OXFORD

Briefings in Bioinformatics, 2022, 1–12
https://doi.org/10.1093/bib/bbac565
Problem Solving Protocol

GraphLncLoc: long non-coding RNA subcellular localization prediction using graph convolutional networks based on sequence to graph transformation

Min Li, Baoying Zhao, Rui Yin[2], Chengqian Lu[3], Fei Guo and Min Zeng

Corresponding author. Min Zeng, Hunan Provincial Key Lab on Bioinformatics, School of Computer Science and Engineering, Central South University, Changsha 410083, China. Tel: +(86) 15874980512; E-mail: zengmin@csu.edu.cn

Abstract

The subcellular localization of long non-coding RNAs (lncRNAs) is crucial for understanding lncRNA functions. Most of existing lncRNA subcellular localization prediction methods use *k*-mer frequency features to encode lncRNA sequences. However, *k*-mer frequency features lose sequence order information and fail to capture sequence patterns and motifs of different lengths. In this paper, we proposed GraphLncLoc, a graph convolutional network-based deep learning model, for predicting lncRNA subcellular localization. Unlike previous studies encoding lncRNA sequences by using *k*-mer frequency features, GraphLncLoc transforms lncRNA sequences into de Bruijn graphs, which transforms the sequence classification problem into a graph classification problem. To extract the high-level features from the de Bruijn graph, GraphLncLoc employs graph convolutional networks to learn latent representations. Then, the high-level feature vectors derived from de Bruijn graph are fed into a fully connected layer to perform the prediction task. Extensive experiments show that GraphLncLoc achieves better performance than traditional machine learning models and existing predictors. In addition, our analyses show that transforming sequences into graphs has more distinguishable features and is more robust than *k*-mer frequency features. The case study shows that GraphLncLoc can uncover important motifs for nucleus subcellular localization. GraphLncLoc web server is available at http://csuligroup.com:8000/GraphLncLoc/.

Keywords: long non-coding RNA, subcellular localization prediction, graph convolutional networks, deep learning, de Bruijn graph

Introduction

Long non-coding RNAs (lncRNAs) are an extremely important class of RNAs, which usually have more than 200 nucleotides. With the rapid development of high-throughput sequencing technology, the cumulative evidence shows that lncRNAs are involved in almost all life cycles of cells [1], including metabolic processes, epigenetic regulation, cell differentiation and apoptosis, chromosomal abnormalities, organ or tissue development [2]. For example, lncRNAs regulate the active state of gene expression by interacting with chromatin-modifying proteins or transcription factors and their specific protein-binding motifs [3]; lncRNA can directly bind to its complementary DNA sequence, forming an RNA–DNA triple structure, which can block the transcription process [4]. In addition, many human diseases are closely associated with mutations or dysregulation of lncRNAs [5], including breast cancer, prostate cancer, hepatocellular carcinoma, colon cancer, bladder cancer, thyroid cancer, lung cancer, ovarian cancer, Alzheimer's disease, diabetes and AIDS. As a result, recent years have witnessed an increasing number of lncRNA function studies in the biological field.

It has been reported that the subcellular localization of lncRNAs is different and the mechanisms of lncRNA subcellular localization are diverse [6]. Understanding the subcellular localizations of lncRNAs can provide valuable insights into their functions [7, 8] . For example, lncRNA PVT1 located in the nucleus leads to elevated MYC levels in cancer by interfering with the phosphorylation of the MYC Thr58 site in the nucleus, thereby increasing MYC stability [9]; lncRNA linc-MD1 located in the cytoplasm can repress miR-133 and thus affect transcription factor activation of muscle-specific gene expression [10]; lncRNAs located in exosomes are thought to mediate cell-to-cell communication via RNA carriers [11]. Therefore, the identification of lncRNA subcellular localization is essential to understand the biological functions of lncRNAs [12].

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 the vice dean and a professor at the School of Computer Science and Engineering, Central South University. Her main research interests include bioinformatics and systems biology.
Baoying Zhao received the Bachelor degree of computer science from Guizhou University, China. She is currently a master student in the School of Computer Science and Engineering, Central South University. Her current research interests include bioinformatics and deep learning.
Rui Yin is a research fellow at the Department of Biomedical Informatics, Harvard Medical School. He received PhD degree from Nanyang Technological University. His research interests focus on data mining and machine learning to make sense of big heterogeneous data for real-world application in biomedical fields.
Chengqian Lu received his PhD degree in computer science from Central South University in 2019. His current research interests include bioinformatics and deep learning.
Fei Guo is a professor in Central South University. Her research interests include Bioinformatics and Computational Biology.
Min Zeng 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.
Received: September 15, 2022. **Revised:** November 4, 2022. **Accepted:** November 20, 2022

The gold standard method for determining RNA subcellular localizations is single-molecule fluorescent in situ hybridization (smFISH) technique. Despite the fact that such image data are perfect for determining lncRNA localization compartment, the technique is expensive, time-consuming and technically challenging. Given these limitations, developing accurate and reliable computational methods to predict lncRNA subcellular localization would be of great value to biologists.

Currently, some computational methods have been proposed to predict lncRNA subcellular localization. To the best of our knowledge, the first predictor is lncLocator [13]. LncLocator feeds raw 4-mer frequency features to stacked autoencoders, and then feeds the high-level abstraction of 4-mer frequency features to two types of classifiers, i.e. random forest (RF) and support vector machine (SVM). Lastly, lncLocator uses a stacked ensemble strategy to combine the results to obtain the final classification probabilities. Su *et al*. [14] proposed iLoc-lncRNA that converts lncRNA sequences into 8-mer frequency features and uses binomial distribution to perform feature selection. Finally, iLoc-lncRNA applies SVM to obtain the output results by using the optimal features. Gudenas and Wang developed a deep learning model called DeepLncRNA [15]. DeepLncRNA uses 2, 3, 4, 5-mer frequency features, RNA-binding motifs and genomic loci, and feeds the combined features to a deep neural network to obtain the final prediction results. Ahmad *et al*. [16] proposed LocateR, using a local depth SVM and selecting 655 optimal *k*-mer frequency features as input features. Fan *et al*. [17] proposed a logistic regression-based machine learning predictor called lncLocPred. LncLocPred uses sequence features including *k*-mer frequency features, PseDNC and Triplet features, and then selects representative features from the combined features. Feng *et al*. [18] proposed lncLocation, which integrates multi-source features including *k*-mer frequency features, physicochemical properties and secondary structure features. Then lncLocation applies feature extraction based on self-encoders and hybrid feature selection methods to select representative features. Zeng *et al*. [19] proposed DeepLncLoc, a text convolutional neural network-based deep learning framework that uses subsequence embedding techniques to encode lncRNA sequences. Recently, lncLocator 2.0 [20] and iLoc-lncRNA 2.0 [21] have been developed. LncLocator 2.0 extracts GloVe embedding vectors from sequences and feeds the embedding vectors into convolutional neural networks (CNN), long short-term memory (LSTM), and multi-layer perception (MLP). iLoc-lncRNA 2.0 uses 8-mer frequency features to encode lncRNA sequences and then uses mutual information-based feature selection and incremental feature selection strategy to select the optimal features.

Although several computational methods have been proposed, most of them rely on *k*-mer frequency features to encode lncRNA sequences. In a machine/deep learning model, how to encode raw lncRNA sequences into discriminative features is the most important issue. However, using *k*-mer frequency features to encode lncRNA sequences has some drawbacks. (i) It only reflects the frequency information and ignores sequence order information; (ii) it cannot capture patterns and motifs of different lengths when *k* is fixed. (iii) Moreover, when *k* is small, the encoding method could not obtain sufficient feature information or is unable to capture useful features, which leads to the underfitting of the prediction model. When *k* is large, the dimensionality of the encoding vector increases exponentially and will make the encoding vector sparse, wasting computational resources and causing potential overfitting problems.

To address these issues, we developed GraphLncLoc, a graph convolutional network-based model using only lncRNA sequences, for predicting lncRNA subcellular localization. Different from the previous studies that encode lncRNA sequences by using *k*-mer frequency features, GraphLncLoc transforms lncRNA sequences into de Bruijn graphs, which can provide more comprehensive information. In the de Bruijn graph, the nodes of the graph are 4-mer units and the direction of the edges is determined by the sequential order. Then, GraphLncLoc uses the pre-trained word2vec embedding vector of 4-mer as node features and assigns weights for edges. GraphLncLoc uses graph convolutional networks (GCN) to learn the latent representations and extract the high-level features from the de Bruijn graph. Finally, GraphLncLoc uses a fully connected layer to perform the prediction task. The core idea is inspired by the de Bruijn graph in genome assembly [22]. Figure 1(A) shows the core idea of our study. The advantages of transforming sequences into graphs are summarized as follows. Figure 1(B) shows the advantages of our method.

