OXFORD

*Briefings in Bioinformatics*, 23(1), 2022, 1–10

https://doi.org/10.1093/bib/bbab360
Problem Solving Protocol

# DeepLncLoc: a deep learning framework for long non-coding RNA subcellular localization prediction based on subsequence embedding

Min Zeng, Yifan Wu, Chengqian Lu, Fuhao Zhang, Fang-Xiang Wu and Min Li

Corresponding author. Min Li, Hunan Provincial Key Lab on Bioinformatics, School of Computer Science and Engineering, Central South University, Changsha, Hunan 410083, China. E-mail: limin@mail.csu.edu.cn

## Abstract

Long non-coding RNAs (lncRNAs) are a class of RNA molecules with more than 200 nucleotides. A growing amount of evidence reveals that subcellular localization of lncRNAs can provide valuable insights into their biological functions. Existing computational methods for predicting lncRNA subcellular localization use *k*-mer features to encode lncRNA sequences. However, the sequence order information is lost by using only *k*-mer features. We proposed a deep learning framework, DeepLncLoc, to predict lncRNA subcellular localization. In DeepLncLoc, we introduced a new subsequence embedding method that keeps the order information of lncRNA sequences. The subsequence embedding method first divides a sequence into some consecutive subsequences and then extracts the patterns of each subsequence, last combines these patterns to obtain a complete representation of the lncRNA sequence. After that, a text convolutional neural network is employed to learn high-level features and perform the prediction task. Compared with traditional machine learning models, popular representation methods and existing predictors, DeepLncLoc achieved better performance, which shows that DeepLncLoc could effectively predict lncRNA subcellular localization. Our study not only presented a novel computational model for predicting lncRNA subcellular localization but also introduced a new subsequence embedding method which is expected to be applied in other sequence-based prediction tasks. The DeepLncLoc web server is freely accessible at http://bioinformatics.csu.edu.cn/DeepLncLoc/, and source code and datasets can be downloaded from https://github.com/CSUBioGroup/DeepLncLoc.

**Key words:** lncRNA; subcellular localization prediction; deep learning; subsequence embedding

**Min Zeng** received the BS degree from Lanzhou University in 2013 and the MS and PhD degrees in system science and computer science from Central South University in 2016 and 2020, respectively. He is currently an Assistant Professor in the School of Computer Science and Engineering, Central South University. His main research interests include bioinformatics, machine learning and deep learning.
**Yifan Wu** received the Bachelor degree of computer science from Jishou University, Jishou, China. He is currently a master student in the School of Computer Science and Engineering, Central South University. His current research interests include bioinformatics and deep learning.
**Chengqian Lu** received his PhD degree in computer science from Central South University in 2019. His current research interests include bioinformatics and deep learning.
**Fuhao Zhang** is working toward the PhD degree in computer science in Central South University, Changsha, China. His current research interests include bioinformatics, machine learning and deep learning.
**Fang-Xiang Wu** is a Professor in the Division of Biomedical Engineering and the Department of Mechanical Engineering at the University of Saskatchewan, Saskatoon, Canada. His current research interests include artificial intelligence, systems biology and bioinformatics.
**Min Li** received the BS degree in communication engineering and the MS and PhD degrees in computer science from Central South University, Changsha, China, in 2001, 2004 and 2008, respectively. She is currently a Professor at the School of Computer Science and Engineering, Central South University. Her main research interests include bioinformatics and system biology.
**Submitted:** 4 June 2021; **Received (in revised form):** 4 August 2021

## Introduction

Non-coding RNAs have attracted lots of attention from researchers and are associated with the development of various human diseases [1, 2]. Long non-coding RNAs (lncRNAs) are a type of non-coding RNA molecules (more than 200 nucleotides) that are transcribed from DNA but not translated into proteins [3, 4]. LncRNAs play an important role in various biological processes including regulation of gene expression, alternative splicing, nuclear organization and genomic imprinting [5–7]. For example, lncRNAs can bind to DNAs, RNAs and proteins, and then perform their functions through these interactions [8]. LncRNAs can act as 'miRNA sponge' to regulate the level of miRNA and then affect the expression of miRNA's target [9]. LncRNAs can regulate transcriptional activity or pathways under specific stimulation [10]. Due to the complexity of molecular functions, lncRNA-related studies are drawing increasing attention [11].

A growing amount of evidence reveals that the subcellular localization of biomacromolecules can provide valuable insights into their functions [12–14]. For example, lncRNA 'XIST', which locates in nucleus, interacts with the nuclear-matrix factor hnRNPU and modulates nuclear architecture and trans-chromosomal interactions [15]. LncRNA 'lincRNA-p21', which locates in cytoplasm, regulates JUNB and CTNNB1 translation in HeLa cells [16]. LncRNA 'ZFAS1', which locates in ribosome, regulates mRNAs encoding proteins from the ribosomal complex [17]. Thus, identification of lncRNA subcellular localizations is very important to understand lncRNA functions [18]. Recently, some large databases of RNA-associated subcellular localization were released. Zhang *et al.* published a database, RNALocate [19], to collect the subcellular localization of different kinds of RNAs, which contains more than 23 100 RNAs with 42 subcellular localizations in 65 species. Mas Ponte *et al*. developed a database called LncATLAS to display the subcellular localization of lncRNAs [20]. Wen *et al.* [21] created a lncRNA subcellular localization database called lncSLdb, which collects 14 973 subcellular localization information of lncRNAs from three species (human, mouse and fruitfly).

However, only a few computational predictors for lncRNA subcellular localization have been proposed. To the best of our knowledge, the first predictor is lncLocator [22]. LncLocator uses 4-mer features and high-level features extracted by stacked autoencoder, and feeds the two kinds of features into two kinds of classifiers [support vector machine (SVM) and random forest (RF)], respectively. Then, lncLocator uses an ensemble strategy to combine the results of different classifiers and get the final prediction. In their training process, lncLocator utilizes a supervised over-sampling algorithm to balance the ratio of different classes. After that, Su *et al.* [23] proposed iLoc-lncRNA which uses 8-mer features to encode lncRNA sequences. Considering the dimension of 8-mer features is too large, iLoc-lncRNA applies a feature selection method based on binomial distribution to select the most optimal features. Then, iLoc-lncRNA feeds the most optimal features into SVM to get the prediction results. Gudenas and Wang proposed DeepLncRNA [24] which uses 2, 3, 4 and 5-mer features to encode lncRNA sequences and adds additional features (RNA–binding motifs and genomic loci). Then, the combined features are fed into a neural network (NN) to obtain the final prediction. Fan *et al.* [25] developed lncLocPred, which selects important features of 5, 6, 8-mer features and combines triplet and PseDNC features. Last, lncLocPred applies a logistic regression (LR) model to make predictions. Wang *et al.* [26] developed an integration SVM model, which uses multiple sequence features including *k*-mer, reverse compliment *k*-mer, nucleic acid composition, di-nucleotide composition, tri-nucleotide composition and *k*-spaced nucleic acid pair to predict multiply subcellular localizations.

Although these computational predictors achieve decent performance, several improvements can still be made. Encoding raw lncRNA sequences into discriminative features is very important in developing machine learning models. The flaw of these predictors is the use of *k*-mer features to encode raw lncRNA sequences. Apparently, using *k*-mer features cannot keep the sequence order information of the raw lncRNA sequence.

