i \in \{A, T\} \\ 0 & \text{if } D_i \in \{C, G\} \end{cases}, \\ \text{NCP}_3(i) = \begin{cases} 1 & \text{if } D_i \in \{A, C\} \\ 0 & \text{if } D_i \in \{G, T\} \end{cases} \quad (2)$$

where i is the position of the deoxynucleotide in the sequence; NCP_1 , NCP_2 and NCP_3 represent the three chemical properties, respectively. By using NCP, a DNA sequence of length L can be transformed into a $3 \times L$ matrix A_2 .

$$A_2 = \begin{bmatrix} \text{NCP}_1(1) & \dots & \text{NCP}_1(L) \\ \vdots & \ddots & \vdots \\ \text{NCP}_3(1) & \dots & \text{NCP}_3(L) \end{bmatrix} \quad (3)$$

Dinucleotide physicochemical properties (DPCP)

Continuous deoxynucleotide combinations have different physicochemical properties, which is an important feature of genome functional element identification [30] and has been used in promoter prediction [31, 32]. In this study, six DPCP, namely twist, tilt, roll, shift, slide and rise, were used to encode DNA sequences. Their values were obtained from previous work [33]. Since their values varied in different ranges, the min-max normalization

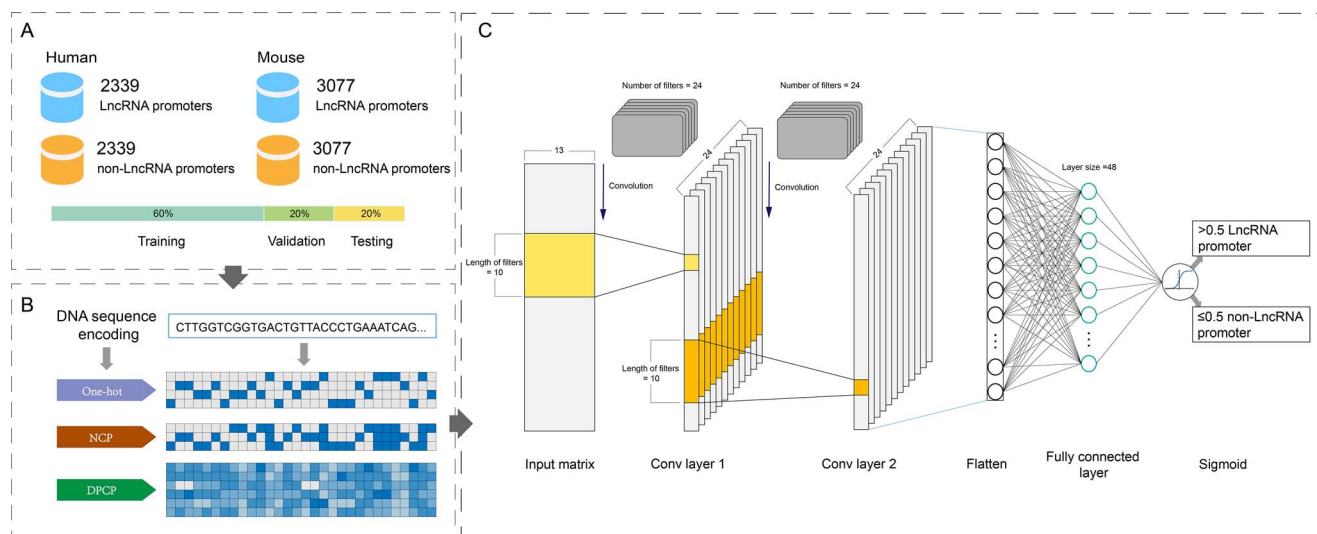


Figure 1. An overview of DeepLncPro. **(A)** Data sets. The dataset contains 2339 positive and 2339 negative samples from human and 3077 positive and 3077 negative samples from mouse. Each sample was intercepted at different lengths from 61 bp to 301 bp with a step of 40 bp. **(B)** Feature encoding. These samples were encoded by using three feature encoding methods. The encoded features were merged into a $13 \times L$ matrix. **(C)** Framework of DeepLncPro. DeepLncPro was built based on convolutional neural network. Each sample got a prediction score, ranging from 0 to 1. If the score was >0.5 , the sequence is predicted as a lncRNA promoter; otherwise, a non-lncRNA promoter.

method was used to scale them into a range of $[0,1]$. Based on the six DPCP, a DNA sequence of length L can be converted into a $6 \times (L-1)$ matrix. In order to make sure that the number of columns of the matrix is the same as that of matrix A_1 and A_2 , a sliding dimer window algorithm was used to calculate the DPCP for each nucleotide [34],

$$\text{DPCP}_n(i) = \frac{X_n(D_{i-1}D_i) + X_n(D_iD_{i+1})}{2} \quad (4)$$

where $\text{DPCP}_n(i)$ represents the n -th physicochemical property for the i -th nucleotide, X_n represents the n -th ($n=1, 2, \dots, 6$) dinucleotide physicochemical properties, which takes the front dimer $D_{i-1}D_i$ and the behind dimer D_iD_{i+1} as input, respectively. The two terminal nucleotides D_1 and D_L only rely on the data of dinucleotides at both ends, respectively. Accordingly, a $6 \times L$ matrix A_3 was obtained, which depicts the sequence in terms of physicochemical properties.

$$A_3 = \begin{bmatrix} \text{DPCP}_1(1) & \cdots & \text{DPCP}_1(L) \\ \vdots & \ddots & \vdots \\ \text{DPCP}_6(1) & \cdots & \text{DPCP}_6(L) \end{bmatrix} \quad (5)$$

Model architectures

In recent years, CNN has been widely used in biological sequence analysis [35–38]. In the present work, we employed CNN to build the DeepLncPro model as well. The implementation of DeepLncPro was based on the deep learning library Pytorch [39]. DeepLncPro contains two 1D convolutional layers with 24 filters with the size of 10, which were determined by performing hyperparameter optimization. Since the rectified linear unit (ReLU) can keep the input values that are positive [40], it was used to combat the vanishing gradient problem. The framework of the proposed model DeepLncPro was shown in Figure 1C.