- It keeps local order information of lncRNA sequences by using a directed graph.
- It can automatically capture patterns and motifs of different lengths in lncRNA sequences by connecting multiple nodes in the graphs to form paths.
- Using aggregation operation can aggregate multiple neighboring nodes to form community and subgraph, and thus can capture global and high-level features of the whole lncRNA sequence.
- It can integrate other types of biological data from different data sources as the node features, and thus can provide a more comprehensive feature encoding for lncRNA sequences.

We conducted extensive experiments to evaluate the performance of GraphLncLoc. Comparison with traditional machine learning classifiers using different *k*-mer frequency features shows the benefits of transforming sequences into graphs. Comparison with different graph construction methods proves the advantages of the proposed graph construction method in GraphLncLoc. Comparison with existing predictors proves the effectiveness of GraphLncLoc in predicting lncRNA subcellular localization. Furthermore, our analysis shows that GraphLncLoc is capable of generating more distinguishable features than *k*-mer frequency features. GraphLncLoc is also more robust than *k*-mer frequency features in lncRNA subcellular localization prediction. The case study shows that GraphLncLoc can find important motifs for nucleus subcellular localization. Finally, we developed a free and user-friendly web server to facilitate the use of GraphLncLoc.

Materials and methods

Datasets

Constructing a high-quality benchmark dataset is the first prerequisite for building a reliable machine/deep learning model. We collected known RNA subcellular localization information from RNALocate v1.0 database [23]. The RNALocate v1.0 database records 42 190 RNA subcellular localization entries with experimental evidence involving nine RNA categories (including lncRNA, csRNA, mRNA, miRNA, piRNA, snRNA, rRNA, snoRNA and tRNA). Among these entries, the RNALocate v1.0 database contains a total of 2383 lncRNA subcellular localization entries. The steps to construct the benchmark dataset are as follows:

1. We retrieved the 2383 lncRNA subcellular localization entries from 42 190 RNA-associated subcellular localization entries.

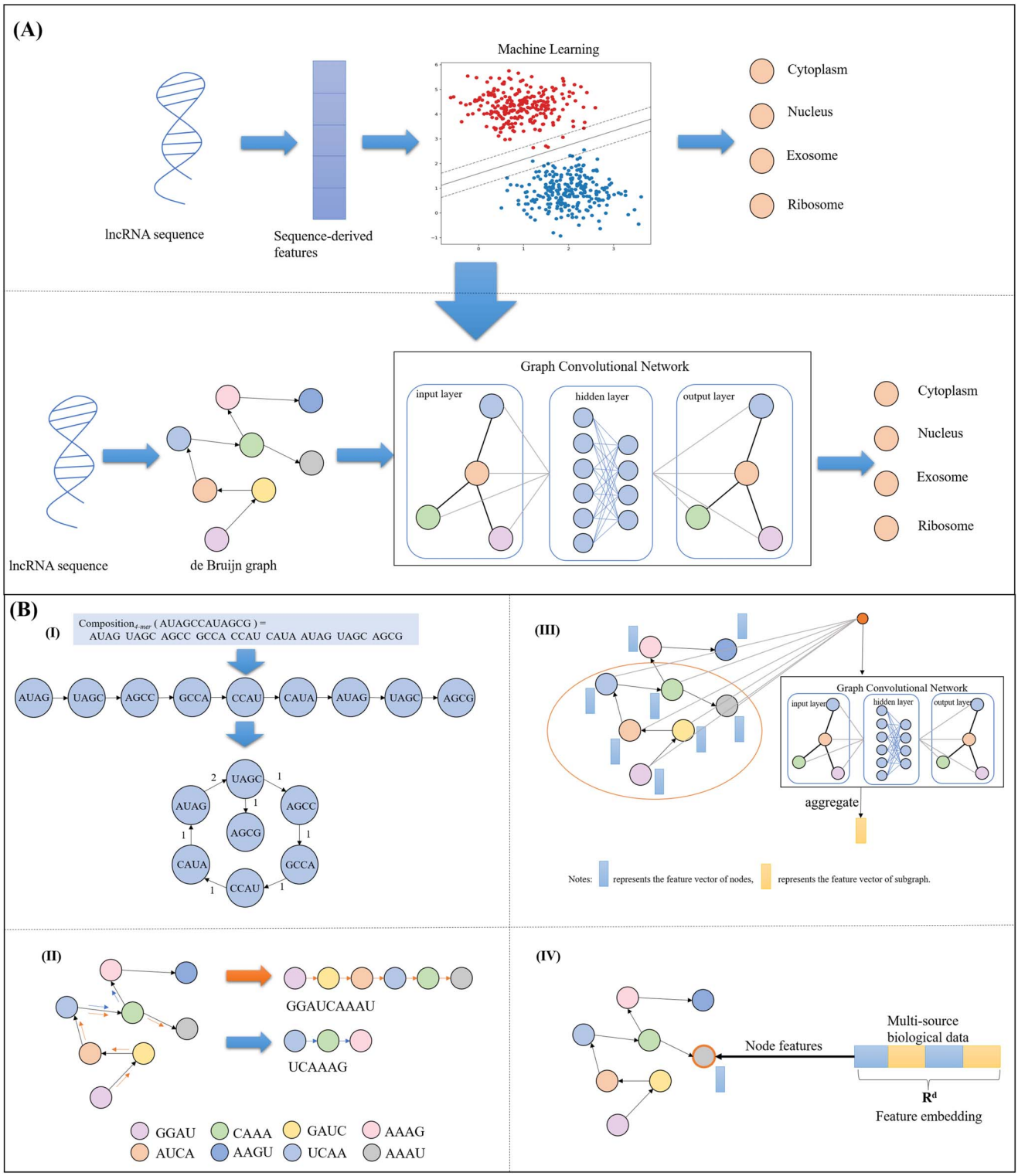

Figure 1. The core idea and advantages of our study. (**A**) The core idea of our study: transforming sequence into graph, which is a new viewpoint of sequence-based lncRNA subcellular localization prediction. (**B**) The advantages of our study. Four advantages of our method: (I) keeping sequence local order information, (II) capturing motifs of different lengths, (III) aggregating multiple nodes to form subgraph to capture high-level features, (IV) integrating other types of biological data easily.

2. Some lncRNAs have multiple subcellular localization entries in the database, we merged these entries with the same gene symbol, and removed some entries without sequence information in NCBI [24] and Ensembl [25].
3. Because most lncRNAs have only one subcellular localization in the database, we selected the lncRNAs that are located in single subcellular localization in the study.

4. To reduce data redundancy, we used the CD-HIT-EST tool [26] to remove redundant sequences at a cut-off value of 80%.
5. The filtered dataset covers seven subcellular localizations. Two of these categories have fewer than ten lncRNAs, so we deleted those lncRNAs which are located in the two subcellular localizations. In addition, considering the ambiguous annotations of the cytoplasm and the cytosol in early literature [27], we only focus on cytoplasm compartment. Thus, we removed those lncRNAs which are located in cytosol.

Finally, we established a benchmark dataset with 769 lncRNAs from four different subcellular localizations including cytoplasm, nucleus, ribosome and exosome (see Supplementary Figure S1). Supplementary Figure S2 shows the subcellular localization distribution of the benchmark dataset.

Overview of GraphLncLoc

The overall framework of GraphLncLoc is illustrated in Figure 2. The input of GraphLncLoc is a lncRNA sequence and the output of GraphLncLoc is the probability of each subcellular localization. The main idea of GraphLncLoc is to transform a lncRNA sequence into a graph and capture high-level features from the graph using GCN. GraphLncLoc consists of four parts: graph construction, node feature extraction, GCN and classification. The graph construction part transforms a lncRNA sequence into a weighted de Bruijn graph. The node feature extraction part is responsible for generating node features for each node in the graph based on the word2vec technique. The GCN part applies GCN to capture high-level features of the graph. On top of the GCN part, there is a fully connected layer with a softmax activation function taking the high-level features as input, which performs four-category subcellular localization prediction.

