Computational and Structural Biotechnology Journal 23 (2024) 2034–2048

Contents lists available at ScienceDirect

Computational and Structural Biotechnology Journal

journal homepage: www.elsevier.com/locate/csbj

Research Article

GP-HTNLoc: A graph prototype head-tail network-based model for multi-label subcellular localization prediction of ncRNAs

Shuangkai Han [a,b], Lin Liu [a,b,*]

[a] School of Information, Yunnan Normal University, Kunming, China
[b] Engineering Research Center of Computer Vision and Intelligent Control Technology, Department of Education of Yunnan Province, China

ARTICLE INFO

Keywords:
Non-coding RNA subcellular localization prediction
Multi-label classification
Class imbalance
Heterogeneous graph representation learning

ABSTRACT

Numerous research results demonstrated that understanding the subcellular localization of non-coding RNAs (ncRNAs) is pivotal in elucidating their roles and regulatory mechanisms in cells. Despite the existence of over ten computational models dedicated to predicting the subcellular localization of ncRNAs, a majority of these models are designed solely for single-label prediction. In reality, ncRNAs often exhibit localization across multiple subcellular compartments. Furthermore, the existing multi-label localization prediction models are insufficient in addressing the challenges posed by the scarcity of training samples and class imbalance in ncRNA dataset. To address these limitations, this study proposes a novel multi-label localization prediction model for ncRNAs, named GP-HTNLoc. To mitigate class imbalance, GP-HTNLoc adopts separate training approaches for head and tail location labels. Additionally, GP-HTNLoc introduces a pioneering graph prototype module to enhance its performance in small-sample, multi-label scenarios. The experimental results based on 10-fold cross-validation on benchmark datasets demonstrate that GP-HTNLoc achieves competitive predictive performance. The average results from 10 rounds of testing on an independent dataset show that GP-HTNLoc outperforms the best existing models on the human lncRNA, human snoRNA, and human miRNA subsets, with average precision improvements of 31.5%, 14.2%, and 5.6%, respectively, reaching 0.685, 0.632, and 0.704. A user-friendly online GP-HTNLoc server is accessible at https://56s8y85390.goho.co.

1. Introduction

Previous studies have identified a large number of non-coding RNAs (ncRNAs) in the mammalian genome, and while it is entirely possible that most of these ncRNAs are transcriptional noise or by-products of RNA processing, there is growing evidence that most of them are functional and provide a variety of regulatory activities in the cell [1]. Recent research underscores the intimate association between ncRNAs and the onset and progression of specific diseases [2]. Particularly, most ncRNAs exhibit varying local concentrations, interacting partners, post-transcriptional modifications, and regulatory pathways in diverse subcellular locations. These differences significantly impact protein synthesis and cellular functions [3,4]. For instance, miR-122, a highly expressed miRNA in the liver, predominantly functions within the cytoplasm of hepatic cells. Its influence on metabolic activities in the liver arises from its binding to target gene mRNAs, thereby regulating their translation or stability [5]. Consequently, the investigation of the subcellular localization of ncRNAs emerges as a crucial avenue for unraveling the functional intricacies and regulatory mechanisms inherent in these molecules. This work provides substantial value for researchers in elucidating gene regulation, cellular functions, and the mechanisms underlying various disorders and biological processes in multicellular organisms.

Although traditional RNA subcellular localization methods, such as Fluorescence in situ hybridization (FISH) [6] or Subcellular fractionation [7], can ensure high localization accuracy, they are only suitable for small-scale studies because they are expensive and time-consuming. Facing the current high-throughput needs, researchers are trying to find some computational methods to enhance the efficiency and reduce the workload of RNA subcellular localization. Supported by a rapidly growing database of RNA subcellular localization[8–11], computational model-based methods for subcellular localization of RNAs have emerged as a focal point of this research domain in recent years.

Non-coding RNAs (ncRNAs) are classified into different categories according to their length, function, and subcellular location, and the ncRNAs that have been widely studied by the biological community

* Correspondence to: School of Information, Yunnan Normal University, China.
E-mail address: liulin@ynnu.edu.cn (L. Liu).

https://doi.org/10.1016/j.csbj.2024.04.052
Received 8 February 2024; Received in revised form 17 April 2024; Accepted 18 April 2024
Available online 3 May 2024

include long noncoding RNAs (lncRNAs), small nucleolar RNAs (snoRNAs), microRNAs (miRNAs), and PIWI-interacting RNAs (piRNAs) [11–13]. For long non-coding RNAs (lncRNAs), Fan et al. developed lncLocPred, a logistic regression-based machine learning predictor specifically designed for predicting the subcellular localization of lncRNAs [14]. Addressing the challenge of limited samples in lncRNA subcellular localization, Cai et al. introduced GM-lncLoc, a meta-learning training model facilitating knowledge transfer through meta-parameters [15]. On the other hand, for microRNAs (miRNAs), Yang et al. proposed MiRGOFS, a functional similarity measure based on Gene Ontology (GO), for predicting miRNA subcellular localization and miRNA-disease associations [16]. Similarly, Zhang et al. developed the iLoc-miRNA model based on bidirectional long short-term memory networks and multi-head self-attention mechanisms. This model is used to predict whether miRNAs are located intracellularly or extracellularly [17]. Furthermore, Wang et al. developed a model utilizing the Hilbert-Schmidt Independence Criterion for Multi-Kernel Learning (MK-HSIC), which targets multiple RNA types including mRNAs, lncRNAs, miRNAs, and snoRNAs [18]. Building upon this approach, Zhou et al. introduced the MKGHKNN model, a multi-kernel graph regularization K local hyperplane distance nearest neighbor model, tailored for lncRNAs, miRNAs, and snoRNAs [19]. Additionally, Bai et al. presented the ncRNALocate-EL model, leveraging natural language processing to extract high-level features from ncRNA sequences, focusing on lncRNAs, miRNAs, and snoRNAs [20]. Overall, various machine learning models are currently blossoming in the field of ncRNA subcellular localization prediction.

Nonetheless, current ncRNA subcellular localization studies still face three major challenges: (1)small-sample Challenge: datasets containing reliable localization information for ncRNAs typically comprise only a few hundred sequences, insufficient for the effective training of machine learning models; (2) Class imbalance challenge: there are significant differences in the number of ncRNA samples involved in different subcellular locations, which makes it very difficult for the model to learn at locations where the number of samples is scarce; (3) Multi-labeling Challenge: the research indicates that a single primary RNA transcript can be utilized to produce multiple proteins [21–23]. Therefore, extending RNA subcellular localization to a multi-label classification problem holds significant practical significance. However, the majority of existing studies focus on single-label localization for ncRNAs. Among the existing predictive models for ncRNA subcellular localization, several resampling techniques are employed in several enhanced machine learning-based localization prediction models to tackle class imbalance [24–26]. Although effective in achieving balance in the number of categories within the training data, these techniques come at the expense of losing the original data distribution features or introducing new noise. In addressing the multi-labeling challenge, a predominant approach employed by the majority of extant methodologies involves the utilization of the One-vs-Rest strategy [18–20]. However, the One-vs-Rest strategy overlooks the relationships between labels, often making it challenging for the model to learn label association information.

In multi-label classification problems in natural language processing (NLP), there is a significant class imbalance. A few of the labels (called head labels) are associated with a large number of documents, while the majority of the labels (called tail labels) are associated with a small number of documents, and the label frequency exhibits a distinct long-tailed distribution. This makes the tail class samples scarce and the model difficult to train [27–29]. To address this problem, Xiao et al. proposed the HTTN model, which employs a transfer learner that transforms class prototypes into classifier parameters, facilitating the transfer of meta-knowledge from data-rich head labels to data-poor tail labels [27]. In HTTN, the method of calculating the average of multiple samples within the same class to represent the prototype of the class is effective in the natural language processing field where there is abundant sample data. However, for the problem of ncRNA subcellular localization prediction where both head and tail class samples are scarce, this method often fails to obtain a high-quality prototype representation, making it difficult to improve the subsequent classification accuracy. To address this challenge, our study proposes a novel model, GP-HTNLoc, which adopts the concept of transfer learning to enhance learning of the sparse tail classes by leveraging meta-knowledge acquired from the data-rich head classes. Specifically, GP-HTNLoc divides localization labels into head labels and tail labels based on the number of samples involved and trains them separately. Additionally, the model introduces an innovative graph prototype module. Unlike HTNN, which computes sample averages to obtain label prototypes, the graph prototype module constructs a heterogeneous graph based on the relationship between ncRNA labels and samples. On this heterogeneous graph, label prototypes are obtained using Heterogeneous Graph Convolutional Networks (HGCN) [30] and the classical random walk algorithm MetaPath2Vec [31]. These prototypes are used to train a transfer learner from the head label prototypes to the parameters of the head classifier. The trained transfer learner can then convert tail label prototypes into tail classifier parameters. The combination of head and tail classifiers forms the final multi-label classifier. For new ncRNA sequences, feeding the feature vector into the trained multi-label classifier yields a binary vector indicating the potential subcellular locations of the sequence.

This study trained and optimized GP-HTNLoc using a 10-fold cross-validation method on a benchmark dataset spanning 12 species and encompassing 9 subcellular locations. To further demonstrate the performance of GP-HTNLoc, we conducted 10 rounds of testing on an independent dataset. The results of these tests indicate that GP-HTNLoc significantly outperforms existing state-of-the-art models. Additionally, our case study highlighted the usability and robustness of GP-HTNLoc in real-world scenarios. Ablation study confirmed the significant contributions of the head-tail network and graph prototype module to the model's performance improvement. Finally, in the snoRNA and lncRNA subsets of the benchmark dataset, we utilized the SHAP (Shapley Additive exPlanation) algorithm [32] to identify and explain the features that have the greatest impact on the model's predictions for each subcellular location. These features, which the model focuses on, may indicate certain localization patterns.

2. Materials and methods

2.1. Experimental datasets

2.1.1. Benchmark dataset

RNALocate database [33] is a widely used database for RNA subcellular localization prediction research. It contains over 42,000 manually curated RNA subcellular localization entries with experimental (FISH and ISH etc.) evidence, involving more than 23,100 RNAs across 42 subcellular locations in 65 species, including Homo sapiens, Mus musculus, and Saccharomyces cerevisiae etc. In this study, we utilized the multi-location ncRNA localization dataset, constructed by Zhou and Wang [18,19] from the RNALocate database, as the benchmark dataset for training and evaluating the effectiveness of our model. This dataset comprises three categories of ncRNA: lncRNA, miRNA, and snoRNA. It covers a total of 12 species, including Homo sapiens, Mus musculus, and Sus scrofa, and spans nine subcellular locations such as Nucleus, Ribosome, and Cytosol. Zhou and Wang initially divided the benchmark dataset into three subsets based on the RNA type, namely the lncRNA dataset, miRNA dataset, and snoRNA dataset. Subsequently, they further selected human-specific sequences from each of the three subsets, resulting in the H_lncRNA dataset, H_miRNA dataset, and H_snoRNA dataset. In total, the benchmark dataset consists of six subsets. Fig. 1 illustrates the distribution of RNA sequence counts across different subcellular locations within each subset. It reveals the class imbalance that exists in each subset. For detailed information on the number of sequences corresponding to different subcellular locations within each subset, please refer to Supplementary Table S1. Information

