CS440 Project Description: RNA-RNA Interaction Prediction

Ali Ebrahimpour Boroojeny and Hamidreza Chitsaz

Department of Computer Science, Colorado State University

http://chitsazlab.org
chitsaz@chitsazlab.org

1 Introduction

In this project you will use machine learning methods to predict RNA-RNA interaction (RRI) from the sequencing-based training data. We use the RISE database [1] as our training and test data. Each row of the database contains a pair of RNAs that supposedly interact with each other. There are many other columns in this dataset for which you can find the information in this website. However, in this dataset, there is no information about the exact transcript of the gene that was involved in the experiment; there is only the gene Ensembl ID and for some rows, only the subsequence that was involved in the interaction. Therefore, we took the gene sequence into consideration, and throughout the rest of this paper, RNA and gene will be used interchangeably. We actually used the gene sequences. The dataset has 14 columns which includes various features such as Ensembl ID of the RNAs, strand of each RNA, publication ID of the experiment, and type of each RNA, etc. The Ensembl ID of each gene was used to download the corresponding nucleotide sequence; for this task, we used the API that Ensembl provides in the Perl language. According to the documentation (which we manually checked for some randomly chosen RNAs), that API returns the sequence on the forward strand, no matter what the actual strand is. Hence, if RNA resides on the reverse strand, it returns the reverse complement of the RNA sequence. To provide the missing data here, Ensembl also provides a flag for the strand which is supposed to be 1 for the forward strand and -1 for the reverse strand. However, as of now the flag feature was not working correctly and always returned 1, even for the genes on the reverse strand. Because the strand information is missing for several rows of the RISE dataset, and we cannot trust the strand flag of the Ensembl API at the time, we use the fact that most of the RNAs occur multiple times in the dataset and in some rows they have the missing features in some other rows. In this way, many missing fields were completed especially the missing strands of those RNAs that had this information at some point in the dataset, provided that they did not appear with contradicting different strands at different rows. We ignored the rows for which we could not find the RNA ID or strand information and also those RNAs that Ensembl API returned "Do not exist in the database" for their sequences. The lengths of RNA sequences present in the final human dataset had a very large range, from 40 nt to 2,473,537 nt. In Fig. the complete histogram of the RNA lengths is shown. The long tail of the histogram is clear in that figure.

The tasks and approaches for solving this problem has been divided to several sections. You may use one of the sections as your project.

2 Methods

Training the network has to be done on the training set which contains 90% of the dataset (dataset has to be split in an stratified manner). Then to choose the best structure and parameters for the neural network, the trained model is evaluated on the evaluation set which contains 6% of the dataset. Once the best model is chosen, the accuracy of the model on classifying the test set, which contains 4% of the data, has to be reported in a table. You have to pick one of the following approaches for your project.

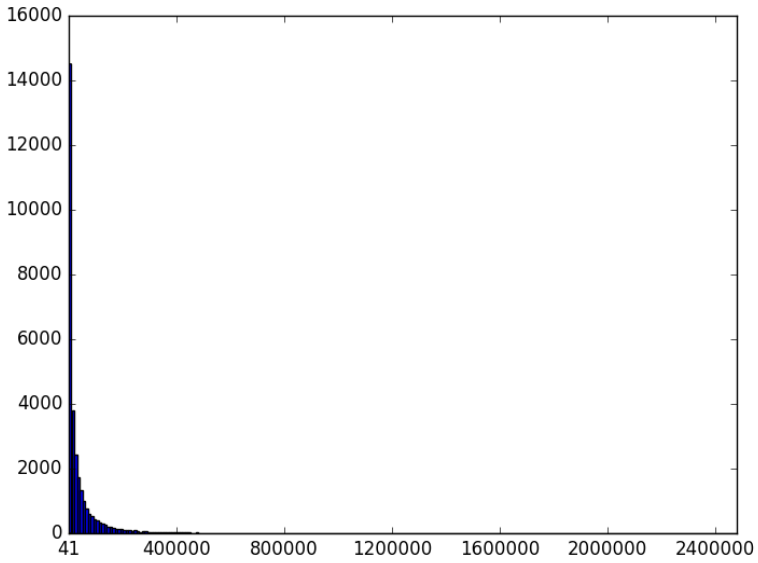

Figure 1: Complete histogram of the lengths of RNAs in the human dataset.