Graph construction

In the graph construction part, we transformed lncRNA sequences into directed graphs. Specifically, we used de Bruijn graph to encode a lncRNA sequence. Given a lncRNA sequence:

$$\text{lncRNA} = \mathrm{N}_1, \mathrm{N}_2, \mathrm{N}_3, \ldots, \mathrm{N}_{L-1}, \mathrm{N}_L \tag{1}$$

where L denotes the length of the lncRNA, N_j is one of the four nucleotide bases (A, C, G and U) at position j of the lncRNA sequence. Its k-mer composition set (here we use 4-mer as an example) is $\{N_1N_2N_3N_4, N_2N_3N_4N_5, N_3N_4N_5N_6, \ldots, N_{L-3}N_{L-2}N_{L-1}N_L\}$. Then we assigned these k-mers to nodes, followed the order of the k-mer composition set (from left to right), added one k-mer at each time, used these k-mers to reconstruct the lncRNA sequence. After the reconstruction process, we glued identically labeled nodes and formed a de Bruijn graph. Then, we assigned each directed edge a weight. The weight of this edge is the frequency of $(k+1)$-mer, which is formed by the two nodes that make up this edge. To reduce the influence of the absolute differences between edge frequencies, we normalized the edge weights in the graph. Formally, e_{ji} denotes the frequency weight of the edge from node j to node i, $N(i)$ denotes the set of neighbor nodes of node i, we normalized the frequency weight with the formula:

$$w_{\text{norm}} = \frac{e_{ji}}{\sqrt{\sum_{q\in N(j)} e_{jq}}\sqrt{\sum_{q\in N(i)} e_{qi}}} \tag{2}$$

Node features

We employed continuous distributed word representations of k-mer as node features in GraphLncLoc. The k-mer units in lncRNA sequences are similar to words in the article, thus using continuous distributed word representations of k-mer can naturally represent the contextual information of nucleotides in lncRNA. Specifically, we used all lncRNA sequences in our benchmark dataset as the corpus, and applied the word2vec technique to obtain the encoding vector of each 4-mer unit in the lncRNA sequence corpus as the node feature vector of the graph. In this study, we used the word2vec technique with the Skip-gram model to predict surrounding context words given a center word. Following the idea of the Skip-Gram model, word2vec technique aims at maximizing the co-occurrence likelihood between a target 4-mer and its contextual 4-mers. By using continuous distributed word representations of k-mer as node features, GraphLncLoc enriches the semantic information of the constructed de Bruijn graph.

Graph convolutional network

After constructing the de Bruijn graph and obtaining the node features, we trained a GCN to extract high-level features from the de Bruijn graph. GCN can refine graph topology and node features by performing convolutional operations on the graph [28]. The working mechanism of GCN to update network parameters is described in the following.

In GCN, the propagation rule can be formulated by the following equation:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{3}$$

where $\tilde{A} = A + I_N$ is the adjacency matrix of the graph with added self- connections. I_N is the identity matrix, $\tilde{D}$ is the degree matrix of $\tilde{A}$, $W^{(l)}$ denotes the weight of thel^{th} layer, $H^{(l)}$ denotes the matrix of activations of thelth layer, σ denotes the non-linear activation function.

The main idea of GCN layer is to learn a transformation function to generate the new embedding matrix $H_i^{(l+1)}$ of node i by aggregating its own features and its neighbors' features considering the normalized edge weights. By stacking multi-layer GCN, we can implement inter-node message passing and capture the high-level features of the graph. Specifically, GCN aggregates the embedding matrixes of all nodes or edges, and takes the average value as the final graph encoding vector. The aggregation formula for averaging its node features is as follows:

$$h_G = \frac{1}{|V|}\sum_{v\in V} h_v \tag{4}$$

where h_G is the encoding vector of graph G, V is the set of all nodes in graph G and h_v is the embedding vector of node v.

Finally, we obtained the graph representation vector h_G of graph G. The high-level features extracted from the de Bruijn graph using GCN are fed into a fully connected layer to perform the classification task.

Implementation details

GraphLncLoc is implemented based on the Pytorch library, and the GCN layer is implemented using Deep Graph Library. The value k of the k-mer node is set to 4. The dimensionality of the node feature vector is 128, which is extracted by using pre-trained word2vec technique with the genism library. The number of the

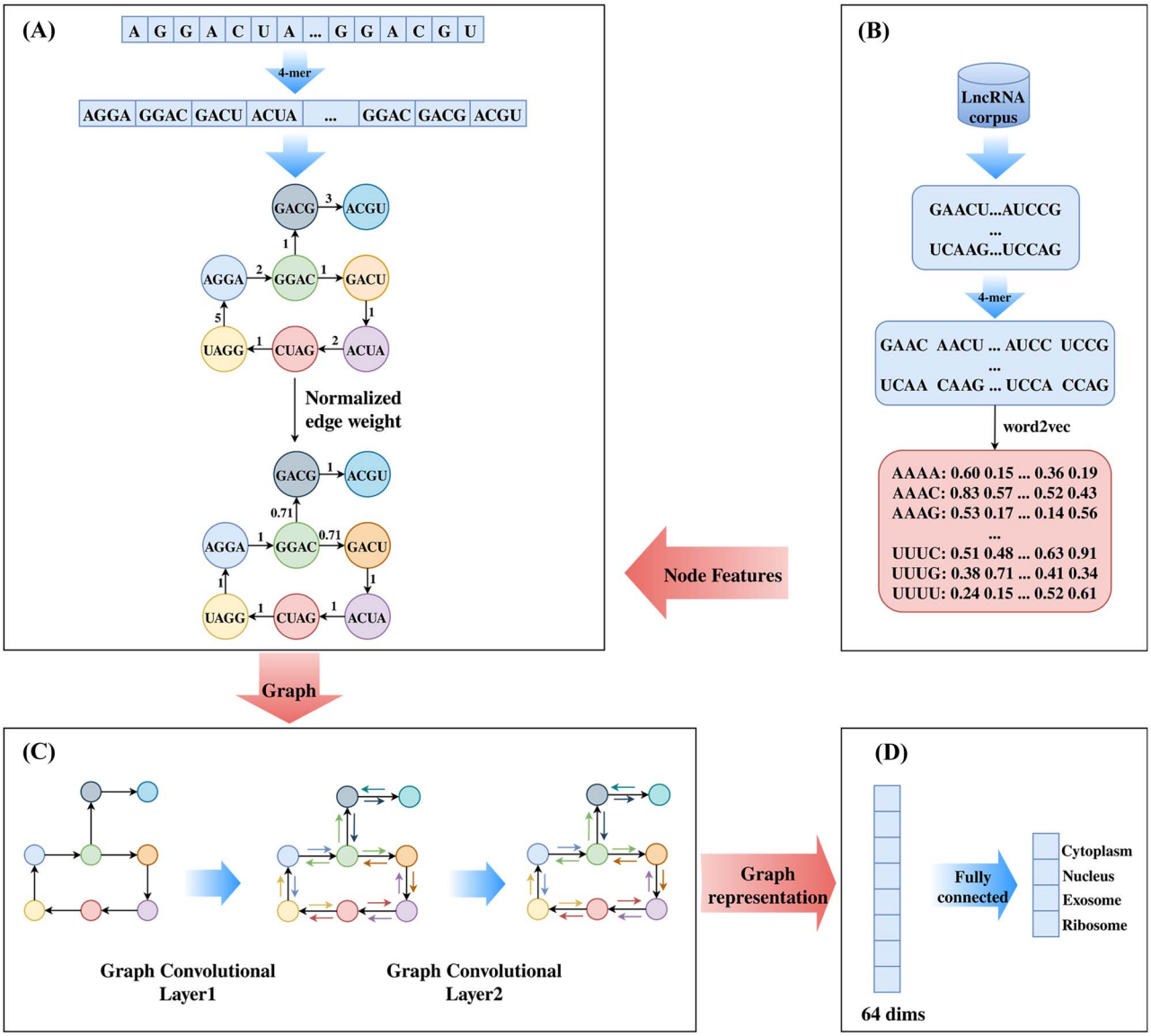

Figure 2. The overall architecture of GraphLncLoc. The network architecture has four parts. (**A**) Graph construction, (**B**) node feature extraction, (**C**) GCN and (**D**) classification. The input is a lncRNA sequence. Through the graph construction part, a lncRNA sequence is transformed into a weighted de Bruijn graph. Then, the node feature extraction part is responsible for generating node features for each node in the graph. The GCN part captures high-level features of the graph by using two layers of GCN. Lastly, a four-category subcellular localization prediction task is performed by a fully connected layer with a softmax activation function.

hidden neurons in GCN is set to 64, the number of input neurons of the final fully connected layer is 64 and the number of output neurons is 4. To avoid overfitting, the dropout rate is set to 0.2 in the fully connected layer.