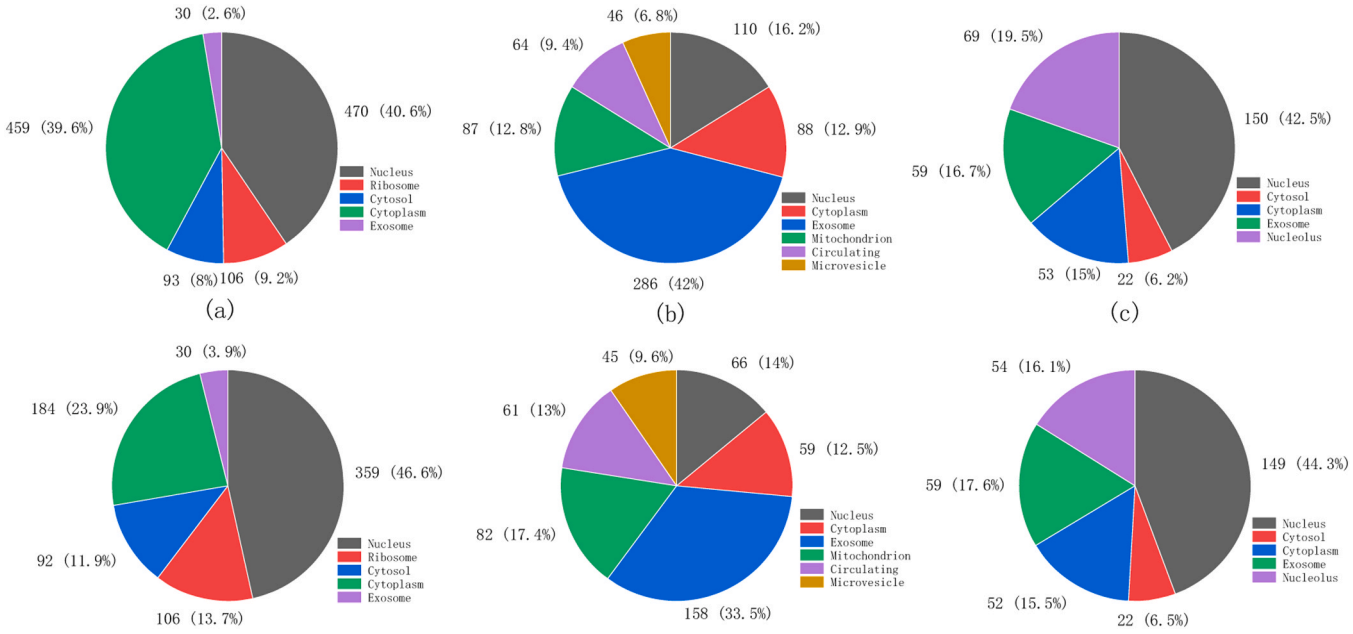

Fig. 1. The class imbalance phenomenon present in the six subsets of the benchmark dataset: (a) LncRNA dataset; (b) miRNA dataset; (c) snoRNA dataset; (d) Human LncRNA dataset; (e) Human miRNA dataset; (f) Human snoRNA dataset.

on the species involved in the first three subsets and the number of sequences corresponding to each species can be found in Supplementary Table S2.

2.1.2. Independent dataset

To further validate the performance of the model proposed in this study, we employed a dataset constructed by Bai et al. [20] from RNALocate v2.0 [10] as an independent dataset to test our model. RNALocate v2.0, an extension of RNALocate, catalogues over 210,000 RNA subcellular localization entries with experimental evidence, encompassing more than 110,000 RNAs from 104 species across 171 subcellular locations. Bai et al. extracted human ncRNA sequences from RNALocate v2.0, excluding sequences overlapping with the dataset by Zhou and Wang et al., to obtain an independent dataset comprising human lncRNA, human miRNA, and human snoRNA. Information regarding the number of sequences per subcellular location in the independent dataset can be found in Supplementary Table S3. Compared with the benchmark dataset, the independent dataset was greatly expanded in terms of the number of human lncRNAs. Fig. 2 illustrates the distribution of human lncRNAs across different subcellular locations within the independent dataset.

2.2. Model architecture

The overall architecture of the GP-HTNLoc model proposed in this study consists of three stages: (i) Imbalanced learning based on graph prototypes; (ii) Fine-tuning; and (iii) Prediction. In the first stage, the primary features of the sequence, denoted as X^S, are first obtained by multiple sequence feature extraction methods. The set of X^S and its corresponding label dataset in the training set are referred to as the Base Data. The Base Data is then partitioned into D_{head} and D_{tail} based on head and tail labels, respectively.

For D_{head}, on the one hand, the primary sequence features X^S is fed into a bidirectional long short-term memory (BiLSTM) network [34] to obtain advanced features represented as $\acute{X}^S$. The head classifier parameters, denoted as M_{head}, are subsequently trained using $\acute{X}^S$. On the other hand, its primary features are directly fed into the proposed graph prototype module in this study. The graph prototype module utilizes the association information between ncRNA samples and labels to construct a heterogeneous graph G. On G, HGCN and MetaPath2Vec are used to learn the graph structure and aggregate sample features to obtain the

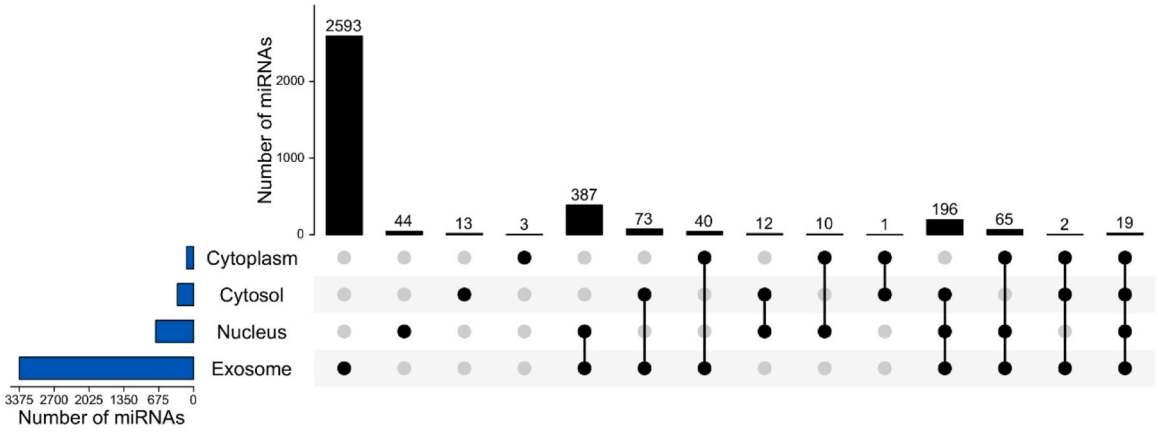

Fig. 2. Distribution of subcellular localization of lncRNAs in independent datasets.

embedding of labeled nodes, denoted as $\widetilde{X}^{L}_{head}$. The transfer learner is then used to learn the mapping from $\widetilde{X}^{L}_{head}$ to M_{head}.

For D_{tail}, the primary features are input into the graph prototype module to obtain the prototype representation of tail class, denoted as $\widetilde{X}^{L}_{tail}$. By inputting $\widetilde{X}^{L}_{tail}$ into the pre-trained Transfer Learner, the parameters for the tail classifier, denoted as M_{tail}, are obtained. Finally, M_{head} and M_{tail} are concatenated to form the complete parameters M for ncRNA multi-label classification.

In the fine-tuning phase, M is lightly trained using tail class samples containing both head and tail labels, called Novel Data. In the prediction stage, after the new ncRNA sequences are acquired with deep sequence features, they are directly input into the fine-tuned M to obtain the prediction of subcellular multi-label localization of ncRNA sequences.

Fig. 3 illustrates the overall workflow of GP-HTNLoc, using the imbalance learning, fine-tuning, and prediction on the LncRNA dataset as an example. Subsequently, this paper will provide a detailed introduction to the workflow of GP-HTNLoc.

2.3. GP-HTNLoc

2.3.1. Deep feature extraction

The deep feature extraction section comprises primary feature extraction and advanced feature extraction. Primary features are obtained from the original ncRNA sequences through one or multiple sequence feature extraction methods, while advanced features are acquired by feeding the primary features into a BiLSTM equipped with an attention mechanism. The following will provide a detailed introduction to the two-level feature extraction process.

In this study, a combination of eight sequence feature extraction methods was employed to derive the primary features of ncRNA sequences. These eight methods include K-mer (comprising 2-mer, 3-mer, 4-mer), Nucleic Acid Composition (NAC), Di-Nucleotide Composition (DNC), Tri-Nucleotide Composition (TNC), Region Coverage Rate (Coverage-Rate), and Fickett score. Given the widespread utilization of K-mer, NAC, DNC, and TNC for biological sequence feature extraction [35–37], detailed descriptions are omitted in this paper, and specific calculation formulas adhere to those described in the study by Zhou et al. [19].

Additionally, it has been shown that some short open reading frames

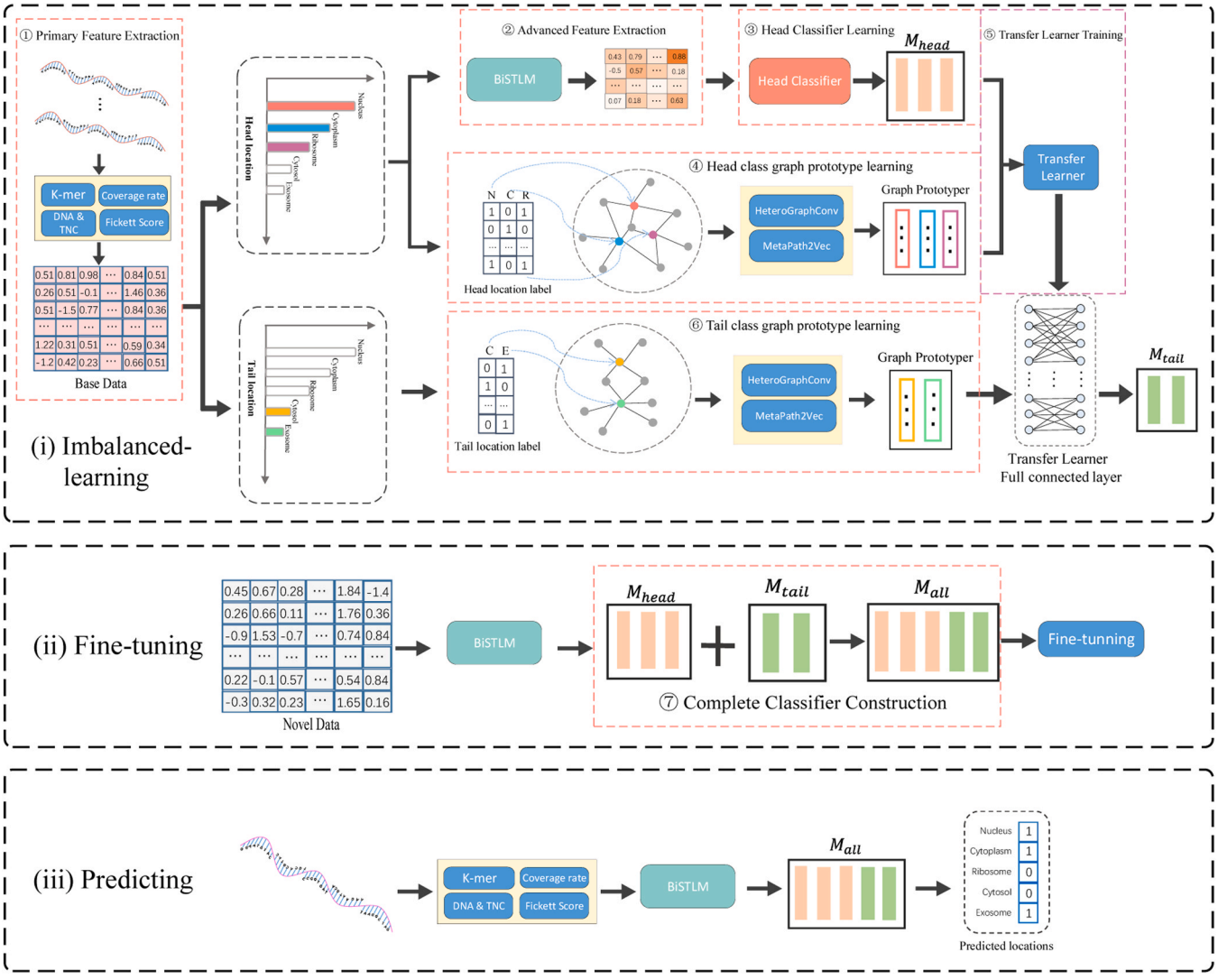

Fig. 3. The overall architecture of GP-HTNLoc comprises three main components: (i) imbalanced learning based on graph prototypes, (ii) fine-tuning, and (iii) prediction. In the unbalanced learning phase, this study introduces a graph prototype module that obtains prototypical representations of labels from head label samples, which in turn trains a transfer learner to transfer rich categorization knowledge from the head class to the sample-scarce tail class (illustrated in the figure using the lncRNA dataset).