To overcome the limitation, we developed DeepLncLoc, a new deep learning-based predictor for subcellular localization of lncRNAs. In the predictor, we proposed a new feature embedding method that keeps the order information of lncRNA sequences. The main idea of the new feature embedding method is encoding a complete lncRNA sequence by using the combination of its subsequence embeddings. In DeepLncLoc, we first divided a sequence into some consecutive subsequences and then extracted the patterns of each subsequence by using an average pooling layer, last combined these patterns to obtain a complete representation of the lncRNA sequence. After obtaining the complete representation, a text convolutional NN (textCNN) was applied to learn high-level features and perform the prediction task. Different from traditional machine learning models with *k*-mer features in previous studies, DeepLncLoc has two advantages: (i) by using the new subsequence embedding method, the input lncRNA sequence keeps the sequence order information; (ii) textCNN has a more powerful capability of high-level feature extraction.

We conducted extensive experiments to evaluate the performance of DeepLncLoc. Comparison with traditional machine learning models using different *k*-mer features demonstrated the advantages of using deep learning structure instead of using traditional machine learning models. Comparison with some popular representation methods indicated the advantages of using subsequence embedding to encode the whole lncRNA sequence. Comparison with existing predictors on an independent test set showed the capability of DeepLncLoc to predict subcellular localization of lncRNAs. Moreover, we investigated the effects of different species. Finally, we developed a user-friendly web server. We anticipate that DeepLncLoc will serve as a useful bioinformatics tool for accurate prediction of lncRNA subcellular localization.

## Materials and Methods

### Datasets

Similar to previous studies, we retrieved known subcellular localization information of lncRNA from RNALocate database [19]. The current version of RNALocate collects 42 190 manually curated RNA-associated subcellular localization entries with experimental evidence. It contains more than 23 100 RNAs with 42 subcellular localizations in 65 species. We generated a benchmark dataset to train and test our model by the following procedure:

(i) All 42 190 manually curated RNA-associated subcellular localization entries are downloaded from RNAlocate database.

Table 1. Distribution of the constructed benchmark dataset

| Subcellular localization | # of samples |
|---|---|
| Cytoplasm | 328 |
| Nucleus | 325 |
| Ribosome | 88 |
| Cytosol | 88 |
| Exosome | 28 |
| Total | 857 |

(ii) Total 2383 manually curated lncRNA-associated subcellular localization entries are selected from 42 190 manually curated RNA-associated subcellular localization entries.

(iii) Some lncRNAs have multiple entries in the extracted entries; we merged these entries with the same gene name. Then, we removed the lncRNAs that do not have sequence information in NCBI and Ensembl.

(iv) Because most lncRNAs only have one subcellular localization, we selected the lncRNAs that are located in one location for model construction in the study.

(v) The filtered dataset covers seven different subcellular localizations. Two of seven subcellular localizations only have a very small number of samples (less than 10). Thus, we removed these lncRNAs that are located in the two subcellular localizations.

Finally, we constructed a benchmark dataset of 857 lncRNAs, covering 5 subcellular localizations including nucleus, cytosol, ribosome, cytoplasm and exosome (see Supplementary Figure S1, see Supplementary Data available online at http://bib.oxfordjournals.org/). Table 1 lists the distribution of the constructed benchmark dataset.

## Limitations of using only *k*-mer features to encode RNA sequences

Before putting raw RNA sequences into a machine learning or deep learning model, RNA sequences need to be encoded as numeric vectors. There are two kinds of widely used RNA sequence embedding methods. The first one is encoding each nucleotide into a 4-dimensional one-hot vector. The A, C, G and U are encoded with a one-hot vector of (1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0) and (0, 0, 0, 1), respectively (Pan *et al*., 2019). Then, the four types of vectors are used to encode RNA sequences. However, using one-hot encoding has two disadvantages in practice. The first disadvantage is that one-hot vector is sparse, i.e. only a small fraction of features contribute to the prediction task. The second disadvantage is that using one-hot encoding is difficult to accurately represent the similarity between different nucleotides. The second method is using *k*-mer features to encode RNA sequences. The *k*-mer feature encoding method is very simple to implement, and it maps lncRNA sequences with variable-length to a vector with a fixed dimension. Thus, *k*-mer feature encoding method is the most widely used method in the prediction of lncRNA subcellular localization. Previous methods (LncLocator [22], iLoc-lncRNA [23], DeepLncRNA [24], lncLocPred [25] and Wang *et al*. [26]) use *k*-mer features for lncRNA embedding. Formally, we assume a lncRNA sequence is represented as

$$\text{lncRNA} = B_1, B_2, B_3, \ldots, B_{L-1}, B_L, \tag{1}$$

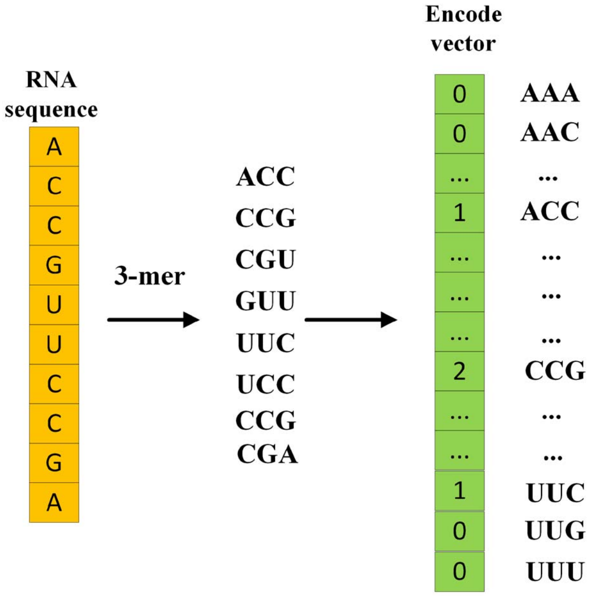

Figure 1. Illustration of the *k*-mer encoding method for single RNA sequence, where *k* is set to 3. The example RNA sequence is 'ACCGUUCCGA', and its 3-mer features are {ACC, CCG, CGU, GUU, UUC, UCC, CCG, CGA}. It should be noted that 'CCG' appears twice, while other 3-mer features (such as 'ACC') appear only once in the 3-mer features, and thus, the vector position which corresponds to 'GGG' is 2 and other 3-mers (such as 'ACC') is 1.

where $L$ denotes the length of the lncRNA; $B_j$ is one of the four nucleotide bases (A, C, G and U) in the $j$ position of the lncRNA sequence.

For a given *k*, *k*-mer features represent the frequency of individual *k*-mer from lncRNA sequences. We take 3-mer as an example, each position can take four nucleotide bases (A, C, G and U), and thus, we have $4^3$, i.e. 64 3-mer features (AAA, AAC, ..., UUG, UUU). Then, we can use a 64-dimensional vector to represent a lncRNA sequence, and each dimension is used to record the frequency of a certain 3-mer. Figure 1 plots the 3-mer encoding method for a single RNA sequence. The *k*-mer feature encoding method is very simple to understand and implement. But there is a disadvantage of using *k*-mer features. Namely, *k*-mer feature encoding method lost order information of the raw lncRNA sequence. *k*-mer features encoding method is only concerned with the occurrence of the *k*-mer and ignores the position of *k*-mer in the raw lncRNA sequence. For example, RNA A is 'ACACACGCGC', 3-mer features of RNA A are {ACA, CAC, ACA, CAC, ACG, CGC, GCG, CGC}, we reverse the RNA sequence to obtain RNA B 'CGCGCACACA' and the 3-mer features of RNA B are {CGC, GCG, CGC, GCA, CAC, ACA, CAC, ACA}. It can be seen that the order of the two RNA sequences is reversed, but their 3-mer features are very similar. The difference between the two 3-mer features is only one 3-mer ('ACG' in RNA A versus 'GCA' in RNA B). When using the 64-dimensional 3-mer vector to encode the two RNA sequences, only two dimensions are different.

