The Author(s) *BMC Bioinformatics* 2017, **18**(Suppl 12):420
DOI 10.1186/s12859-017-1819-1

BMC Bioinformatics

RESEARCH Open Access

Fusing multiple protein-protein similarity networks to effectively predict lncRNA-protein interactions

Xiaoxiong Zheng[1], Yang Wang[5], Kai Tian[1], Jiaogen Zhou[4], Jihong Guan[3], Libo Luo[2] and Shuigeng Zhou[1,2*]

From 12th International Symposium on Bioinformatics Research and Applications (ISBRA 2016)
Minsk, Belarus. 5-8 June 2016

Abstract

Background: Long non-coding RNA (lncRNA) plays important roles in many biological and pathological processes, including transcriptional regulation and gene regulation. As lncRNA interacts with multiple proteins, predicting *lncRNA-protein interactions* (lncRPIs) is an important way to study the functions of lncRNA. Up to now, there have been a few works that exploit *protein-protein interactions* (PPIs) to help the prediction of new lncRPIs.

Results: In this paper, we propose to boost the prediction of lncRPIs by fusing multiple *protein-protein similarity networks* (PPSNs). Concretely, we first construct four PPSNs based on protein sequences, protein domains, protein GO terms and the STRING database respectively, then build a more informative PPSN by fusing these four constructed PPSNs. Finally, we predict new lncRPIs by a random walk method with the fused PPSN and known lncRPIs. Our experimental results show that the new approach outperforms the existing methods.

Conclusion: Fusing multiple protein-protein similarity networks can effectively boost the performance of predicting lncRPIs.

Keywords: lncRNA-Protein Interaction, Random walk, Similarity network fusion

Background

Long non-coding RNAs (lncRNAs in short), one type of non-protein coding transcripts longer than 200 nucleotides, play important roles in complex biological processes, ranging from transcriptional regulation, epigenetic gene regulation to disease identification [1]. Up to date, a number of lncRNAs have been identified, such as HOTAIR [2], MALAT-1 [3] and Xist [4], but most of them are still unknown. Researches have shown that most lncRNAs can exert their functions by interfacing with multiple corresponding RNA binding proteins [5]. Therefore, predicting *lncRNA-protein interactions* (lncRPIs) is an important way to study the functions of lncRNAs.

In the literature, there are more and more works that employ machine learning methods to predict the interactions between RNAs/ncRNAs/lncRNAs and proteins. For example, Muppirala et al. [6] proposed the RPISeq method for identifying *RNA-protein interactions* (RPIs) by using Random Forest (RF) and Support Vector Machine (SVM) classifiers trained with features of protein and RNA sequences. Bellucci et al. [7] proposed the catRAPID method by using a number of physicochemical features, including hydrogen bonding, van der Waals interaction and secondary structure, for predicting lncRPIs. Wang et al. [8] developed an extended Naive Bayes classifier for predicting RPIs using only protein and RNA sequence information. Lu et al. [9] developed the LncPro tool for lncRPI prediction based on Van der Waals propensities, hydrogen bonding and secondary structures extracted from lncRNA-protein pairs.

*Correspondence: sgzhou@fudan.edu.cn
[1]Shanghai Key Lab of Intelligent Information Processing, and School of Computer Science, Fudan University, 220 Handan Road, Shanghai 200433, China
[2]The Bioinformatics Lab at Changzhou NO. 7 People's Hospital, Changzhou, Jiangsu 213011, China
Full list of author information is available at the end of the article

Cheng et al. [10] proposed the PRIPU approach that uses only positive and unlabeled examples to predict RPIs. Recently, Suresh et al. [11] employed SVM to predict ncRNA-protein interactions (ncRPIs) by using the information of sequences and predicted structural peculiarities of proteins and RNAs, and Cheng et al. [12] proposed to boost the performance of protein-RNA interaction prediction by selecting high-quality negative samples.

In addition, there are some works that identify lncRPIs from the perspective of network. That is, to construct networks by using known lncRPIs and PPIs as well as lncRNA-lncRNA interactions. For example, Yang et al. [13] first constructed a heterogeneous network of lncRNAs and proteins, and then employed HeteSim [14] — a pair-wise random walk model that can evaluate n between heterogeneous objects, to evaluate the connection possibilities between lncRNAs and proteins in the network, and thus identify new lncRPIs. However, in their heterogeneous network, PPIs were represented in 0/1 style. That is, if two involved proteins interact, there is an edge between them with weight 1; Otherwise there is no edge. Li et al. [15] proposed the LPIHN method to infer new lncRPIs, which is roughly similar to the method above. They also constructed a heterogenous network of lncRNAs and proteins. But the LPIHN method is different from the method of Yang et al. [13] in three aspects: 1) lncRNA-lncRNA interactions are considered in the heterogenous network; 2) PPIs are represented by protein-protein similarity based on the PPI confidence data from the STRING database; and 3) the random walk with restart model (instead of HeteSim) was used. One common point of these two works is that the only used protein-related information was PPI data