(ORFs) of ncRNAs have the potential to encode micropeptides [38], and the 3′-UTR (3′ untranslated region) and 5′-UTR (5′ untranslated region) in non-ORF regions are equally biologically important as two important segments of RNA [39]. Therefore, the Region Coverage Rate (Coverage-Rate) feature used in this study actually encompasses five dimensions: the overall ORF coverage rate, 3′-UTR coverage rate and 5′-UTR coverage rate, the cytosine (C) content difference and guanine (G) content difference between 3′-UTR and 5′-UTR. The content difference refers to the subtraction of the number of C/G in two sequences, and the formula for coverage rate calculation is as follows:

$$Coverage_{ORF} = \frac{ORF(S)}{l(S)} \tag{1}$$

$$Coverage_{3'\mathrm{UTR}/5'-\mathrm{UTR}} = \frac{3'\mathrm{UTR}(S)/5'\mathrm{UTR}(S)}{l(S)} \tag{2}$$

Here, S represents an RNA sequence, ORF(S) represents the total length of the Open ORF, $l(S)$ denotes the total length of the RNA sequence. 3′-UTR(S)/5′-UTR(S) represents the length of the 3′-UTR or 5′-UTR region.

Fickett and colleagues proposed that codons may exhibit asymmetric biases and nucleotide content, which can be utilized to distinguish between non-coding and protein-coding regions of a sequence [40]. Based on this observation, they introduced the Fickett Score to characterize the distinctiveness in nucleotide content and position. For example, the position frequency of a certain nucleotide P can be described by the following formula:

$$\begin{cases} P_1 = \sum_{i=0}^{L/3} A_{3i+1} \\ P_2 = \sum_{j=0}^{L/3} A_{3j+2} \\ P_3 = \sum_{k=0}^{L/3} A_{3k+3} \end{cases} \tag{3}$$

where L represents the total number of nucleotides in the sequence,

$$A_i = \begin{cases} 1 & P \text{ is appeared at position } i \\ 0 & else \end{cases}$$

the final position frequency of nucleotide P is expressed as:

$$F_{result} = \frac{MAX(P_1, P_2, P_3)}{MAX(P_1, P_2, P_3) + 1} \quad , F \in \{A, G, C, T(U)\} \tag{4}$$

The Fickett Score feature utilized in this study actually encompasses eight dimensions: the positional frequency information of adenine (A), guanine (G), cytosine (C), and thymine (T) (or uracil (U)), as well as their respective percentages in the sequence.

So far, we have introduced eight methods used in this study to extract primary features from ncRNA sequences. Table 1 provides the final dimensions of each feature and assigns an identifier to each feature to simplify the description in the experimental section.

After obtaining the primary features $X^S = \{x_1^S, \ldots, x_i^S, \ldots x_m^S\}$ of the ncRNA sequence through the above multiple sequence feature extraction methods, we use a BiLSTM with attention mechanism [27] to further extract the deeper features of the sequence $\acute{X}^S = \{\acute{x}_1 S, \ldots, \acute{x}_i^S, \ldots \acute{x}_m S\}$,

$$\acute{x}^S = BiLSTM(x^S) \tag{5}$$

Table 1
Feature dimensions and identifiers.

Features	Dimensions	identifiers
2-mer	16	A
3-mer	64	B
4-mer	256	Z
Coverage-Rate	5	C
NAC	4	N
DNC	16	D
TNC	64	T
Fickett Score	8	F

2.3.2. Head classifier learning

In order to circumvent excessively complex classifier models that fail to yield robust generalization performance on datasets with limited samples, this study exclusively employs a single-layer perceptron devoid of bias terms as the classifier. The classifier solely consists of the weight matrix $M_{head} \in \mathbb{R}^{d \times l_{head}}$, where l_{head} represents the number of head labels. The head classifier is trained utilizing the deep features extracted from sequences in D_{head}, denoted as $\acute{X}^S = \{\acute{x}_1 S, \ldots, \acute{x}_i^S, \ldots \acute{x}_m S\}$,

$$\hat{y} = sigmoid\left(\acute{x}^S M_{head}\right) \tag{6}$$

The weights of the head classifier, denoted as M_{head} are learned by minimizing the cross-entropy loss function. After multiple epochs of training on D_{head}, we save the parameters of M_{head} that yield the best performance as the final head classifier parameters. The head classifier is trained on D_{head}, which contains a substantial number of ncRNA samples, resulting in its superior classification capability. In the subsequent steps of the workflow, we aim to transfer this enhanced classification capability of the head classifier to the tail classifier, using graph prototype module and transfer learning ideology.

2.3.3. Graph prototype module

Graph representation learning is an important branch in the field of machine learning that aims to represent nodes and edges in graph data, such as social networks, recommendation systems, and bioinformatics, as vectors or embeddings. This helps in performing downstream tasks such as node classification, link prediction, community detection and recommendation [41]. Researchers have developed various graph embedding techniques to convert original graph data into high-dimensional vectors. Examples include graph convolutional networks (GCNs) [42], variational graph autoencoders (VGAEs) [43], graph attention networks (GATs) [44], and random walk-based models such as DeepWalk [45] and Node2Vec [46]. For general homomorphic graphs these methods show excellent graph embedding capabilities, but for heteromorphic graphs the above methods are not up to the task. Considering the complexity of heterogeneous graphs, researchers have developed representation learning methods specialized for heterogeneous graphs, such as Heterogeneous Graph Convolutional Networks (HGCN), MetaPath2Vec, etc., and these methods have achieved excellent performances in heterogeneous graph embedding in different scenarios [47].

In this study, we innovatively propose a graph prototype module, which utilizes the association information between ncRNA labels and samples to construct a heterogeneous graph, learns a high-dimensional embedding of labeled nodes on the heterogeneous graph using HGCN and MetaPath2Vec, and uses this high-dimensional embedding as a category prototype for subsequent model training. Fig. 4 illustrates the workflow of the graph prototype module, taking the head class of the LncRNA dataset as an example. The process is delineated into the following steps:

Step 1 – Graph Construction: A two-dimensional table can be generated from the localization label set $Y = \{y_i \in \{0,1\}^n\}$, as depicted in Fig. 4. In the table, rows represent different LncRNA samples, and columns represent different localization labels. If the position at the i-th row and j-th column is 0, it indicates an association between the LncRNA sample v_i^S and the label v_j^L. In this case, a directed edge is established from v_i^S to v_j^L to signify that the sample belongs to the label, denoted as

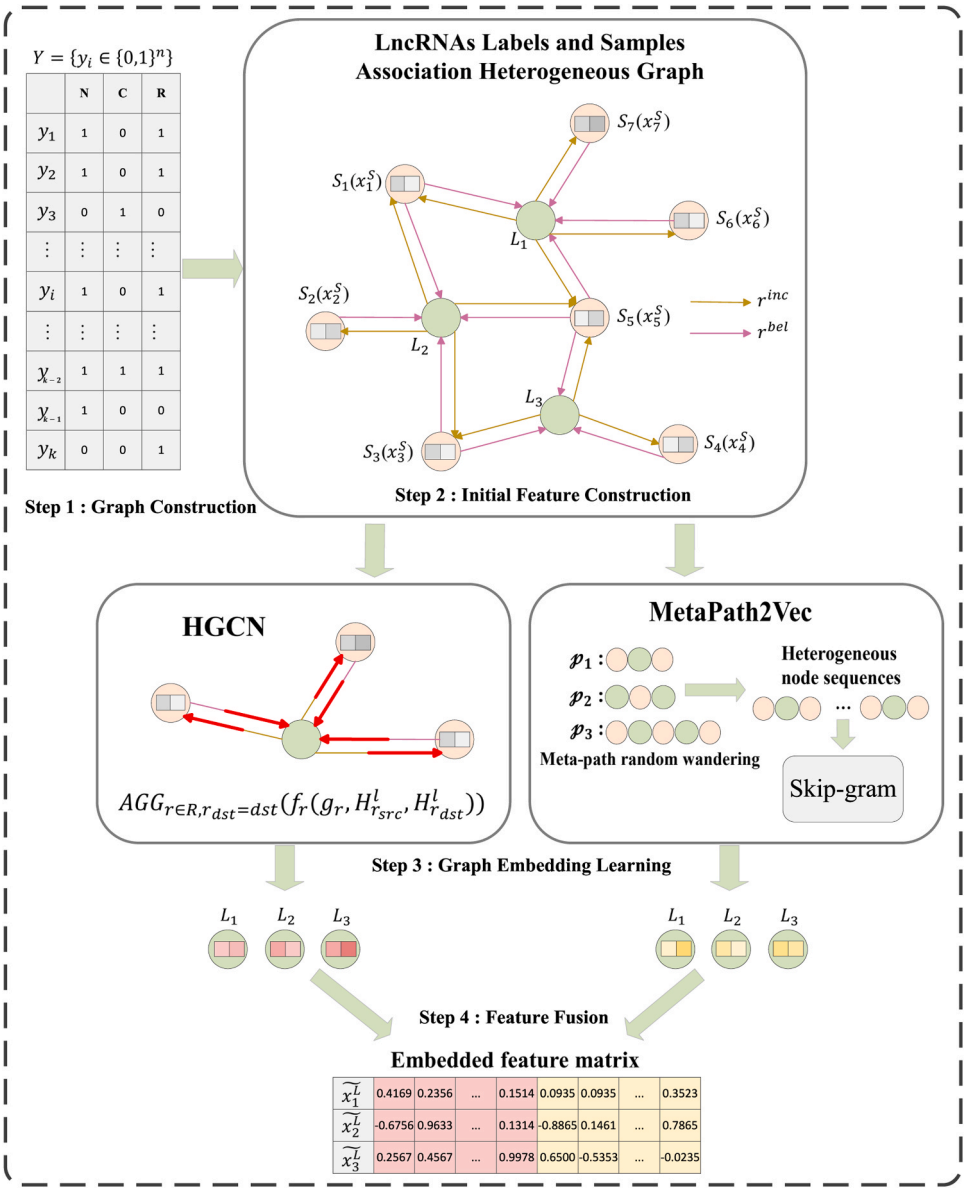

Fig. 4. Workflow diagram of the graphical prototype module. Firstly, the labels and samples association heterogeneous graph is constructed from the label matrix, then the sample nodes are initialized with deep sequence features and the label nodes are initialized with the standard normal distribution. Following this, the label node embeddings are learned on the heterogeneous graph by using HGCN and MetaPath2Vec, respectively. Finally, the two types of embeddings are fused to obtain the final label embeddings as labeling prototypes.

r_i^{bel}. Similarly, a directed edge is established from v_j^L to v_i^S to represent the relationship indicating that the label includes the sample, denoted as r_i^{inc}. If a position in the label table is 0, it implies the absence of an association between the corresponding row and column samples and localization labels. In this scenario, no edge relationship is established between the corresponding label and sample nodes. The final result is the heterogeneous graph G representing the association between LncRNA labels and samples, denoted as $G = (V, R, X)$, where $V \in \{V^S, V^L\}$. V^S represents LncRNA sample nodes (depicted as orange circles in Fig. 4), $V^S = \{v_1^S, v_2^S, \ldots v_m^S\}$, where m is the total number of LncRNA samples. V^L represents localization label nodes (depicted as green circles in Fig. 4), $V^L = \{v_1^L, v_2^L, \ldots v_n^L\}$, where n is the total number of LncRNA localization label nodes. R represents relationships between the two types of nodes, $R \in \{r^{bel}, r^{inc}\}$, $r^{bel} = \{r_1^{bel}, r_2^{bel}, \ldots r_m^{bel}\}$, and $r^{inc} = \{r_1^{inc}, r_2^{inc}, \ldots r_m^{inc}\}$. In Fig. 4, r^{bel} is indicated by brown arrows, and r^{inc} is represented by purple arrows.