## Subsequence embedding

In order to tackle the limitation, we proposed an effective subsequence embedding method to keep the sequence order information of lncRNAs. The main idea is that we split a lncRNA sequence into some consecutive subsequences with no overlap and then extract the patterns of each subsequence; last, we

**Table 2.** Frequently used notations and their descriptions in this paper

| Notation | Description |
|---|---|
| $k$ | the length of $k$-mer |
| $n$ | the number of subsequences |
| $S_i$ | the $i$th subsequence of the raw lncRNA |
| $L_{si}$ | the length of the $i$th subsequence |
| $D$ | the dimension of the pre-trained vector |

combine these patterns to obtain a complete representation of the lncRNA sequence. In this way, we can keep the sequence order information. The idea is motivated by spatial pyramid pooling-net [27]. He *et al.* proposed spatial pyramid pooling-net to obtain the features from arbitrary sub-images to generate fixed-length representations for the entire image. We transferred and modified their idea to encode lncRNA sequences.

First, we give the frequently used notations and their descriptions in Table 2. We split a lncRNA sequence into $n$ consecutive subsequences, and thus, we denote a lncRNA sequence as another representation form

$$\text{lncRNA} = S_1, S_2, S_3, \ldots, S_{n-1}, S_n, \tag{2}$$

where $n$ is the number of subsequences and $S_i$ is the $i$th subsequence. We denote $L_{si}$ as the length of $S_i$. After dividing a lncRNA sequence into $n$ subsequences, the next step is encoding these subsequences. In our study, we used a word embedding technique to encode subsequences. Word embedding techniques have shown promise in many natural language processing (NLP) applications including text classification, sentiment analysis and part-of-speech tagging. The core idea is as follows: we used word2vec embedding like NLP training word vectors, all RNAs in our dataset formed the corpus, the $4^k$ types of $k$-mer formed the vocabulary and each RNA is the sentence in the corpus. We treated each $k$-mer in $k$-mer splitting sequence as a 'word' in the sentence, and pre-trained language model with lncRNA sequences in our dataset to obtain the distribution representation of $k$-mer by using word2vec technique, last used the distribution representation of $k$-mer features to represent subsequences. We take 2-mer as an example to show the process. $k$ is set to 2, the stride window is set to 1 and a lncRNA sequence can be split into a $k$-mer sequence. There are 16 types of 2-mer {'AA', 'AC', 'AG', 'AU', 'CA', 'CC', 'CG', 'CU', 'GA', 'GC', 'GG', 'GU', 'UA', 'UC', 'UG', 'UU'}, and the 16 types of 2-mer formed the vocabulary. RNA A is 'ACACACGCGC' and 2-mer splitting of RNA A is {AC, CA, AC, CA, AC, CG, GC, CG, GC}; 2-mer splitting of RNA A is treated as a sentence, and each 2-mer in the splitting ('AC', 'CA', ..., 'GC') is treated as 'word'. Then, we trained a $k$-mer language model *KM* through word2vec technique, and then the vector of each $k$-mer $V_{k\text{-mer}}$ is obtained by the $k$-mer model *KM*.

$$V_{k-\text{mer}} = KM\,(k - \text{mer}). \tag{3}$$

Finally, RNA A can be represented as $\{V_{AC}, V_{CA}, V_{AC}, V_{CA}, V_{AC}, V_{CG}, V_{GC}, V_{CG}, V_{GC}\}$. Word2vec is a popular word embedding technique [28], and its variant algorithms are widely used in network learning field [29–31]. It aims at learning a dense vector automatically for each word in a corpus. The word2vec technique has two models: skip-gram and continuous bag of words model. The skip-gram model uses the central word to predict context words. In the training process, we maximized the co-occurrence likelihood function of the central word and corresponding context words. In our study, we used gensim library to learn $k$-mer features of lncRNA sequences [32]. The parameter $k$ is chosen from $\{1, 2, 3, 4, 5, 6\}$ to find the best splitting way of lncRNAs.

The steps of subsequence embedding (see the subsequence embedding part in Figure 2) are described as follows:

(i) We first built the $k$-mer corpus, which consists of all $k$-mer sequences built by splitting lncRNA sequences.
(ii) We used gensim library to learn representation vectors of $k$-mer of all lncRNA sequences.
(iii) For a given lncRNA, we split it into $n$ subsequences, where the length of each subsequence is $L_{si}$.
(iv) According to the $k$-mer splitting of lncRNA, we found the pre-trained vector of each $k$-mer and then combined these vectors into a matrix as the representation of a subsequence.

Last, we converted each lncRNA subsequence into a matrix whose dimension is $D \times (L_{si} - k + 1)$, which is the actual input for our deep learning model.

## Network architecture

So far, we have obtained the representation of each subsequence. The question then arises: how can we predict the subcellular localization by using the representation of subsequences. We have $n$ subsequences, and the representation of each subsequence is a matrix whose dimension is $D \times (L_{si} - k + 1)$. If we put them together directly, the dimension is $n \times D \times (L_{si} - k + 1)$, which has two disadvantages. First, the length of different subsequence $L_{si}$ in different lncRNA sequences is not the same. If we put them together directly, we must pad them to the same length. It means we have to fill a lot of zeros at the end of the raw sequence, which brings many meaninglessness in representation vectors. Second, the dimension is too large after putting them together directly, which causes a lot of computational waste. To tackle the two limitations, we use an average pooling layer to extract the patterns in each channel of the subsequence. By using the average pooling layer, the dimension of each subsequence is reduced from $D \times (L_{si} - k + 1)$ to $D$. It can be seen that $D$ is the dimension of the pre-trained vector of $k$-mer and has no relationship with the length of lncRNA subsequence $L_{si}$. By using this method, we do not need to pad with zeros and reduce the dimension.

After obtaining the representation of each subsequence by using an average pooling layer, we combined them together to obtain the complete representation of the whole lncRNA sequence. Then the next step is predicting the subcellular localization. TextCNN is a kind of powerful deep learning network structure that is used for text classification. Traditional CNNs are two-dimensional CNNs that are used to process two-dimensional image data. Actually, a text can be treated as a one-dimensional image, so that we can use one-dimensional CNN to extract the features of the text. TextCNN uses a one-dimensional convolutional layer and a max-pooling layer to extract the features of sequence [33]. Inspired by its success in bioinformatics [34], we used textCNN to extract the features of the complete representation. Specifically, we have $n$ subsequences, and the representation of each subsequence is $D$. We