In this paper, to boost the performance of lncRPI prediction we propose to fuse multiple protein-protein similarity networks (PPSNs), and integrate the fused PPSN with known lncRPIs to construct a more informative heterogeneous network, on which new lncRPIs are inferred. Concretely, we first use protein sequences, protein domains, GO terms and the STRING database respectively to construct four PPSNs, then employ the *similarity network fusion* (SNF) algorithm [16] to combine the four PPSNs into a fused PPSN. Following that, a heterogeneous lncRNA-protein network based on the fused PPSN and known lncRPIs is built. Finally, the HeteSim algorithm is performed on the heterogeneous lncRNA-protein network to infer new lncPRIs. Extensive experiments show that our approach outperforms not only the existing method but also those using only one PPSN.

Methods

In this section, we introduce the lncRPI data used in our study and present the details of our method.

Datasets

We extracted Homo sapiens ncRNA-protein interactions from NPInter (v2.0) [17] and Homo sapiens lncRNA from the ncRNA database NONCODE (v4.0) [18]. Then we retrieved lncRNA-protein interactions by manually filtering these interactions not involving lncRNAs. We also gave up the lncRNAs that interacts only one protein, because such interactions cannot be validated by leave-one-out cross validation (LOOCV). Finally, we got an lncRPI dataset that consists 4467 lncRPIs, involving 1050 unique lncRNAs and 84 unique proteins.

Overview of our method

The pipeline of our method is shown in Fig. 1. The rectangles represent lncRNAs and circles represent proteins. On the right side of figure, four squares mean different protein-protein similarity networks (PPSNs) with different similarity metrics. We use Similarity Network Fusion (SNF) algorithm to fuse them to get a more informative PPSN. Then we construct a heterogenous lncRNA-protein network based on known lncRPIs and the fused PPSN. The green solid lines are known lncRPIs and red solid lines are the similarity of proteins. Finally, we perform the HeteSim algorithm on the heterogenous network to predict novel lncRPIs.

As most lncRNAs do not show the same pattern of high interspecies conservation as protein-coding genes [19]. To avoid difficulties caused by low conservation, we predict lncRNA-protein interactions from the perspective of interaction network. However, with a few lncRNAs' crosstalk reported, our lncRNA-protein interaction network considers of only a lncRNA-protein sub-network and a protein-protein sub-network. We adopt the HeteSim [14] to predict novel lncRPIs on the heterogenous network.

Protein-protein similarity computation

Many sources are available to evaluate the similarity between proteins. In this paper, we calculate protein-protein similarity by using protein sequences, protein domains, protein functional annotations (or GO terms), and PPI confidence data from the STRING database v10.0 [20]. As protein sequences, domains and GO terms are different types of data with different biological implications, we employ different methods to compute the similarity between any pair of such data items. The computation results in four similarity matrices, denoted as Seqs, Pfam, Go and STRING, corresponding to four *protein-protein similarity networks* (PPSNs).

Sequence similarity (Seqs)

Protein sequences are obtained from the UniProt database [21]. We compute the sequence similarity between

