OXFORD

Briefings in Bioinformatics, 2022, 1–9

https://doi.org/10.1093/bib/bbac527

Problem Solving Protocol

Predicting potential interactions between lncRNAs and proteins via combined graph auto-encoder methods

Jingxuan Zhao, Jianqiang Sun, Stella C. Shuai, Qi Zhao and Jianwei Shuai

Corresponding authors: Qi Zhao. E-mail: zhaoqi@lnu.edu.cn; Jianwei Shuai. E-mail: jianweishuai@xmu.edu.cn

Abstract

Long noncoding RNA (lncRNA) is a kind of noncoding RNA with a length of more than 200 nucleotide units. Numerous research studies have proven that although lncRNAs cannot be directly translated into proteins, lncRNAs still play an important role in human growth processes by interacting with proteins. Since traditional biological experiments often require a lot of time and material costs to explore potential lncRNA–protein interactions (LPI), several computational models have been proposed for this task. In this study, we introduce a novel deep learning method known as combined graph auto-encoders (LPICGAE) to predict potential human LPIs. First, we apply a variational graph auto-encoder to learn the low dimensional representations from the high-dimensional features of lncRNAs and proteins. Then the graph auto-encoder is used to reconstruct the adjacency matrix for inferring potential interactions between lncRNAs and proteins. Finally, we minimize the loss of the two processes alternately to gain the final predicted interaction matrix. The result in 5-fold cross-validation experiments illustrates that our method achieves an average area under receiver operating characteristic curve of 0.974 and an average accuracy of 0.985, which is better than those of existing six state-of-the-art computational methods. We believe that LPICGAE can help researchers to gain more potential relationships between lncRNAs and proteins effectively.

Keywords: LncRNA, protein, lncRNA-protein interactions, graph auto-encoder, variational graph auto-encoder

Introduction

A considerable number of genes in the genome are transcribed into RNAs that do not code for proteins in genetic regulation. They are named as noncoding RNA (ncRNA) by researchers [1]. In the past few decades, ncRNA has been ignored by researchers because it violates the central dogma. Nowadays, a growing number of evidence suggest that lncRNAs can participate in a series of biological processes, such as genetic performance regulation [2], disease progression [3], immune response [4], etc. Furthermore, one of the most important ways for lncRNA to perform its biological function is interacting with relevant proteins. For example, HOTAIR, one of the earliest known lncRNAs shown to be associated with cancer, can interact with the polycomb group protein PRC2 to participate in chromatin modification complexes [5]. ANRIL is an antisense lncRNA upregulated in prostate cancer that can interact with chromobox 7 protein [6]. In recent years, the emergence of high-throughput technologies and the development in various biological experimental approaches have led to the expansion of the lncRNA world. This means that we can confirm the relationships between certain lncRNAs and proteins through high-throughput technologies such as RNA compete [7], RIP-Chip [8], MS2 trapping [9], etc. However, the large-scale experiments to identify lncRNA–protein interactions (LPIs) are often burdensome. Fortunately, by taking advantage of the large amount of accumulated experimental data and the rich property information of lncRNAs and proteins, we are able to predict the potential interactions between lncRNAs and proteins using computational methods.

During recent years, many kinds of research studies such as miRNA–lncRNA interactions prediction [10, 11], miRNA–disease associations prediction [12–14], metabolite–disease associations prediction [15] and circRNA-disease associations prediction [16, 17] have been carried out in bioinformatics. These studies have promoted the development of methods for predicting LPI to a certain extent. In 2011, Muppirala *et al.* developed a computational model called RPISeq [18], which applied sequence features of lncRNAs and proteins and made use of Support Vector Machine (SVM) and Random Forests (RF) as its two subclassifiers. In 2013, Lu *et al.* proposed a matrix computation method called LncPro for LPI prediction [19], which integrated hydrogen-bonding features, secondary structure features, Vander Waal's interaction features, and digitized these features by Fisher's linear discriminant analysis. In 2015, Suresh *et al.* proposed RPI-Pred [20], a SVM-based method that applies both RNA secondary structure features and protein three-dimensional structural features for LPI prediction. In 2017, Liu *et al.* developed a neighborhood regularized logistic matrix factorization algorithm for LPI prediction named LPI-NRLMF [21]. In 2018, Hu *et al.* integrated three models of SVM, RF and eXtreme Gradient Boosting (XGB) [22] using a linear ensemble strategy, and applied the sequence features extracted by three different methods to construct the HLPI-Ensemble model [23]. In the same year, Zhang *et al.* proposed a sequence-based feature projection ensemble learning frame named SFPEL-LPI [24]. They combined multiple lncRNA–lncRNA similarities, multiple protein–protein similarities and multiple sequence features with a feature projection ensemble learning frame. In 2019, Yi *et al.* proposed

Jingxuan Zhao is a graduate student in University of Science and Technology Liaoning. His research interests include bioinformatics and deep learning.
Jianqiang Sun is an associate professor at Linyi University. His research interests include bioinformatics and deep learning.
Stella C. Shuai is an undergraduate student in Northwestern University. Her research interests include bioinformatics and biomedicine.
Qi Zhao is a professor in University of Science and Technology Liaoning. His research interests include bioinformatics, complex network and machine learning.
Jianwei Shuai is a professor at the Department of Physics, Xiamen University. His research interests include biophysics, deep learning and bioinformatics.

Received: September 2, 2022. **Revised:** October 23, 2022. **Accepted:** November 6, 2022

LPI-Pred [25], which used a word2vec model [26] to obtain word embedding vectors of lncRNA sequences and protein sequences as features, and also predicted potential LPIs by RF classifier.

In addition, network-based methods have also achieved efficient advancements [27, 28]. In 2016, Ge *et al.* proposed LPBNI [29], which predicted LPIs in matrix iterative form on bipartite networks via a two-step propagation process. In 2017, Hu *et al.* presented an eigenvalue transformation-based semi-supervised link prediction approach (named LPI-ETSLP) to uncover the possible relationship between lncRNAs and proteins [30]. In 2018, Zhao *et al.* developed a semi-supervised LPI predictive model based on random walk called RWLPAP [31]. RWLPAP made full use of LPI information, lncRNA similarity information and protein similarity information to gain more accurate prediction. Soon afterwards, Zhang *et al.* proposed a linear neighborhood propagation method named LPLNP [32]. It calculated the linear neighborhood similarity and transferred it into the interaction space. In the same year, Zhao *et al.* proposed IRWNRLPI [33], which synthesized two algorithms, random walk and neighborhood regularized logistic matrix factorization, to obtain a potential LPI scoring matrix. Zhao *et al.* also developed a bipartite network projection recommended algorithm (named LPI-BNPRA) for this task [34]. Later, Zhang *et al.* proposed LPGNMF [35], which added graph regularization to the non-negative matrix factorization to further improve the model performance for LPI prediction. In 2020, Zhou *et al.* proposed a LPI prediction algorithm based on similarity kernel fusion and Laplacian regularized least squares algorithm to predict potential LPI interactions, called LPI-SKF [36].