The loss function in GraphLncLoc is the focal loss of the non-α-balanced form [29]. The focal loss function can tackle the problem of imbalanced data distribution. It is defined as follows:

$$\text{Focal Loss} = -y(1-p)^{\gamma}\log(p) - (1-y)\,p^{\gamma}\log(1-p) \tag{5}$$

where p is the predicted probability value of the sample, y is the true value and the γ parameter is set to 2. In the training process, Adam optimizer is applied to optimize the focal loss function, the learning rate of the optimizer is set to 0.003, the batch size of the sample is 8.

Results

Evaluation metrics

To evaluate the performance of GraphLncLoc, we used Accuracy (ACC), Macro Precision, Macro Recall, Macro F1-score and area under the receiver operator characteristic (ROC) curve (AUC) as evaluation metrics.

$$\text{Precision}_{(i)} = \frac{\text{TP}_{(i)}}{\text{TP}_{(i)} + \text{FP}_{(i)}} \tag{6}$$

$$\text{Macro Precision} = \frac{1}{n}\sum_{i=1}^{n}\text{precision}_{(i)} \tag{7}$$

$$\text{Recall}_{(i)} = \frac{\text{TP}_{(i)}}{\text{TP}_{(i)} + \text{FN}_{(i)}} \tag{8}$$

$$\text{Macro Recall} = \frac{1}{n}\sum_{i=1}^{n}\text{recall}_{(i)} \tag{9}$$

$$\text{Macro } F1\text{-score} = \frac{1}{n}\sum_{i=1}^{n}\frac{2 * \text{precision}_{(i)} * \text{recall}_{(i)}}{\text{precision}_{(i)} + \text{recall}_{(i)}} \quad (10)$$

where $\mathbf{TP}_{(i)}$, $\mathbf{FP}_{(i)}$, $\mathbf{FN}_{(i)}$ represent the number of true positives, false positives and false negatives of class i, $\mathbf{precision}_{(i)}$ and $\mathbf{recall}_{(i)}$ represent the precision and recall of class i, and n is the number of classes.

Hyper-parameter optimization

There are many hyper-parameters that affect the model performance, such as the value k of the k-mer node, the dimensionality of the pre-trained word2vec embedding vector, the number of the hidden neurons in GCN, the batch size, the learning rate and the dropout rate. In the study, what we care about most is the effects of transforming lncRNA sequences into de Bruijn Graphs on computational results. Thus, we considered the value k of the k-mer node, the dimensionality of the pre-trained word2vec embedding vector, and the number of the hidden neurons in GCN as the major tuning hyper-parameters. The value k of the k-mer node was chosen from {2, 3, 4, 5, 6}, the dimensionality of the pre-trained word2vec embedding vector was chosen from {64, 128}, and the number of the hidden neurons in GCN was chosen from {64, 128}. A grid search strategy was applied to find the best combination of the three hyper-parameters. Finally, the best performance is achieved when the three hyper-parameters are set to 4, 128 and 64, respectively. Because the value k makes a non-negligible impact on the performance of GraphLncLoc. Supplementary Figure S3 shows the performance of different k. The other hyper-parameters are unchanged as follows: the dimensionality of the pre-trained word2vec embedding vector is 128, and the number of the hidden neurons in GCN is 64. We can observe that the results of 4-mer outperform other k-mers. When k is set to 4, the ACC and Macro F1-score are the highest, and the AUC competitive among all the results. The possible explanation is that 2-mer and 3-mer are unable to capture the essential features while 5-mer and 6-mer are sparser than 4-mer.

Comparison with traditional machine learning classifiers using different k-mer frequency features

In previous studies, k-mer frequency features are the most commonly used features. To evaluate the performance of the GraphLncLoc and show the advantages of transforming sequences into graphs for lncRNA subcellular localization prediction, we compared GraphLncLoc with traditional machine learning models with different k-mer frequency features. The compared models include SVM, RF, logistic regression (LR) and simple neural network (NN). We implemented these machine learning models using the scikit-learn (v1.0.1) library. For the SVM, RF and LR models, we used the default parameters in the scikit-learn library. For NN, the number of neurons in the input, hidden and output layers are set to 4^k, 64 and 4, respectively. The parameter k for the k-mer frequency features is set from 3, 4, 5 and 6. We used the average results of the 5-fold cross-validation to evaluate the performance. The results are shown in Table 1.

From Table 1, we first focus on the results of machine learning models. In terms of Macro F1-score, SVM, RF, LR and NN achieve the highest Macro F1-score at $k=3$, $k=3$, $k=6$ and $k=5$, respectively. The results indicate that different machine learning classifiers have their preferred k value for achieving the best performance. Second, indicates better performance regarding all evaluation metrics compared with other machine learning classifiers using k-mer frequency features. The best machine learning classifier is RF model with $k=3$, which obtains 0.572 in ACC, 0.391 in Macro F1-score, 0.511 in Macro Precision and 0.380 in Macro Recall. GraphLncLoc outperforms RF model with 3-mer in terms of ACC (0.612), Macro F1-score (0.506), Macro Precision (0.691) and Macro Recall (0.475). In summary, the results demonstrate that GraphLncLoc performs better than these traditional machine learning classifiers using different k-mer frequency features, which shows the advantages of using graph vectors.

Comparison with deep learning baseline models

To demonstrate the effectiveness of GraphLncLoc (word2vec (4-mer) + 5-mer frequency features + GCN + MLP), we compared it with two deep learning baseline models.

1. Baseline 1: word2vec (4-mer) + 5-mer frequency features + CNN + MLP, it converts lncRNA sequences to word embedding learned by word2vec technique, and extracts 5-mer frequency features, followed by a CNN layer and an MLP layer to predict the subcellular localizations.
2. Baseline 2: word2vec (4-mer) + 5-mer frequency features + LSTM + MLP, it converts lncRNA sequences to word embedding learned by word2vec technique, and extracts 5-mer frequency features, followed by a LSTM layer and an MLP layer to predict the subcellular localizations.

Table 2 shows the performance comparison between GraphLncLoc and the two deep learning baseline models. It is worth noting that the input features are same in the three models. From Table 2, we can observe that GraphLncLoc achieves the best performance. More specifically, in terms of ACC, GraphLncLoc improves about 5.52% compared to the CNN-based model, and improves about 8.13% compared to the LSTM-based model. In terms of Macro F1-score, GraphLncLoc improves about 25.87% compared to the CNN-based model and improves about 19.06% compared to the LSTM-based model. These results demonstrate the effectiveness of our model.

Comparison with existing predictors on an independent test set

To further evaluate the performance of GraphLncLoc in predicting lncRNA subcellular localization, we compared GraphLncLoc with existing predictors and evaluated them using an independent test set which is provided by DeepLncLoc. Considering GraphLncLoc predicts four subcellular localization categories including cytoplasm, nucleus, ribosome and exosome, we removed the samples which belong to cytosol from the independent test set. To reduce data redundancy between our constructed benchmark dataset and the independent test set, we merged the two datasets and used the CD-HIT-EST tool [26] with a cut-off value of 40% to remove redundant sequences in the independent test set. Finally, the independent test set contains 20 sequences from cytoplasm, 20 sequences from nucleus, 10 sequences from ribosome and 7 sequences from exosome (see Supplementary Figure S4).