Step 2 – Initial Feature Construction: $X \in \{X^S, X^L\}$, where X^S represents the features corresponding to all LncRNA sample nodes, $X^S = \{x_1^S, x_2^S, \ldots x_m^S\}$. X^L represents the features corresponding to all label nodes, $X^L = \{x_1^L, x_2^L, \ldots x_n^L\}$. Initially, low-level features are extracted from the original sequences of each LncRNA sample to form X^S. As for the label features X^L, they are initialized using a standard normal distribution.

Step 3 – Graph Embedding Learning: The embedding learning for label nodes is conducted on the heterogeneous graph G using both HGCN and MetaPath2Vec methods. The HGCN part performs graph convolution operations[30] on different relations $R \in \{r^{bel}, r^{inc}\}$ of the heterogeneous graph separately, if multiple relations share the same target node type, their results are aggregated using the specified method, and if the relational graph has no edges, the module is not invoked, the formal definition is shown in Eq. 7. Eventually, after multiple rounds of propagation the sample feature information and graph structure information are aggregated on the labeled nodes to get

the label embedding specific to HGCN.

$$H_{r_{dst}}^{(l+1)} = AGG_{r\in R, r_{dst}=dst}(f_r(g_r, H_{r_{src}}^{l}, H_{r_{dst}}^{l})) \tag{7}$$

In this context, l is the convolution layer of heterogeneous graph, $H_r^l src$ represents the embedding of source node at l-th layer. g_r denotes the subgraph on the heterogeneous graph G that contains the relationship r. The convolution result on relationship r is obtained through the processing module $f_r(\bullet)$. Subsequently, the convolution results across multiple relationships are aggregated using the aggregation function $AGG(\bullet)$, producing the embedding of target node at $l+$ 1-th layer, labeled as $H_{r_{dst}}^{l+1}$.

In MetaPath2Vec, the model utilizes meta paths to guide the process of random walks, generating heterogeneous node sequences. These node sequences are then inputted into Skip-Gram [48], enabling vectorization of the sequences and producing embeddings for the label nodes. In our task, the meta path "SLS" indicates that a certain LncRNA sample (S) belongs to a specific localization label (L), which in turn includes another (or the same) sample (S). Furthermore, meta paths are often used symmetrically, which helps in recursive guidance during random walks [31]. In this study, on the heterogeneous graph that associates ncRNA samples with labels, "SLS", "SLSLS", "LSL" and "LSLSL" and various other meta paths to guide the random tour of graph embedding learning. The experimental results show that when the meta path "SLSLS" is used, the performance is slightly better, and there is no significant difference in the final model performance of the other meta paths.

The labeled embedding $\widetilde{X}^{LH}$, which pools the features of the samples, was obtained from HGCN, and the labeled embedding $\widetilde{X}^{LM}$, which is enriched with information about the graph structure, was obtained from MetaPath2Vec, and is formally described as follows:

$$\widetilde{X}^{LH} = \mathrm{HGCN}(\mathrm{G}, \quad X^S) \tag{8}$$

$$\widetilde{X}^{LM} = MatePath2Vec(G, X^S) \tag{9}$$

Step 4 – Feature Fusion: To comprehensively leverage both sample feature information and graph structural details, this study horizontally concatenates the aforementioned two types of label embeddings to serve as the embedding for each label node. The formal description is as follows:

$$\widetilde{X}_{head}^{L} = cat\left[\widetilde{X}_{H}^{L} : \widetilde{X}_{M}^{L}\right] \tag{10}$$

With the above four steps, we obtain labeled embeddings that contain rich information. In this study, $\widetilde{X}_{head}^{L}$ is used as a prototype of the head class for subsequent training of the GP-HTNLoc model.

2.3.4. Transfer learner training

Up to this point, our study has trained the parameters M_{head} for the head classifier using sequence deep features on the relatively abundant samples in D_{head}. Additionally, the graph prototype module has been utilized to construct prototypes for the head class. To learn the mapping relationship from class prototypes to classifier parameters, we have established a transfer learner on the head class. The transfer learner consists of a single-layer perceptron without bias terms. The learnable weight matrix $W_{transfer} \in \mathbb{R}^{d\times d}$ serves as the trainable parameters for the transfer learner. The training of the transfer learner parameters $W_{transfer}$ involves minimizing the following loss function $\mathscr{L}_t$:

$$\mathscr{L}_t = \sum_{j=1}^{l_{head}} ||m_{head}^{j} - W_{transfer}\widetilde{X}_{j}^{L}||^2 \tag{11}$$

Where $m_{head}^{j} \in M_{head}$ represents the parameters of the head class classifier, and $\widetilde{X}_{j}^{L} \in \widetilde{X}_{head}^{L}$ denotes the prototype for the head class.

2.3.5. Complete classifier construction

On D_{tail}, the graph prototype module is similarly employed to obtain prototype representations $\widetilde{X}_{tail}^{L}$ corresponding to the tail labels. Subsequently, the mapping from the tail class prototypes to the parameters of the tail classifier is accomplished using the previously trained transfer learner,

$$\widehat{m}_{tail}^{z} = W_{transfer}\widetilde{X}_{z}^{L} \tag{12}$$

The prototype representation $\widetilde{X}_{z}^{L} \in \widetilde{X}_{tail}^{L}$corresponds to the tail class, where $\widetilde{X}_{tail}^{L}$ represents the prototypes for the tail class. Additionally, $\widehat{m}_{tail}^{z} \in \widehat{M}_{tail}$denotes the parameters of the tail class classifier. The construction of the final classifier parameters is outlined as follows:

$$M = cat[M_{head} : \widehat{M}_{tail}] \tag{13}$$

So far, we have resolved the imbalance in the head and tail categories of the ncRNA dataset and initially constructed a subcellular multilabel localization predictor.

2.3.6. Fine-tuning and prediction

The complete classifier M, obtained by directly concatenating M_{head} and $\widehat{M}_{tail}$, is likely to exhibit excellent performance solely on samples involving either head or tail class labels. However, due to the lack of optimization on data containing both head and tail class labels simultaneously, it may not achieve the anticipated performance on samples simultaneously involving both label types. In response, this study designed a fine-tuning module, utilizing data containing both head class labels and tail class labels simultaneously, called Novel Data, to further optimize the parameters of the complete classifier M.

During the prediction phase, for a given novel ncRNA sequence, after extracting primary features through multiple sequence feature extraction methods, these primary features are fed into a BiLSTM network equipped with an attention mechanism to obtain advanced features $x^{'S}$. The corresponding label set $\widehat{y}$ for this ncRNA sequence can then be obtained through the following process:

$$\widehat{y} = sigmoid\left(x^{'S} M\right) \tag{14}$$

Where M represents the parameters of the complete multi-label classifier after fine-tuning. For $\widehat{y} \in \{0,1\}^n$, n represents the number of subcellular locations to be predicted, and $\widehat{y}$ provides the final subcellular localization prediction results of the ncRNA in a binary format.

2.4. Evaluation metrics

Distinguished from typical binary and multiclass problems, multi-label classification entails unique evaluation metrics [49]. In this study, six commonly employed metrics in multi-label classification problems [18–20] are utilized to assess the performance of GP-HTNLoc. The six evaluative metrics include Average Precision (AP), Hamming Loss(L_h), One-Error(E_{noe}), Coverage (Cov), Accuracy (ACC), and Ranking Loss(L_r). For the aforementioned evaluation metrics, higher values in Average Precision and Accuracy signify superior model performance, while smaller values in Hamming Loss, One-Error, Coverage, and Ranking Loss indicate better model performance.

2.5. Implementation details

GP-HTNLoc is implemented based on PyTorch[50], with the HeteroGraphConv (HGCN) and MetaPath2Vec modules in the graph prototype module implemented using Deep Graph Library (DGL)[51]. Both the head class classifier and fine-tuning utilize BCELoss for training, while the transformation learner is trained using MSELoss. We employed the Adam optimizer with a learning rate of 0.00001, along with default

beta1 and beta2 values for model training.

3. Results

3.1. Comparison with different features

To determine the optimal feature combination, this study conducted experiments on the subsets of lncRNA, miRNA, and snoRNA from the benchmark dataset, using the eight primary feature extraction methods. These features were extracted from the raw RNA sequences using Python code, the identifier for each feature is shown in Table 1. During the feature combination experiments, the other model parameters are set to their initial values: For subsets with five labels (lncRNA subset and snoRNA subset), the head classes were set to two classes, and the tail classes were set to three classes. For the subset with six labels (miRNA subset), the head classes were set to three classes, and the tail classes were set to three classes. The dimension of the graph prototype was set to 64 dimensions obtained through HGCN and 64 dimensions obtained through MetaPath2Vec, totaling 128 dimensions. Fine-tuning 0 epochs, i.e., not utilizing the fine-tuning module, and HGCN was trained for two epochs.

With the other parameters fixed, we initially extracted vector representations of the eight features from the original ncRNA sequences and inputted their different combinations into the model to obtain ncRNA localization prediction results. For this process, our study employed 10-fold cross-validation to evaluate the performance of different feature combinations. Specifically, 10-fold cross-validation involves partitioning the dataset into 10 roughly equal-sized subsets, where each subset is used as the validation set once while the remaining nine subsets are used for training. This process is repeated 10 times, with each subset used exactly once as the validation set. The final result is the average of the performance metrics obtained from the 10 repetitions. This technique reduces variance by using multiple train-test splits and provides a more reliable estimate of model performance.

During the experimental process, due to the considerable number of feature combinations, we initially evaluated the performance of each feature independently and prioritized considering the top-ranking features for subsequent combinations. Table 2 lists the superior features and their combinations based on the average results obtained from 10 times 10-fold cross-validation. From the table, it can be observed that on the lncRNA dataset, the feature combination ABCDTF (2-mer, 3-mer, Coverage-Rate, DNC, TNC, Fickett Score) demonstrates the highest performance, achieving an *ACC* of 0.468 and an *AP* of 0.715. On the snoRNA dataset, feature T (TNC) attains the best performance with an *ACC* of 0.453 and an *AP* of 0.721. In addition, feature N (NAC) on the miRNA dataset achieves the highest performance with an *ACC* of 0.463 and an *AP* 0.703.

3.2. Comparison with different parameters

3.2.1. Different fine-tuning epochs and HGCN training epochs

In Section 2.3.6, this paper elucidates the necessity of the fine-tuning stage. Considering that different degrees of fine-tuning may have varying effects on model performance, this section conducts code experiments on the fine-tuning module under the premise of fixing other model parameters to their initial values and utilizing the optimal feature combinations of various ncRNAs. We set different numbers of fine-tuning epochs for the lncRNA, snoRNA, and miRNA subsets of the benchmark dataset to run the model. Fig. 5A presents the average AP values of 10 times 10-fold cross-validations for each subset at different fine-tuning epochs. It can be observed that fine-tuning for one epoch indeed leads to performance improvement, but as the number of fine-tuning epochs increases, the model performance tends to decrease. Therefore, the optimal degree of fine-tuning is maintained at one epoch.

Similarly, under the condition of fixing other model parameters to their initial values and using the optimal feature combinations, we conducted experiments to investigate the impact of different numbers of training epochs for HGCN on model performance. Fig. 5B shows the average AP values from 10 times 10-fold cross-validations at different numbers of HGCN training epochs on the three subsets of the benchmark dataset. From the figure, it can be seen that on the lncRNA subset, the maximum AP is achieved when HGCN is trained for four epochs, reaching 0.726. On the miRNA and snoRNA subsets, the maximum AP is achieved when HGCN is trained for three epochs, with values of 0.719 and 0.731, respectively. When HGCN is trained for more than five epochs, the model performance gradually declines.