Nowadays, deep learning methods are gradually surpassing the original methods for LPI prediction due to their high efficiency and high accuracy. In 2016, Pan *et al.* developed a computational method named IPMiner [37]. It made use of a stacked autoencoder to mine hidden features from sequence information of lncRNAs and proteins, and sent the hidden features into RF models to gain prediction results. In 2018, Yang *et al.* introduced a comprehensive lncRNA identification and functional annotation tool called LncADeep [38], which is based on deep neural networks using both sequence and structural information to predict potential LPI. In 2020, Zhang *et al.* proposed a deep learning model based on convolutional neural networks called LPI-CNNCP [39], which applied copy padding trick on sequence information to unify the sequences to a fixed dimension. In 2021, Li *et al.* proposed a LPI prediction model based on multi-channel capsule networks called Capsule-LPI [40], which can integrate multiple features of lncRNAs and proteins to participate in prediction. In the same year, Jin *et al.* developed an end-to-end deep learning model based on GAEs and collaborative training to predict potential interactions between lncRNAs and proteins, called LPIGAC [41]. Not long after this, Shen *et al.* proposed a GNN-based approach to predict interactions between ncRNAs and proteins, called NPI-GNN [42]. This can make predictions based on sequence information and network information. Later, Tian *et al.* introduced a deep forest model with cascade forest structure named LPIDF to predict new LPIs [43].

Generally, the traditional LPI prediction methods are divided into two categories. The first category usually starts with sequence information to extract high-dimensional digital feature vectors, and then the sequence feature vectors are sent to the machine learning classifier to obtain potential predictions. The other type makes full use of interaction information and similarity information for prediction by constructing a heterogeneous network composed of an adjacency matrix containing LPI information and similarity networks of lncRNAs and proteins. However, there are some obvious disadvantages in the previous methods. First, none of these methods fully exploit the rich topological information in the lncRNA–protein graph for label prediction [44]. Second, the feature extraction process and the label prediction process of these methods are divided into two separate parts and therefore lack connections. To solve the existing difficulties, we propose a combined GAE based method called LPICGAE to predict potential LPI. In our study, we first send the extracted high-dimensional features of lncRNAs and proteins into two variational GAEs (VGAEs) to obtain effective representations, then use GAEs to implement the label propagation process. Lastly, we optimize the losses in these two processes alternately to gain the final predicted interaction matrix. The reasons for using the combination of VGAE and GAE are as follows: the interaction prediction task of lncRNAs and proteins can be regarded as a label propagation problem on a complex biological network [45], while GAE and VGAE are presenting excellent performance in the feature representation of nodes on graph structure data, leading to their wide application in the graph structure data problem of bioinformatics [46, 47]. Additionally, we can deeply integrate the feature extraction process and the label prediction process through this combined model. After model construction, we conduct a 5-fold cross-validation (5-fold CV) experiment to evaluate the performance of LPICGAE and verify the robustness of LPICGAE on an external validation dataset. Meanwhile, we also do case studies. All the results show that LPICGAE is an efficient model for LPI prediction task.

Materials and methods

Data preparation

In this study, we adopt two datasets for model training and testing. The first dataset which we call dataset1 is extracted from the NPInter v2.0 database [48]. NPInter is a database that integrates the experimental interaction of ncRNA and multiple biomolecules, which covers the majority of known human LPIs. Another dataset named dataset2 is extracted from the lncRNome database [49]. It is used as an external validation dataset. The lncRNome database is a comprehensive knowledge base for human lncRNA. Beyond that, to enhance the persuasiveness of using an external validation dataset, we remove the overlap between dataset1 and dataset2.

Dataset1 includes 8112 interactions among 3046 lncRNAs and 136 proteins. Dataset2 includes 2729 interactions among 1184 lncRNAs and 9 proteins. The sequence information of lncRNAs and proteins mentioned above are obtained from NONCODE v3.0 database [50] and UniProt database [51], respectively.

Adjacency matrix construction and feature extraction

LPICGAE needs the adjacency matrix and feature matrix as input, and then outputs the predicted interaction score of each lncRNA–protein pair. The adjacency matrix stores the interaction information between lncRNAs and proteins. Supposing that the number of lncRNAs is m, and the number of proteins is n, we use $A_{m\times n}$ to represent the adjacency matrix. $A_{ij} = 1$indicates that lncRNAi has a known interaction with protein j, otherwise $A_{ij} = 0$. Since the sequence information can correspond to a unique lncRNA (protein), we obtain the numerical feature vector of lncRNA or protein through sequence information. For each lncRNA (protein), we adopt a Doc2Vec model [52], which is widely used in the field