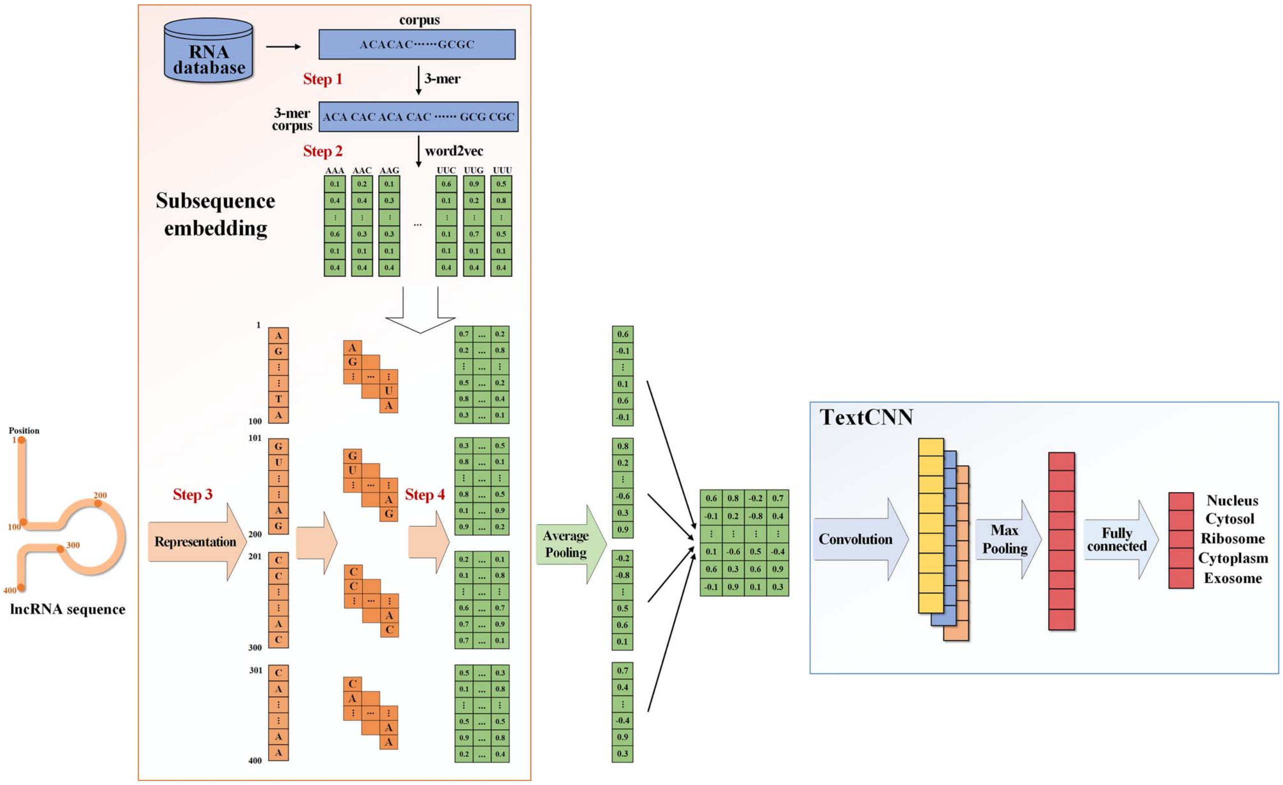

**Figure 2.** Illustration of the deep NN structure. This figure is only an example. The network structure consists of three parts: subsequence embedding, an average pooling layer and a textCNN. The input is a lncRNA sequence with a length of 400. The lncRNA sequence is split into four subsequences. The sequence embedding part has four steps. After subsequence embedding, we use an average pooling layer to extract the patterns of each subsequence. Then, we combine these patterns together to obtain a matrix as the representation of the whole lncRNA sequence. Last, a textCNN is employed to learn high-level features and perform the prediction task.

combined them together to form a matrix whose dimension is $n \times D$ to represent the whole sequence. The representation of the lncRNA sequence can be treated as a one-dimensional image, the width is $n$, the height is 1 and the channel is $D$. To extract high-level features, textCNN uses three convolutional kernels (sizes = 1, 3, 5) to capture the correlation of adjacent nucleotides. Then, textCNN performs a max-pooling layer on all channels to obtain the most remarkable features and reduce the dimension of the output vector. Last, the output vectors of the max-pooling layer are concatenated together as the input of a fully connected layer with a softmax function to perform the final prediction. Figure 2 gives a schematic view of the whole network structure.

## Evaluation metrics

Similar to previous studies [22–24], we used accuracy (ACC), Macro F-measure and area under the receiver operator characteristic curve (AUC) as evaluation metrics to evaluate DeepLncLoc and other methods in the study.

$$\text{Accuracy} = \frac{\text{Num (Pred} = \text{Label)}}{\text{Num (samples)}} \tag{4}$$

$$\text{precision}^{(i)} = \frac{\text{TP}^{(i)}}{\text{TP}^{(i)} + \text{FP}^{(i)}} \tag{5}$$

$$\text{recall}^{(i)} = \frac{\text{TP}^{(i)}}{\text{TP}^{(i)} + \text{FN}^{(i)}} \tag{6}$$

$$\text{Macro F} - \text{measure} = \frac{1}{m}\sum_{i=1}^{m} \frac{2 * \text{precision}^{(i)} * \text{recall}^{(i)}}{\text{precision}^{(i)} + \text{recall}^{(i)}}, \tag{7}$$

where $\text{TP}^{(i)}$, $\text{FP}^{(i)}$ and $\text{FN}^{(i)}$ represent the number of true positives, false positives and false negatives of the class $i$, respectively.

## Implementation details

DeepLncLoc is implemented with PyTorch [35]. The loss function used in DeepLncLoc is the focal loss of non-$\alpha$-balanced form [36]. It is used for object detection to address this class imbalance problem. It is defined as follows:

$$\text{Focal Loss} = -\frac{1}{m}\sum y\left(1 - y_{\text{pred}}\right)^{\gamma} \log\left(y_{\text{pred}}\right), \tag{8}$$

where $m$ is the number of training samples, $y$ is the true label, $y_{\text{pred}}$ is the predicted label and $\gamma$ is the focusing parameter (we set $\gamma$ to 2, according to Lin's paper [36]).

Skip-gram model [28] is used to pre-train the vectors of $k$-mer for embedding. In textCNN, three convolutional kernels (sizes = 1, 3, 5, filter number = 128) are used to extract the high-level features of adjacent nucleotides. The fully connected layer in the classification part has 384 neurons. To avoid overfitting, dropout rates of 0.3 and 0.5 are applied in the embedding layer and the fully connected layer, respectively. Finally, we trained DeepLncLoc using the Adaptive Momentum optimizer; the initial learning rate is set to 0.001.

## Results and Discussion

### Hyper-parameter optimization for DeepLncLoc

We used 5-fold cross-validation (5-fold CV) to tune the hyper-parameters of DeepLncLoc based on the value of Macro F-measure. In our model, many hyper-parameters affect the computational results, such as the parameter $k$, the number of subsequences, the dimension of the pre-trained vector of $k$-mer, initial learning rate and kernel sizes. In the study, we cared about most is the effect of subsequence embedding on computational results. Thus, we considered the parameter $k$, the number of subsequences $n$ and the dimension of the pre-trained vector of $k$-mer $D$ as the major tuning hyper-parameters. A grid search strategy was applied to find the best combination of the three hyper-parameters. The parameter $k$ was chosen from {1, 2, 3, 4, 5, 6}, the number of subsequences $n$ was chosen from {16, 32, 64, 128, 256} and the dimension of pre-trained vector $D$ was chosen from {64, 128}. We tuned these hyper-parameters to find the final model parameters (see Supplementary Table S1, see Supplementary Data available online at http://bib.oxfordjournals.org/). From Supplementary Table S1, see Supplementary Data available online at http://bib.oxfordjournals.org/, it is very hard to determine the parameters directly. We analyzed and found that the performance is unstable when $k$ and $n$ are too high or too low. In order to ensure the generalization of DeepLncloc, $k$, $n$ and $D$ are set to 3, 64 and 64, respectively. In this setting, the ACC, Macro F-measure and AUC obtained by DeepLncLoc are 0.548, 0.421 and 0.820, respectively.