2.1 Vector Representation Using the K-mer Composition

To get such vector representation of each input sequence, two different methods can be used. Both of them resemble the notion of "Bag of Words" which is a common method in Natural Language Processing. In the first representation, each sequence is mapped to an integer vector of the size of all possible k-mers, with entries equal to the number of occurrences of the corresponding k-mer in the input sequence. The second representation is similar to the first one but is a $0-1$ binary vector which only represents whether a k-mer exists in the sequence. Very large and very small RNAs introduced some complexity and probably some noise to the dataset with both representations. So we should also use methods for normalizing the numbers in each vector. Another way of dealing with this problem is to limit the dataset to the ones whose size are less than a threshold. For example, if we only consider genes with length up to 40k nt, it provides a much shorter range of lengths while still includes more than 72% of the genes and most of the samples.

Once the vector representation of the samples are prepared for each RNA, you may generate a vector representation for each sample of the dataset by concatenating the corresponding vector of its two constituents RNAs. These inputs can then be provided to a simple fully connected neural network for classification.

Once you are done with these steps, there is another experiment that you should implement. For each k-mer, we want to consider the 4 k-mers that have the same last $k-1$ characters as their first $k-1$ ones and a new character as the last one. We want to also consider the 4 k-mers that share the first $k-1$ of the original k-mer as their last $k-1$ ones and have another character as their first character. As an example, when $k=3$, for the k-mer ACG we want to consider the eight k-mers: $CGA, CGC, CGG, CGT, AAC, CAC, GAC, TAC$. Finally, We want to add a 1D convolutional layer with filters of size 9 to our fully connected network. Each of these filtered will be applied to each set of 9 k-mers.

2.2 Convolutional Neural Nets and Recurrent Neural Nets

First, read the DeepBind paper [2].

For interaction of a pair of RNAs, the convolution filters and computations can be changed as shown in Figures 2 and 3.

In this project you can also download and use the codes of the DeepBind project.

To work on this project, you will be not only predicting if two RNAs interact, but also finding the section at which the interaction occurs. Therefore, you will also need to know the interaction site of each pair of

$$S = \left[\begin{array}{cccc|cccc} .25 & .25 & .25 & .25 & .25 & .25 & .25 & .25 \\ .25 & .25 & .25 & .25 & .25 & .25 & .25 & .25 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ .25 & .25 & .25 & .25 & .25 & .25 & .25 & .25 \\ .25 & .25 & .25 & .25 & .25 & .25 & .25 & .25 \end{array}\right]$$

Figure 2: a motif of length 4 in cases of having 2 sequences.

$$X_{i,k} = \sum_{j=1}^{m} \sum_{l=1}^{4} \left(S_{i+j,l} M_{k,j,l} + S'_{i+j,l} M_{k,j,l+4} \right) w_{l,l+4}$$

Figure 3: The resulting equation in case of having 2 sequences.

RNAs. We will provide you with this information by giving you the start and the end index of the interaction windows.

2.3 Recurrent Neural Network

For this project you will use the representations of the papers [3, 4] to apply these methods for a pair of sequences.

2.4 Graph Convolutional Networks

In this section, like the previous one, you will be predicting the region in which the interaction occurs. You have to consider windows of limited size and in that region consider each nucleotide as a node of a graph. Then, connect each node to the ones that it can pair with. If you choose a constant size for the windows, then you will have a graph for each pair of windows from each pair of RNAs. By using the information about the interacting windows and considering them as the positive samples and considering the rest of the pairs of the windows as negative lengths, you can use Graph Convolutional Networks to perform learning on the training data and make the predictions on the windows of the test samples.

References

[1] J. Gong, D. Shao, K. Xu, Z. Lu, Z. J. Lu, Y. T. Yang, and Q. C. Zhang. RISE: a database of RNA interactome from sequencing experiments. *Nucleic Acids Res.*, Oct 2017.

[2] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33(8):831, 2015.

[3] Zhen Shen, Wenzheng Bao, and De-Shuang Huang. Recurrent neural network for predicting transcription factor binding sites. *Scientific reports*, 8(1):15270, 2018.

[4] Xiaoyong Pan, Peter Rijnbeek, Junchi Yan, and Hong-Bin Shen. Prediction of rna-protein sequence and structure binding preferences using deep convolutional and recurrent neural networks. *BMC genomics*, 19(1):511, 2018.