Machine Learning Engineer Nanodegree

Capstone Project

LeeXA
20th December 2017

I. Definition

a. Project Background (Biochemistry)

All biology depends on nucleotide and amino acid sequences which spontaneously fold to form 3D atomic biochemical structures, with the former relatively less studied. (Wan et al. 2011). However, RNA structure has increasingly been found to play a significant role in regulating cellular activities where two examples include RNA polymerase, riboswitches whose function is to make a RNA copy of DNA and sense metabolic molecules respectively (Wan et al. 2011) and there is a need to study RNA structure.

The way RNA forms structure is through physical interactions (Van der Waals interaction, hydrogen bonds, ionic pairing) (Lounnas et al. 2013), an RNA sequence is able to spontaneously fold to its structure in the cell. These structures can be experimentally determined if one packs them into a crystal and subsequently resolved by X-ray diffraction. The experimental structure is subsequently deposited into the Protein Data Bank (https://www.rcsb.org/pdb/home/home.do).

Among the 100 thousand experimental structures deposited into the Protein Data Bank, only three thousand contains RNA -- RNA is inherently flexible and difficult to resolve to a good resolution (Ke 2004). There is now a need to develop of various computational tools to predict RNA tertiary structure, where notable developments include statistical mechanics and machine translation methods (Das and Baker 2007; Popenda et al. 2012). It must be noted that RNA contains at least 20 * length atoms and the 3D space to explore when predicting an RNA structure is not a trivial task.

One area at least to my knowledge not attempted is deep learning, despite having successfully applied to a similar class of biological molecules – proteins which are largely analogous to RNA in structure and function (Wang et al. 2017). It is hoped that deep learning can be used to help predict RNA structure or least reduce the complexity of RNA structure problem for other algorithms.

II. Problem Statement

a. Output of model

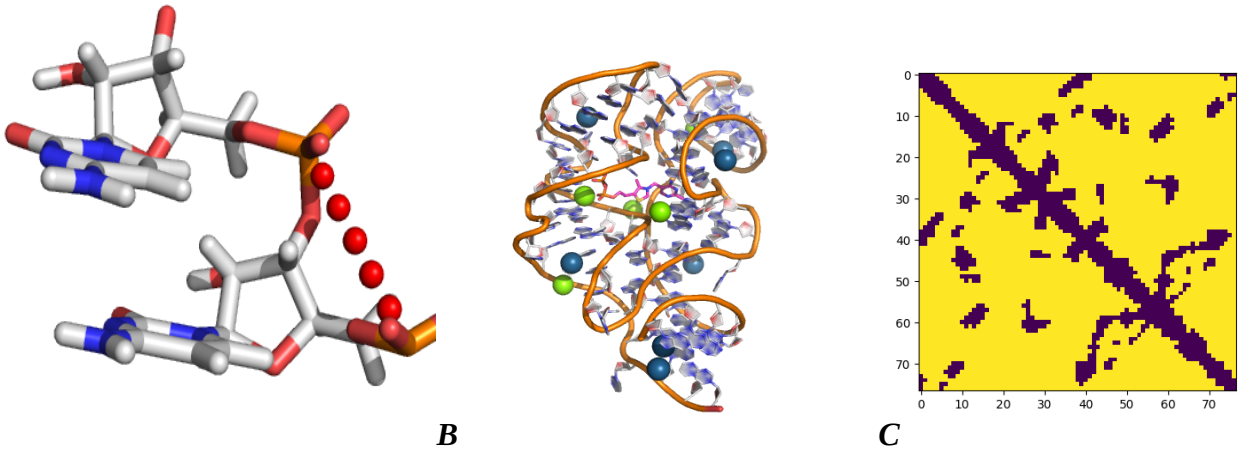
To predict the final structure of RNA, I would be predicting pair-wise atom euclidean distances between basic building blocks of RNA, also called ***residues***. The distance matrix is shown in matrix **D** (refer to formula 1). The matrix is zero-diagonal symmetric. The matrix has dimensions length * length , with length being number of residues in RNA sequence in the pair-wise distance matrix **D**.

Figure 1a shows exactly what distances are measured, which is the distance between two RNA phosphate atoms, while figure 1b shows an entire RNA structure with 70 residues and a complex fold for you to gain an intuition of the complexity of the problem.