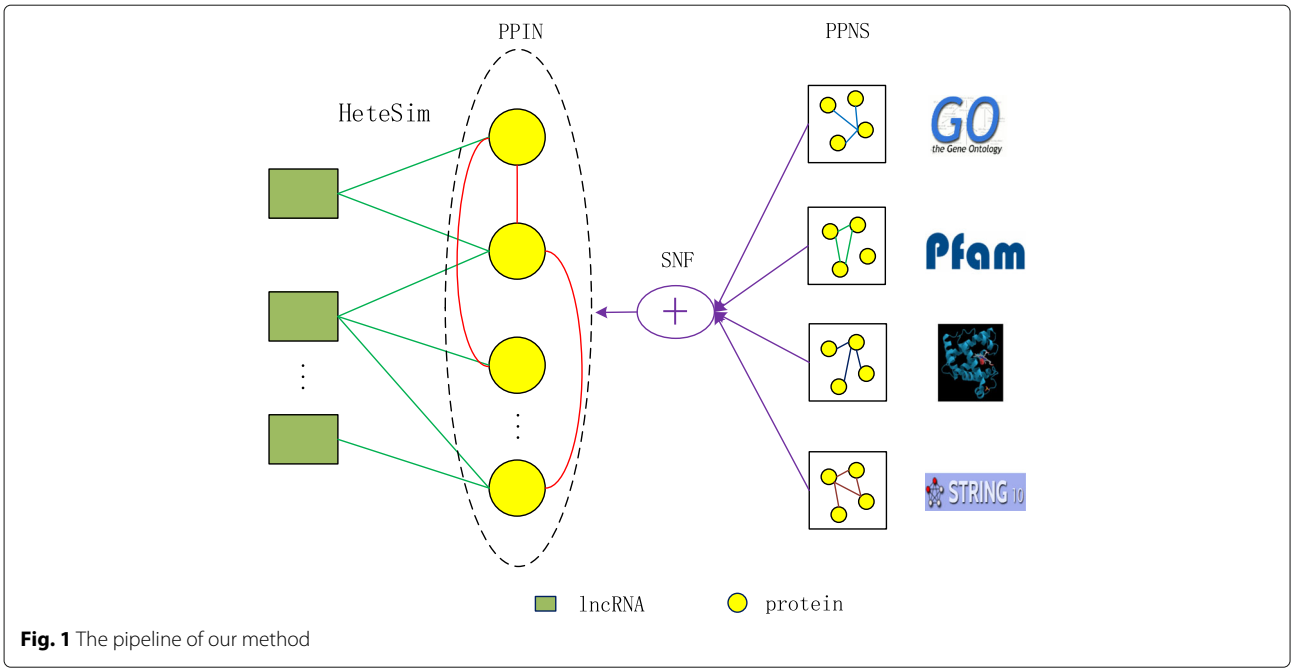

Fig. 1 The pipeline of our method

two proteins using a normalized version of Smith-Waterman score [22]. The normalized Smith-Waterman score between proteins p_i and p_j is $nsw(p_i,p_j) = \frac{sw(p_i,p_j)}{\sqrt{sw(p_i,p_i)}\sqrt{sw(p_j,p_j)}}$ where $sw_{(.,.)}$ means the original Smith-Waterman score. By applying this operation to any protein pair of p_i and p_j, we can obtain their sequence similarity as $SS(p_i,p_j) = (nsw(p_i,p_j) + nsw(p_j,p_i))/2$.

Functional annotation semantic similarity (Go)

GO annotations are downloaded from the GO database [23]. Semantic similarity between any pair of proteins is calculated based on the overlap of the GO terms associated with the two proteins [24]. All three types of GO are used in the computation as similar RNAs are expected to interact with proteins that act in similar biological processes, or have similar molecular functions or reside in similar cell compartments. We compute the Jaccard value [25] with respect to the GO terms of each pair of proteins as their similarity. The Jaccard score between two term sets t_i and t_j of proteins p_i and p_j is defined as $\frac{|t_i \cap t_j|}{|t_i \cup t_j|}$, which is the ratio of the number of common terms between proteins p_i and p_j to the total number of terms of p_i and p_j, which is used as the functional annotation semantic similarity $FS(p_i,p_j)$ of proteins p_i and p_j.

Protein domain similarity (Pfam)

Protein domains are extracted from the Pfam database [26]. Each protein is represented by a domain fingerprint (binary vector) whose elements encode the presence or absence of each retained Pfam domain by 1 or 0, respectively. We compute the Jaccard value of any two proteins p_i and p_j with their domain fingerprints as their similarity $DS(p_i,p_j)$.

STRING similarity (String)

STRING is a database of known and predicted interactions which currently covers 9643763 proteins from 2031 organisms [20]. It provides a confidence score for the interaction of any two interacting proteins, and the highest score is 999. We use the confidence scores to evaluate the similarities between interacting proteins. Formally, for proteins p_i and p_j, their similarity is $String(p_i,p_j) = confidence\_score(p_i,p_j)/999$.

Fusing protein-protein similarity networks

As each protein-protein similarity matrix (network) computed above may contain noise, here we fuse these four matrices (network) to get a more informative and reliable matrix (or network). The similarity network fusion (SNF) algorithm [16] is employed. SNF can derive useful information even from a small number of samples, and is robust to noise and data heterogeneity. It is a nonlinear message-passing based method that iteratively updates each network and makes it more and more similar to the other networks.

A PPSN can be represented as a graph $G = (V,E)$ where $V=\{v_1, v_2, \cdots, v_n\}$ corresponds to the set of proteins in the network and E corresponds to the set of edges, each of which has a similarity weight. We denote the corresponding similarity matrix as $\boldsymbol{W}$ where $\boldsymbol{W}(i,j)$ is the similarity

between proteins v_i and v_j. To compute the fused matrix (network) from the four protein similarity matrices, we define a full and sparse kernel on each matrix. The full kernel is a normalized weight matrix $\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{W}$, $\boldsymbol{D}$ is a diagonal matrix and $\boldsymbol{D}(i,i) = \sum_j \boldsymbol{W}(i,j)$. To avoid numerical instability since $\boldsymbol{P}$ involves self-similarities on the diagonal entries of $\boldsymbol{W}$, a better normalization is as follows [16]:

$$\boldsymbol{P}(i,j) = \begin{cases} \dfrac{\boldsymbol{W}(i,j)}{2\sum_{k\neq i}\boldsymbol{W}(i,k)}, j \neq i \\ 1/2, j = i \end{cases} \quad (1)$$

We denote protein v_i's neighbours as N_i and use k nearest neighbors (kNN) to measure the local affinity as follows:

$$\boldsymbol{S}(i,j) = \begin{cases} \dfrac{\boldsymbol{W}(i,j)}{\sum_{k\in N_i}\boldsymbol{W}(i,k)}, j \in N_i \\ 0, otherwise \end{cases} \quad (2)$$