### Comparison with traditional machine learning classifiers with different $k$-mer features

Considering that traditional machine learning classifiers with $k$-mer features are widely used in the prediction of lncRNA subcellular localization, we compared DeepLncLoc with four traditional machine learning models including SVM, RF, LR and simple NN. We implemented all machine learning models in scikit-learn (v 0.21.1) library in Python. For SVM, we used rbf kernel. For LR and RF, we used the default parameters in scikit-learn. For NN, the input, hidden and output layers use $4^k$, 64 and 5 neurons, respectively. The parameter $k$ in these machine learning models was chosen from {3, 4, 5, 6}. We did not consider the lower and higher $k$ because much lower or higher $k$ will increase the risk of underfitting or overfitting. For example, the dimension of 2-mer features is $4^2$, i.e. 16, which hardly encodes the diversity of all sequences in the database. In this case, the model has a high risk of underfitting. The dimension of 7-mer features is $4^7$, i.e. 16 384, which is far beyond the number of all samples. In this case, the model has a high risk of overfitting. The results are shown in Table 3.

From Table 3, first noted that the performance of each machine learning model with different $k$-mer features is different. We can see that the best performance of SVM and RF is achieved when $k = 5$ and 4, respectively. For LR and NN, the highest ACC, Macro F-measure and AUC are achieved when $k = 3$, 6 and 3, respectively. Second, all evaluation metrics obtained by DeepLncLoc are higher than other machine learning classifiers. The ACC and Macro F-measure of DeepLncLoc are significantly higher than the other machine learning methods. The AUC of DeepLncLoc is slightly higher than the other machine learning methods. Figure 3 plots the ROC curves of DeepLncLoc and other machine learning methods with the highest AUC. As we can see, DeepLncLoc outperforms traditional machine learning models for four subcellular localizations (cytoplasm, nucleus, exosome, cytosol). However, in ribsome, DeepLncLoc is higher than SVM, RF and NN but slightly worse than LR, for which DeepLncLoc achieves an AUC of 0.657, slightly lower than that of LR (AUC = 0.675). In summary, these results indicate that DeepLncLoc outperforms traditional machine learning models with $k$-mer features for most subcellular localizations.

### Comparison with different lncRNA representation methods

In this study, we are focused on the lncRNA representation for subcellular localization prediction of lncRNAs. To prove the effectiveness of subsequence embedding, we compared the performance of our proposed method and several popular lncRNA representation methods. Specifically, we changed the representation method and kept the textCNN structure, and developed some variant networks.

(i) One-hot + textCNN, it embeds A, C, G, U using one-hot encoding, followed by textCNN structure to predict subcellular localization.
(ii) Word2vec (1-mer) + textCNN, it embeds A, C, G, U using word2vec, which are fed into textCNN structure to output subcellular localization.
(iii) Word2vec (3-mer) + textCNN, it embeds 3-mer features using word2vec, followed by textCNN structure to predict subcellular localization.

In the three variant networks, considering that most of sequences are shorter than 6000, the length of all lncRNA sequences is normalized to 6000. Sequences longer than 6000 are truncated and those shorter than 6000 are padded with zeros. We used 5-fold CV and reported the classification performance in Table 4. As we can see, with the help of subsequence embedding, our method obtains the highest ACC, Macro F-measure and AUC, which shows that our method consistently surpasses the other representation methods. Taking AUC as an example, on average, our method shows the AUC 4.7, 6.5, 3.9% higher than one-hot + textCNN, word2vec (1-mer) + textCNN and word2vec (3-mer) + textCNN, respectively. Besides, our method outperforms the four variant networks on ACC and Macro F-measure. This observation confirms the effectiveness of our method.

### Comparison with existing predictors

The 5-fold CV was applied in our previous experiments. To further evaluate the performance of DeepLncLoc in predicting the subcellular localization of lncRNAs, we compared DeepLncLoc with existing predictors by using a stand-alone test set.

We selected current predictors follow these criteria: (i) availability of web server or stand-alone version; (ii) input that only needs lncRNA sequences and (iii) outputs that include predictive scores for subcellular localization. Consequently, lncLocator and iLoc-lncRNA satisfy these criteria. LncLocator can predict 5 subcellular localizations of lncRNAs, including nucleus, cytoplasm, cytosol, ribosome and exosome. iLoc-lncRNA can predict 4 subcellular localizations of lncRNAs, including nucleus, cytoplasm, ribosome and exosome. We used the web servers of lncLocator (available at http://www.csbio.sjtu.edu.cn/bioinf/lncLocator/) and iLoc-lncRNA (available at http://lin-group.cn/server/iLoc-LncRNA/download.php) for comparison.

We compared DeepLncLoc with the two predictors (lncLocator and iLoc-lncRNA) by using an independent test set. The test set was created from another lncRNA subcellular localization database lncSLdb and recent literature, since

**Table 3.** Performance of DeepLncLoc and different machine learning models with different *k*-mer features

| | Model | ACC | Macro F-measure | AUC |
|---|---|---|---|---|
| *k* = 3 | SVM | 0.481 ± 0.021 | 0.224 ± 0.027 | 0.794 ± 0.008 |
| | RF | 0.480 ± 0.042 | 0.305 ± 0.037 | 0.777 ± 0.011 |
| | LR | 0.497 ± 0.025 | 0.267 ± 0.048 | 0.813 ± 0.004 |
| | NN | 0.527 ± 0.032 | 0.324 ± 0.033 | 0.808 ± 0.009 |
| *k* = 4 | SVM | 0.486 ± 0.010 | 0.223 ± 0.011 | 0.808 ± 0.007 |
| | RF | 0.508 ± 0.024 | 0.327 ± 0.037 | 0.788 ± 0.009 |
| | LR | 0.469 ± 0.029 | 0.289 ± 0.043 | 0.775 ± 0.017 |
| | NN | 0.481 ± 0.023 | 0.325 ± 0.048 | 0.769 ± 0.015 |
| *k* = 5 | SVM | 0.499 ± 0.013 | 0.271 ± 0.031 | 0.811 ± 0.006 |
| | RF | 0.497 ± 0.019 | 0.282 ± 0.011 | 0.786 ± 0.014 |
| | LR | 0.446 ± 0.023 | 0.290 ± 0.035 | 0.728 ± 0.020 |
| | NN | 0.461 ± 0.048 | 0.321 ± 0.064 | 0.736 ± 0.018 |
| *k* = 6 | SVM | 0.496 ± 0.040 | 0.245 ± 0.013 | 0.809 ± 0.015 |
| | RF | 0.483 ± 0.044 | 0.280 ± 0.043 | 0.772 ± 0.020 |
| | LR | 0.479 ± 0.033 | 0.335 ± 0.046 | 0.767 ± 0.012 |
| | NN | 0.506 ± 0.039 | 0.345 ± 0.053 | 0.759 ± 0.021 |
| DeepLncLoc | | **0.548** ± **0.038** | **0.421** ± **0.033** | **0.820** ± **0.017** |

*Note*: The best performance values are highlighted in bold.