We selected existing predictors following these criteria: (1) there is an available web server or stand-alone version; (2) the input only requires lncRNA sequences; (3) the output contains predicted probabilities for subcellular localization. As a result, lncLocator, iLoc-lncRNA, Locate-R, DeepLncLoc and iLoc-lncRNA 2.0 satisfy these criteria. We did not compare GraphLncLoc with lncLocator 2.0 because lncLocator 2.0 only provides the predicted cytoplasm/nucleus relative concentration index (CNRCI) values instead of probabilities. LncLocator and DeepLncLoc predict five subcellular locations,

Table 1. Performance comparison of GraphLncLoc and different machine learning models using different *k*-mer frequency features

	Model	ACC	Macro precision	Macro recall	Macro F1-score
k = 3	SVM	0.521	0.301	0.306	0.257
	RF	0.572	0.511	0.380	0.391
	LR	0.462	0.303	0.311	0.304
	NN	0.391	0.276	0.278	0.260
k = 4	SVM	0.520	0.300	0.305	0.256
	RF	0.572	0.523	0.373	0.377
	LR	0.450	0.305	0.311	0.306
	NN	0.398	0.309	0.302	0.285
k = 5	SVM	0.518	0.297	0.304	0.254
	RF	0.572	0.535	0.364	0.360
	LR	0.490	0.367	0.354	0.356
	NN	0.468	0.413	0.326	0.318
k = 6	SVM	0.516	0.299	0.303	0.252
	RF	0.564	0.530	0.355	0.346
	LR	0.536	0.401	0.359	0.360
	NN	0.485	0.333	0.307	0.271
GraphLncLoc		**0.612**	**0.691**	**0.475**	**0.506**

Note: The best performance values are highlighted in bold.

Table 2. Performance comparison of GraphLncLoc with the deep learning baseline models

Deep learning baseline models	ACC	Macro precision	Macro recall	Macro F1-score
Baseline 1	0.580	0.510	0.394	0.402
Baseline 2	0.566	0.557	0.425	0.425
GraphLncLoc	**0.612**	**0.691**	**0.475**	**0.506**

Note: The best performance values are highlighted in bold.

Table 3. Performance comparison of GraphLncLoc with existing predictors on the test set

Predictor	ACC	Macro precision	Macro recall	Macro F1-score
lncLocator	0.421	0.374	0.325	0.289
iLoc-lncRNA	0.509	0.524	0.470	0.474
iLoc-lncRNA 2.0	0.404	0.454	0.384	0.385
Locate-R	0.368	0.362	0.321	0.321
DeepLncLoc	0.561	0.673	0.543	0.582
GraphLncLoc	**0.579**	**0.736**	**0.557**	**0.584**

Note: The best performance values are highlighted in bold.

including nucleus, cytoplasm, cytosol, ribosome and exosome. iLoc-lncRNA, Locate-R and iLoc-lncRNA 2.0 predict four subcellular localizations, including nucleus, cytoplasm, ribosome and exosome. To ensure a fair comparison, when comparing with lncLocator and DeepLncLoc, we merged the output probabilities of cytoplasm and cytosol as the output probability of cytoplasm. Supplementary Table S1 shows the detailed prediction results of GraphLncLoc with five existing predictors on the test set, Table 3 shows the performance comparison of GraphLncLoc with five existing predictors, Supplementary Figure S5 shows the confusion matrices of GraphLncLoc with five existing predictors, and Figure 3 shows the ROC curves of GraphLncLoc and five existing predictors.

From the results in Table 3, it can be seen that GraphLncLoc outperforms the other predictors in terms of all evaluation metrics. GraphLncLoc achieves 0.579 in ACC, which is significantly higher than lncLocator (0.421), iLoc-lncRNA (0.509), Locate-R (0.368), DeepLncLoc (0.561) and iLoc-lncRNA 2.0 (0.404), respectively. The other evaluation metrics (Macro Precision, Macro Recall, Macro F1-score) and Figure 3 indicates similar results. These results demonstrate that GraphLncLoc is an effective tool to predict lncRNA subcellular localization.

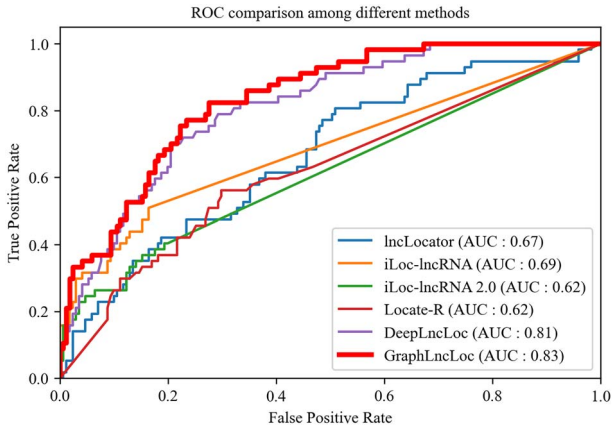

Figure 3. ROC curves of GraphLncLoc and existing predictors on the test set.

Effects of different species

In addition, we investigated the effects of different species on classification results. The benchmark dataset covers six different species, and Supplementary Table S2 shows the species

distribution. From Supplementary Table S2, the *Mus musculus* group contains 391 lncRNAs, the *Homo sapiens* group contains 373 lncRNAs and the other four species groups only have 1 or 2 lncRNAs. Furthermore, we analyzed the lncRNA subcellular localization distribution of two species (*H. sapiens* and *M. musculus*) in Supplementary Figure S6. We found that *M. musculus* only have two types of subcellular localizations (cytoplasm and nucleus) and *H. sapiens* have four types of subcellular localizations. We evaluated the performance of GraphLncLoc on the two species. Supplementary Table S3 shows the precision, recall and F1-score for each subcellular localization of *H. sapiens* and *M. musculus* groups, and Supplementary Figure S7 plots the ACC and AUC of these two species. From Supplementary Table S3, we can observe that the F1-score of *H. sapiens* is lower than that of *M. musculus* for cytoplasm, while the F1-score of *H. sapiens* is higher than that of *M. musculus* for nucleus. As shown in Supplementary Figure S7, the ACC and AUC of the *H. sapiens* group are 0.555 and 0.727, respectively, which is slightly lower than those of the *M. musculus* group (0.670 and 0.863).

Effects of different graph construction methods

In the study, we focus on how to represent lncRNA sequences as graphs for lncRNA subcellular localization prediction. To investigate the effectiveness and necessity of graph construction in GraphLncLoc, we conducted an ablation study by substituting one component from the model and evaluated the performance. We highlighted two aspects: node features and weight normalization. Specifically, we compared our proposed graph representation with other different graph representations.

(1) One-hot node features: it uses one-hot coding method to encode the node feature vector of 4-mer nodes in the graph.
(2) Without weight normalization: it assigns weights to edges using the original frequency without normalization.
(3) Normalized weight (in-degree): it assigns the weights to edges only considering the aggregating messages by each node's in-degrees. Formally, the formula is as follows:

$$w_{\text{norm}} = \frac{e_{ji}}{\sum_{q \in N(i)} e_{qi}} \tag{11}$$

where e_{qi} denotes the frequency weight of the edge from node q to node i, $N(i)$ denotes the set of neighbor nodes of node i.

Table 4 shows the performance of GraphLncLoc and its variant graph construction methods. From Table 4, we can observe that when using one-hot coding node features, ACC, Macro F1-score and AUC decreased by 20.7%, 32.4% and 6.5%, respectively, emphasizing the importance of word2vec technique. Moreover, we can observe that without weight normalization, ACC, Macro F1-score and AUC decreased by 5.7%, 16.2% and 0.6%, respectively, which demonstrates the importance of weight normalization. Moreover, we used different methods to normalize the edge weights. The 'in-degree' type normalization method only considers the relationship of node's in-degree. GraphLncLoc takes into account both the relationship of node's in-degree and out-degree. The results demonstrate that using our normalization method is better. Compared to the 'in-degree' type, ACC and Macro F1-score improved by about 5.2% and 6.3%, and AUC only drops slightly. In summary, the model of using word2vec vectors as node features and considering both the relationship between node's in-degree and out-degree in weight normalization has the best performance, which proves the effectiveness of our method.

t-SNE visualization of graph vectors and different feature representation methods

To show the differences between graph vectors and different coding/representation features, we visualized the embedding spaces of them by projecting them into two dimensions using the t-distributed stochastic neighbor embedding (t-SNE). Figure 4 displays the t-SNE visualization of 4-mer frequency features, 5-mer frequency features, combining 4-mer embedding vectors with 4-mer frequency features, combining 4-mer embedding vectors with 5-mer frequency features and graph vectors. The different subcellular localization classes are marked with different colors. As shown in Figure 4(a) and (b), the two figures are very similar, the four types of samples in the feature space are distributed close, which suggests that the *k*-mer frequency features are not distinguishable features. From Figure 4(c) and (d), we can see that the four types of samples in the feature space are distributed loose. From Figure 4(e), we noted that the four types of samples are distributed more clearly compared to other feature representation methods. The results demonstrate the benefits of applying graph vectors, implying that the learned graph vectors can clearly distinguish between different subcellular localizations.