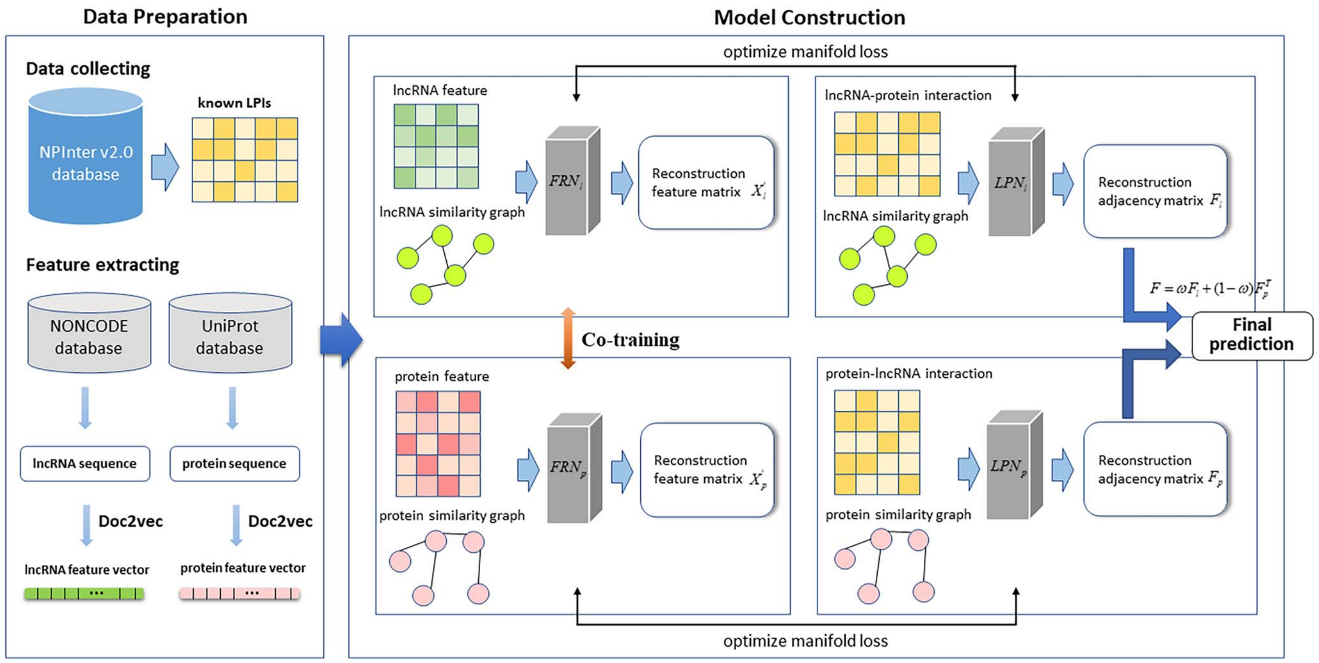

Figure 1. The workflow of LPICGAE.

of natural language processing to generate its 300-dimensional feature vector.

In the Doc2Vec model, feature representation of continuous lncRNA (protein) sequences is based upon the assumption that a set of lncRNA (protein) sequences comprises a document. In particular, each sequence is considered as a sentence written in a biological language, suggesting that the corresponding biological function can be semantically interpreted. As for training data (termed as corpus), we utilize non-redundant lncRNA sequences and protein sequences from GENCODE v39 database [53]. After collecting the training data, we break such biological sequences into non-overlapping residue segments (k-mers) as biological words. Then we use these k-mer residue segments (words) and the complete sequences (sentences) to train the Doc2vec model. All the word and sentence vectors are trained by using stochastic gradient descent and backpropagation to update weight parameters iteratively. After training, the output sentence vectors are used as our lncRNA (protein) sequence features.

Combined graph auto-encoders to predict potential human lncRNA-protein interactions

Our model consists of two main parts. The first part is the feature reconstruction network, and the other part is the label prediction network. For convenience, we name the feature reconstruction network as *FRN* and the label prediction network as *LPN*. *FRN* is used to capture efficient low-dimensional representation from high-dimensional features through inferring representation from the feature matrix of lncRNAs and proteins, respectively. In our model, *FRN* is divided into lncRNA feature reconstruction network FRN_l and protein feature reconstruction network FRN_p. Meanwhile *LPN* is applied to infer unknown interactions from known interactions. *LPN* is divided into lncRNA label prediction network LPN_l and protein label prediction network LPN_p. *FRN* is implemented with a VGAE [54], and the *LPN* is implemented with a GAE [54]. The workflow chart of LPICGAE is shown in Figure 1.

Graph auto-encoder

The traditional auto-encoder (AE) is a neural network with an encoder and a decoder. The encoder takes the high-dimensional vector x as input and converts it into a low-dimensional representation z, while the decoder extracts the low-dimensional representation z and returns the reconstructed vector $\hat{x}$. The loss function measures information lost between x and $\hat{x}$.

The purpose of GAE is to apply traditional AE to graph structure data and use GAE to reconstruct the input graph. GAE makes use of graph convolution neural network (GCN) as an encoder to obtain the latent representation of nodes in the input graph. The embedding of all nodes Z can be expressed as

$$Z = \mathbf{GCN}\ X, A = \tilde{A}\ \mathbf{ReLU}\left(\tilde{A}XW_0\right) W_1 \tag{1}$$

$$\tilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{2}$$

where A is the adjacency matrix, $\tilde{A}$ is the normalized adjacency matrix, X is the feature matrix, D is the degree matrix obtained from the adjacency matrix and W_0 and W_1 are the initialized parameter matrix and the updated parameter matrix after the GCN layer, respectively.

After getting the embedding Z from the encoder, GAE uses an inner-product decoder to reconstruct the original graph. The advantage of using the inner product decoder is that after obtaining the hidden embedding Z, we want to find a way to learn the similarity of each row in Z to generate the output adjacency matrix. The vect or inner product can calculate the cosine similarity of two vectors, which allows us to obtain a distance measure when the vector size is unchanged. Therefore, we use vector inner product to gain the reconstructed adjacency matrix by learning the similarity of each row in Z.

$$\hat{A} = \sigma\left(ZZ^T\right) \tag{3}$$

where $\sigma\ ()$ denotes the nonlinear convolution layer and $\hat{A}$ represents the reconstructed adjacency matrix. In this manner we can get the reconstructed A after GAE.

Variational graph auto-encoder

The difference between variational auto-encoder (VAE) and AE is that the input vector x in VAE is embedded as a distribution rather than a fixed vector, so the embedding Z is taken from a distribution rather than directly generated from the encoder. The distribution obtained by the encoder is usually parameterized into a normal distribution $N(\mu, \sigma^2)$. Then the following equation is utilized to get embedding Z generated with μ and σ. This process is called the reparameterization trick, and the reconstructed vector $\hat{x}$ is finally obtained through the decoder.

$$Z = \mu + \sigma * \varepsilon, \varepsilon \sim \mathrm{N}\,(0, 1) \tag{4}$$

The purpose of VGAE is to apply the VAE to graph structure data and use VGAE to construct new graphs or infer graphs. The encoder of VGAE is usually composed of two layers of GCN. An adjacency matrix and a feature matrix are used as input to generate a low-dimensional representation in the first layer, while the second layer catches the low-dimensional representation and generates μ and $\mathbf{log}\sigma^2$. Then the embedding Z can be calculated using a reparameterization trick through Equation (4). This process can be implemented by the following formulas:

$$\overline{X} = \mathbf{GCN}\,(X, A) = \mathbf{ReLU}\left(\tilde{A}XW_0\right) \tag{5}$$

$$\mu, \mathbf{log}\,\sigma^2 = \mathbf{GCN}_{\mu,\sigma}\,(X, A) = \tilde{A}\overline{X}W_1 \tag{6}$$

where $\overline{X}$ is the representation obtained from the first layer of encoder and $GCN_{\mu,\sigma}$ is the layer that generates μ and σ. Similar to GAE, we also define the decoder of VGAE by an inner-product decoder according to Equation (3), after which we can get the reconstruction graph.