We think that the similarities between a protein and its neighbours are more reliable than the similarities between the protein with remote ones. Through graph diffusion, the similarities can be disseminated to remote proteins. Matrix $\boldsymbol{P}$ carries all information of the protein-protein similarity network and $\boldsymbol{S}$ carries local similarity information of the network. Then, we can do iterative computation as follows:

$$\boldsymbol{P}_t^{(i)} = \boldsymbol{S}^{(i)} \times \left(\frac{\sum_{k\neq i}\boldsymbol{P}_{t-1}^{(k)}}{m-1}\right) \times (\boldsymbol{S}^{(i)})^T, i = 1,2,3,4, \quad (3)$$

where $\boldsymbol{P}_t^{(i)}$ is the i^{th} similarity matrix (network) after t ($\geq$0) iterations, $\boldsymbol{S}^{(i)}$ is the kNN matrix of the i^{th} similarity matrix (network). m is the number of PPSNs used, here m=4. As $\boldsymbol{S}$ is the kNN neighbour matrix of $\boldsymbol{P}$, it contains the most important information of $\boldsymbol{P}$ and also alleviates the noise effect in $\boldsymbol{P}$. At each iteration, each similarity matrix (network) can get reliable information from the other similarity matrices (networks) and also updates itself with the other similarity matrices (networks). After t iterations, the fused matrix (network) is computed as follows:

$$\boldsymbol{P} = \left(\sum_{i=1}^{m}\boldsymbol{P}_t^{(i)}\right)/m. \quad (4)$$

Note that we normalize matrix $\boldsymbol{P}_t$ after each iteration to ensure the matrix is full rank and each protein is more similar to itself than the other proteins.

Evaluating relevance score in a lncrna-protein network

With the known lncRPIs and the fused protein-protein similarity network, we build a lncRNA-protein heterogenous network, on which a random walk model HeteSim [14] is employed to infer new lncRPIs. HeteSim is to evaluate the relevance between a pair of lncRNA and protein, and a large relevance score means a high possibility that the lncRNA and protein interacts.

Given a schema $S = (A,R)$ where A is a set of object types and R is a set of relationships. A lncRNA-protein network is defined as a directed graph $\boldsymbol{G} = (\boldsymbol{V},\boldsymbol{E})$ with an object-type mapping function $\phi : \boldsymbol{V} \rightarrow A$ and a edge-relationship mapping function $\psi : \boldsymbol{E} \rightarrow R$. Each object $v \in \boldsymbol{V}$ belongs to one particular object type $\phi(v) \in A$, and each link $e \in \boldsymbol{E}$ belongs to a particular relationship $\psi(e) \in R$. The schema S depicts the object types and the relationships existing among object types. For example, a relationship existing from type A to type B, denoted as $A \xrightarrow{R} B$, A and B are termed the source type and target type of relationship R. In this paper, there are two object types: lncRNA and protein, and three possible relationships: lncRNA-protein, protein-protein, and lncRNA-lncRNA. Here, we consider only the former two relationships. An object may be a concrete protein or lncRNA, and two objects can be connected via different paths that have different meanings.

In the heterogeneous network, a *relevance path* along a sequence of object types $A_1, A_2, \cdots, A_{l+1}$ can be denoted as $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$, the *composite relationship* between A_1 and A_{l+1} is denoted as $R = R_1 \circ R_2 \circ \cdots \circ R_l$ where $\circ$ denotes the relationship between two object types. For two objects o_1 and o_2 with a composite relationship $R = R_1 \circ R_2 \circ \cdots \circ R_l$, HeteSim iteratively evaluates the relevance score between them as follows:

$$HeteSim(o_1,o_2|R_1 \circ R_2 \circ \cdots \circ R_l) = \frac{1}{|O(o_1|R_1)||I(o_2|R_l)|} \sum_{i=1}^{|O(o_1|R_1)|}\sum_{j=1}^{|I(o_2|R_l)|} HeteSim(O_i(o_1|R_1), I_j(o_2|R_l)|R_2 \circ \cdots \circ R_{l-1}), \quad (5)$$

where $O(o_1|R_1)$ is the out-neighbours of o_1 based on relationship R_1, $I(o_2|R_l)$ is the in-neighbours of o_2 based on relationship R_l, $O_i(o_1|R_1)$ / $I_j(o_2|R_l)$ indicate the i^th / j^th object in $O(o_1|R_1)$ / $I(o_2|R_l)$, $|\cdot|$ means the size of a set.