Robustness analysis between GraphLncLoc and *k*-mer frequency features

To further show the advantages of transforming sequences into graphs, we conducted some experiments to test the robustness of GraphLncLoc and *k*-mer frequency features. From the perspective of machine learning model design, if a lncRNA sequence changes slightly, a robust feature representation method should basically remain unchanged. In other words, a robust feature representation method should be resistant to the minor changes that actually occur during the input data [30, 31]. if a perturbed input results in the model outputting an incorrect answer with high confidence, then the model is considered with poor generalization and hard to be applied to the samples out of the training dataset. Thus, a robust model should keep the stability of prediction performance under small input perturbations ideally. Based on the theory, we performed robustness analysis to measure stability quantitatively. First, we generated a 'mutated' dataset from the original dataset by introducing three mutation actions including insertion, deletion and mutation. Specifically, the 'mutated' dataset is generated as follows:

1. Set a point mutation rate *M*;
2. For each nucleotide in a lncRNA sequence, we randomly generate a probability. If the probability is larger than the point mutation rate *M*, the nucleotide keeps the same; if the probability is smaller than or equal to the point mutation rate *M*, we randomly execute one of three actions to change the nucleotide.

Action 1 (insertion): we randomly insert a nucleotide (A, U, C, G) before the nucleotide;
Action 2 (deletion): we delete the nucleotide in the sequence;
Action 3 (mutation): we randomly change the nucleotide to another three types of nucleotides.

3. Repeat step 2 for all lncRNA sequences in our benchmark dataset, until all sequences have been 'mutated.'

It is worth noting that during the generating process, the labels of lncRNA sequences are not changed. After the generation process, we obtained a 'mutated' dataset from the original dataset.

Table 4. The performance of GraphLncLoc and its variant graph construction methods

Representation method	ACC	Macro F1-score	AUC
One-hot node features	0.485	0.342	0.753
Without weight normalization	0.577	0.424	0.800
Normalized weight (in-degree)	0.580	0.474	**0.811**
Our method	**0.612**	**0.506**	0.805

Note: The best performance values are highlighted in bold.

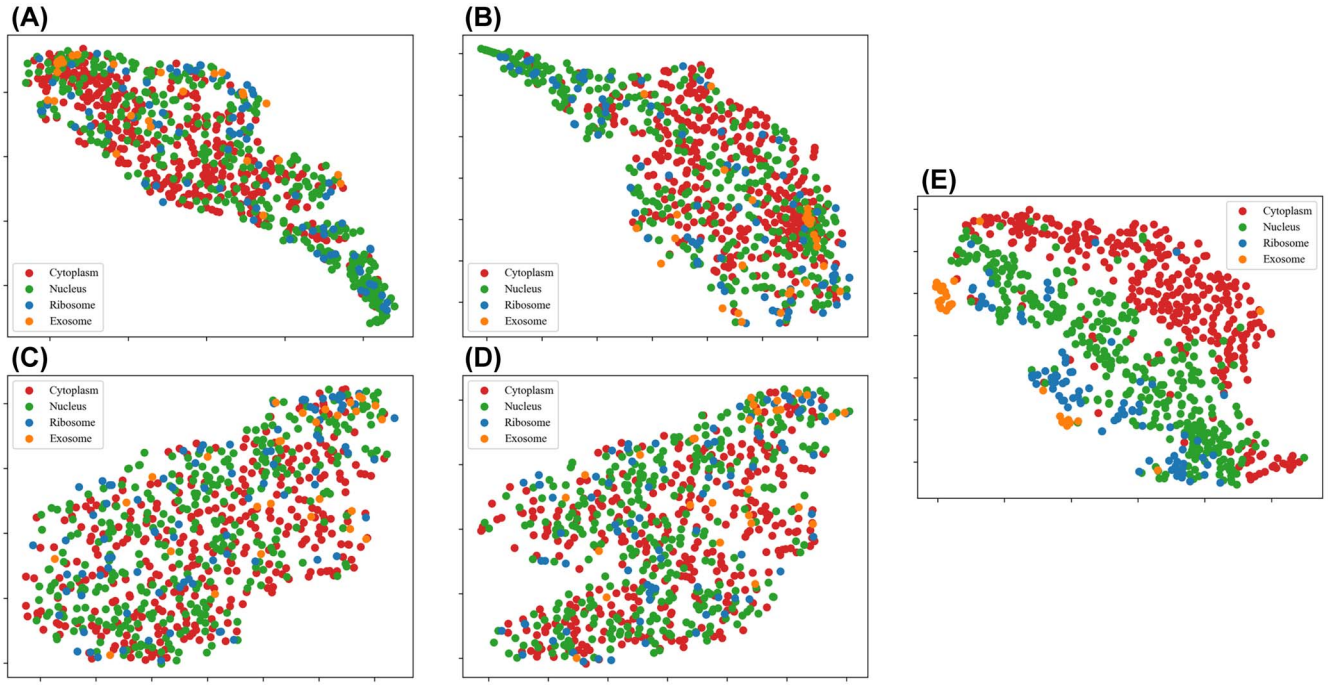

Figure 4. T-SNE visualization of 4-mer frequency features, 5-mer frequency features, combining 4-mer embedding vectors with 4-mer frequency features, combining 4-mer embedding vectors with 5-mer frequency features and graph vectors. Each dot represents a sample and its color represents its true class. (**A**) using 4-mer frequency features, (**B**) using 5-mer frequency features, (**C**) combining 4-mer embedding vectors with 4-mer frequency features, (**D**) combining 4-mer embedding vectors with 5-mer frequency features, (**E**) using graph vectors.

Then, we used GraphLncLoc and *k*-mer frequency features to encode the sequences in the 'mutated' dataset, and compared the differences with the original dataset. Because GraphLncLoc uses 4-mer as the node, and RF model achieves the best performance in traditional machine learning models (see Table 1), we used 4-mer + RF as the baseline for comparison. We queried some databases and found that the human genome mutation rate is estimated to be about 1×10^{-8}. However, a low mutation rate on input sequences basically has no effect on the machine learning model. Thus, we set the point mutation rates of 0.001 and 0.0001 to see the differences between the original and 'mutated' datasets. The results are shown in Figure 5.

From Figure 5, we can observe that the results are basically unchanged when M is 0.0001. When M is 0.001, in terms Macro F1-score, 4-mer + RF decreases from 0.377 to 0.325 (about 13.8%) while GraphLncLoc decreases only from 0.506 to 0.493 (about 2.6%). We could discern that the robustness of GraphLncLoc is better than 4-mer + RF, which implies the robustness of using graph vectors is better than using *k*-mer frequency features. The other evaluation metrics (Macro Precision and Macro Recall) indicate similar results. Thus, the evaluation of GraphLncLoc on the 'mutated' dataset confirmed its robustness.

Case study

To further show the ability to capture motifs of GraphLncLoc, we performed a case study on lncRNA RP11-57H14.2. The organism of lncRNA RP11-57H14.2 is *H. sapiens* and the tissue of lncRNA RP11-57H14.2 is K562 cell line. The subcellular localization of lncRNA RP11-57H14.2 is nucleus. After transforming the sequence into a graph by our method, the constructed graph has 236 nodes and 545 edges. We found a known motif associated with subcellular localization. According to Zhang *et al.* [32], motif AGCCC acts as a general nucleus localization signal. Since motif AGCCC is a 5-mer motif and the constructed graph used 4-mer as the nodes, we measured the importance of motif AGCCC by measuring the importance of node AGCC and node GCCC. Specifically, we obtained all vectors for each node in the constructed graph, and then used mean aggregation to obtain a new vector of any two adjacent nodes in the constructed graph. Lastly, we used the cosine function to calculate the similarity between the new vector and the whole graph vector. The output value is treated as the importance score which is in the range of 0 to 1. The larger score of the importance score is, the more important the corresponding motif is. Figure 6(a) shows the top 10 most important 5-mer motifs captured by GraphLncLoc. We can see that motif AGCCC obtains the highest importance score. Moreover, motif AGCCA is the sec-