In DeepLncPro, the convolution operation was equivalent to using a sliding window to extract motifs from the sequence with

high activation values. Hence, the first convolutional layer detects motifs in the sequence, and the second convolutional layer depicts the associations between the extracted motifs from a longer scale [41]. The subsequent layer of the model is a fully connected layer and is used to integrate the information of the whole sequence. Finally, the probability obtained from the sigmoid function was used to make predictions. The first convolutional layer can be mathematically represented as the following [42, 43],

$$\text{Conv}(M)_{ij} = \text{ReLU} \left(\sum_{s=0}^{S-1} \sum_{n=0}^{N-1} W_{s,n}^j M_{i+s,n} \right) \quad (6)$$

where M indicates matrix encoding the sequence, i is the index of the output location and j is the index of the filter. Each convolutional filter W^j is an $S \times N$ matrix, where S is the filter size (determined by hyper-parameter optimization) and N is the number of input channels (determined by encoding strategy). For the first convolutional layer, N is the input dimension and equals to 13 (the combination of the three coding methods). The ReLU is expressed as the following,

$$\text{ReLU}(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \quad (7)$$

Similarity, the convolution of the second layer can be mathematically expressed as,

$$\text{Conv}(M')_{ij} = \text{ReLU} \left(\sum_{s=0}^{S'-1} \sum_{n=0}^{N'-1} W'_{s,n}{}^j M'_{i+s,n} \right) \quad (8)$$

where M' is a $24 \times ((L-S)/k+1)$ matrix combined from the output of the first convolution layer, i is the index of the output location and j is the index of the filter. k is the convolution stride and equals to 1. Each convolutional filter W'^j is an $S' \times N'$ matrix, where S' is the filter size, N' is the number of filters in the first convolutional layer and equals to 24 (determined by hyper-parameter optimization).

Hyper-parameter optimization and model selection

In order to obtain models with better performance and generalization capability, we performed hyper-parameter optimization. To make the training process more stable, the Adam algorithm [44] was applied to automatically determine the learning rate based on the batch gradient descent. The random search method was used to determine the hyperparameters including learning rate, number of neurons, size of convolutional layers and number of filters.

In the hyper-parameter optimization process, we first trained a basic model by selecting a set of hyperparameters within a reasonable range (see details in [Supplementary Table S1](#) available online at <http://bib.oxfordjournals.org/>). Then, by keeping the other hyperparameters fixed, a certain hyperparameter was searched in the given range. According to the performance obtained from the validation dataset, an optimal hyperparameter was selected. This process was repeated until all hyperparameters were optimized. Once all hyperparameters were determined, they were used to train DeepLncPro again on the training and validation datasets. It should be pointed out that only the combination of hyperparameters with the highest accuracy in the validation set was retained.

Performance evaluation

The threshold dependent metrics, namely sensitivity (Sn), specificity (Sp), accuracy (Acc) and Matthew's correlation coefficient (MCC) [45] were used to evaluate the performance of the model, which were defined as the following:

$$\begin{aligned} Sn &= \frac{TP}{TP+FN} \times 100\% \\ Sp &= \frac{TN}{TN+FP} \times 100\% \\ Acc &= \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \\ MCC &= \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}} \end{aligned} \quad (9)$$

where TP, FN, TN and FP denote true positive, false negative, true negative and false positive, respectively.

In addition, we also used the receiver operating characteristic (ROC) curve [46] and the area under the curve (AUC) as the threshold independent metrics to objectively evaluate the performances of DeepLncPro and existing methods.

Motif extraction

In order to make DeepLncPro interpretable, we used the same method as in deepRAM [41] to extract the motifs from its first convolutional layer. For each filter in the first convolutional layer, according to our preliminary test, we extracted sequence segments, which could activate the filter with the activation value greater than 65% of the filter's maximum value. By stacking these segments, we computed the nucleotide frequencies and obtained the position weight matrix (PWM) which was considered as the local motif captured by DeepLncPro. Afterwards, the correlation between the PWM and the transcription factor binding motifs in the JASPAR [47] database was calculated by using TOMTOM [48].