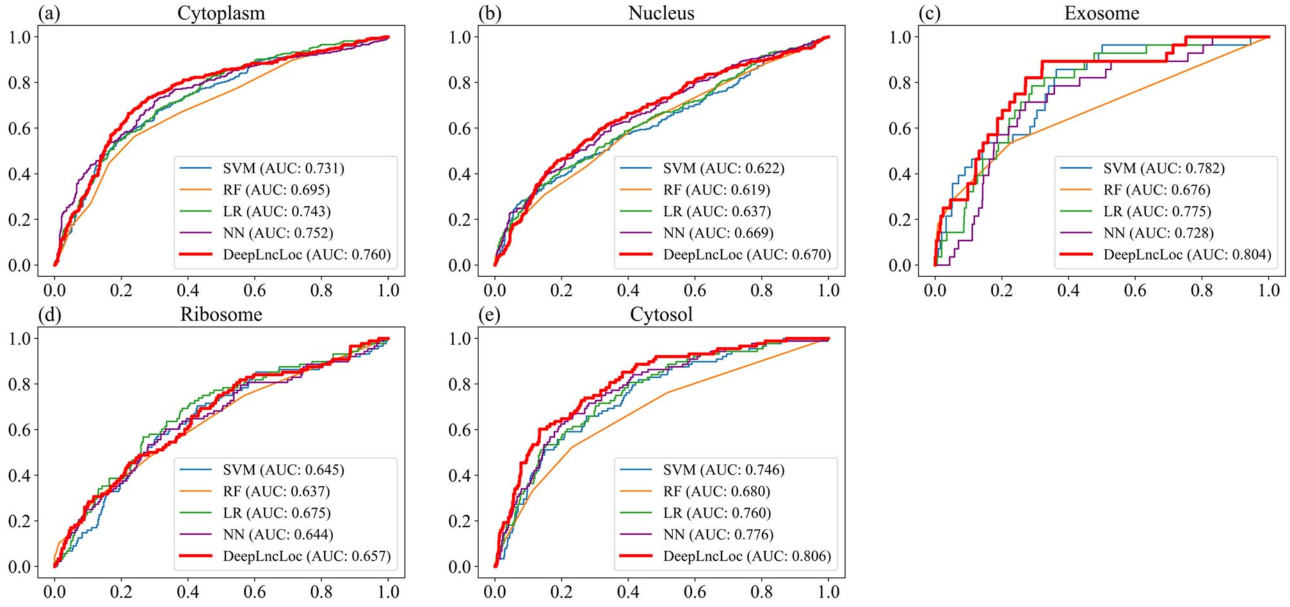

**Figure 3.** The ROC curves of DeepLncLoc, SVM (*k* = 5), RF (*k* = 4), LR (*k* = 3) and NN (*k* = 3) for each class. (**A**) Cytoplasm, (**B**) Nucleus, (**C**) Exosome, (**D**) Ribosome, (**E**) Cytosol.

**Table 4.** Performance of subsequence embedding and different lncRNA representation methods

| Representation method | ACC | Macro F-measure | AUC |
|---|---|---|---|
| One-hot | 0.481 ± 0.029 | 0.254 ± 0.048 | 0.783 ± 0.021 |
| Word2vec (1-mer) | 0.483 ± 0.045 | 0.314 ± 0.028 | 0.770 ± 0.011 |
| Word2vec (3-mer) | 0.504 ± 0.029 | 0.367 ± 0.033 | 0.789 ± 0.014 |
| Subsequence embedding | **0.548** ± **0.038** | **0.421** ± **0.033** | **0.820** ± **0.017** |

*Note*: The best performance values are highlighted in bold.

lncSLdb database only collects five subcellular localizations: nucleus, chromosome, cytoplasm, nucleoplasm and ribosome, and does not have records in the subcellular localization of cytosol and exosome. Thus, we randomly selected some samples from three subcellular localizations (nucleus, cytoplasm and ribosome) in lncSLdb database. To obtain other samples from the subcellular localization of cytosol and exosome, we searched some recent literature in the PubMed database using the following keywords: lncRNA and each subcellular localization, and then obtained lncRNA sequences from NCBI database. we used the cd-hit tool to remove the redundant sequences with a cutoff of 90%. Last, the test set contains 20 samples from cytoplasm, 20 samples from nucleus, 10 samples from ribosome, 10 samples from cytosol and 7 samples from exosome

**Table 5.** Comparison of the prediction performance of DeepLncLoc with lncLocator and iLoc-lncRNA on the test set

| Predictor | Macro Precision | Macro Recall | Macro F-measure | ACC |
|---|---|---|---|---|
| lncLocator | 0.282 | 0.310 | 0.283 | 0.373 |
| iLoc-lncRNA | 0.488 | 0.445 | 0.458 | 0.507 |
| DeepLncLoc (5 classes) | 0.702 | 0.524 | 0.563 | 0.537 |
| DeepLncLoc (4 classes) | 0.675 | 0.543 | 0.560 | 0.537 |

**Table 6.** Precision, recall and F-measure of DeepLncLoc and lncLocator for each class on the test set

| Predictor | lncLocator | | | DeepLncLoc | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | $F_1$ | Precision | Recall | $F_1$ |
| Cytoplasm | 0.484 | 0.750 | 0.588 | 0.778 | 0.350 | 0.483 |
| Nucleus | 0.308 | 0.200 | 0.242 | 0.400 | 0.800 | 0.533 |
| Ribosome | 0.333 | 0.200 | 0.250 | 0.500 | 0.400 | 0.444 |
| Cytosol | 0.286 | 0.400 | 0.333 | 0.833 | 0.500 | 0.625 |
| Exosome | 0.000 | 0.000 | 0.000 | 1.000 | 0.571 | 0.727 |

*Note*: $F_1$ represents F-measure.

**Table 7.** Precision, recall and F-measure of DeepLncLoc and iLoc-lncRNA for each class on the test set

| Predictor | iLoc-lncRNA | | | DeepLncLoc | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | $F_1$ | Precision | Recall | $F_1$ |
| Cytoplasm | 0.553 | 0.700 | 0.618 | 0.800 | 0.400 | 0.533 |
| Nucleus | 0.467 | 0.350 | 0.400 | 0.400 | 0.800 | 0.533 |
| Ribosome | 0.333 | 0.300 | 0.316 | 0.500 | 0.400 | 0.444 |
| Exosome | 0.600 | 0.429 | 0.500 | 1.000 | 0.571 | 0.727 |

*Note*: $F_1$ represents F-measure.

(see Supplementary Table S2, see Supplementary Data available online at http://bib.oxfordjournals.org/).