As we consider only *lncRNA-protein relationship* (*lp* in short) and *protein-protein relationship* (*pp* in short), so we have

$$HeteSim(lncRNA_i, p_j|lp) = \begin{cases} 1 & \text{if they interact with each other;} \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

$$HeteSim(p_i,p_j|pp) = sim(p_i,p_j). \quad (7)$$

For relationship $A \xrightarrow{R} B$, we define U_{AB} is a normalized adjacent matrix along the row vector between type

A and type B based on relationship R. What is more, V_{AB} is the normalized matrix along the column vector, which is the transition probability matrix of $B \rightarrow A$ based on inverse relationship R^{-1}. So, we can get $U_{AB} = V'_{BA}$ and $V_{AB} = U'_{BA}$ [14], where V'_{BA} is the transpose of V_{BA}. Given a relevance path $P = A_1A_2 \cdots A_{l+1}$. The reachable probability matrix PM for path P is defined as $PM_P = U_{A_1A_2}U_{A_2A_3} \cdots U_{A_lA_{l+1}}$ (PM in short). $PM(i,j)$ represents the probability of object $i \in A_1$ reaching object $j \in A_{l+1}$ based on path P. So the relevance between objects in A_1 and A_{l+1} based on the relevance path P is:

$$\begin{aligned} &HeteSim(A_1, A_{l+1}|P) \\ &\quad = HeteSim(A_1, A_{l+1}|P_LP_R) \\ &\quad = U_{A_1A_2} \cdots U_{A_{mid-1}M} V_{MA_{mid+1}} \cdots V_{A_lA_{l+1}} \\ &\quad = U_{A_1A_2} \cdots U_{A_{mid-1}M} U'_{A_{mid+1}A_M} \cdots U'_{A_{l+1}A_l} \\ &\quad = U_{A_1A_2} \cdots U_{A_{mid-1}M} (U_{A_{l+1}A_l} \cdots U_{A_{mid+1}A_M})' \\ &\quad = PM_{P_L} PM'_{P_R^{-1}} \end{aligned} \tag{8}$$

where M is the middle position node type of A_1 and A_{l+1}. So above equation shows the inner product of matrices of two probability distributions that A_1 reaches M and A_{l+1} reaches M.

For two instances o_1 and o_2 of type A_1 and type A_{l+1}, we can get their normalized relevance score:

$$HeteSim(o_1, o_2|P) = \frac{PM_{P_L}(o_1,:)\, PM'_{P_R^{-1}}(o_2,:)}{\sqrt{\|PM_{P_L}(o_1,:)\| \|PM'_{P_R^{-1}}(o_2,:)\|}} \tag{9}$$

where $PM_{P_L}(o_1,:)$ is the row that object o_1 lies in the matrix PM_{P_L} [14].

Results and discussion

In our experiments, the leave-one-out cross validation (LOOCV) is used to evaluate the proposed method. The baseline is the method proposed by Yang et al. [13] where PPIs were modeled as a binary network. As for our method, there are 15 settings: 4 settings of using only one similarity matrix, Seqs, Pfam, Go and STRING, respectively; 6 settings of fusing two similarity matrices, corresponding to Seqs+Pfam, Seqs+Go, Seqs+String, Pfam+Go, Pfam+String, and Go+String; 4 settings of fusing three similarity matrices, including Seqs+Pfam+Go, Seqs+Pfam+String, Seqs+Go+String, and Pfam+Go+String; and 1 setting of fusing the four similarity matrices, i.e., Seqs+Pfam+Go+String. For reference simplicity, we denote the baseline method as *Binary*, and denote our method under different settings by the setting names, such Seqs, Seqs+Pfam, Seqs+Pfam+Go, Seqs+Pfam+Go+String, etc. Actually, the String case of our method is roughly similar to the LPIHM method [15]. So essentially we compare our method with both the method in [13] and the LPHIM method in [15]. Because HeteSim is a path-constrained relevance measure, the selection of path is very important. In Yang et al.'s work, they chose *lncRNA-protein-protein* (LPP) as their relevance path and achieved better performance than other pathes. In our work, we also choose it as the relevance path.

To evaluate the prediction performance, the *receiver operating characteristic* (ROC) curve is generated for each experimental setting, and AUC (the area under the ROC curve) is calculated, which is widely used in assessing prediction performance and its value falls between 0 and 1. The maximum value 1 means a perfect prediction, and 0.5 means a random guess.

We first compare our method with only one similarity matrix to the baseline method *Binary*, the results are presented in Fig.2. The black solid line is the ROC curve of *Binary* and the other colored lines are the ROC curves of our method with different similarity matrices. In Fig.2, we can see that the performance of *String* is better than that of *Binary*, which shows that weighted PPI network is more helpful than binary PPI network. Moreover, the results of *Go*, *Pfam* and *Seqs* are all better than that of *String*. This may be because String is less reliable than the other similarity networks.

We then compare the performance of our method under different experimental settings, the results are shown in Figs. 3, 4 and 5.