Result and discussion

Effect of sequence length and encoding schemes on model performance

To determine the optimal sequence length and encoding schemes for predicting lncRNA promoters, the effects of sequence

lengths and encoding schemes on the model performance were investigated. For this aim, we built different models based on the combinations of different types of sequence lengths and encoding schemes. In order to obtain a model with satisfactory generalizability, the training data from human and mouse were combined together to train the models. For each model, its hyper-parameters were optimized according to the procedures introduced in Hyper-parameter optimization and model selection section. The accuracies of the models for identifying human and mouse lncRNA promoters in the validation set were shown in [Figure 2](#). The corresponding sensitivity, specificity and Matthew's correlation coefficient were listed in [Supplementary Tables S2](#) and [S3](#) available online at <http://bib.oxfordjournals.org/>. It was found that the model based on the sequence length of 181 bp and the combinations of the three kinds of encoding schemes obtained the best accuracies of 87.07% and 87.73% for identifying lncRNA promoters in both human and mouse, respectively. Accordingly, based on the above obtained optimal sequence length (181 bp), combinational encoding method and the best hyper-parameters ([Supplementary Table S4](#) available online at <http://bib.oxfordjournals.org/>), the DeepLncPro was developed for predicting lncRNA promoters in both human and mouse. In addition, we also evaluated the models trained by the data either from human or mouse and reported the results in [Supplementary Figure S1](#), [Supplementary Tables S5](#) and [S6](#) available online at <http://bib.oxfordjournals.org/>. The obtained results demonstrated that the performances of these models were all lower than that of DeepLncPro.

Comparison with classical machine learning methods

Considering that machine learning methods were widely used in DNA sequence elements identification, we compared DeepLncPro with five classical ML methods, namely random forest (RF), logistic regression (LR), k-nearest neighbors (KNN), support vector machine (SVM) and eXtreme Gradient Boosting (XGBoost). The three input matrices of DeepLncPro were flattened into a 13 L dimensional vector and used as the input of RF, LR, KNN, SVM and XGBoost. The evaluating metrics of DeepLncPro and ML models for identifying human and mouse lncRNA promoters in the test dataset were listed in [Table 2](#). DeepLncPro obtained the best accuracies of 86.21% and 86.82% for identifying human and mouse lncRNA promoters, respectively. We also plotted the ROC curves of DeepLncPro and the machine learning methods in [Figure 3](#). It was found that DeepLncPro obtained the best AUCs of 0.928 and 0.931, and outperformed the other machine learning models for predicting lncRNA promoters in both human and mouse.

Comparison with the existing predictor

To further illustrate its superiority, we compared DeepLncPro with the existing predictor ncPro-ML [24]. For a fair comparison, the two predictors were all validated on the same test set. As shown in [Table 3](#), the accuracy of DeepLncPro were 4.57% and 3.74% higher than that of ncPro-ML for identifying human and mouse lncRNA promoters, respectively. The corresponding sensitivity, specificity and Matthew's correlation coefficient were improved 8.47%, 0.65% and 0.10 in human, and 7.12%, 0.36% and 0.08 in mouse, respectively. These results demonstrated that the DeepLncPro is more superior to identify human and mouse lncRNA promoters.

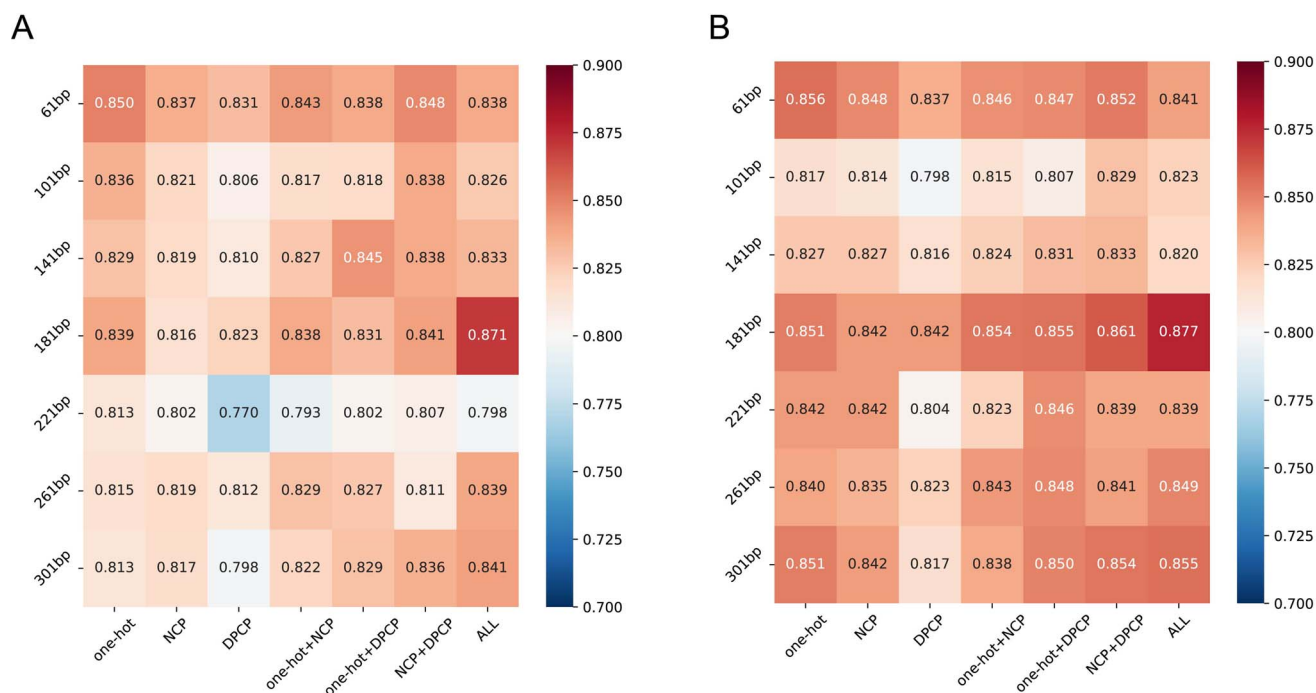


Figure 2. Performance of the models based on different sequence lengths and encoding schemes. The vertical coordinate represents the sequence length ranging from 61 to 301 bp. The horizontal coordinate represents different encoding schemes, including one-hot, NCP, DPCP and their combinations. **(A)** The predictive accuracies of different models for identifying lncRNA promoters in human; **(B)** The predictive accuracies of different models for identifying lncRNA promoters in mouse.