The confusion matrices of DeepLncLoc and lncLocator are shown in Supplementary Figure S2, see Supplementary Data available online at http://bib.oxfordjournals.org/. Since iLoc-lncRNA treats cytoplasm and cytosol as one category, it only predicts four classes (nucleus, cytoplasm, ribosome and exosome). To make the comparison fair, we treated cytoplasm and cytosol as one category when we compared DeepLncLoc with iLoc-lncRNA. The confusion matrices of DeepLncLoc and iLoc-lncRNA are shown in Supplementary Figure S3, see Supplementary Data available online at http://bib.oxfordjournals.org/. In Supplementary Figures S2 and S3, see Supplementary Data available online at http://bib.oxfordjournals.org/, each row represents the true class, whereas each column represents the predicted class. The diagonal elements represent the number of samples that are predicted correctly. Out of the 67 lncRNAs, our method predicted correct subcellular localization for 36 of them, which is far more accurate than lncLocator (25) and slightly higher than iLoc-lncRNA (34). The results of DeepLncLoc, lncLocator and iLoc-lncRNA are reported in Table 5. Clearly, the accuracy of DeepLncLoc is higher than lncLocator and iLoc-lncRNA. The Macro Precision, Macro Recall and Macro F-measure of DeepLncLoc (5 classes) are 0.702, 0.524 and 0.563, respectively, which are significantly higher than those of lncLocator (0.282, 0.310 and 0.283). Similar results are observed when we compared DeepLncLoc (4 classes) with iLoc-lncRNA. All results suggested that the DeepLncloc may serve as a useful tool to predict the subcellular localization of lncRNAs. We gave the detailed prediction results of DeepLncLoc, lncLocator and iLoc-lncRNA on the test set (see Supplementary Table S3, see Supplementary Data available online at http://bib.oxfordjournals.org/). Precision, recall, F-measure of DeepLncLoc, lncLocator and iLoc-lncRNA for each class on the test set are reported in Tables 6 and 7. We observed that the F-measures of DeepLncLoc for nucleus, ribosome, cytosol and exosome are higher than those of lncLocator, and the F-measure of DeepLncLoc for cytoplasm is lower than that of lncLocator. This phenomenon has been observed when we compared DeepLncLoc with iLoc-lncRNA. In addition, we also noted that none of samples in exosome have been correctly recognized by lncLocator, which leads to very bad prediction results for exosome. A possible explanation is that there are too many samples of cytoplasm in the training set of lncLocator and iLoc-lncRNA. The machine learning model will naturally give more preference to cytoplasm, resulting in a bad performance for the other classes. Thus, lncLocator and iLoc-lncRNA tend to classify other subcellular localizations to cytoplasm.

## The effects of different species

In addition, we investigated whether the type of species has an impact on classification results. The dataset covers six different species and the species distribution of lncRNAs is shown in Supplementary Table S4, see Supplementary Data available online at http://bib.oxfordjournals.org/. Four species only have one or two lncRNAs; thus, we only used two species (*Homo sapiens* and *Mus musculus*) for analysis. *Homo sapiens* group contains 461 lncRNAs and *M. musculus* group contains 391 samples. Supplementary Figure S4, see Supplementary Data available online at http://bib.oxfordjournals.org/, plots the performance of DeepLncLoc on the two species. As shown in this figure, the ACC and AUC of *H. sapiens* group are 0.547 and 0.823, respectively, which is slightly higher than those of *M. musculus* group (0.503 and 0.774).

### DeepLncLoc web server

A web server that implements DeepLncLoc is freely available at http://bioinformatics.csu.edu.cn/DeepLncLoc/. DeepLncLoc requires a lncRNA sequence with more than 200 and less than 100 000 nucleotides as input. Then, user click on the submit button to see the predicted results. The results have one table and one sentence and will be shown on the screen of the computer. The table has five columns and each column represents the name of subcellular localization and corresponding probability. Last, the final predicted subcellular localization is marked red to show. Usually, DeepLncLoc takes less than 5 s to predict the subcellular localization of a lncRNA sequence.

## Conclusion

In this study, we proposed DeepLncLoc, an open-source deep learning model, for predicting subcellular localization of lncRNAs. Unlike many previous computational methods, which use *k*-mer features to encode raw lncRNA sequences, DeepLncLoc proposes a novel subsequence embedding method to encode lncRNA sequences. Compared with previous studies, DeepLncLoc has two novel design ideas: (i) it can keep the sequence order information of lncRNA sequences by using subsequence embedding; (ii) using textCNN can automatically capture high-level features from the combination of the patterns of all subsequences. Our extensive results showed that DeepLncLoc outperforms all traditional machine learning models with different *k*-mer features and existing state-of-the-art predictors. We believe that DeepLncLoc can serve as a useful tool to predict the subcellular localization of lncRNAs.

While our results are promising, several improvements can still be made. We would like to point out the following limitations of DeepLncLoc:

(i) Because the majority of lncRNAs in RNALocate database only have one subcellular localization, thus, we only chose the lncRNAs that only have one subcellular localization for training and testing in this study. However, in reality, many lncRNAs have multiple subcellular localizations. Therefore, in future work, if we can collect more labeled lncRNAs with multiple subcellular localizations, we can expand the dataset to train a more powerful model.

(ii) We only used lncRNA sequence-based features in our model for training and did not consider other biological information. There are some useful features that could be integrated for better predicting the subcellular localization [37, 38]. For example, Gudenas *et al.* [24] used *k*-mer features, RNA–binding motifs and genomic loci to predict the subcellular localization of lncRNAs. Thus, in the future, we plan to incorporate other biological information to deep NNs.

(iii) To reduce computational cost and runtime, we did not use a very complex deep learning model to extract features and perform the classification task. With the development of deep learning techniques, more and more powerful network architecture will be proposed. Therefore, using more powerful network structure to predict the subcellular localization is a promising future direction [39].

(iv) Classification for the minority class of subcellular localization (e.g. ribosome) is a challenging problem. This could be due to two reasons. First, there are too few samples in the minority class, which causes that our model cannot capture the patterns of the minority class. Second, the class distribution is imbalanced: the classifier tends to bias to the majority class (e.g. nucleus) and hence leads to a loss of predictive performance for the minority class [40].

The variable-length of lncRNA sequences is hard to address in most existing computational methods. Even though our analysis was limited to predicting the subcellular localization of lncRNAs, we obtained promising results. We believe that the subsequence embedding method in DeepLncLoc can be used as a general representation method of RNA and DNA sequences. It is expected to be applied to other related variable-length sequence problems, such as prediction of mRNA subcellular localization [41], prediction of DNA N4-methylcytosine sites [42], RNA shape prediction [43] and transcription factor binding site prediction [44].

### Key Points

- A novel deep learning architecture named DeepLncLoc is developed to predict lncRNA subcellular localization.
- A new subsequence embedding method is proposed to keep the sequence order information.
- TextCNN is used to capture high-level features from the combination of the patterns of all subsequences.
- Extensive experiments demonstrate that DeepLncLoc achieves better performance than the existing methods.
- A user-friendly web server is established.

## Supplementary data

Supplementary data are available online at *Briefings in Bioinformatics* and https://github.com/CSUBioGroup/DeepLncLoc.

## Funding

National Natural Science Foundation of China under Grant (No. 62102457); Hunan Provincial Science and Technology Program (2019CB1007).

## Availability and implementation

We provided a user-friendly web server that is freely available at http://bioinformatics.csu.edu.cn/DeepLncLoc/. All the code and datasets can be downloaded from https://github.com/CSUBioGroup/DeepLncLoc.

## References

1. Zhang Y, Lei X, Fang Z, *et al.* CircRNA-disease associations prediction based on metapath2vec++ and matrix factorization. *Big Data Min Anal* 2020;**3**:280–91.
2. Fang Z, Lei X. Prediction of miRNA-circRNA associations based on k-NN multi-label with random walk restart on a heterogeneous network. *Big Data Min Anal* 2019;**2**:261–72.