3.2.2. Different head and tail class divisions

In GP-HTNLoc, a training strategy that separates head and tail categories is employed to address the issue of significant imbalances in the number of samples for ncRNA subcellular localization. However, the degree of class imbalance varies from one dataset to another, and it is not possible to give uniform criteria for class division. To address this challenge, this study explores the performance of GP-HTNLoc under various head-tail class division scenarios for benchmark dataset. The specific scenarios are outlined in Table 3. In Table 3, the "divisions" column represents the strategies for dividing head and tail categories. For instance, "2 + 3" indicates that, after arranging categories (subcellular localization labels) in descending order based on sample quantities, the top two categories with the highest sample counts are designated as head categories, while the bottom three categories with fewer samples are considered tail categories. Other division methods follow a similar principle. It is worth noting that the head classifier can only be trained when the number of head classes is greater than 1. Training the classifier with only one class is not feasible for classification. Therefore, divisions like "1 + 4" and "1 + 5" will not occur in the head-tail class division.

The model parameters are set as follows: in the graph prototype part, 64 dimensions are obtained through HGCN and MetaPath2Vec each, totaling 128 dimensions; the number of epochs for HGCN training and

Table 2
Performance of different feature combinations on ncRNA datasets.

Dataset	Metrics	AB	N	D	T	Z	ABC	CDT	ABDTF	ABCDTF
lncRNA	ACC	0.401	0.366	0.387	0.420	0.445	0.413	0.430	0.453	**0.468**
	AP	0.701	0.651	0.669	0.687	0.710	0.696	0.702	0.707	**0.715**
	E_{one}	0.448	0.458	0.458	0.461	0.435	0.445	0.439	0.433	**0.427**
	L_r	0.227	0.243	0.238	0.227	0.211	0.217	0.212	0.209	**0.207**
snoRNA	ACC	0.430	0.433	0.441	**0.453**	0.445	0.437	0.442	0.446	0.432
	AP	0.711	0.702	0.703	**0.721**	0.708	0.708	0.705	0.711	0.698
	E_{one}	0.286	**0.279**	0.283	**0.279**	0.280	0.282	0.285	0.285	0.295
	L_r	0.266	0.251	0.259	**0.246**	0.248	0.251	0.250	0.252	0.262
miRNA	ACC	0.459	**0.463**	0.459	0.446	0.451	0.436	0.461	0.439	0.409
	AP	0.679	**0.703**	0.700	0.695	0.696	0.668	0.698	0.674	0.646
	E_{one}	0.351	**0.347**	0.353	0.368	0.365	0.369	0.357	0.408	0.425
	L_r	0.245	**0.239**	0.242	0.247	0.258	0.259	0.247	0.267	0.283

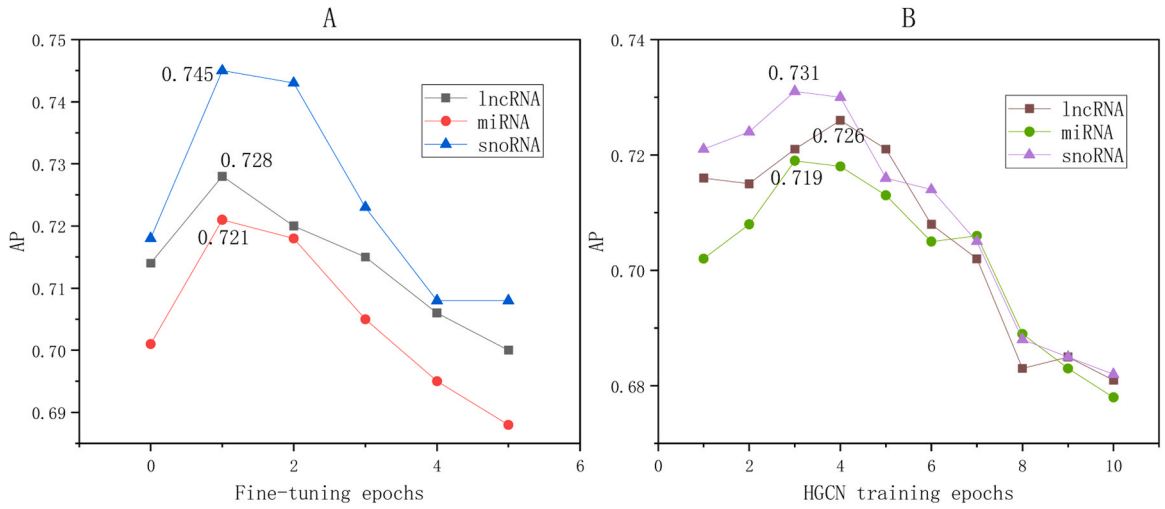

Fig. 5. The average AP of 10 times 10-fold cross-validation under different parameters of GP-HTNLoc is shown. (A) Different fine-tuning epoch numbers, (B) Different HGCN training epoch numbers.

Table 3
Performance of different head-tail class division strategies.

Dataset	divisions	*ACC*	*AP*	E_{one}	L_r	L_h	*Cov*
lncRNA	2 + 3	0.480	0.736	0.422	0.225	0.093	0.959
	3 + 2	**0.508**	**0.760**	**0.388**	**0.189**	**0.072**	**0.899**
	4 + 1	0.472	0.732	0.425	0.230	0.100	0.979
snoRNA	2 + 3	0.467	0.755	0.235	0.206	0.099	1.585
	3 + 2	**0.493**	**0.781**	**0.227**	**0.198**	**0.089**	**1.535**
	4 + 1	0.472	0.757	0.240	0.213	0.102	1.578
miRNA	3 + 3	0.479	0.733	0.335	0.183	0.096	1.416
	4 + 2	**0.517**	**0.759**	**0.298**	**0.169**	**0.077**	**1.328**
	2 + 4	0.399	0.653	0.450	0.288	0.128	1.589
	5 + 1	0.467	0.728	0.342	0.187	0.098	1.412
H_lncRNA	2 + 3	0.481	0.732	0.389	0.221	0.098	1.107
	3 + 2	**0.502**	**0.756**	**0.367**	**0.208**	**0.082**	**1.065**
	4 + 1	0.467	0.721	0.427	0.236	0.105	1.119
H_snoRNA	2 + 3	0.488	0.767	0.209	0.197	0.091	1.539
	3 + 2	**0.510**	**0.798**	**0.199**	**0.189**	**0.083**	**1.499**
	4 + 1	0.483	0.767	0.214	0.201	0.091	1.537
H_miRNA	3 + 3	0.469	0.762	0.323	0.203	0.098	1.421
	4 + 2	**0.493**	**0.780**	**0.303**	**0.185**	**0.087**	**1.379**
	2 + 4	0.402	0.662	0.443	0.276	0.128	1.587
	5 + 1	0.473	0.758	0.318	0.203	0.099	1.426

fine-tuning is set to the optimal value, using the best feature combination on each subset. Based on this, we obtain the average values of each metric through 10 times 10-fold cross-validation, and the specific results are presented in Table 3. From the table, it can be seen that on the lncRNA and H_lncRNA subsets, partitioning with three head classes and two tail classes ("3 +2") achieves the best performance. On the miRNA and H_miRNA subsets, partitioning with four head classes and two tail classes ("4 +2") exhibits the best performance across all metrics. As for the snoRNA and H_snoRNA subsets, partitioning with three head classes and two tail classes ("3 +2") leads to the best performance for GP-HTNLoc.

3.2.3. *Different graph prototype dimensions*

In the graph prototype module of GP-HTNLoc, the graph prototypes are obtained by combining graph embeddings from HGCN and MetaPath2Vec. The dimensionality of the embeddings obtained by each method determines the total dimensionality of the graph prototype. To investigate the impact of different prototype dimensions on the performance of GP-HTNLoc, under the premise of using the optimal feature combination, optimal fine-tuning, HGCN training epochs, and optimal head-tail class division, this study conducted 10 times 10-fold cross-validation experiments on the ncRNA, snoRNA, and miRNA subsets of the benchmark dataset. Fig. 6 presents the experimental results, where the alphabetical labels (A-L) on the X-axis identify different prototype combination methods. "HGCN_dim" represents the embedding dimension obtained through HGCN, while "Meta_dim" represents the embedding dimension obtained through MetaPath2Vec. The height of the stacked bar chart represents the final prototype total dimensionality obtained by combining the embeddings from both methods. The line plots with different colors depict the model performance of GP-HTNLoc on different graph prototype dimensions and data subsets (AP and ACC values are the average results of 5 times 10-fold cross-validation). Specific experimental data can be found in Supplementary Table S4.

From Fig. 6, it can be observed that initially, the model's performance on each subset improves with an increase in the dimensionality of graph embeddings. When the MetaPath2Vec embedding dimension is 156, the HGCN embedding dimension is 100, and the total prototype dimension is 256, the model consistently achieves the best performance across the three subsets. Subsequently, as the dimensions of the two embedding components continue to increase, the model's performance gradually declines. In summary, when the embedding dimension of HGCN is 100 and the embedding dimension of MetaPath2Vec is 156, the

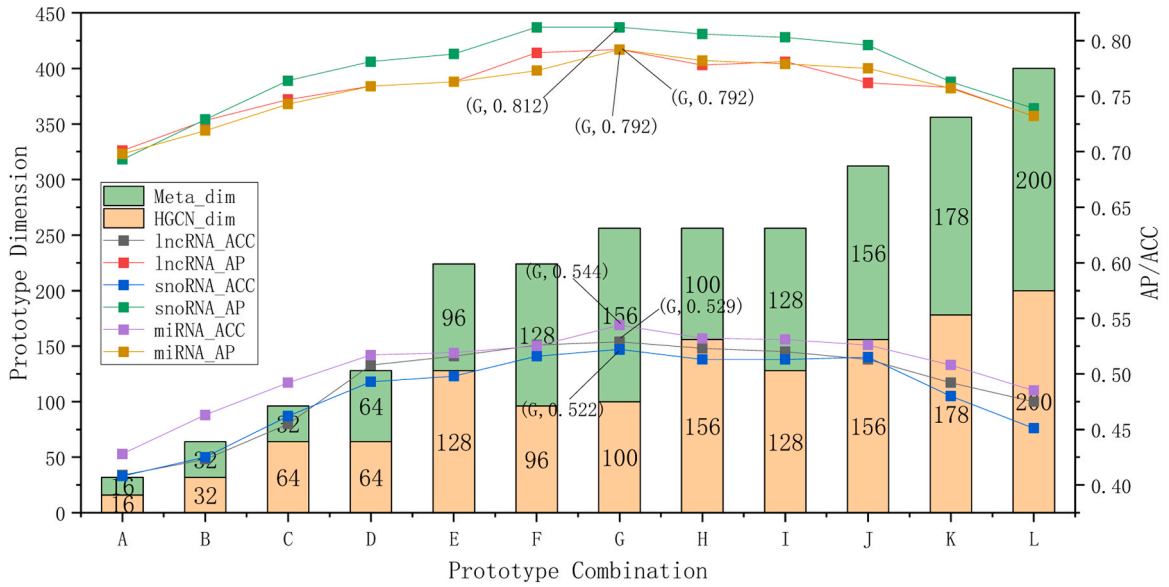

Fig. 6. Performance of GP-HTNLoc on three subsets of the benchmark dataset under different graph prototype dimensions.

total dimensionality of the graph prototype is 256, GP-HTNLoc exhibits optimal performance.

3.3. Performance analysis

We have now obtained the optimal parameters for GP-HTNLoc in all aspects. To demonstrate the final performance of the model, this study evaluated the model performance through 10 times of 10-fold cross-validation on the benchmark dataset's six subsets. Table S5 in the supplementary materials provides the average, confidence intervals, and standard deviations (at a significance level of 0.05) obtained from the 10 times 10-fold cross-validation. From the table, it can be observed that GP-HTNLoc achieved small variances and biases across all six subsets, demonstrating its stable performance.