Table 2. Performance of DeepLncPro and different machine learning models for identifying lncRNA promoters in test set

Method	Species	Sn(%)	Sp(%)	Acc(%)	MCC
RF	Human	85.90%	83.55%	84.72%	0.69
	Mouse	82.60%	83.74%	83.17%	0.66
LR	Human	83.97%	78.63%	81.30%	0.63
	Mouse	83.25%	81.63%	82.44%	0.65
KNN	Human	81.41%	62.18%	71.79%	0.44
	Mouse	86.83%	66.50%	76.67%	0.54
SVM	Human	82.91%	79.49%	81.20%	0.62
	Mouse	84.72%	85.37%	85.04%	0.70
XGBoost	Human	85.26%	83.33%	84.29%	0.69
	Mouse	85.69%	85.53%	85.61%	0.71
CNN	Human	89.74%	82.69%	86.22%	0.73
	Mouse	88.78%	84.88%	86.83%	0.74

Table 3. Comparison of the prediction performance of DeepLncPro with ncPro-ML based on the test set

Species	Name	Sn(%)	Sp(%)	Acc(%)	MCC
Human	ncPro-ML	81.27%	82.04%	81.65%	0.63
	DeepLncPro	89.74%	82.69%	86.22%	0.73
Mouse	ncPro-ML	81.66%	84.52%	83.09%	0.66
	DeepLncPro	88.78%	84.88%	86.83%	0.74

Model interpretation and visualization

To explain the performance of the proposed model, we extracted and visualized the inputs and outputs from all layers of DeepLncPro, namely the original inputs, outputs of the first convolutional layer, outputs of the second convolutional layer and outputs of the fully connected layer. To facilitate understanding these features, the UMAP [49] was used to show the distribution

of positive and negative samples. It was found that the positive and negative samples couldn't be separated in the feature space formed by the original input features (Figure 4A). However, the margins between positive and negative samples were more clearly separated in the feature space based on the output features of the first and second convolutional layers (Figure 4B and C). The positive and negative samples could be more clearly

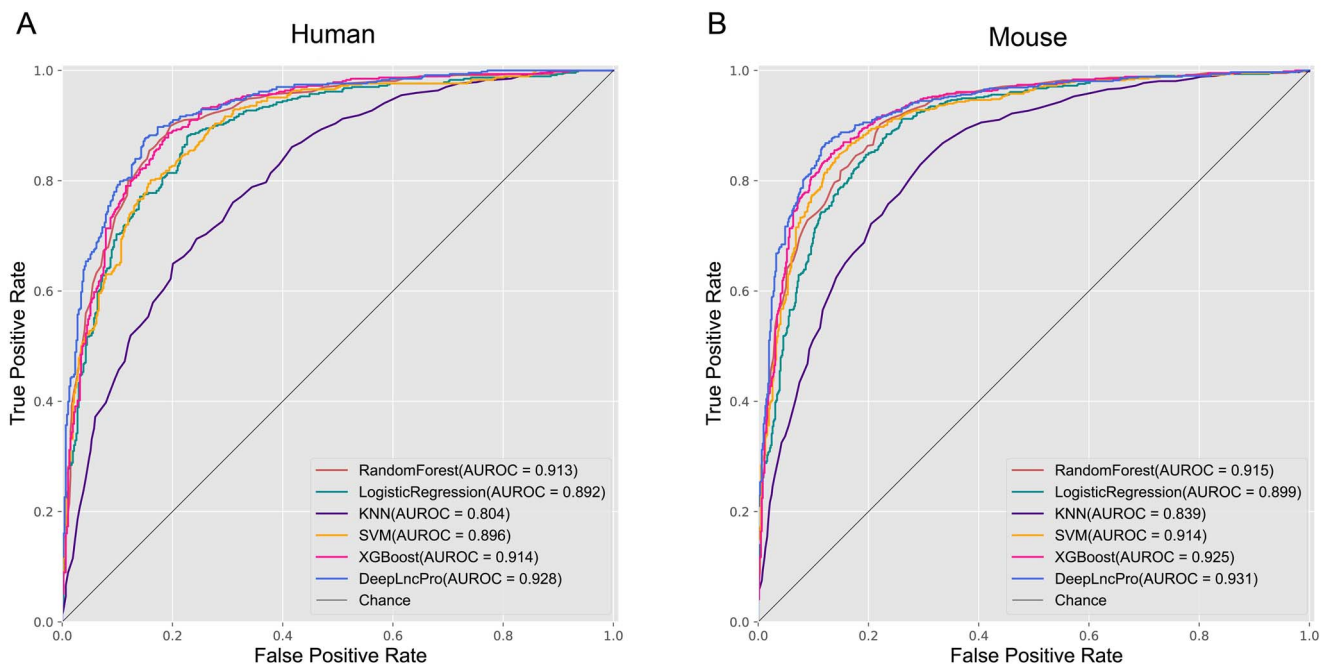


Figure 3. The ROC curves of DeepLncPro, RF, LR, KNN, SVM and XGBoost validated in the test dataset. (A) The ROC curves for identifying lncRNA promoters in human. (B) The ROC curves for identifying lncRNA promoters in mouse.