Collaborative training and loss function

In the feature reconstruction process, we define FRN_l and FRN_p to represent lncRNA feature reconstruction network and protein feature reconstruction network. We use G_l and X_l as the input of FRN_l, and G_p and X_p as the input of FRN_p. The adjacency matrix of graph G_l and G_p can be constructed through calculating the Euclidean distance among feature vectors first and then finding 10-nearest neighborhoods of each node. X_l and X_p represent the feature matrix of lncRNA and protein, respectively. Naturally, we use X_l' and X_p' to represent the reconstructed feature matrix after *FRN*.

Due to the fact that the *FRN* module is implemented with VGAE, the loss function of *FRN* consists of two parts [54]. The first part is feature matrix reconstruction loss $Loss_{fr}$, while another part is KL divergence $Loss_{KL}$. They are calculated by the following equations:

$$Loss_{FRN} = Loss_{fr} + Loss_{KL} \tag{7}$$

$$Loss_{fr} = \frac{1}{2}\left\|X - X'\right\|_F^2 \tag{8}$$

$$Loss_{KL} = -\sum_{i,j}\frac{1}{2}\left(1 + 2\,\mathbf{log}\,\sigma_{ij} - \mu_{ij}^2 - \sigma_{ij}^2\right) \tag{9}$$

Supposing that Z_l and Z_p are representations learned from FRN_l and FRN_p, respectively, we use $Loss_c$ to represent the collaborative training loss. Then we use $Loss_f$ defined by the follow equations to represent the total loss of co-training feature reconstruction module.

$$Loss_c = \frac{1}{2}\left\|Z_l Z_p^T - A\right\|_F^2 \tag{10}$$

$$Loss_f = \omega Loss_{fl} + (1 - \omega)\, Loss_{fp} + \varphi Loss_c \tag{11}$$

Here, $Loss_{fl}$ and $Loss_{fp}$ are the loss of FRN_l and FRN_p computed by Equation (7), respectively. $\omega \in \left(0, 1\right)$ is the weight parameter set for balancing the information between lncRNA and protein. φ is the weight parameter representing the proportion of $Loss_c$ in $Loss_f$, which is initialized to 1e-3 and will be updated during model training. FRN_l and FRN_p can be trained simultaneously by optimizing $Loss_f$.

Similarly, in the label prediction module, we apply G_l and A as the input of LPN_l, and G_p and A^T as the input of LPN_p. The total loss of label prediction module $Loss_l$ can be calculated by the following equation.

$$Loss_l = \omega Loss_{ll} + (1 - \omega)\, Loss_{lp} \tag{12}$$

Here, $Loss_{ll}$ and $Loss_{lp}$ are the loss of LPN_l and LPN_p, respectively. The *LPN* module is implemented with GAE, whose loss is usually represented by the reconstruction loss [54]. In our model, we not only use the reconstruction loss, but also add the manifold loss when generating the loss of *LPN*. The manifold loss $Loss_m$ is the mean square error between the hidden representations of *FRN* and *LPN*. Previous research suggests that graph neural networks are significantly correlated to label propagation [55]. Meanwhile, label propagation leads to a manifold regularization problem, that is, samples with higher feature similarities are closer on the manifold and tend to share the same labels. This theory proves that representations learned by graph neural networks should follow manifold constraint. In addition, Xu *et al.* proved that the representations of two GAEs which share one same graph as input are often similar in manifold constraint [56]. Therefore, we add manifold loss to the total loss of LPN, thus improving the efficiency of obtaining better feature embedding:

$$Loss_{LPN} = Loss_{lr} + \tau Loss_m \tag{13}$$

$$Loss_{lr} = -\sum_{ij} A_{ij}\,\mathbf{log}\,F_{ij} \tag{14}$$

$$Loss_m = \frac{1}{2}\left\|Z - Z^{'}\right\|_F^2 \tag{15}$$

where $Loss_{lr}$ represents the reconstruction loss between the input adjacency matrix and the output adjacency matrix in *LPN*. τ is the weight parameter representing the proportion of $Loss_m$ in $Loss_{LPN}$, whose value is set equal to φ. Then, we optimize $Loss_f$ and $Loss_l$ alternately to gain the final predicted score matrix. In brief, LPN_l products F_l, LPN_p outputs F_p. Finally, we generate the predicted interaction matrix by the following equation:

$$F = \omega F_l + (1 - \omega)\, F_p^T \tag{16}$$

Table 1. The performance of LPICGAE under different values of hyperparameters

Hyperparameters	AUC	AUPR	ACC	F1-score	Precision
lr					
0.1	0.9576	0.5956	0.9729	0.4232	0.2118
0.01	0.9577	0.5609	0.9790	0.4141	0.1927
0.001	0.9453	0.5828	0.9805	0.4290	0.2123
num					
128	0.9440	0.4923	0.9770	0.3892	0.1705
256	0.9577	0.5609	0.9790	0.4141	0.1927
512	0.9653	0.6611	0.9815	0.5185	0.2877
η					
1e-4	0.8654	0.2872	0.9760	0.2222	0.0763
1e-5	0.9577	0.5609	0.9790	0.4141	0.1927
1e-6	0.9768	0.7419	0.9826	0.5845	0.3589
ω					
0.3	0.9298	0.4713	0.9779	0.4028	0.1820
0.4	0.9431	0.5076	0.9781	0.4010	0.1809
0.5	0.9577	0.5609	0.9790	0.4141	0.1927
0.6	0.9625	0.5982	0.9799	0.4620	0.2348
0.7	0.9658	0.6556	0.9815	0.5338	0.2997