To further validate the performance of GP-HTNLoc, this study conducted a comparative analysis with state-of-the-art models in the subcellular multi-label RNA localization field, based on the final results of the aforementioned 10-fold cross-validation (repeated 10 times). The comparative models include MKSVM proposed by Wang et al. [18] and MKGHkNN proposed by Zhou et al. [19], and the metrics of these two methods use the reported values of the optimal parameter case in their papers. The results, as shown in Table 4, demonstrate that GP-HTNLoc achieves an *ACC* of 0.528 on the lncRNA dataset, surpassing the current state-of-the-art model by 9.4%. Moreover, its average *AP* reaches 0.792, indicating a 3.3% improvement over the leading model. For the H_lncRNA dataset, GP-HTNLoc reached 0.527 on ACC, outperforming the best existing model by 9.3%, and 0.785 on AP, outperforming the existing model by 2.6%. Furthermore, GP-HTNLoc exhibits superior performance across all metrics on the H_snoRNA and snoRNA dataset in comparison to the current state-of-the-art model. On the miRNA dataset GP-HTNLoc outperforms the state-of-the-art model on four metrics except *ACC* and *Cov*. On the H_miRNA dataset our model outperforms existing models on *AP*, E_{one}, *Cov*, L_r, and the differences between our model and the state-of-the-art model are relatively small on *ACC* and L_h.

We conducted a statistical significance t-test on the average precision and accuracy of GP-HTNLoc and state-of-the-art methods through 10 times 10-fold cross-validation, as shown in Tables 5 and 6. Our approach achieves significantly higher average precision than the SOTA methods across all subsets of the benchmark dataset ($P < 0.05$). Although the accuracy of GP-HTNLoc is slightly lower than the SOTA methods in the miRNA and H_miRNA subsets (with negative t-statistics), it remains significantly higher than the SOTA methods in the remaining four subsets ($P < 0.05$). In addition, this study provides the macro-average F1 score, micro-average F1 score, and average F1 score for each subcellular

Table 4
Comparison of GP-HTNLoc performance with state-of-the-art models.

Datasets	Models	*ACC*	*AP*	E_{one}	*Cov*	L_r	L_h
LncRNA	GP-HTNLoc	**0.528**	**0.792**	**0.367**	**0.882**	**0.167**	**0.062**
	MKSVM	0.434	0.757	0.402	0.934	0.183	0.066
	MKGHkNN	0.434	0.759	0.401	0.932	0.183	0.069
snoRNA	GP-HTNLoc	**0.522**	**0.812**	**0.219**	**1.526**	**0.188**	**0.076**
	MKSVM	0.515	0.800	0.251	1.594	0.205	0.082
	MKGHkNN	0.519	0.808	0.236	1.569	0.200	0.082
miRNA	GP-HTNLoc	0.545	**0.792**	**0.291**	1.311	**0.163**	**0.068**
	MKSVM	**0.582**	0.787	0.310	1.311	0.175	0.073
	MKGHkNN	0.514	0.789	0.311	**1.308**	0.174	0.074
H_lncRNA	GP-HTNLoc	**0.527**	**0.785**	**0.342**	0.998	**0.183**	**0.067**
	MKSVM	0.418	0.754	0.367	1.180	0.216	0.069
	MKGHkNN	0.434	0.759	0.401	**0.932**	**0.183**	0.069
H_snoRNA	GP-HTNLoc	**0.532**	**0.821**	**0.193**	**1.486**	**0.180**	**0.075**
	MKSVM	0.515	0.800	0.251	1.594	0.205	0.082
	MKGHkNN	0.519	0.808	0.236	1.569	0.200	0.082
H_miRNA	GP-HTNLoc	0.513	**0.795**	**0.281**	**1.291**	**0.160**	0.079
	MKSVM	**0.514**	0.791	0.286	1.462	0.169	0.081
	MKGHkNN	**0.514**	0.789	0.311	1.308	0.174	**0.074**

Table 5
P-Values of GP-HTNLoc and state-of-the-art models Under Average Precision on benchmark dataset.

Models	lncRNA	snoRNA	miRNA	H_lncRNA	H_snoRNA	H_miRNA
MKSVM	1.46E-10	4.35E-05	3.65E-05	3.75E-11	3.17E-06	0.033820
MKGHKNN	2.47E-10	0.036750	0.001487	1.81E-10	0.000144	0.004567

Table 6
P-Values of GP-HTNLoc and state-of-the-art models Under Accuracy on benchmark dataset.

Models	lncRNA	miRNA	snoRNA	H_lncRNA	H_miRNA	H_snoRNA
MKSVM	3.31E-12	1.35E-08*	3.74E-05	1.20E-14	0.718775*	9.06E-06
MKGHKNN	3.31E-12	6.37E-08	0.010764	5.00E-14	0.718775*	7.62E-05

Note: When marked with an asterisk (*), it indicates that the t-statistic corresponding to the p-value is negative.

location based on 10 times of 5-fold cross-validation on the benchmark dataset for GP-HTNLoc and MK-GHKNN in Supplementary Table S10. It is observed that GP-HTNLoc outperforms MK-GHKNN on all subsets except for the H_miRNA subset. Overall, our proposed GP-HTNLoc demonstrates improvement over existing methods in general.

To further validate the performance of GP-HTNLoc, this study conducted testing on an independent dataset constructed by Bai et al. Specifically, considering that each subset in the independent dataset contains only four localization labels for prediction, while each subset in the benchmark dataset contains five or six labels, we modified the benchmark dataset by excluding labels not present in the independent dataset, resulting in a new benchmark dataset. After conducting experiments on the lncRNA, snoRNA, and miRNA subsets of the new benchmark dataset (experimental data can be found in Supplementary Table S6), we specified the model's head-tail class partitioning as "2 + 2" format. and the other parameters were set to their previously optimized values obtained during the initial experiments. Subsequently, we retrained GP-HTNLoc using the new benchmark dataset. Upon completion of training, we tested the model on the independent dataset. Table 7 presents the average performance metrics of GP-HTNLoc after 10 rounds of testing and compares them with those of the state-of-the-art models. Supplementary Tables S7 and S8 respectively present the p-values of the t-tests for statistical significance of AP and ACC. It can be observed that all evaluation metrics of GP-HTNLoc on the three subsets of the independent dataset outperform those of existing state-of-the-art models, which once again demonstrates the improvement of our model over existing ones.

3.4. Case Study

In order to further substantiate the reliability of GP-HTNLoc in practical scenarios of ncRNA multi-label localization prediction, we obtained CARL lncRNA (Cardiac Apoptosis-Related Long Non-coding RNA, NCBI:66774|Ensembl:ENSMUSG00000097638) from RNALocate v2[10]. The RNA Symbol for CARL is Carlr. It should be noted that this particular lncRNA did not appear in the benchmark dataset used to train the model in this study. Research by Castellanos-Rubio et al. indicates that under basal conditions in mouse macrophages, Carlr predominantly exhibits nuclear localization, while after LPS stimulation, a majority of Carlr transcripts localize within the cytoplasm [52], suggesting the true subcellular localization of CARL lncRNA encompasses both the nucleus and cytoplasm. We conducted a case study using this lncRNA as an example to assess GP-HTNLoc. Employing the five-label classifier trained on the LncRNA subset of benchmark dataset in Section 2.1.1, GP-HTNLoc predicted the multi-location subcellular localization of CARL lncRNA. The prediction process revealed sigmoid probability values for five subcellular locations—Nucleus, Cytoplasm, Ribosome, Cytosol, and Exosome—as follows: 0.553, 0.562, 0.173, 0.197, and 0.194, respectively. Using a threshold of 0.5 for class assignment, the final prediction resulted in Nucleus and Cytoplasm, aligning perfectly with the true subcellular locations of this lncRNA. Moreover, the considerable discrepancy in probability values between positive and negative classes in the sigmoid further underscores the discriminative capability of GP-HTNLoc.

Table 7
The performance of GP-HTNLoc compared to the state-of-the-art models on the independent dataset.

subset	Models	*ACC*	*AP*	*Cov*	L_r	L_h	E_{one}
human lncRNA	GP-HTNLoc	**0.386**	**0.685**	**1.062**	**0.286**	**0.213**	**0.497**
	MKSVM	0.084	0.370	2.931	0.882	0.445	0.794
	MK-GHkNN	0.062	0.370	2.937	0.887	0.427	0.808
human snoRNA	GP-HTNLoc	**0.357**	**0.632**	**1.197**	**0.323**	**0.343**	**0.289**
	MKSVM	0.098	0.490	1.281	0.400	0.447	0.918
	MK-GHkNN	0.129	0.480	1.500	0.468	0.458	0.878
human miRNA	GP-HTNLoc	**0.303**	**0.704**	**1.587**	**0.339**	**0.381**	**0.482**
	MKSVM	0.214	0.644	1.624	0.375	0.476	0.652
	MK-GHkNN	0.252	0.648	1.715	0.398	0.443	0.582

3.5. Ablation Study

In order to confirm the role of each component of GP-HTNLoc, this study conducted an ablation study of GP-HTNLoc based on 10 10-fold cross-validation on the benchmark dataset. We remove the graph prototype module from GP-HTNLoc, obtain the prototype by calculating the sample average, and denote the model as AP-HTNLoc. Further, we remove the graph prototype module and the module with separate head-tail training from GP-HTNLoc, and replace it by inputting the set of the best features into BiLSTM and then put it directly into the head classifier for classification training (the head classifier output dimensions are increased accordingly), and the model is denoted as BiL-C. Fig. 7 presents the box plots of accuracy (ACC) and average precision (AP) for the aforementioned experiments on the lncRNA, snoRNA, and miRNA subsets. These box plots depict inter-group differences using P-values. It can be observed that training with separate head and tail patterns moderately improved the model's performance. The introduction of the graph prototype module further enhanced the model's performance, and these enhancements are generally significant (P-value < 0.005). In the above ablation experiments, the average values of each metric after conducting 10-fold cross-validation repeated 10 times on all subsets of the benchmark dataset can be found in Supplementary Table S9.

3.6. Model interpretation

Shapley additive explanations (SHAP) is a method used to explain the predictions of machine learning models. It is based on the concept of

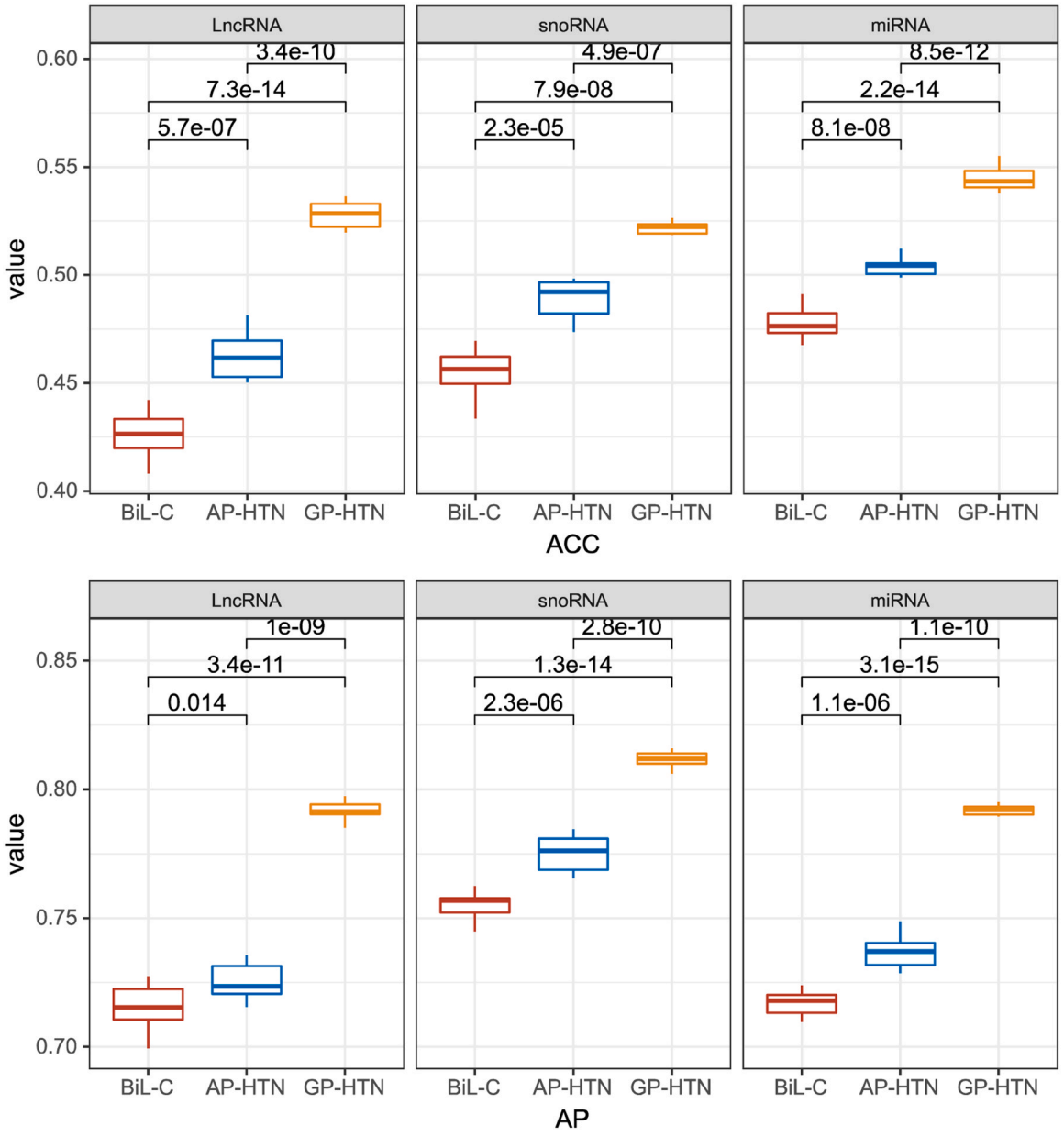

Fig. 7. Box plots of the average precision and accuracy of the ablation experiments for each component of GP-HTNLoc obtained based on 10 ten-fold cross-validation on three subsets of the benchmark dataset (lncRNAs, snoRNAs, miRNAs). Differences between groups are shown as p-values.GP-HTN denotes the final model GP-HTNLoc in this paper,AP-HTN denotes the AP-HTNLoc model, and BiL-C denotes the model composed of BiLSTM and classifiers.