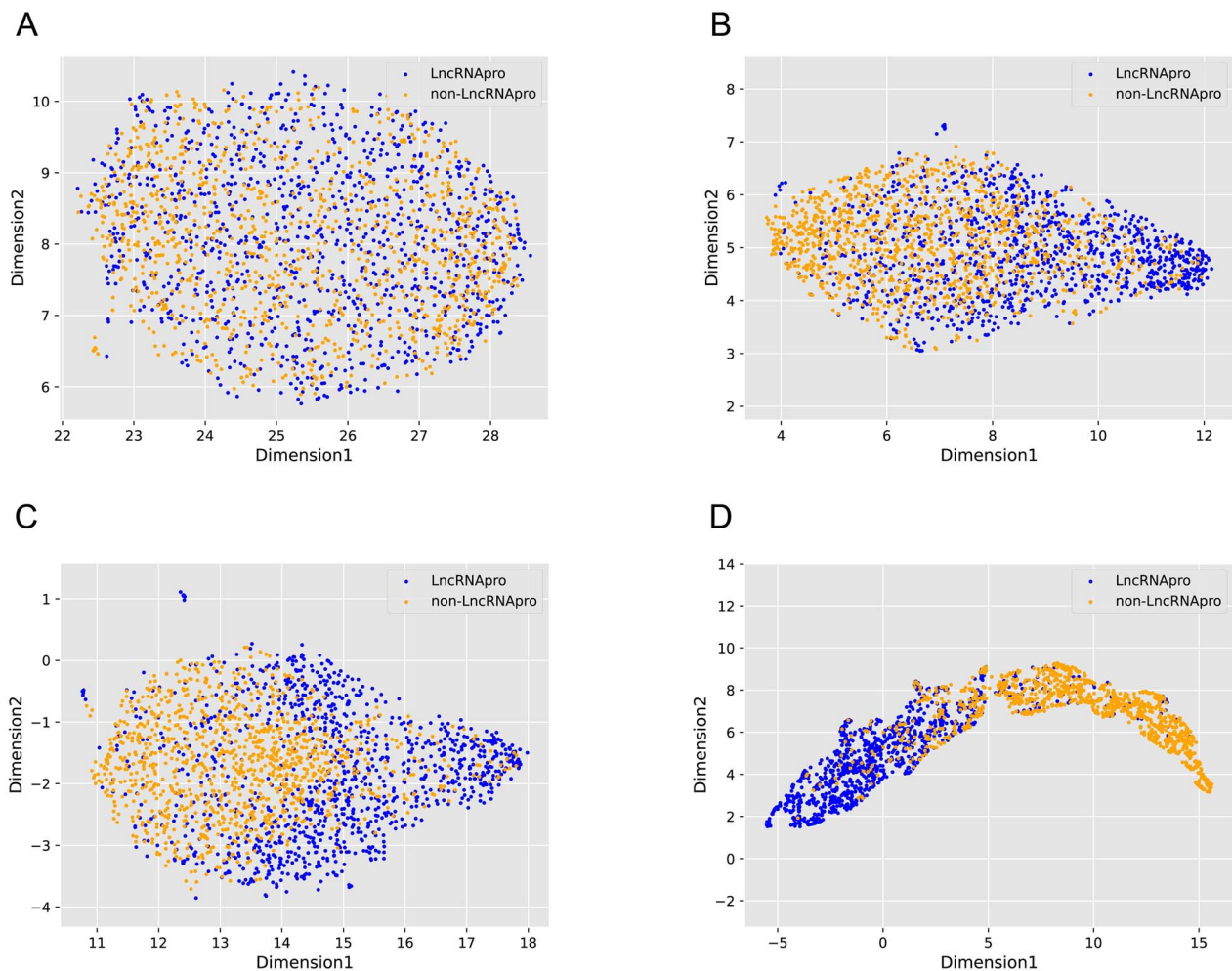


Figure 4. Distribution of positive and negative samples in the 2D feature space. The blue and orange dots represent positive and negative samples, respectively. (A) The feature space of the original input features. (B) The feature space based on the outputs of the first convolutional layer. (C) The feature space based on the outputs of the second convolutional layer. (D) The feature space based on the outputs of the fully connected layer.