Table 2. The main hyperparameters of LPICGAE

Hyperparameters	Values
Learning rate (**lr**)	1e-3
Number of nodes in the hidden layers (**num**)	512
Weight decay rate (η)	1e-6
Weight parameter (ω)	0.7

To summarize, in the loss function construction of LPICGAE, we have used different losses to constitute the entire loss. The feature reconstruction loss and KL divergence loss in FRN are used to represent the loss in the feature reconstruction process; their compatibility has been derived in the previous study [54]. The collaborative training loss in the FRN module is used to realize collaborative training between the lncRNA feature extraction process and the protein feature extraction process. The adjacency matrix reconstruction loss in the LPN module is used to represent the loss in the label prediction process. It has been derived in the previous study [54]. The manifold loss between the feature representations of FRN and LPN is used to deeply integrate the feature extraction process and the label prediction process. It has been derived in the previous study [55]. Each loss of function plays its own role. All the losses are aimed at enhancing the ability of LPICGAE to gain more efficient feature representations and realizing better predictions.

Results

Performance evaluation

We choose the 5-fold CV to evaluate the performance of LPICGAE on our dataset. In this method, the known LPIs are randomly divided into five equal parts and one of them is selected as a testing set each time while the remaining four parts are used as the training set. In this study, we use area under receiver operating characteristic curve (AUC), area under PR curve (AUPR), accuracy (ACC), f1-score (F1) and precision (Pre) to measure the performance of LPICGAE. Among them, the ROC curve is receiver operating characteristic curve, whose abscissa and ordinate are FPR (false positive rate) and TPR (true positive rate), respectively. The PR curve is precision-recall curve, whose abscissa and ordinate are precision rate and recall rate, respectively. The formulas of these metrics mentioned above are as follows:

$$TPR = \frac{TP}{TP + FN} \tag{17}$$

$$FPR = \frac{FP}{TN + FP} \tag{18}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{19}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{20}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{21}$$

$$\text{F1} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{22}$$

In the above formulas, TP and TN represent the number of positive and negative samples that are correctly predicted, respectively. FP and FN indicate the number of positive and negative samples that are wrongly predicted, respectively.

In LPICGAE, we apply several hyperparameters during the model construction process, including *lr* for learning rate, η for weight decay rate, *num* for number of nodes in the hidden layers and ω for weight parameter to balance the importance of information obtained from lncRNA space and protein space.

We investigate the impact on the performance of LPICGAE by controlling the hyperparameters mentioned above within a certain range. With the adjustment of the hyperparameters, the performance of LPICGAE is presented in Table 1. Finally, we get the optimal hyperparameter values combination after multiple experiments, which are shown in Table 2.

Comparison with other methods

To assess the performance of LPICGAE, we compare LPICGAE with six current state-of-the-art LPI prediction models, namely LPISKF, LPBNI, LPICNNCP, HLPI-Ensemble, LPI-NRLMF and LPIDF. In order to enhance the persuasiveness of the comparative experiments, our comparative models cover network-based methods,

Table 3. The performance of LPICGAE and six comparison methods in 5-fold CV under dataset1

Method	AUC	AUPR	ACC	F1-score	Precision
LPICGAE	0.9740	0.7688	0.9851	0.6397	0.4238
LPISKF	0.9650	0.6080	0.8488	0.6410	0.5960
LPBNI	0.8382	0.6716	0.8211	0.6960	0.6144
LPICNNCP	0.9492	0.9441	0.9104	0.9194	0.8669
LPIDF	0.9694	0.9583	0.9344	0.9374	0.8965
LPI-NRLMF	0.9606	0.9596	0.8289	0.8480	0.8644
HLPI-Ensemble	0.9644	0.6312	0.9100	0.9085	0.9096

machine learning-based methods and deep learning-based methods. Table 3 shows the performance of each model in 5-fold CV under dataset1.

- LPISKF [36] is a LPI prediction algorithm based on similarity kernel fusion and Laplacian regularized least squares algorithm to predict potential LPI.
- LPBNI [29] constructed an lncRNA–protein bipartite network inference to predict LPIs.
- LPICNNCP [39] is a novel convolutional neural network method with a copy-padding trick for LPI prediction.
- HLPI-Ensemble [23] employed three mainstream machine learning algorithms of SVM, RF and XGB by ensemble strategy to predict LPIs.
- LPI-NRLMF [21] mapped the LPI matrix to the lncRNA similarity matrix and the protein similarity matrix to predict the possibility of LPIs.
- LPIDF [43] is a deep forest model with cascade forest structure designed to find new LPIs.

From Table 3 we can see that the performance of LPICGAE outdoes the other six methods under dataset1. LPICGAE yields an average AUC of 0.9740, which is 0.9%, 13.58%, 2.48%, 0.46%, 1.34% and 0.96% higher than that of LPISKF, LPBNI, LPICNNCP, LPIDF, LPI-NRLMF and HLPI-Ensemble, respectively. We also show the ROC curves of LPICGAE and the other six methods in 5-fold CV under dataset1 in Figure 2. The average ACC of LPICGAE is 0.9851, which is 13.63%, 16.4%, 7.47%, 5.07%, 15.62% and 7.51% higher than that of LPISKF, LPBNI, LPICNNCP, LPIDF, LPI-NRLMF and HLPI-Ensemble, respectively. However, LPICGAE does not outperform other methods on AUPR, F1-score and Precision. This may be due to an imbalance in the adjacency matrix graph made by dataset1, in which most edges tend to connect with a small part of nodes. It is difficult for the graph neural network method to perform its best performance on the imbalanced graph.

Performance on external validation dataset

To verify the robustness of LPICGAE, we do the same 5-fold CV experiment under dataset2, which is used as an external validation dataset in our work. The performance comparison of LPICGAE and other models in 5-fold CV under dataset2 is shown in Table 4.

From Table 4, it can be concluded that LPICGAE exhibits a robust performance on the external validation dataset. Specifically, LPICGAE achieves an average AUC of 0.9734, which is 1.13%, 23.55%, 12.3%, 5.95%, 3.7% and 15.42% higher than that of LPISKF, LPBNI, LPICNNCP, LPIDF, LPI-NRLMF and HLPI-Ensemble, respectively. The average AUPR of LPICGAE is 0.9421, which is 28.2%, 38.44%, 11.01%, 4.2%, 0.45% and 34.17% higher than that of LPISKF, LPBNI, LPICNNCP, LPIDF, LPI-NRLMF and HLPI-Ensemble. Similarly, we can figure out that the ACC and F1-score of LPICGAE are also higher than those of all other models. Significantly, the AUPR of LPICGAE under dataset2 is much higher than that under dataset1. This is due to data in dataset2 being more balanced with a positive and negative sample ratio of 1:2.9. In contrast, the ratio in dataset1 reaches 1:50.