To be specific about matrix **D**, since distance based regression may be difficult, I would predict categorical values instead of actual euclidean distances. The other reason why I used distance categorical values is that I would be able to assign probabilities on the predictions. These distance categories are short and long which are <16Å and >16Å respectively and rationale for the 16Å distance is in the analysis. 1Å is a unit of measurement of the field, which equals 10^{-10}m. The distance matrix hence is a tensor of size (Length * Length * 1 for two categories) shown in figure 1C. With a good categorical distance matrix, the RNA can be very easily modeled using various modeling software and thus predicting a good distance matrix is the first step to modeling RNA structure.

Formula 1, Pairwise distance matrix

$$\boldsymbol{D} = \begin{bmatrix} 0 & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & 0 & d_{23} & \cdots & d_{2n} \\ d_{31} & d_{32} & 0 & \cdots & d_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & d_{n3} & \cdots & 0 \end{bmatrix}$$



A ***B*** ***C***

Figure 1 : A). Two RNA residues and the distance between phosphate atoms shown in red spheres. B). Example of an RNA molecule and its complex structural fold (2cky). C). Categorical distance matrix, (dark, light for short and long distances respectively). This RNA is the TPP riboswitch which is a potential antibiotic target.

b. Inputs of model

The model was trained in batch size of 1 and for simplicity sake, the batch axis is omitted from the report but present in the model.

i. Sequence inputs

RNA sequences consist of only four different building blocks, A, U, G and C, where the sequence will be one-hot encoded into A, U, G and C in formula 2. Extra features using a variety of experimental and theoretical methods will be added to help the neural network model. The sequence input is a matrix with dimensions (Number_of_features * length , formula 2). This matrix will undergo 1D convolution finally transformed to a (length * length * channels) tensor using inverse singular value decomposition.

The features are the following :
1. Residue A, U, G, C and others one hot encoded (5 features)
2. Predicted secondary structure and its associated base pair probabilities, positional entropy, mean free energy structure from ViennaRNA (Lorenz et al. 2011). (4 features)
3. Familial consensuses sequence[1] and confidence of prediction from infernal (Nawrocki, Kolbe, and Eddy 2009). (6 features)

$$\boldsymbol{Inputs} = \begin{bmatrix} \textit{Legend} & \textit{Residue}\,1 & \textit{Residue}\,2 & \ldots & \ldots & \ldots & \ldots \\ A & 1 & 0 & 0 & 0 & 0 & 0 \\ U & 0 & 1 & 0 & 0 & 0 & 0 \\ G & 0 & 0 & 1 & 0 & 1 & 1 \\ C & 0 & 0 & 0 & 1 & 0 & 0 \\ \textit{Others} & 0 & 0 & 0 & 0 & 0 & 0 \\ \textit{Extra Feature A} & 2.5 & 3.5 & 4.5 & -2 & 2 & 0.5 \\ \textit{Extra Feature B} & 9 & 0.5 & -4.5 & 2 & 9 & -9.5 \end{bmatrix}$$

Formula 2 : Sequence Features for RNA sequence AUGCGG

Firstly, the biological and physical significance of using the first five features (A, U, G, C, others) is that sequence determines structure (Wang et al. 2017) and thus it is possible to predict structure (and its distance matrix) from the sequence alone. Secondly, given that secondary structure predictions are rather robust with an MCC of 0.96-1.00 (Popenda et al. 2012), I used ViennaRNA to predict our secondary structure. This is because secondary structure helps to determine and simplify the problem by defining which residues are close together and form RNA basepairs, however this feature is insufficient to determine the entire fold of the RNA. Thirdly, I used features from the consensus sequences of it's RNA family. RNA can be very different in sequence, yet adopt similar structure if they are from the same RNA family. This information from the consensus sequence will help predict related RNAs that differ very much in sequence but are from the same family.

ii. Pairwise inputs

There are certain features that are already (length * length) without the matrix transformation which transforms (Number_of_features * length) to (length * length). These features include which pairs of residues form secondary structures (also known as base pairs), predicted reliably by ViennaRNA. The pairwise features are obtained from the sequence and consensuses sequence. Figure 3 gives an example of a pairwise residues. They will give 21 (length * length) features. It is hoped this can help prediction of certain combination of residue pairs that interact strongly together.