Shapley values from game theory, which determine the influence of each feature on the model's output [32]. SHAP calculates the contribution of each feature value to the model prediction by applying permutations of feature values to different prediction scenarios. This method allows us to understand how each feature affects the model's decision-making process, thereby explaining the model's prediction results. Currently, SHAP has been widely applied in bioinformatics with promising results [53–55].

In this study, we utilize Shapley values to explain the feature importance during the prediction process of GP-HTNLoc on the snoRNA subset of the benchmark dataset. Fig. 8 and Supplementary Fig. S1 depict the top 15 important TNC feature segments predicted by GP-HTNLoc for snoRNA subcellular localization. In Figures 8AB and Supplementary Fig. S1A-C, each point represents an instance, with values ranging from small to large and colors from blue to red. A higher Shapley value for an instance indicates a greater contribution of that feature to supporting the localization of the instance to the corresponding subcellular location, and vice versa. Fig. 8C presents the top 15 features ranked by the sum of absolute Shapley values for the five subcellular locations. From the figure, it can be observed that different features have varying impacts on the prediction of different subcellular locations. However, certain features, such as AAG, CCA, and AGC, consistently exert considerable effects across different subcellular localizations. These segments may represent a portion of the proteins interacting specifically with snoRNA for subcellular localization or may be associated with the protein functions involved in forming snoRNP (snoRNA ribonucleoprotein complex[56]). Fig. 8D shows three motifs obtained from the snoRNA subset of the benchmark dataset using the MEME suite (https://meme-suite.org/meme/index.html). These motifs maybe represent structural domains, binding sites, or other functional elements (or be associated with them). It can be observed that there are many overlapping segments between these motifs and the top-ranking features based on the Shapley values, such as ATG, CTG, CAG, etc. This demonstrates that GP-HTNLoc has identified potential important structures

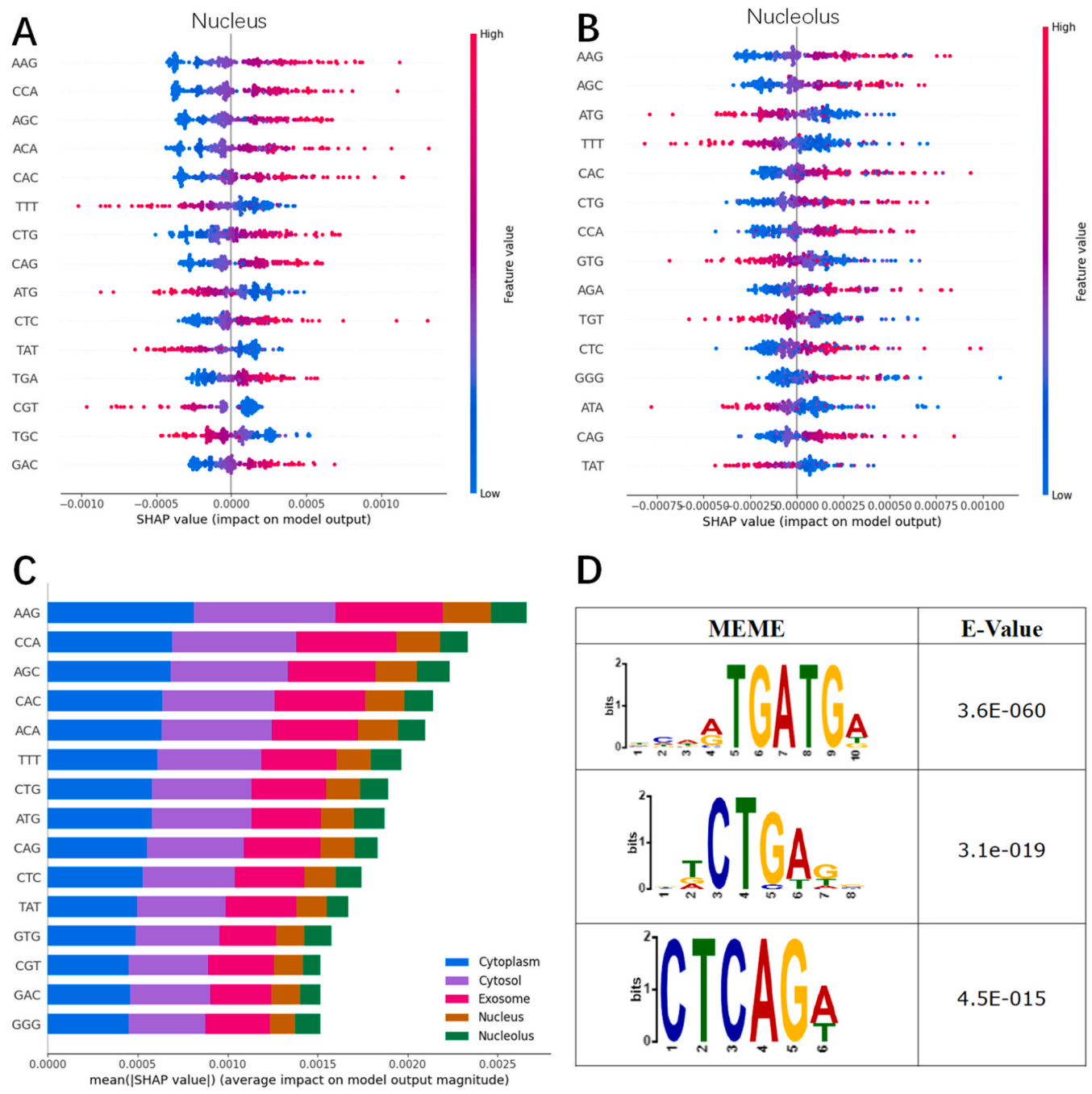

Fig. 8. Model interpretation based on Shapley values and motif analysis on the snoRNA subset of the benchmark dataset. (A) Top 15 features for the nucleus, (B) top 15 features for the nucleolus, (C) top 15 features among all five subcellular localizations. (D) Three motifs obtained from the MEME suite.

within snoRNA sequences and used them for subcellular localization prediction.

Samarsky et al. found that the box C motif (UGAUGA) and box D motif (GUCUGA) are necessary and sufficient for nucleolar targeting in yeast and mammals [57]. From Fig. 8B, we can observe that the segments AGA and CAG make significant contributions to the prediction of Nucleolus localization by the model. Additionally, the AGA segment appears only in Fig. 8B, and these two segments show a high degree of overlap with the complementary segments (CAGACT) of the box D motif. This further demonstrates the ability of GP-HTNLoc to identify key structures.

We utilized SHAP on the lncRNA subset of the benchmark dataset to obtain the top 15 features that have the greatest impact on GP-HTNLoc predictions, as shown in Supplementary Material Fig. S2. It can be observed that the Fickett score feature is crucial for the subcellular localization prediction of lncRNA across five subcellular locations, suggesting that the positional frequency information of the four basic nucleotides and their proportions in the sequence have a considerable impact on lncRNA subcellular localization. In addition, Lubelsky et al. found that the repetitive pattern RCCTCCC (where R represents A/G) drives lncRNA localization to the nucleus [58], which corresponds to the 3-mer fragments TCC and GCC in Fig. S2.

4. Discussion

This study introduces a novel computational model, GP-HTNLoc, for

predicting the subcellular localization of ncRNAs. While there have been over a dozen studies in this field previously, most of them have not specifically addressed the issues of imbalanced datasets and sparse training samples, which are prevalent in ncRNA subcellular localization prediction. GP-HTNLoc, however, focuses on addressing these challenges. Experimental results based on 10-fold cross-validation on benchmark datasets and 10-round testing on independent datasets demonstrate that GP-HTNLoc generally outperforms existing state-of-the-art models. We found that the separate training of head and tail classes is helpful in mitigating the dataset imbalance issue in ncRNA subcellular localization prediction. As observed from Supplementary Table S10, our model consistently achieves higher F1 scores for tail classes compared to existing models. Additionally, ablation studies confirm the effectiveness of both the head-tail network and the graph prototype module in enhancing GP-HTNLoc. The head-tail network typically contributes to a 2–3% improvement in accuracy (ACC) and average precision (AP), while the graph prototype module typically contributes to a 4–6% improvement in ACC and AP. It is worth noting that GP-HTNLoc performs comparably to existing models on miRNA sequences from benchmark datasets, unlike the substantial advantages observed on lncRNA and snoRNA sequences. This may be due to the shorter length of miRNA sequences (typically only 18–25nt), which may limit the ability of BiLSTM to capture sufficient contextual information.

Although GP-HTNLoc has shown improvements over existing models, it still has limitations. While the benchmark dataset includes human-specific sequences for training and validation, the dataset is biased towards human sequences, with fewer sequences from other species. This bias is also present in the original RNA localization databases, limiting our ability to explore sequences from a broader range of species. Therefore, based on the current experimental results, we cannot determine whether ncRNA subcellular localization exhibits species specificity. In future studies, we hope to obtain more ncRNA sequences from different species to thoroughly investigate this interesting question.

Additionally, recent studies have emphasized the importance of cell line specificity in studying subcellular environments. For example, Lin et al. noted that lncRNAs are often expressed in a tissue-specific manner, and their subcellular localization depends on the tissue or cell line in which they are expressed [59]. Li et al. highlighted the strong influence of cellular environment on essential proteins, which exhibit significant differences between different cell lines [60]. Therefore, in future research, we plan to incorporate cell line specificity into our model to further explore the prediction of ncRNA subcellular multi-label localization.

5. Conclusion

This study introduces a novel ncRNA subcellular multi-label localization model, GP-HTNLoc, which distinguishes itself from existing single-label localization models by simultaneously predicting multiple potential subcellular locations for each ncRNA sequence. In order to solve the common class imbalance problem in RNA datasets, we adopt a training strategy in which the head class, which has more samples, and the tail class, which has fewer samples, are trained separately during the training process. In GP-HTNLoc, we innovatively introduce a graph prototype module, which requires only a small number of samples to obtain high-quality label prototype representations. These prototype representations contain rich graph structural information and label association information, laying a solid foundation for the final localization prediction.

Experimental results on benchmark datasets and independent datasets demonstrate the superiority of GP-HTNLoc over existing models. Additionally, our case studies highlight the robustness and applicability of GP-HTNLoc in real-world scenarios for multi-label subcellular localization prediction of ncRNAs. Results from ablation studies show that both the head-tail network and the graph prototype module play critical roles in improving the performance of GP-HTNLoc. Finally, we provide explanations of the model's predictions from the perspective of feature importance based on SHAP, aiming to offer more relevant insights to biologists.