First, we consider the cases of fusing two different similarity matrices, their corresponding ROC curves are presented in Fig. 3. Each color curve indicates the ROC curve of our method of fusing two specific similarity

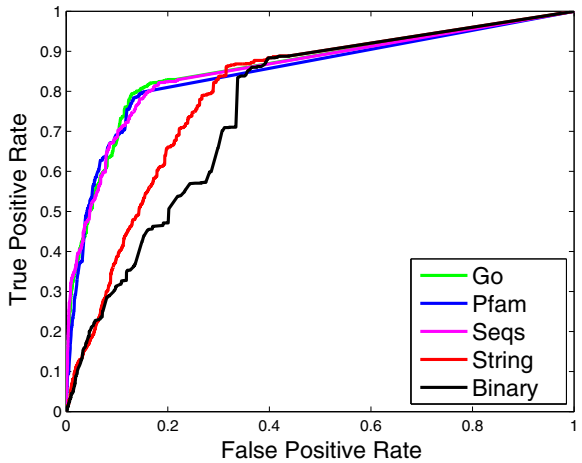

Fig. 2 ROC curves of Binary and our method using only one of different similarity matrices

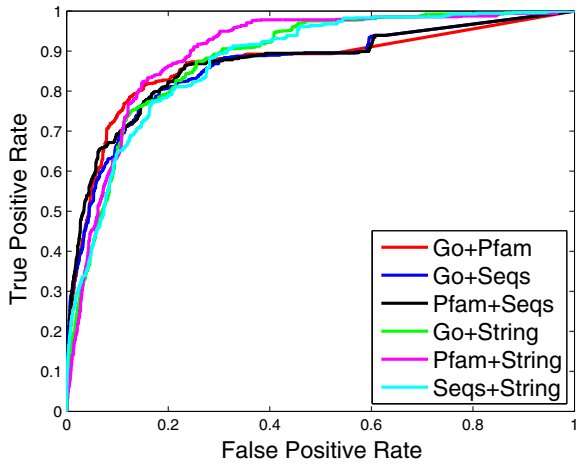

Fig. 3 ROC curves of our method when fusing two different matrices

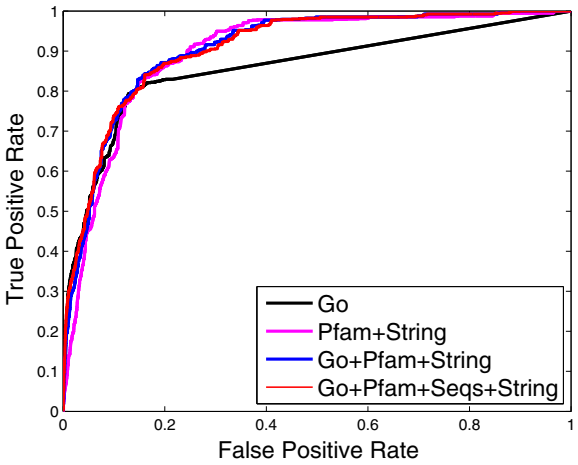

Fig. 5 ROC curves of the best performance when using one matrix and fusing two, three and four matrices, respectively

matrices. We can see that fusing the String similarity matrix with any other similarity matrix can achieve better performance than fusing any other two similarity matrices. This may be because STRING database has many PPIs that contain much complementary information to the other similarity matrices. We achieve the best performance when fusing the Pfam similarity matrix and the String similarity matrix.

Then, we check the cases that fuse three different similarity matrices, their ROC curves are shown in Fig. 4. We get the best performance when fusing Go, Pfam and String similarity matrices. Similar to Fig. 3, we can see that the performance when fusing the String similarity matrix with any two other similarity matrices is better than that of Go+Pfam+Seqs.

Finally, we consider the case that fuses all four similarity matrices, and present its ROC curve in Fig. 5. For the convenience of comparison, in Fig. 5 we also plot the best results when using one similarity matrix, and fusing two and three different similarity matrices. It can be seen that the performance of Go+Pfam+Seqs+String is better than the performance of the other settings (Go, Pfam+String, Go+Pfam+String). This illustrates that network fusion can really extract complementary information from different networks to achieve better prediction performance.

To more clearly compare the baseline method *Binary* and our method under different settings, we give all of their AUC values in Fig. 6. Here, the bars of similar color means using the same number of similarity matrices. We can see that 1) all the AUC values of our method under different settings are larger than that of using a binary PPI network; 2) As more matrices are fused, the AUC value becomes larger. For example, the AUC value of Go+Pfam+String is 0.9066, which is bigger than the AUC values of Go+Pfam, Go+String and Pfam+String. And when fusing all the four matrices, the corresponding AUC value is the largest (0.9068). This shows that by fusing multiple matrices we can get a more reliable and informative matrix or network.