1 Consensus sequence is the representative sequence of the RNA family

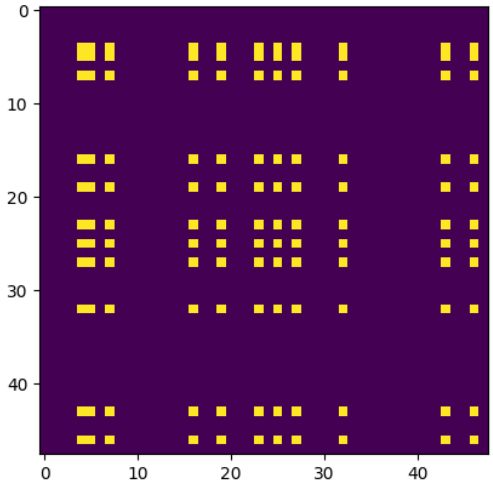


Figure 2 – Here shows presence of both A and G residues among residue pairs depicted by yellow dot.

iii). Distance matrix

I got the distance matrix from 491 RNA structures, with RNAs from 35 to 1500 residues long. I removed very small <35 residues and very large RNAs due to the former being usually uninteresting and the latter too large to model. This number is much reduced compared a total of 3000 structures due to redundancies in the protein data bank as each structure may occur multiple times. I used a manually curated list of non-redundant structures found here of version **2.141** to ensure our data is not biased by duplicates.

III. Methodology

a. Data Preprocessing in detail

For your reference, the preprocessed variables are listed in the table below and will be referred when described the data preprocessing.

	Sequence Features	D, Categorical Distance Matrix	Masking Matrix	Pairwise Features	List of names (string)
TRAIN	data2_x	data2_y	data2_y_nan	data2_y_ss	data2_name
VALIDATION	data2_x_val	data2_y_val	data2_y_nan_val	data2_y_ss_val	data2_name_val
TEST	data2_x_test	data2_y_test	data2_y_nan_test	data2_y_ss_test	data2_name_test

Table 1 : Names of inputs used in code.

We got the categorical distance matrix from the euclidean distance matrix from the RNA. I download the non-redundant RNA structure and sequence from the PDB database using wget which can be done with the following command for a pdb id 2CKY from figure 1.

wget https://files.rcsb.org/view/2CKY.cif
wget http://www.rcsb.org/pdb/download/viewFastaFiles.do?structureIdList=2CKY&compressionType=uncompressed

The cif files contain the cordinates of experimentally resolved atoms, and I only used the phosphate atom to find distances between RNA residues. There might be more than one RNA chain in the structure, and the non-redundant database specifies the correct chain I should use. I coverted the euclidean distances to categories <16Å and >=16Å. Since an RNA might have certain atoms not experimentally resolved, I also noted down which entries are missing from the distance matrix to be

used to mask out the loss to be further explained in the ‘metric’ section. The sequence and distance matrix will be stored in data_lim5000_nan.npy.zip.

To make the sequence features, we will have 15 features split into three parts.

i. We noted down the sequence of each RNA. We made a one hot encoded vector of A,U,G,C and others forming five features.

ii. To calculate the secondary structure features, I used the ViennaRNA to predict the secondary structure with the following command.

RNAfold -p < 2cky.seq > 2cky.dbn (it also produces 2cky.dp.ps which contain other features)
mountain.pl 2cky.dp.ps > $i.three_features

The final output is the secondary structure and also three features relating to secondary structure being associated base pair probabilities, positional entropy, mean free energy structure. The former is stored in data_lim5000_ss.npy.zip and latter stored in data_lim5000_extra.npy.

iii. The last six features are from the RFAM, which describe the sequence of a representative member of its family if it exist. We made a one hot encoded vector of A,U,G,C and none forming five features, where the entry of the vector will be 1 or 0.5 depending on the certainty of the prediction. The last feature will the certainty of the prediction. Missing values will have a value of 100. This was stored in data_lim5000_MSA.npy.