3. Consortium EP. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature* 2007;**447**:799.
4. Lu C, Yang M, Luo F, *et al.* Prediction of lncRNA–disease associations based on inductive matrix completion. *Bioinformatics* 2018;**34**:3357–64.
5. Moran VA, Perera RJ, Khalil AM. Emerging functional and mechanistic paradigms of mammalian long non-coding RNAs. *Nucleic Acids Res* 2012;**40**:6391–400.
6. Zeng M, Lu C, Zhang F, *et al.* SDLDA: lncRNA-disease association prediction based on singular value decomposition and deep learning. *Methods* 2020;**179**:73–80.
7. Zeng M, Lu C, Fei Z, *et al.* DMFLDA: a deep learning framework for predicting IncRNA–disease associations. *IEEE/ACM Trans Comput Biol* 2020. 10.1109/TCBB.2020.2983958.
8. Esteller M. Non-coding RNAs in human disease. *Nat Rev Genet* 2011;**12**:861.
9. DiStefano JK. The emerging role of long noncoding RNAs in human disease. *Methods Mol Biol* 2018;**1706**:91–110.
10. Wang KC, Chang HY. Molecular mechanisms of long noncoding RNAs. *Mol Cell* 2011;**43**:904–14.
11. Lu C, Yang M, Li M, *et al.* Predicting human lncRNA-disease associations based on geometric matrix completion. *IEEE J Biomed Health Inform* 2019;**24**:2420–9.
12. Carlevaro-Fita J, Johnson R. Global positioning system: understanding long noncoding RNAs through subcellular localization. *Mol Cell* 2019;**73**:869–83.
13. Shen Y, Ding Y, Tang J, *et al.* Critical evaluation of web-based prediction tools for human protein subcellular localization. *Brief Bioinform* 2020;**21**:1628–40.
14. Shen Y, Tang J, Guo F. Identification of protein subcellular localization via integrating evolutionary and physicochemical information into Chou's general PseAAC. *J Theor Biol* 2019;**462**:230–9.
15. Hacisuleyman E, Goff LA, Trapnell C, *et al.* Topological organization of multichromosomal regions by the long intergenic noncoding RNA firre. *Nat Struct Mol Biol* 2014;**21**:198.
16. Yoon J-H, Abdelmohsen K, Srikantan S, *et al.* LincRNA-p21 suppresses target mRNA translation. *Mol Cell* 2012;**47**:648–55.
17. Hansji H, Leung EY, Baguley BC, *et al.* ZFAS1: a long noncoding RNA associated with ribosomes in breast cancer cells. *Biol Direct* 2016;**11**:62.
18. Voit EO, Martens HA, Omholt SW. 150 years of the mass action law. *PLoS Comput Biol* 2015;**11**:e1004012.
19. Zhang T, Tan P, Wang L, *et al.* RNALocate: a resource for RNA subcellular localizations. *Nucleic Acids Res* 2016;**45**:D135–8.
20. Mas-Ponte D, Carlevaro-Fita J, Palumbo E, *et al.* LncATLAS database for subcellular localization of long noncoding RNAs. *RNA* 2017;**23**:1080–7.
21. Wen X, Gao L, Guo X, *et al.* lncSLdb: a resource for long noncoding RNA subcellular localization. *Database* 2018;**2018**. 10.1093/database/bay085.
22. Cao Z, Pan X, Yang Y, *et al.* The lncLocator: a subcellular localization predictor for long non-coding RNAs based on a stacked ensemble classifier. *Bioinformatics* 2018;**34**:2185–94.
23. Su Z-D, Huang Y, Zhang Z-Y, *et al.* iLoc-lncRNA: predict the subcellular location of lncRNAs by incorporating octamer composition into general PseKNC. *Bioinformatics* 2018;**34**:4196–204.
24. Gudenas BL, Wang L. Prediction of lncRNA subcellular localization with deep learning from sequence features. *Sci Rep* 2018;**8**:16385.
25. Fan Y, Chen M, Zhu QJIA. lncLocPred: predicting LncRNA subcellular localization using multiple sequence feature information. *IEEE Access* 2020;**8**:124702–11.
26. Wang H, Ding Y, Tang J, *et al.* Identify RNA-associated subcellular localizations based on multi-label learning using Chou's 5-steps rule. *BMC Genomics* 2021;**22**:1–14.
27. He K, Zhang X, Ren S, *et al.* Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans Pattern Anal Mach Intell* 2015;**37**:1904–16.
28. Mikolov T, Chen K, Corrado G, *et al.* Efficient estimation of word representations in vector space, arXiv preprint. In: *arXiv:1301.3781*, 2013. 10 September 2013, preprint: not peer reviewed.
29. Meng X, Xiang J, Zheng R, *et al.* DPCMNE: detecting protein complexes from protein-protein interaction networks via multi-level network embedding. *IEEE/ACM Trans Comput Biol Bioinform* 2021. 10.1109/TCBB.2021.3050102.
30. Zhou R, Lu Z, Luo H, *et al.* NEDD: a network embedding based method for predicting drug-disease associations. *Bmc Bioinformatics* 2020;**21**. 10.1186/s12859-020-03682-4.
31. Xiang J, Zhang N-R, Zhang J-S, *et al.* PrGeFNE: predicting disease-related genes by fast network embedding. *Methods* 2021;**192**:3–12.
32. Rehurek R, Sojka P. Software framework for topic modelling with large corpora. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer, 2010. Valletta, Malta: European Language Resources Association (ELRA).
33. Kim Y. Convolutional neural networks for sentence classification, arXiv preprint. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. Doha, Qatar: Association for Computational Linguistics. pp. 1746–51.
34. Zeng M, Zhang F, Wu F-X, *et al.* Protein–protein interaction site prediction through combining local and global features with deep neural networks. *Bioinformatics* 2019;**36**:1114–20.
35. Paszke A, Gross S, Chintala S, *et al. Automatic differentiation in pytorch*, 2017.
36. Lin T-Y, Goyal P, Girshick R, *et al. Focal loss for dense object detection*. In: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–8. Venice, Italy: IEEE.
37. Zhang F, Song H, Zeng M, *et al.* DeepFunc: a deep learning framework for accurate prediction of protein functions from protein sequences and interactions. *Proteomics* 2019;**19**: 1900019.
38. Zeng M, Li M, Fei Z, *et al.* A deep learning framework for identifying essential proteins by integrating multiple types of biological information. *IEEE/ACM Trans Comput Biol Bioinform* 2019;**18**:296–305. 10.1109/TCBB.2019.2897679.
39. Zeng M, Li M, Fei Z, *et al.* Automatic ICD-9 coding via deep transfer learning. *Neurocomputing* 2019;**324**:43–50.
40. He H, Garcia EA. Learning from imbalanced data. *IEEE Trans Knowl Data Eng* 2008;**21**:1263–84.
41. Yan Z, Lécuyer E, Blanchette M. Prediction of mRNA subcellular localization using deep recurrent neural networks. *Bioinformatics* 2019;**35**:i333–42.
42. 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* 2018;**35**:1326–33.
43. Mautner S, Montaseri S, Miladi M, *et al.* ShaKer: RNA SHAPE prediction using graph kernel. *Bioinformatics* 2019;**35**:i354–9.
44. Shen Z, Bao W, D-SJSr H. Recurrent neural network for predicting transcription factor binding sites. *Sci Rep* 2018;**8**:1–10.