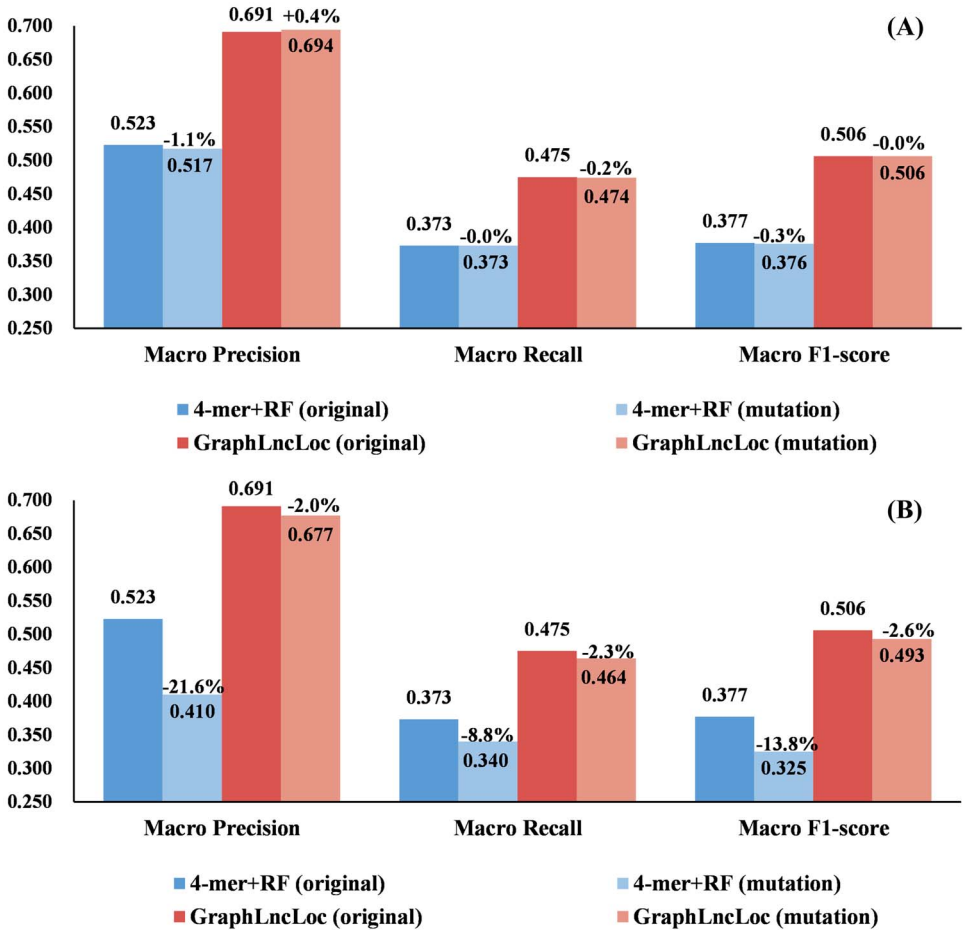

Figure 5. The performances of GraphLncLoc on original and 'mutated' datasets under different mutation rates. (**A**) Mutation rate is 0.0001, (**B**) mutation rate is 0.001.

ond important motif. The two motifs have the same 4-mer 'AGCC,' which shows the importance of a small community (the core node is AGCC). Before we conducted the experiment, we envisioned that the importance score has a strong relationship with the 5-mer frequency, i.e. 5-mers with higher frequency tend to be more important. Thus, we plotted the 5-mer frequency distribution of lncRNA RP11-57H14.2 in Figure 6(b). The frequency of motif AGCCC is 4 in lncRNA RP11-57H14.2. We can observe that there are several motifs whose frequency is greater than or equal to AGCCC. The observation implies that the importance score does not have a strong relationship with frequency, which indicates that our network structure really captured the important motifs. In summary, our results are consistent with Zhang *et al.*'s findings [32].

GraphLncLoc web server

To facilitate researchers using GraphLncLoc to predict lncRNA subcellular localization, a user-friendly web server was developed. Users can access this web server by visiting http://csuligroup.com:8000/GraphLncLoc/. A step-by-step guide is given as follows.

Step 1: Type a query lncRNA sequence into the input box. The GraphLncLoc web server accepts the input sequence with the length from 200 to 100 000. The form of the input sequence should be in FASTA format. Users can access the example sequence by clicking the Example button.

Step 2: After typing a lncRNA sequence, click the Submit button to submit the lncRNA sequence to GraphLncLoc. GraphLncLoc usually takes less than 5 s to calculate the predicted probability of the lncRNA subcellular localization.

Step 3: The results are shown in a table with five columns, column 1 is the sequence ID, columns 2–5 are the four subcellular localizations and the corresponding predicted probabilities. Finally, the final predicted positions are marked in red and displayed at the bottom of the table.

Discussion

In this study, we proposed GraphLncLoc, a GCN-based deep learning model for predicting lncRNA subcellular localization prediction. Compared with previous studies, GraphLncLoc has two main novel design ideas: (i) GraphLncLoc is the first method to transform lncRNA sequences into graphs in lncRNA subcellular localization prediction; (ii) to extract high-level features of the constructed graph, GraphLncLoc applies GCN to learn the latent representations. To evaluate the performance of GraphLncLoc, we conducted extensive experiments and the results show that GraphLncLoc outperforms traditional machine learning methods and existing predictors. Finally, we developed a user-friendly web server. We believe that GraphLncLoc is an effective tool for predicting lncRNA subcellular localization.

Although the prediction performance of GraphLncLoc is promising, there are still several limitations of the model.

(1) We only used lncRNA sequence information as node features, and did not consider integrating other biological information. Obviously, utilizing some relevant biological information can better predict lncRNA subcellular localization. Therefore, in future work, if we can collect more useful

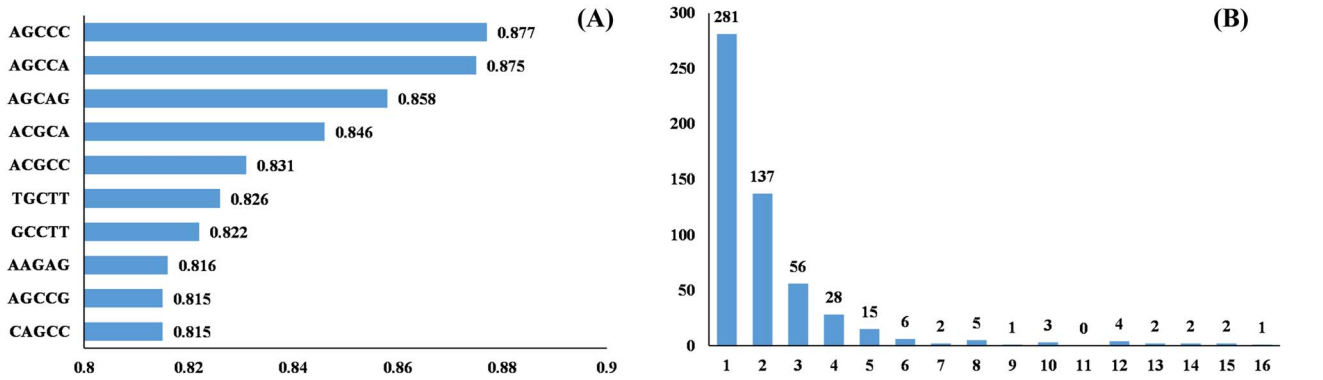

Figure 6. (**A**) Top 10 most important motifs captured by GraphLncLoc. The x-axis shows the importance scores and the y-axis shows the top 10 most important 5-mer motifs captured by GraphLncLoc. (**B**) The 5-mer frequency distribution of lncRNA RP11-57H14.2. The x-axis shows the frequencies of 5-mer and the y-axis shows the counts of 5-mers for each frequency.

lncRNA biological information, we can enrich the feature representation to train a more powerful model.

(2) To reduce the computational time and cost, we did not attempt to use complex GCN models to extract high-level features from sequence information. With the rapid development of deep learning and natural language processing fields, many powerful encoders and network architectures will be proposed. Therefore, we will continue to follow the development of cutting-edge deep learning techniques and try to use more powerful network architecture to predict the lncRNA subcellular localization.

(3) We used a fixed k in our study to construct the de Bruijn graph. Although the constructed graph has the ability to connect nodes to form paths, and then capture specific patterns or motifs, it fails to capture too long dependencies. Therefore, considering using different values of k would be possible to provide multi-scale information, which is a promising future direction.