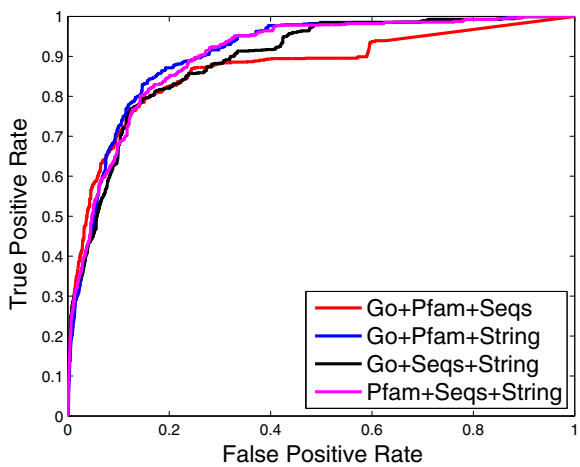

Fig. 4 ROC curves of our method when fusing three different matrices

Conclusion

In this paper, we proposed a new approach to predicting lncRPIs by fusing four protein-protein similarity networks, which were computed with protein sequences, protein domains, protein functional annotations of GO, and the PPI confidence scores from the STRING database. The similarity network fusion (SNF) algorithm and the random walk on heterogeneous network model HeteSim were employed. Our experimental results show that the proposed method outperforms the existing method and

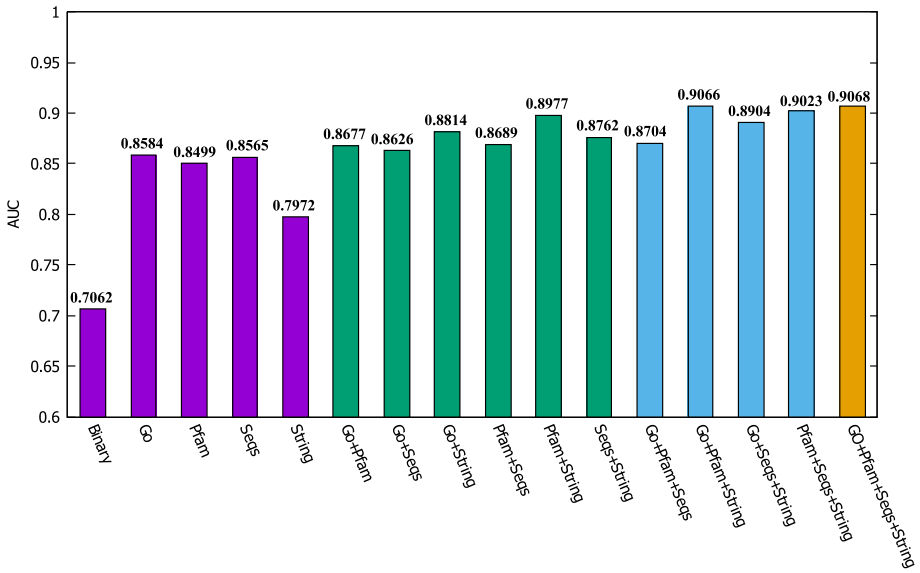

Fig. 6 The AUC values of the baseline method and our method under different settings

those cases when using only one protein-protein similarity network. For future work, on the one hand, we will explore other advanced network fusion methods to fuse more available data sources for further boosting the performance of lncRPI prediction; On the other hand, we will include the lncRNA-lncRNA interactions into the prediction procedure.

Acknowledgments
A 2-page abstract has been published in Lecture Notes in Computer Science (LNCS): Bioinformatics Research and Applications.

Funding
National Natural Science Foundation of China (NSFC) (grant No. 61272380) for data collection, manuscript writing, and publication cost; The National Key Research and Development Program of China (grant No. 2016YFC0901704) for data collection and analysis; NSFC (No. 61672113) and the Program of Shanghai Subject Chief Scientist (15XD1503600) for manuscript writing. No funding body played any role in design/conclusion.

Availability of data and materials
The datasets used and/or analysed during the current study are available at http://admis.fudan.edu.cn/projects/pipi.html.

Authors' contributions
SG and JH designed the research, analyzed the results and revised the manuscript. XX developed the algorithms, carried out experiments, and drafted the manuscript. JG and LB were involved in data analysis and revising the paper. YW and KT prepared data and coded some of the algorithms. All authors read and approved the final manuscript.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Author details
[1]Shanghai Key Lab of Intelligent Information Processing, and School of Computer Science, Fudan University, 220 Handan Road, Shanghai 200433, China. [2]The Bioinformatics Lab at Changzhou NO. 7 People's Hospital, Changzhou, Jiangsu 213011, China. [3]Department of Computer Science and Technology, Tongji University, 4800 Cao'an Road, Shanghai 201804, China. [4]The institute of subtropical Agriculture, China Academy of Sciences, 444 Yuandaer Road, Mapoling, Changsha 410125, China. [5]School of Software, Jiangxi Normal University, 99 Ziyang Avenue, Nanchang 330022, China.