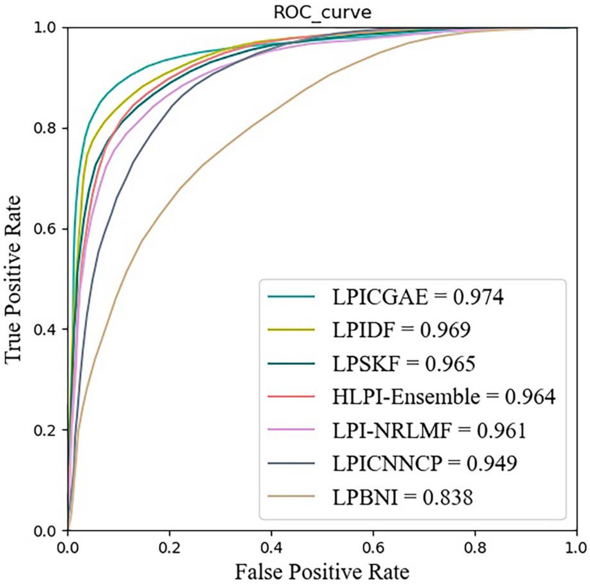

Figure 2. ROC curves of LPICGAE and six comparison methods in 5-fold CV under dataset 1.

Case study

To further prove the ability of LPICGAE in identifying novel LPIs, our model is implemented on several case studies. The benchmark dataset which we use during model training and testing is NPInter v2.0 database. FUS (fused in sarcoma) is a kind of protein integrally involved in amyotrophic lateral sclerosis and frontotemporal dementia, which can be searched in UniProt database by index P35637 [57]. We take the top 10 possible lncRNAs related to protein FUS with the highest probability from the predictive results by LPICGAE. It is important to note that these 10 interactions are not included in the NPInter v2.0 database.

Table 5 shows that the existence of these 10 interactions is confirmed in the NPInter v4.0 database [58] updated in 2019, which is direct evidence proving the utility of LPICGAE. For example, the interaction between lncRNA NONHSAG026396 and protein P35637 predicted by LPICGAE is not recorded in the NPInter v2.0 database, but it is complemented in NPInter v4.0 database by

Table 4. The performance of LPICGAE and six comparison methods in 5-fold CV under dataset2

Method	AUC	AUPR	ACC	F1-score	Precision
LPICGAE	0.9734	0.9421	0.9305	0.8534	0.7871
LPISKF	0.9621	0.6601	0.7461	0.7079	0.8418
LPBNI	0.7379	0.5577	0.7888	0.3852	0.7050
LPICNNCP	0.8504	0.8320	0.7474	0.6443	0.6578
LPIDF	0.9139	0.9001	0.8457	0.8495	0.8289
LPI-NRLMF	0.9364	0.9376	0.8070	0.8167	0.8506
HLPI-Ensemble	0.8192	0.6004	0.7643	0.7787	0.7340

Table 5. The top 10 predicted results of protein FUS related lncRNAs based on LPICGAE

Species	lncRNA ID	Protein ID	Confirmed	PMID
Homo sapiens	NONHSAG026396	P35637	Yes	23023293
Homo sapiens	NONHSAG005685	P35637	Yes	23023293
Homo sapiens	NONHSAG035903	P35637	Yes	23023293
Homo sapiens	NONHSAG000047	P35637	Yes	23023293
Homo sapiens	NONHSAG020957	P35637	Yes	22081015
Homo sapiens	NONHSAG035788	P35637	Yes	23023293
Homo sapiens	NONHSAG012986	P35637	Yes	23023293
Homo sapiens	NONHSAG039294	P35637	Yes	23023293
Homo sapiens	NONHSAG022006	P35637	Yes	23023293
Homo sapiens	NONHSAG048962	P35637	Yes	23023293

PMID 23023293. Therefore, we believe that LPICGAE has the ability to discover new LPIs.

Discussion and conclusion

With the rapid development of biomedicine during recent decades, researchers have gained a broader understanding of the molecular functions of lncRNAs. More and more studies have shown that lncRNAs are closely related to many human complex diseases. It is worth noting that lncRNAs usually exert their biological functions by interacting with related proteins rather than translating into fixed proteins following the central dogma. High-throughput sequencing technologies and large-scale biological experiments have been developed to help researchers explore the molecular functions of lncRNAs. However, how to obtain more LPI information efficiently and quickly has still been a difficult problem.

In this study, we introduce a novel deep learning framework (LPICGAE) based on combined GAEs to predict potential LPI interactions. First, we send the feature networks and similarity graphs of lncRNAs (proteins) into VGAE to implement the feature reconstruction. Second, the lncRNA–protein adjacency matrix and the similarity graph mentioned above are sent into GAE to reconstruct the adjacency matrix. Finally, we alternately perform the two processes and optimize the manifold loss between the two hidden layer embeddings to strengthen the ability of our model to acquire higher quality prediction. In the training process of LPICGAE, we also take the lncRNA space and the protein space as two independent processes and perform collaborative training to make full use of existing information. After model training, we can obtain the predictive score of each lncRNA–protein pair from the final adjacency matrix. We compare LPICGAE with six classical LPI prediction methods in 5-fold CV. We also apply an external validation dataset to verify the robustness of our model. The results show that LPICGAE exhibits the best comprehensive performance.

The ideal predictive ability of LPICGAE mainly depends on the following factors. First, none of the previous methods proposed for this task fully exploit the topological information in the stage of feature extraction, while LPICGAE makes full use of the sequence feature information of lncRNAs (proteins) and topological information on the LPI graph by applying VGAE for feature extraction. Second, the feature extraction process and the label prediction process of previous methods are divided into two separate parts and therefore lack of connections. LPICGAE can deeply integrate the feature extraction process and the label prediction process, and obtain higher quality feature embeddings by optimizing the manifold loss between two GAEs. Third, we apply collaborative training in LPICGAE, which helps gain balanced information between lncRNA space and protein space.

