Predicting RNAcompete binding from RNA bind-n-seq data

**Eden Mironi**

**Moria Rais Alon**

## 

## Introduction

RNAcompete and RNA Bind-n-seq are two pivotal techniques in studying RNA-protein interactions. RNAcompete, a high-throughput platform, profiles the binding preferences of RNA sequences to a diverse set of RNA-binding proteins. On the other hand, RNA Bind-n-seq delves into the binding landscapes between RNA molecules and proteins, generating a wealth of experimental data on binding affinities and patterns.

RNA motifs are conserved patterns of nucleotides that play a significant role in the recognition and interaction between RNA molecules and proteins. The presence of specific motifs within RNA sequences can provide crucial insights into the binding preferences of RNA-binding proteins and aid in the interpretation of binding affinities. The Predicting compete Binding from RNA Bind-n-seq Data project recognizes the significance of motifs as essential determinants of binding interactions and aims to integrate motif analysis into our predictive modeling approach.

## Goal

The primary objective of this project is to develop a binding model for each RBP, which will be subsequently utilized to score RNAcompete probes binding intensities. The program is designed to provide binding intensities for the RNAcompete probes in the testing data.

## Datasets

1. **RBNS\_training**  
   This file comprises 240,000 RNA sequences.   
   The file format is as follows: TTAGTGTAAGATCGATAAAT 1   
   Each file is labeled with the RBNS number and the corresponding concentration of the RBNS in the file. Greater condensation indicates a higher likelihood of possessing more binding regions. The numeric value at the end of each sequence indicates the frequency of the binding sequence, although it is currently not in use.
2. **RBNS\_test**   
   Similar to RBNS\_training, this dataset is intended for testing purposes.
3. **RNAcompete\_sequences**  
   This file consists of 240,000 lines of sequences that the protein can potentially bind to.
4. **RNCMPT\_training**   
   The files within this folder contain intensity values corresponding to the RNA sequences that the protein binds to. Each line contains a numerical value, such as 0.116567456256324, which can be positive or negative.

## Model description

What Have We Done?   
In broad terms, we constructed the training process around multiclassification. This entails the initial classification of sequence concentrations following one-hot encoding of the sequences.

Following the classification step, we employed an aggregation function to compute the intensity of each RBP. The resulting intensity values guided us in calculating the Pearson value, achieved through the comparison with the actual values provided in the RNCMPT\_training folder.

### CNN Model Structure

The create\_model function constructs a convolutional neural network (CNN) for multiclassification. It takes parameters like the number of output classes (num\_classes), lists of kernel sizes and numbers of kernels for convolutional layers, and the configuration of dense layers (dense\_layers). The function first creates input layers for each kernel size, followed by convolutional layers with ReLU activations and subsequent global max-pooling layers. Normalized kernel outputs are concatenated, and dense layers with ReLU activation are added based on the specified *`dense\_layers`*. The final output layer employs sigmoid activation for predicted outputs. The model is compiled using the Keras functional API, and its architecture is printed. This adaptable function provides a way to create CNN architectures tailored to different multiclassification tasks.

We divided the training and test data to be in relation to 80%/20%.

### Calculation methods

For predicting the intensity values we used the aggregation formula that was suggested by “DeepSELEX: inferring DNA-binding preferences from HT-SELEX data using multi-class CNNs”[1]:

max(cycle4)+max(cycle3)−min(cycle0).

The output of the multi-classification model is as of the concentrations filtered by the file name.

## Results

After many methods and changes, we received the following results:

| 1 | 0.101917742 |
| --- | --- |
| 2 | 0.102205556 |
| 3 | -0.021017695 |
| 4 | -0.015373073 |
| 5 | 0.031388915 |
| 6 | -0.004193626 |
| 7 | -0.036843609 |
| 8 | -0.00232283 |
| 9 | -0.090984041 |
| 10 | 0.061757221 |
| 11 | 0.078534531 |
| 12 | -0.072480661 |
| 13 | 0.256018564 |
| Average | 0.029892846 |

## Performance

Our development environment was Google Collab to develop the code. We received the following metrics:

We used `psutil` library to calculate the performance values.

Run time: ~4-7 minutes

The run time and calculation were pretty short.

Memory usage: ~2645.875 MB

CPU usage: 7.9%

## Conclusions

We attempted several approaches for the training process, including changing the model's structural parameters like batch size and the activation function for the activation layers, for instance. The relu activation function yielded the best results with the optimal values.

Initially, we experimented with padding sequences instead of using sliding windows. However, the approach that proved most effective for us was the utilization of sliding windows. (We have retained the code in case we decide to explore a different approach in the future.)

We are determined to investigate the reasons behind the results we obtained, even though we closely adhered to the core concepts presented during the course.

## References

1. Maor Asif, Yaron Orenstein. (2020). DeepSELEX: inferring DNA-binding preferences from HT-SELEX data using multi-class CNNs. <https://academic.oup.com/bioinformatics/article/36/Supplement_2/i634/6055905>
2. Ilan Ben-Bassat, Benny Chor, Yaron Orenstein. (2018). A deep neural network approach for learning intrinsic protein-RNA binding preferences. <https://academic.oup.com/bioinformatics/article/34/17/i638/5093226>