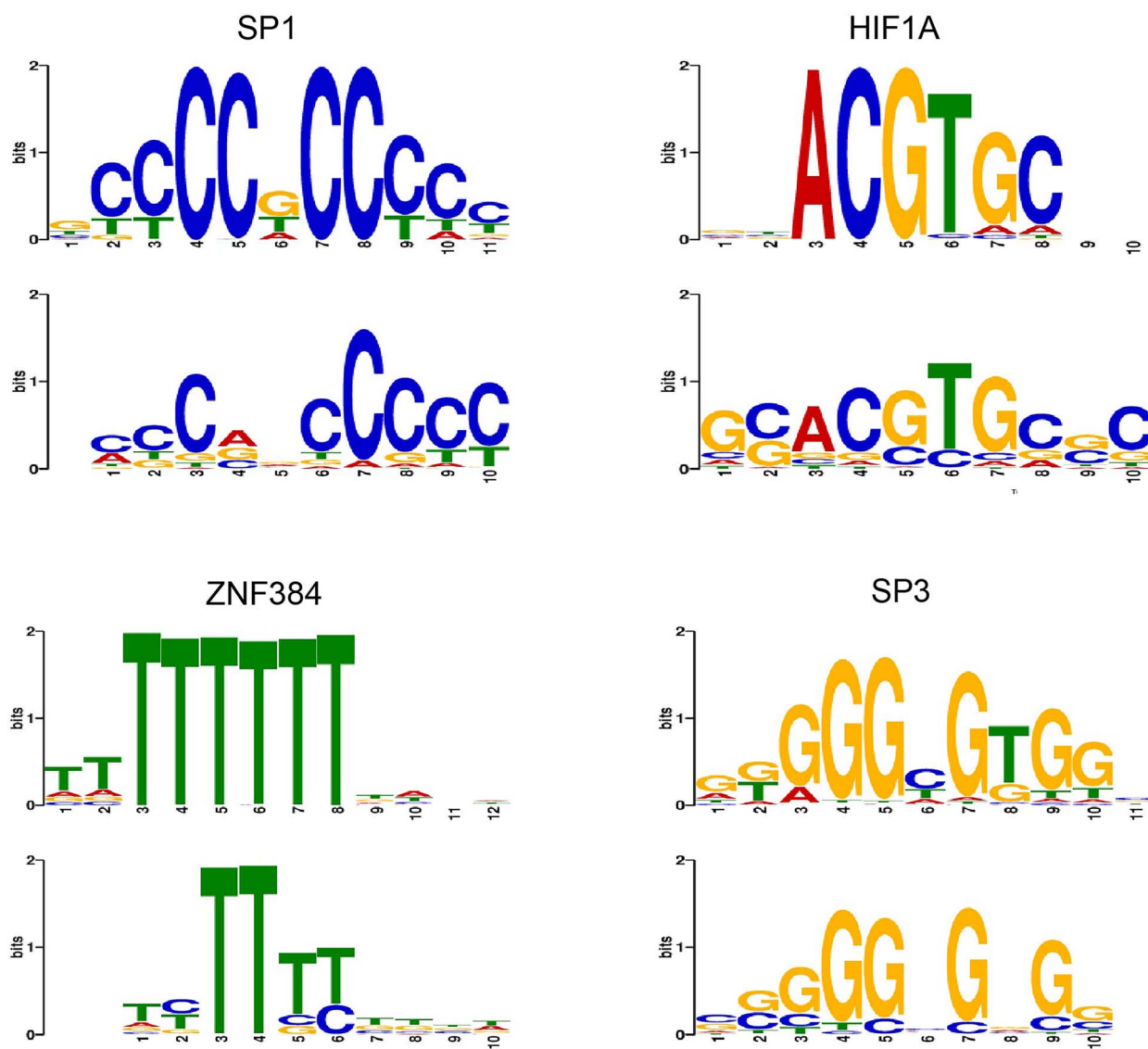


Figure 5. The four representative motifs extracted by DeepLncPro in human lncRNAs. The motifs correspond to the binding sites of the transcription factors SP1 ($P=9.17e-06$), HIF1A ($P\text{-value}=1.61e-06$), ZNF384 ($P\text{-value}=3.83e-05$) and SP3 ($P\text{-value}=6.89e-05$). In each case, the top panel was the known motif in the JASPAR database, and the bottom panel was the motif extracted by DeepLncPro.

separated based on the output features of the fully connected layer (Figure 4D). These results demonstrated the ability of the proposed model in extracting potential features, which help to learn a better decision margin for identifying lncRNA promoters.

To demonstrate the ability of DeepLncPro in capturing informative motifs, we calculated the PWM (see Motif extraction section for details) to analyze the extracted motifs from the 24 filters of the first convolutional layer. The TOMTOM was then used to map the motifs learned from each filter to known transcription factor (TF) binding motifs in the JASPAR database. Finally, we obtained 87 and 85 known motifs in JASPAR with $P\text{-value} < 0.05$ in human and mouse (Supplementary Tables S7 and S8 available online at <http://bib.oxfordjournals.org/>), respectively. The representative binding motifs of the four TFs (SP1; HIF1A, ZNF384 and SP3) obtained from human and mouse were shown in Figure 5 and Supplementary Figure S2 available online at <http://bib.oxfordjournals.org/>. In each case, the top panel was the known motif in the JASPAR database, and the bottom panel

was the motif extracted by DeepLncPro. It was observed that the representative motifs in mouse were very similar to those in human. As indicated by the hTFtarget database [50], the transcription factors SP1, HIF1A, ZNF384 and SP3 were all involved in the regulation of lncRNA expression, which demonstrated the biological significance of DeepLncPro.

Conclusion

lncRNA plays important regulatory roles in various biological processes. Accurate identification of lncRNA promoter is helpful to understand its regulatory mechanisms. In order to improve the model performance and provide model explainability in promoter prediction, we proposed a deep learning based model, called DeepLncPro, to identify lncRNA promoters in human and mouse. A series of comparative experiments demonstrated that DeepLncPro is superior to the state-of-the-art machine learning methods and existing models for identifying lncRNA promoters.

The excellent performance of DeepLncPro is attributed to the informative features extracted from the convolutional layers. By mapping these features to JASPAR database, it was found that they are the known transcription factor binding motifs, which provides the interpretability of DeepLncPro. An open-source tool for DeepLncPro was provided at <https://github.com/zhangtian-yang/DeepLncPro>, which will stimulate further studies on lncRNA promoter identification.

It should be pointed out that only the sequence-derived information was used in DeepLncPro, which is not enough to capture the information depicting promoters. It has been reported that the data from both ATAC-seq and ChIP-seq are also key signals in promoter regions [51, 52]. Therefore, in the future work, we need to collect and integrate these data for identifying lncRNA promoters.