In summary, GP-HTNLoc makes significant advancements in the field of ncRNA subcellular localization. We believe that GP-HTNLoc will greatly facilitate our biological understanding of ncRNA functions and mechanisms, further elucidating the regulatory patterns of ncRNAs in disease occurrence and progression.

CRediT authorship contribution statement

Shuangkai Han: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft. **Lin Liu:** Conceptualization, Funding acquisition, Methodology, Project administration, Resources, Supervision, Writing – review & editing.

Declaration of Competing Interest

All authors disclosed no relevant relationships.

Data Availability

The experimental codes and data sets for this study can be downloaded from https://github.com/han-skai/GP-HTNLoc.

Acknowledgement

This work was supported by the Applied Basic Research Project in Yunnan Province (grant no. 202201AT070042), the project funding of the "Support Program of Xingdian Talents", the National Natural Science Foundation of China (grant no. 61862067, U1902201), Yunnan Provincial Science and Technology Department-Yunnan University Double First-Class Joint Fund Key Projects (grant no. 2019FY003027) and National Key R&D Program of China (grant no. 2022YFC2602500).

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.csbj.2024.04.052.

References

[1] Fu XD. Non-coding RNA: a new frontier in regulatory biology. Natl Sci Rev 2014;1 (2):190–204.
[2] Sheng N, Huang L, Lu Y, et al. Data resources and computational methods for lncRNA-disease association prediction. Comput Biol Med 2023:106527.
[3] Savulescu AF, Brackin R, Bouilhol E, et al. Interrogating RNA and protein spatial subcellular distribution in smFISH data with DypFISH. Cell Rep Methods 2021;1 (5).
[4] Zappulo A, Van Den Bruck D, Ciolli Mattioli C, et al. RNA localization is a key determinant of neurite-enriched proteome. Nat Commun 2017;8(1):583.
[5] Jopling CL, Schütz S, Sarnow P. Position-dependent function for a tandem microRNA miR-122-binding site located in the hepatitis C virus RNA genome. Cell Host Microbe 2008;4(1):77–85.
[6] Moffitt J.R., Zhuang X. RNA imaging with multiplexed error-robust fluorescence in situ hybridization (MERFISH)[M]//Methods in enzymology. Academic Press, 2016, 572: 1–49.
[7] Jagannathan S, Nwosu C, Nicchitta CV. Analyzing Subcellular mRNA localization via cell fractionation. Methods Mol Biol 2011;714:301.
[8] Mas-Ponte D, Carlevaro-Fita J, Palumbo E, et al. LncATLAS database for subcellular localization of long noncoding RNAs. Rna 2017;23(7):1080–7 (Xiao W, Lin G, Guo X, et al).
[9] Wen X, Gao L, Guo X, et al. lncSLdb: a resource for long non-coding RNA subcellular localization. Database 2018;2018. bay085.https:// doi. org/ 10.1093/ datab ase/ bay085.
[10] Cui T, Dou Y, Tan P, et al. RNALocate v2. 0: an updated resource for RNA subcellular localization with increased coverage and annotation. Nucleic Acids Res 2022;50(D1). D333-D339.
[11] Kozomara A, Birgaoanu M, Griffiths-Jones S. miRBase: from microRNA sequences to function. Nucleic Acids Res 2019;47(D1):D155–62.

[12] Xiao L, Wang J, Ju S, et al. Disorders and roles of tsRNA, snoRNA, snRNA and piRNA in cancer. J Med Genet 2022;59(7):623–31.
[13] Peng Y, Li J, Zhu L. Cancer and non-coding RNAs. Nutritional epigenomics. Academic Press; 2019. p. 119–32.
[14] Fan Y, Chen M, Zhu Q. lncLocPred: predicting LncRNA subcellular localization using multiple sequence feature information. IEEE Access 2020;8:124702–11.
[15] Cai J, Wang T, Deng X, et al. GM-lncLoc: LncRNAs subcellular localization prediction based on graph neural network with meta-learning. BMC Genom 2023; 24(1):52.
[16] Yang Y, Fu X, Qu W, et al. MiRGOFS: a GO-based functional similarity measurement for miRNAs, with applications to the prediction of miRNA subcellular localization and miRNA–disease association. Bioinformatics 2018;34 (20):3547–56.
[17] Zhang ZY, Ning L, Ye X, et al. iLoc-miRNA: extracellular/intracellular miRNA prediction using deep BiLSTM with attention mechanism. Brief Bioinforma 2022; 23(5):bbac395.
[18] 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 Genom 2021;22(1): 1–14.
[19] Zhou H, Wang H, Tang J, et al. Identify ncRNA subcellular localization via graph regularized $ k $ k-local hyperplane distance nearest neighbor model on multi-kernel learning. IEEE/ACM Trans Comput Biol Bioinforma 2021;19(6):3517–29.
[20] Bai T, Liu B. ncRNALocate-EL: a multi-label ncRNA subcellular locality prediction model based on ensemble learning. Brief Funct Genom 2023. elad007.
[21] Wan S, Duan Y, Zou Q. HPSLPred: an ensemble multi-label classifier for human protein subcellular location prediction with imbalanced source. Proteomics 2017; 17(17-18):1700262.
[22] Ying-Ying X, Fan Y, Hong-Bin S. Incorporating organelle correlations into semi-supervised learning for protein subcellular localization prediction. Bioinformatics 2016;32(14):14.
[23] Shen HB, Chou KC. Virus-mploc: a fusion classifier for viral protein subcellular location prediction by incorporating multiple sites. J Biomol Struct Dyn 2010;28 (2):175–86.
[24] 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.
[25] Ahmad A, Lin H, Shatabda S. Locate-R: subcellular localization of long non-coding RNAs using nucleotide compositions. Genomics 2020;112(3):2583–9.
[26] Wang Y, Zhu X, Yang L, et al. IDDLncLoc: subcellular localization of LncRNAs based on a framework for imbalanced data distributions. Interdiscip Sci: Comput Life Sci 2022;14(2):409–20.
[27] Xiao L., Zhang X., Jing L., et al. Does head label help for long-tailed multi-label text classification[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2021, 35(16): 14103–14111.
[28] Chen J., Li X., Xi J., et al. Rare Codes Count: Mining Inter-code Relations for Long-tail Clinical Text Classification[C]//Proceedings of the 5th Clinical Natural Language Processing Workshop. 2023: 403–413.
[29] Yu P., Ji H. Shorten the Long Tail for Rare Entity and Event Extraction[C]// Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics. 2023: 1331–1342.
[30] Lu Z, Zhong H, Tang L, et al. Predicting lncRNA-disease associations based on heterogeneous graph convolutional generative adversarial network. PLoS Comput Biol 2023;19(11):e1011634.
[31] Dong Y., Chawla N.V., Swami A. metapath2vec: Scalable representation learning for heterogeneous networks[C]//Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. 2017: 135–144.
[32] Lundberg SM, Erion G, Chen H, et al. From local explanations to global understanding with explainable AI for trees. Nat Mach Intell 2020;2(1):56–67.
[33] Zhang T, Tan P, Wang L, et al. RNALocate: a resource for RNA subcellular localizations. Nucleic Acids Res 2017;45(D1):D135–8.
[34] Siami-Namini S., Tavakoli N., Namin A.S. The performance of LSTM and BiLSTM in forecasting time series[C]//2019 IEEE International conference on big data (Big Data). IEEE, 2019: 3285–3292.
[35] Bonidia RP, Domingues DS, Sanches DS, et al. MathFeature: feature extraction package for DNA, RNA and protein sequences based on mathematical descriptors. Brief Bioinforma 2022;23(1):bbab434.
[36] 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 Genom 2021;22: 1–14.
[37] Dao FY, Lv H, Zhang D, et al. DeepYY1: a deep learning approach to identify YY1-mediated chromatin loops. Brief Bioinforma 2021;22(4):bbaa356.
[38] Ye M, Zhang J, Wei M, et al. Emerging role of long noncoding RNA-encoded micropeptides in cancer. Cancer Cell Int 2020;20(1):1–9.
[39] Chang CC, Liu TY, Lee YT, et al. Genome-wide analysis of lncRNAs in 3′-untranslated regions: CR933609 acts as a decoy to protect the INO80D gene. Int J Oncol 2018;53(1):417–33.
[40] Fickett JW. Recognition of protein coding regions in DNA sequences[J]. Nucleic Acids Res 1982;10(17):5303–18.
[41] Chen F, Wang YC, Wang B, et al. Graph representation learning: a survey[J]. APSIPA Trans Signal Inf Process 2020;9:e15.
[42] Nguyen T, Nguyen GTT, Nguyen T, et al. Graph convolutional networks for drug response prediction[J]. IEEE/ACM Trans Comput Biol Bioinforma 2021;19(1): 146–54.
[43] Qiu YL, Zheng H, Gevaert O. Genomic data imputation with variational auto-encoders[J]. GigaScience 2020;9(8):giaa082.
[44] Lai B, Xu J. Accurate protein function prediction via graph attention networks with predicted structure information[J]. Brief Bioinforma 2022;23(1):bbab502.
[45] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations [C]. Proc 20th ACM SIGKDD Int Conf Knowl Discov data Min 2014:701–10.
[46] Grover A, Leskovec J. node2vec: Scalable feature learning for networks[C]. Proc 22nd ACM SIGKDD Int Conf Knowl Discov data Min 2016:855–64.
[47] Wang X, Bo D, Shi C, et al. A survey on heterogeneous graph embedding: methods, techniques, applications and sources[J]. IEEE Trans Big Data 2022;9(2):415–36.
[48] McCormick C. Word2vec tutorial-the skip-gram model[J]. Apr-2016.[Online]. Available: http://mccormickml. com/2016/04/19/word2vec-tutorial-the-skip-gram-model, 2016.
[49] Zhang ML, Zhou ZH. ML-KNN: A lazy learning approach to multi-label learning[J]. Pattern Recognit 2007;40(7):2038–48.
[50] Imambi S, Prakash KB, Kanagachidambaresan GR. PyTorch[J]. Program Tensor: Solut Edge Comput Appl 2021:87–104.
[51] Wang M.Y. Deep graph library: Towards efficient and scalable deep learning on graphs[C]//ICLR workshop on representation learning on graphs and manifolds. 2019.
[52] Castellanos-Rubio A, Kratchmarov R, Sebastian M, et al. Cytoplasmic form of Carlr lncRNA facilitates inflammatory gene expression upon NF-κB activation[J]. J Immunol 2017;199(2):581–8.
[53] Li F, Chen J, Ge Z, et al. Computational prediction and interpretation of both general and specific types of promoters in Escherichia coli by exploiting a stacked ensemble-learning framework. Brief Bioinform 2021;22:2126–40.
[54] Li F, Guo X, Jin P, et al. Porpoise: a new approach for accurate prediction of RNA pseudouridine sites. Brief Bioinform 2021;22(6):bbab245.
[55] Li F, Guo X, Xiang D, et al. Computational analysis and prediction of PE_PGRS proteins using machine learning. Comput Struct Biotechnol J 2022;20:662–74.
[56] Huang Zh, Du Yp, Wen Jt, et al. snoRNAs: functions and mechanisms in biological processes, and roles in tumor pathophysiology. Cell Death Discov 2022;8:259. https://doi.org/10.1038/s41420-022-01056-8.
[57] Samarsky DA, Fournier MJ, Singer RH, et al. The snoRNA box C/D motif directs nucleolar targeting and also couples snoRNA synthesis and localization[J]. EMBO J 1998.
[58] Lubelsky Y, Ulitsky I. Sequences enriched in Alu repeats drive nuclear localization of long RNAs in human cells[J]. Nature 2018;555(7694):107–11.
[59] 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[J]. Bioinformatics 2021;37(16):2308–16.
[60] Li Y, Zeng M, Zhang F, et al. DeepCellEss: cell line-specific essential protein prediction with attention-based interpretable deep learning[J]. Bioinformatics 2023;39(1):btac779.