(4) We only considered the lncRNAs that are associated to only one subcellular localization. However, in reality, many lncRNAs have multiple subcellular localizations. For example, lncRNA SNHG1 displays cytoplasmic distribution in human HCT116 colon cancer cells. Upon DNA damage stress, they are retained in the nucleus compartment. Although some computational methods have been developed, few of them are designed for lncRNAs with multiple subcellular localizations. Thus, considering lncRNAs with multiple subcellular localizations is necessary and useful. In the future, we expect to collect more labeled lncRNAs with multiple subcellular localizations, and then we can use more samples to train a more powerful model. In addition, we will change some components in our model to make it from a multi-class classifier to a multi-label classifier, such as changing the cross-entropy loss function to the binary cross-entropy loss function, and changing the softmax function to the sigmoid function in the output layer.

Most existing computational methods struggle to deal with variable-length lncRNA sequences. We proposed a novel paradigm for lncRNA subcellular localization prediction by transforming sequences into graphs. We believe that the encoding method in GraphLncLoc can be used as a general representation method for biological sequences. It is expected to be applied to other biological sequence prediction problems, such as mutation prediction of influenza viruses [33], drug–protein prediction [34], essential gene prediction [35] and binding-site prediction [36, 37].

Key Points

- We proposed a sequence-based graph convolutional network model called GraphLncLoc to predict lncRNA subcellular localization.
- GraphLncLoc transforms lncRNA sequences into de Bruijn graphs, which can keep local order information of lncRNA sequences and automatically capture patterns and motifs of different lengths in lncRNA sequences.
- Extensive experiments demonstrated that GraphLncLoc achieves better performance than the existing predictors.
- Our analyses showed that transforming sequences into graphs has more distinguishable features and is more robust than k-mer frequency features. The case study showed that GraphLncLoc can uncover important motifs for nucleus subcellular localization.
- The user-friendly web server of GraphLncLoc is available at http://csuligroup.com:8000/GraphLncLoc/.

Supplementary data

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

Data availability

GraphLncLoc web server is available at http://csuligroup.com:8000/GraphLncLoc/. Code is available at https://github.com/CSUBioGroup/GraphLncLoc.

Funding

This work was supported in part by the National Natural Science Foundation of China under Grant (No. 62102457, No. 62002390), Hunan Provincial Science and Technology Program (2019CB1007), Scientific Research Fund of Hunan Provincial Education Department (No. 22B0153, No. 22A0007).

References

1. Lu Q, Ren S, Lu M, *et al*. Computational prediction of associations between long non-coding RNAs and proteins. *BMC Genomics* 2013;**14**(1):1–10.

2. Kretz M, Siprashvili Z, Chu C, *et al.* Control of somatic tissue differentiation by the long non-coding RNA TINCR. *Nature* 2013;**493**(7431):231–5.
3. Wu Z, Liu X, Liu L, *et al.* Regulation of lncRNA expression. *Cell Mol Biol Lett* 2014;**19**(4):561–75.
4. Martianov I, Ramadass A, Serra Barros A, *et al.* Repression of the human dihydrofolate reductase gene by a non-coding interfering transcript. *Nature* 2007;**445**(7128):666–70.
5. 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.
6. Chen L-L. Linking long noncoding RNA localization and function. *Trends Biochem Sci* 2016;**41**(9):761–72.
7. Carlevaro-Fita J, Johnson R. Global positioning system: understanding long noncoding RNAs through subcellular localization. *Mol Cell* 2019;**73**(5):869–83.
8. Cabili MN, Dunagin MC, McClanahan PD, *et al.* Localization and abundance analysis of human lncRNAs at single-cell and single-molecule resolution. *Genome Biol* 2015;**16**(1):1–16.
9. Tseng Y-Y, Moriarity BS, Gong W, *et al.* PVT1 dependence in cancer with MYC copy-number increase. *Nature* 2014;**512**(7512):82–6.
10. Cesana M, Cacchiarelli D, Legnini I, *et al.* A long noncoding RNA controls muscle differentiation by functioning as a competing endogenous RNA. *Cell* 2011;**147**(2):358–69.
11. Chakrabortty SK, Prakash A, Nechooshtan G, *et al.* Extracellular vesicle-mediated transfer of processed and functional RNY5 RNA. *RNA* 2015;**21**(11):1966–79.
12. Voit EO, Martens HA, Omholt SW. 150 years of the mass action law. *PLoS Comput Biol* 2015;**11**(1):e1004012.
13. 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**(13):2185–94.
14. Su ZD, Huang Y, Zhang ZY, *et al.* iLoc-lncRNA: predict the subcellular location of lncRNAs by incorporating octamer composition into general PseKNC. *Bioinformatics* 2018;**34**(24):4196–204.
15. Gudenas BL, Wang L. Prediction of lncRNA subcellular localization with deep learning from sequence features. *Sci Rep* 2018;**8**(1):16385.
16. Ahmad A, Lin H, Shatabda S. Locate-R: subcellular localization of long non-coding RNAs using nucleotide compositions. *Genomics* 2020;**112**(3):2583–9.
17. Fan Y, Chen M, Zhu Q. lncLocPred: predicting lncRNA subcellular localization using multiple sequence feature information. *IEEE Access* 2020;**8**:124702–11.
18. Feng S, Liang Y, du W, *et al.* lncLocation: efficient subcellular location prediction of long non-coding RNA-based multi-source heterogeneous feature fusion. *Int J Mol Sci* 2020;**21**(19):7271.
19. Zeng M, Wu Y, Lu C, *et al.* DeepLncLoc: a deep learning framework for long non-coding RNA subcellular localization prediction based on subsequence embedding. *Brief Bioinform* 2022;**23**(1):bbab360.
20. Lin Y, Pan X, Shen HB. lncLocator 2.0: a cell-line-specific subcellular localization predictor for long non-coding RNAs with interpretable deep learning. *Bioinformatics* 2021;**37**:2308–16.
21. Zhang Z-Y, Sun ZJ, Yang YH, *et al.* Towards a better prediction of subcellular location of long non-coding RNA. *Front Comp Sci* 2022;**16**(5):1–7.
22. Compeau PE, Pevzner PA, Tesler G. How to apply de Bruijn graphs to genome assembly. *Nat Biotechnol* 2011;**29**(11):987–91.
23. Zhang T, Tan P, Wang L, *et al.* RNALocate: a resource for RNA subcellular localizations. *Nucleic Acids Res* 2017;**45**(D1):D135–8.
24. Pruitt KD, Tatusova T, Maglott DR. NCBI reference sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res* 2005;**33**(suppl_1):D501–4.
25. Cunningham F, Allen JE, Allen J, *et al.* Ensembl 2022. *Nucleic Acids Res* 2022;**50**(D1):D988–95.
26. Huang Y, Niu B, Gao Y, *et al.* CD-HIT suite: a web server for clustering and comparing biological sequences. *Bioinformatics* 2010;**26**(5):680–2.
27. Wang D, Zhang Z, Jiang Y, *et al.* DM3Loc: multi-label mRNA subcellular localization prediction and analysis based on multi-head self-attention mechanism. *Nucleic Acids Res* 2021;**49**(8):e46–6.
28. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907. 2016.
29. 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*. 2017:2980–8.
30. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572. 2014.
31. Huber PJ. *Robust statistics*. In: *International Encyclopedia of Statistical Science*. Springer, 2011, 1248–51.
32. Zhang B, Gunawardane L, Niazi F, *et al.* A novel RNA motif mediates the strict nuclear localization of a long noncoding RNA. *Mol Cell Biol* 2014;**34**(12):2318–29.
33. Yin R, Luusua E, Dabrowski J, *et al.* Tempel: time-series mutation prediction of influenza A viruses via attention-based recurrent neural networks. *Bioinformatics* 2020;**36**(9):2697–704.
34. Wu Y, Gao M, Zeng M, *et al.* BridgeDPI: a novel graph neural network for predicting drug–protein interactions. *Bioinformatics* 2022;**38**(9):2571–8.
35. Li Y, Zeng M, Wu Y, *et al.* Accurate prediction of human essential proteins using ensemble deep learning. *IEEE/ACM Trans Comput Biol Bioinform* 2021;**PP**:1.
36. Zeng M, Zhang F, Wu FX, *et al.* Protein–protein interaction site prediction through combining local and global features with deep neural networks. *Bioinformatics* 2020;**36**(4):1114–20.
37. Zhang F, Shi W, Zhang J, *et al.* PROBselect: accurate prediction of protein-binding residues from proteins sequences via dynamic predictor selection. *Bioinformatics* 2020;**36**(Supplement_2):i735–44.