Authors' contributions

W.C. and T.Y.Z. conceived and designed the work. T.Y.Z., Q.T. and F.L.N. performed the data collection and analysis, visualized the results. T.Y.Z. and Q.T. collected the data. T.Y.Z., Q.Z. and W.C. wrote the manuscript. All authors read and approved the final manuscript.

Data availability

The data and code that support the findings of this study are available at <https://github.com/zhangtian-yang/DeepLncPro>.

Key Points

- A convolutional neural network based model, called DeepLncPro, is proposed to identify human and mouse lncRNA promoters.
- Comparative studies demonstrated that DeepLncPro outperforms existing models for identification of lncRNA promoters.
- DeepLncPro is capable of capturing transcription factor binding sites, which facilitates its biological interpretation.
- An open-source tool for DeepLncPro is provided at <https://github.com/zhangtian-yang/DeepLncPro>.

Supplementary data

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

Funding

Natural Science Foundation of Sichuan (No. 2022NSFSC1770), National Natural Science Foundation of China (No. 31771471), Natural Science Foundation of Foundation of Education Department of Liaoning Province (No. LJKZ0280).

References

1. Boon RA, Jae N, Holdt L, et al. Long noncoding RNAs: from clinical genetics to therapeutic targets? *J Am Coll Cardiol* 2016;**67**:1214–26.
2. Zhu J, Fu H, Wu Y, et al. Function of lncRNAs and approaches to lncRNA-protein interactions. *Sci China Life Sci* 2013;**56**:876–85.
3. Rinn JL, Chang HY. Long noncoding RNAs: molecular modalities to organismal functions. *Annu Rev Biochem* 2020;**89**:283–308.
4. Tripathi V, Shen Z, Chakraborty A, et al. Long noncoding RNA MALAT1 controls cell cycle progression by regulating the expression of oncogenic transcription factor B-MYB. *PLoS Genet* 2013;**9**:e1003368.
5. Tripathi V, Ellis JD, Shen Z, et al. The nuclear-retained noncoding RNA MALAT1 regulates alternative splicing by modulating SR splicing factor phosphorylation. *Mol Cell* 2010;**39**:925–38.
6. Stavropoulos N, Lu N, Lee JT. A functional role for Tsix transcription in blocking Xist RNA accumulation but not in X-chromosome choice. *Proc Natl Acad Sci U S A* 2001;**98**:10232–7.
7. Kretz M, Siprashvili Z, Chu C, et al. Control of somatic tissue differentiation by the long non-coding RNA TINCR. *Nature* 2013;**493**:231–5.
8. Gong C, Maquat LE. lncRNAs transactivate STAU1-mediated mRNA decay by duplexing with 3' UTRs via Alu elements. *Nature* 2011;**470**:284–8.
9. Mourtada-Maarabouni M, Pickard MR, Hedge VL, et al. GAS5, a non-protein-coding RNA, controls apoptosis and is downregulated in breast cancer. *Oncogene* 2009;**28**:195–208.
10. Esposito R, Bosch N, Lanzos A, et al. Hacking the cancer genome: profiling therapeutically actionable long non-coding RNAs using CRISPR-Cas9 screening. *Cancer Cell* 2019;**35**:545–57.
11. Kim J, Piao HL, Kim BJ, et al. Long noncoding RNA MALAT1 suppresses breast cancer metastasis. *Nat Genet* 2018;**50**:1705–15.
12. McPherson R, Pertsemliadis A, Kavaslar N, et al. A common allele on chromosome 9 associated with coronary heart disease. *Science* 2007;**316**:1488–91.
13. Johnson R. Long non-coding RNAs in Huntington's disease neurodegeneration. *Neurobiol Dis* 2012;**46**:245–54.
14. Kopp F, Mendell JT. Functional classification and experimental dissection of long noncoding RNAs. *Cell* 2018;**172**:393–407.
15. Bansal M, Kumar A, Yella VR. Role of DNA sequence based structural features of promoters in transcription initiation and gene expression. *Curr Opin Struct Biol* 2014;**25**:77–85.
16. Fulton DL, Sundararajan S, Badis G, et al. TFCat: the curated catalog of mouse and human transcription factors. *Genome Biol* 2009;**10**:R29.
17. Valen E, Sandelin A. Genomic and chromatin signals underlying transcription start-site selection. *Trends Genet* 2011;**27**:475–85.
18. Dahl JA, Collas P. MicroChIP—a rapid micro chromatin immunoprecipitation assay for small cell samples and biopsies. *Nucleic Acids Res* 2008;**36**:e15.
19. Kim JW, Zeller KI, Wang Y, et al. Evaluation of myc E-box phylogenetic footprints in glycolytic genes by chromatin immunoprecipitation assays. *Mol Cell Biol* 2004;**24**:5923–36.
20. Lin H, Deng EZ, Ding H, et al. iPro54-PseKNC: a sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition. *Nucleic Acids Res* 2014;**42**:12961–72.
21. Oubounyt M, Louadi Z, Tayara H, et al. DeePromoter: robust promoter predictor using deep learning. *Front Genet* 2019;**10**:286.
22. Wang S, Cheng X, Li Y, et al. Image-based promoter prediction: a promoter prediction method based on evolutionarily generated patterns. *Sci Rep* 2018;**8**:17695.
23. Alam T, Islam MT, Househ M, et al. DeepCNPP: deep learning architecture to distinguish the promoter of human long non-coding RNA genes and protein-coding genes. *Stud Health Technol Inform* 2019;**262**:232–5.
24. Tang Q, Nie F, Kang J, et al. ncPro-ML: an integrated computational tool for identifying non-coding RNA promoters in multiple species. *Comput Struct Biotechnol J* 2020;**18**:2445–52.