However, LPICGAE also suffers from some limitations. First, there is still a gap between the performances of LPICGAE under a balanced dataset and a highly unbalanced dataset, which results in some metrics in our study that are not ideal. Second, LPICGAE needs to be retrained when the dataset expands with new lncRNAs (proteins), which will affect the efficiency of LPICGAE. In the future, we will focus on collecting higher quality LPI datasets and exploring better feature extraction methods to improve the prediction performance of LPI.

Key Points

- We present a new deep learning algorithm (LPICGAE) based on combined graph auto-encoder methods for predicting potential lncRNA-protein interactions.
- We apply collaborative training in LPICGAE, which help gain balanced information between lncRNA space and protein space.
- LPICGAE can obtain higher quality feature embeddings by optimizing the manifold loss between two kinds of graph auto-encoders.

- Compared with existing state-of-the-art methods, LPIC-GAE achieves higher predictive accuracy.

Data availability

The source code and datasets are available online at https://github.com/zhaoqi106/LPICGAE.

Funding

Ministry of Science and Technology of the People's Republic of China (grant 2021ZD0201900); National Natural Science Foundation of China (grants 11805091, 12090052 and 11874310); Foundation of Education Department of Liaoning Province (grant LJKZ0280); Major projects in Fujian Province (grant 2020Y4001).

References

1. Djebali S, Davis CA, Merkel A, *et al.* Landscape of transcription in human cells. *Nature* 2012;**489**:101–8.
2. Guttman M, Amit I, Garber M, *et al.* Chromatin signature reveals over a thousand highly conserved large non-coding RNAs in mammals. *Nature* 2009;**458**:223–7.
3. Chen X. Predicting lncRNA-disease associations and constructing lncRNA functional similarity network based on the information of miRNA. *Sci Rep* 2015;**5**:13186.
4. Wapinski O, Chang HY. Long noncoding RNAs and human disease. *Trends Cell Biol* 2011;**21**:354–61.
5. Wu Y, Zhang L, Wang Y, *et al.* Long noncoding RNA HOTAIR involvement in cancer. *Tumour Biol* 2014;**35**:9531–8.
6. Yap KL, Li S, Munoz-Cabello AM, *et al.* Molecular interplay of the noncoding RNA ANRIL and methylated histone H3 lysine 27 by polycomb CBX7 in transcriptional silencing of INK4a. *Mol Cell* 2010;**38**:662–74.
7. Ray D, Kazan H, Chan ET, *et al.* Rapid and systematic analysis of the RNA recognition specificities of RNA-binding proteins. *Nat Biotechnol* 2009;**27**:667–70.
8. Keene JD, Komisarow JM, Friedersdorf MB. RIP-Chip: the isolation and identification of mRNAs, microRNAs and protein components of ribonucleoprotein complexes from cell extracts. *Nat Protoc* 2006;**1**:302–7.
9. Gong C, Popp MW, Maquat LE. Biochemical analysis of long non-coding RNA-containing ribonucleoprotein complexes. *Methods* 2012;**58**:88–93.
10. Zhang L, Liu T, Chen H, *et al.* Predicting lncRNA-miRNA interactions based on interactome network and graphlet interaction. *Genomics* 2021;**113**:874–80.
11. Zhang L, Yang P, Feng H, *et al.* Using network distance analysis to predict lncRNA-miRNA interactions. *Interdiscip Sci* 2021;**13**: 535–45.
12. Chen X, Zhu CC, Yin J. Ensemble of decision tree reveals potential miRNA-disease associations. *PLoS Comput Biol* 2019;**15**:e1007209.
13. Chen X, Sun LG, Zhao Y. NCMCMDA: miRNA-disease association prediction through neighborhood constraint matrix completion. *Brief Bioinform* 2021;**22**:485–96.
14. Chen X, Li TH, Zhao Y, *et al.* Deep-belief network for predicting potential miRNA-disease associations. *Brief Bioinform* 2021;**22**:bbaa186.
15. Sun F, Sun J, Zhao Q. A deep learning method for predicting metabolite–disease associations via graph neural network. *Brief Bioinform* 2022;**23**:bbac266.
16. Zhao Q, Yang Y, Ren G, *et al.* Integrating bipartite network projection and KATZ measure to identify novel CircRNA-disease associations. *IEEE Trans Nanobiosci* 2019;**18**:578–84.
17. Wang CC, Han CD, Zhao Q, *et al.* Circular RNAs and complex diseases: from experimental results to computational models. *Brief Bioinform* 2021;**22**:bbab286.
18. Muppirala UK, Honavar VG, Dobbs D. Predicting RNA-protein interactions using only sequence information. *BMC Bioinformatics* 2011;**12**:1–11.
19. Lu Q, Ren S, Lu M, *et al.* Computational prediction of associations between long non-coding RNAs and proteins. *BMC Genomics* 2013;**14**:1–10.
20. Suresh V, Liu L, Adjeroh D, *et al.* RPI-Pred: predicting ncRNA-protein interaction using sequence and structural information. *Nucleic Acids Res* 2015;**43**:1370–9.
21. Liu H, Ren G, Hu H, *et al.* LPI-NRLMF: lncRNA-protein interaction prediction by neighborhood regularized logistic matrix factorization. *Oncotarget* 2017;**8**:103975.
22. Chen T, Guestrin C. XGBoost: a scalable tree boosting system ACM SIGKDD international conference on knowledge discovery and data mining. *ACM* 2016;785–94.
23. Hu H, Zhang L, Ai H, *et al.* HLPI-Ensemble: prediction of human lncRNA-protein interactions based on ensemble strategy. *RNA Biol* 2018;**15**:797–806.
24. Zhang W, Yue X, Tang G, *et al.* SFPEL-LPI: Sequence-based feature projection ensemble learning for predicting LncRNA-protein interactions. *PLoS Comput Biol* 2018;**14**:e1006616.
25. Yi HC, You ZH, Cheng L, *et al.* Learning distributed representations of RNA and protein sequences and its application for predicting lncRNA-protein interactions. *Comput Struct Biotechnol J* 2020;**18**:20–6.
26. Le Q, Mikolov T. Distributed representations of sentences and documents. In: *Proceedings of the 31st International Conference on Machine Learning*, PMLR, 2014;**32**:1188–96.
27. Shen L, Liu F, Huang L, *et al.* VDA-RWLRLS: An anti-SARS-CoV-2 drug prioritizing framework combining an unbalanced bi-random walk and Laplacian regularized least squares. *Comput Biol Med* 2021;**140**:105119.
28. Peng L, Wang F, Wang Z, *et al.* Cell-cell communication inference and analysis in the tumour microenvironments from single-cell transcriptomics: data resources and computational strategies. *Brief Bioinform* 2022;**23**:bbac234.
29. Ge M, Li A, Wang M. A Bipartite network-based method for prediction of long non-coding RNA-protein interactions. *Genomics Proteomics Bioinformatics* 2016;**14**:62–71.
30. Hu H, Zhu C, Ai H, *et al.* LPI-ETSLP: lncRNA-protein interaction prediction using eigenvalue transformation-based semi-supervised link prediction. *Mol Biosyst* 2017;**13**:1781–7.
31. Zhao Q, Liang D, Hu H, *et al.* RWLPAP: random walk for lncRNA-protein associations prediction. *Protein Pept Lett* 2018;**25**:830–7.
32. Zhang W, Qu Q, Zhang Y, *et al.* The linear neighborhood propagation method for predicting long non-coding RNA–protein interactions. *Neurocomputing* 2018;**273**:526–34.
33. Zhao Q, Zhang Y, Hu H, *et al.* IRWNRLPI: integrating random walk and neighborhood regularized logistic matrix factorization for lncRNA-protein interaction prediction. *Front Genet* 2018;**9**: 239.
34. Zhao Q, Yu H, Ming Z, *et al.* The bipartite network projection-recommended algorithm for predicting long non-coding RNA-protein interactions. *Mol Ther Nucleic Acids* 2018;**13**:464–71.