Published: 16 October 2017

References

1. Ponting CP, Oliver PL, Reik W. Evolution and functions of long noncoding RNAs. Cell. 2009;136(4):629–41.
2. Rinn JL, Kertesz M, Wang JK, Squazzo SL, Xu X, Brugmann SA, et al. Functional demarcation of active and silent chromatin domains in human HOX loci by noncoding RNAs. Cell. 2007;129(7):1311–23.
3. Tripathi V, Ellis JD, Shen Z, Song DY, Pan Q, Watt AT, et al. The nuclear-retained noncoding RNA MALAT1 regulates alternative splicing by modulating SR splicing factor phosphorylation. Mol Cell. 2010;39(6):925–38.
4. Kohlmaier A, Savarese F, Lachner M, Martens J, Jenuwein T, Wutz A. A chromosomal memory triggered by Xist regulates histone methylation in X inactivation. PLoS Biol. 2004;2(7):e171.
5. Mercer TR, Dinger ME, Mattick JS. Long non-coding RNAs: insights into functions. Nat Rev Genet. 2009;10(3):155–9.
6. Muppirala UK, Honavar VG, Dobbs D. Predicting RNA-protein interactions using only sequence information. BMC Bioinforma. 2011;12(1):1.
7. Bellucci M, Agostini F, Masin M, Tartaglia GG. Predicting protein associations with long noncoding RNAs. Nat Methods. 2011;8(6):444–5.
8. Wang Y, Chen X, Liu ZP, Huang Q, Wang Y, Xu D, et al. De novo prediction of RNA–protein interactions from sequence information. Mol BioSyst. 2013;9(1):133–42.
9. Lu Q, Ren S, Lu M, Zhang Y, Zhu D, Zhang X, et al. Computational prediction of associations between long non-coding RNAs and proteins. BMC Genomics. 2013;14(1):1.

10. Cheng Z, Zhou S, Guan J. Computationally predicting protein-RNA interactions using only positive and unlabeled examples. J Bioinforma Comput Biol. 2015;13(03):1541005.
11. Suresh V, Liu L, Adjeroh D, Zhou X. RPI-Pred: predicting ncRNA-protein interaction using sequence and structural information. Nucleic Acids Res. 2015;43(3):1370–9.
12. Cheng Z, Huang K, Wang Y, Liu H, Guan J, Zhou S. Selecting high-quality negative samples for effectively predicting protein-RNA interactions. BMC Syst Biol. 2017;11(S-2):9:1–9:11.
13. Yang J, Li A, Ge M, Wang M. Prediction of interactions between lncRNA and protein by using relevance search in a heterogeneous lncRNA-protein network. In: Proceedings of 34th, Chinese Control Conference (CCC'15). New York: IEEE; 2015. p. 8540–4.
14. Shi C, Kong X, Huang Y, Yu PS, Wu B. HeteSim: A General Framework for Relevance Measure in Heterogeneous Networks. IEEE Trans Knowl Data Eng. 2014;26(10):2479–92.
15. Li A, Ge M, Zhang Y, Peng C, Wang M. Predicting Long Noncoding RNA and Protein Interactions Using Heterogeneous Network Model. BioMed Res Int. 2015;2015:671950.
16. Wang B, Mezlini AM, Demir F, Fiume M, Tu Z, Brudno M, et al. Similarity network fusion for aggregating data types on a genomic scale. Nat Methods. 2014;11(3):333–7.
17. Yuan J, Wu W, Xie C, Zhao G, Zhao Y, Chen R. NPInter v2.0: an updated database of ncRNA interactions. Nucleic Acids Res. 2014;42(D1): D104—D108.
18. Xie C, Yuan J, Li H, Li M, Zhao G, Bu D, et al. NONCODEv4: exploring the world of long non-coding RNA genes. Nucleic Acids Res. 2014;42(D1): D98—D103.
19. Johnsson P, Lipovich L, Grandér D, Morris KV. Evolutionary conservation of long non-coding RNAs; sequence, structure, function. Biochim Biophys Acta (BBA) Gen Subj. 2014;1840(3):1063–71.
20. Szklarczyk D, Franceschini A, Wyder S, Forslund K, Heller D, Huerta-Cepas J, et al. STRING v10: protein–protein interaction networks, integrated over the tree of life. Nucleic Acids Res. 2015;43(Database issue):D447–52.
21. Consortium TU. Update on activities at the Universal Protein Resource (UniProt) in 2013. Nucleic Acids Res. 2013;41(D1):D43—D47.
22. Smith TF, Waterman MS. Identification of common molecular subsequences. J Mol Biol. 1981;147(1):195–7.
23. Consortium GO, et al. Gene Ontology annotations and resources. Nucleic Acids Res. 2013;41(D1):D530—D535.
24. Wu X, Pang E, Lin K, Pei ZM. Improving the Measurement of Semantic Similarity between Gene Ontology Terms and Gene Products: Insights from an Edge- and IC-Based Hybrid Method. PLoS ONE. 2013;05;8(5): e66745.
25. JACQUART P. Nouvelles recherches sur la distribution florale. Bull Soc Vand Sci Nat. 1908;0:44.
26. Finn RD, Bateman A, Clements J, Coggil P, Eberhardt RY, Eddy SR, Heger A, Hetherington K, Holm L, Mistry J. Pfam: the protein families database. Nucleic Acids Res. 2014;42(D1):D222–D30.