25. Meylan P, Dreos R, Ambrosini G, et al. EPD in 2020: enhanced data visualization and extension to ncRNA promoters. *Nucleic Acids Res* 2020;**48**:D65–9.
26. Sahu B, Hartonen T, Pihlajamaa P, et al. Sequence determinants of human gene regulatory elements. *Nat Genet* 2022;**54**:283–94.
27. Xiong Y, He X, Zhao D, et al. Modeling multi-species RNA modification through multi-task curriculum learning. *Nucleic Acids Res* 2021;**49**:3719–34.
28. Li K, Carroll M, Vafabakhsh R, et al. DNAcycP: a deep learning tool for DNA cyclizability prediction. *Nucleic Acids Res* 2022;**50**:3142–54.
29. Chen W, Yang H, Feng P, et al. iDNA4mC: identifying DNA N4-methylcytosine sites based on nucleotide chemical properties. *Bioinformatics* 2017;**33**:3518–23.
30. Guo SH, Deng EZ, Xu LQ, et al. iNuc-PseKNC: a sequence-based predictor for predicting nucleosome positioning in genomes with pseudo k-tuple nucleotide composition. *Bioinformatics* 2014;**30**:1522–9.
31. Brick K, Watanabe J, Pizzi E. Core promoters are predicted by their distinct physicochemical properties in the genome of *Plasmodium falciparum*. *Genome Biol* 2008;**9**:R178.
32. Abeel T, Saeys Y, Bonnet E, et al. Generic eukaryotic core promoter prediction using structural features of DNA. *Genome Res* 2008;**18**:310–23.
33. Chen W, Zhang X, Brooker J, et al. PseKNC-General: a cross-platform package for generating various modes of pseudo nucleotide compositions. *Bioinformatics* 2015;**31**:119–20.
34. Greenbaum JA, Pang B, Tullius TD. Construction of a genome-scale structural map at single-nucleotide resolution. *Genome Res* 2007;**17**:947–53.
35. Aoki G, Sakakibara Y. Convolutional neural networks for classification of alignments of non-coding RNA sequences. *Bioinformatics* 2018;**34**:i237–44.
36. Abbas Z, Tayara H, Chong KT. 4mCPred-CNN-prediction of DNA N4-methylcytosine in the mouse genome using a convolutional neural network. *Genes (Basel)* 2021;**12**:296.
37. Liu K, Cao L, Du P, et al. im6A-TS-CNN: identifying the N(6)-methyladenine site in multiple tissues by using the convolutional neural network. *Mol Ther Nucleic Acids* 2020;**21**:1044–9.
38. Yang S, Wang Y, Lin Y, et al. LncMirNet: predicting LncRNA-miRNA interaction based on deep learning of ribonucleic acid sequences. *Molecules* 2020;**25**:4372.
39. Paszke A, Gross S, Massa F et al. PyTorch: an imperative style, high-performance deep learning library. In: 2019 *Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada: NIPS, 2019. 8026–37.
40. Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. In: 2010 *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, Ft. Lauderdale, FL, USA: PMLR, 2011. 315–23.
41. Trabelsi A, Chaabane M, Ben-Hur A. Comprehensive evaluation of deep learning architectures for prediction of DNA/RNA sequence binding specificities. *Bioinformatics* 2019;**35**:i269–77.
42. Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*. *IEEE Neural Networks Council* 1994;**5**:157–66.
43. Mattioli F, Porcaro C, Baldassarre G. A 1D CNN for high accuracy classification and transfer learning in motor imagery EEG-based brain-computer interface. *J Neural Eng* 2021;**18**:006053.
44. Kingma D, Ba J. Adam: a method for stochastic optimization. In: 2015 *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA. arXiv:1412.6980.
45. Tang Q, Nie F, Kang J, et al. mRNALocator: Enhance the prediction accuracy of eukaryotic mRNA subcellular localization by using model fusion strategy. *Mol Ther* 2021;**29**:2617–23.
46. Hirschfeld G, von Glischinski M, Thiele C. Optimal cycle thresholds for coronavirus disease 2019 (COVID-19) screening-receiver operating characteristic (ROC)-based methods highlight between-study differences. *Clin Infect Dis* 2021;**73**:e852–3.
47. Mathelier A, Zhao X, Zhang AW, et al. JASPAR 2014: an extensively expanded and updated open-access database of transcription factor binding profiles. *Nucleic Acids Res* 2014;**42**:D142–7.
48. Gupta S, Stamatoyannopoulos JA, Bailey TL, et al. Quantifying similarity between motifs. *Genome Biol* 2007;**8**:R24.
49. Becht E, McInnes L, Healy J, et al. Dimensionality reduction for visualizing single-cell data using UMAP. *Nat Biotechnol* 2018;**37**:38–44.
50. Zhang Q, Liu W, Zhang HM, et al. hTFtarget: a comprehensive database for regulations of human transcription factors and their targets. *Genom Proteom Bioinform* 2020;**18**:120–8.
51. Li YY, Li XC, Yang YS, et al. TRlnc: a comprehensive database for human transcriptional regulatory information of lncRNAs. *Brief Bioinform* 2021;**22**(2):1929–39.
52. Wang F, Bi XF, Wang YZ, et al. ATACdb: a comprehensive human chromatin accessibility database. *Nucleic Acids Res* 2021;**49**(D1):D55–64.