35. Zhang T, Wang M, Xi J, *et al.* LPGNMF: predicting long noncoding RNA and protein interaction using graph regularized nonnegative matrix factorization. *IEEE/ACM Trans Comput Biol Bioinform* 2020;**17**:189–97.
36. Zhou YK, Hu J, Shen ZA, *et al.* LPI-SKF: predicting lncRNA-protein interactions using similarity kernel fusions. *Front Genet* 2020;**11**:615144.
37. Pan X, Fan YX, Yan J, *et al.* IPMiner: hidden ncRNA-protein interaction sequential pattern mining with stacked autoencoder for accurate computational prediction. *BMC Genomics* 2016;**17**:582.
38. Yang C, Yang L, Zhou M, *et al.* LncADeep: an ab initio lncRNA identification and functional annotation tool based on deep learning. *Bioinformatics* 2018;**34**:3825–34.
39. Zhang SW, Zhang XX, Fan XN, *et al.* LPI-CNNCP: prediction of lncRNA-protein interactions by using convolutional neural network with the copy-padding trick. *Anal Biochem* 2020;**601**:113767.
40. Li Y, Sun H, Feng S, *et al.* Capsule-LPI: a LncRNA-protein interaction predicting tool based on a capsule network. *BMC Bioinformatics* 2021;**22**:246.
41. Jin C, Shi Z, Zhang H, *et al.* Predicting lncRNA-protein interactions based on graph autoencoders and collaborative training. *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* 2021;38–43.
42. Shen ZA, Luo T, Zhou YK, *et al.* NPI-GNN: predicting ncRNA-protein interactions with deep graph neural networks. *Brief Bioinform* 2021;**22**:bbab051.
43. Tian X, Shen L, Wang Z, *et al.* A novel lncRNA-protein interaction prediction method based on deep forest with cascade forest structure. *Sci Rep* 2021;**11**:18881.
44. Pan X, Hu L, Hu P, *et al.* Identifying protein complexes from protein-protein interaction networks based on fuzzy clustering and GO semantic information. *IEEE/ACM Trans Comput Biol Bioinform* 2022;**19**:2882–93.
45. Hu L, Pan XY, Tang ZH, *et al.* A fast fuzzy clustering algorithm for complex networks via a generalized momentum method. *IEEE Trans Fuzzy Syst* 2022;**30**:3473–85.
46. Hu L, Zhang J, Pan X, *et al.* HiSCF: leveraging higher-order structures for clustering analysis in biological networks. *Bioinformatics* 2021;**37**:542–50.
47. Zhao BW, Hu L, You ZH, *et al.* HINGRL: predicting drug-disease associations with graph representation learning on heterogeneous information networks. *Brief Bioinform* 2022;**23**: bbab515.
48. Yuan J, Wu W, Xie C, *et al.* NPInter v2.0: an updated database of ncRNA interactions. *Nucleic Acids Res* 2014;**42**:D104–8.
49. Bhartiya D, Pal K, Ghosh S, *et al.* lncRNome: a comprehensive knowledgebase of human long noncoding RNAs. *Database (Oxford)* 2013;**2013**:bat034.
50. Bu D, Yu K, Sun S, *et al.* NONCODE v3.0: integrative annotation of long noncoding RNAs. *Nucleic Acids Res* 2012;**40**:D210–5.
51. UniProt C. UniProt: a hub for protein information. *Nucleic Acids Res* 2015;**43**:D204–12.
52. Yang X, Yang S, Li Q, *et al.* Prediction of human-virus protein-protein interactions through a sequence embedding-based machine learning method. *Comput Struct Biotechnol J* 2020;**18**: 153–61.
53. Frankish A, Diekhans M, Jungreis I, *et al.* Gencode 2021. *Nucleic Acids Res* 2021;**49**:D916–23.
54. Kipf TN, Welling M. Variational graph auto-encoders. *arXiv e-prints* 2016; arXiv:1611.07308.
55. Li QM, Wu XM, Liu H *et al.* Label efficient semi-supervised learning via graph filtering. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 9574–83.
56. Xu B, Shen H, Cao Q, *et al.* Graph convolutional networks using heat kernel for semi-supervised learning. 2020; arXiv:2007.16002.
57. Lagier-Tourenne C, Polymenidou M, Hutt KR, *et al.* Divergent roles of ALS-linked proteins FUS/TLS and TDP-43 intersect in processing long pre-mRNAs. *Nat Neurosci* 2022;**15**: 1488–97.
58. Teng X, Chen X, Xue H, *et al.* NPInter v4.0: an integrated database of ncRNA interactions. *Nucleic Acids Res* 2020;**48**:D160–5.