For the pair-wise features, preprocessing was done in the code below to make a tensor of size (length * length * 10) . This tensor notes down presence of certain residue pairs. I did this for both the sequence and representative sequence of the family. With the RNA pair-wise secondary structure matrix, I get a tensor of (length * length * 21).

```
pair_wise_res = {('A','A') : 0, ('U','U') : 1, ('G','G') : 2,
                 ('C','C') : 3, ('A','U') : 4, ('A','G') : 5,
                 ('A','C') : 6, ('G','U') : 7, ('C','U') : 8,
                 ('C','G') : 9}
mat_pairres_con = np.zeros((len(temp1[1]),len(temp1[1]),10))
for ii in range(0,len(temp1[1])):
    for jj in range(ii,len(temp1[1])):
        temp_paires = sorted((data4[i][1][ii].upper(),
                              data4[i][1][jj].upper()))
        if tuple(temp_paires) in pair_wise_res:
            index = pair_wise_res[tuple(temp_paires)]
            mat_pairres_con[ii,jj,index] = 1
            mat_pairres_con[jj,ii,index] = 1
```

Code 1 : Code to make the pair-wise sequence matrix

v) Data augmentation

The train data will also be augmented by sub-sampling 35, 50, 75, 100, 150, 200, 500 residues of the entire structure and also subsampling three halves, five thirds and 7 quarters of the data. This increases the size of the training set by 30 fold. A 500 residue limit was set due to memory constraints and this means RNA >= 500 residues long are all in the train set.

b. Data test split

I made a train,validation, test split of 370 : 110 : 11 structures. Model is trained on the train set, training stopped when performance is optimized int the validation set and model finally evaluated against the 11 test structures. These 11 structures were part of an RNA modeling competition and benchmarks against the best methods.

c. Metrics

Model predicts the categorical distance matrix and the objective function will be the average sigmoid log loss of the distance matrix. The diagonals and missing data of the sigmoid loss function will be masked with 0 since they should not contribute to the loss with the former being easy to predict and the latter having no labels.

The model however will be evaluated using balanced accuracy with formula (TP/Positives + TN/negatives)/2. Balanced accuracy is a balanced metric unaffected by the size of the RNA. This is because the pair-wise distance categorical matrix contains mostly negatives and larger RNAs contains relatively more negatives compared to a smaller RNA since the number of matrix entries scales with N^2, but the number of interactions scales roughly by N since an RNA residue can only maximally interact with a fixed number of different residues. The balanced accuracy thus gives a metric that can be compared across RNAs of different sizes.

The probability threshold will be 0.3, where distances predicted with 30% probability will be deemed sufficient to be called a close distance. This was seen in the cross validations where a lower threshold results in a higher balanced accuracy.

To obtain the predictions, I performed a sigmoid transformation on the logits using this function : tf.nn.sigmoid(out), where out is the logits. I filled the diagonals as 0 (close distances) and added its the categorical distance matrix by its transpose to obtain a symmetrical matrix and applied the 30% threshold to obtain the predictions. Note : $M + M^T$ is a symmetric matrix

d. Choosing a distance threshold of 16Å

I plotted all the distances between residues and found that there are three peaks in 8Å, 12Å and 18Å. I tried different models with different distances. The 8Å and 12Å model suffered from severe class imbalance and the categorical distance matrix did not look informative judging from experience which is why I settled for a 16Å model.

e. Implementation

I would be using deep learning to predict the categorical distance matrix from the sequence and features generated from the sequence. Deep learning automates features and current image processing techniques are now state of the art and exceed human performance in certain cases (Dodge and Karam 2017). I believe a CNN is the easiest and most direct way to solve the problem because the RNA sequence varies in length in two orders of magnitude and existing machine learning methods cannot easily take in inputs of varying sizes. Furthermore, the problem is extremely complicated and almost impossible to model by 'expert intuition'. This makes is difficult to design features to model using traditional machine learning methods. Given the extreme difficulty for a different machine learning model and the robustness of CNN in image processing, I decided to use other existing modeling methods as a benchmark instead of machine learning to be elaborated in the next section.

One popular CNN framework is that of inception3 (Szegedy et al. 2016) which uses a large effective convolutional filter while using less parameters, as it dissects a large convolution kernel to various smaller kernels. Since the inception3 framework has been shown to work for many cases and is relatively easy to implement, I would use these inception units for my problem.

The deep learning model can be summarized in figure 3 and it contains five inception3 units. There were two types of inputs : i) residue level features and ii) pairwise features, and both types of features undergo 1D and 2D convolution respectively.

Firstly, the residue level features was followed by a two 1D inception3 networks, with the first being having filter size of (15x1) with 'valid' padding to transform the inputs from (15 * Length * 1) to (1 * length * 64) followed by 1D convolution with 'same' padding along the length axis with the

final tensor to be (1 * length * 96). This is because the matrix is spatially correlated along the length axis, but not along the features axis as different features have no obvious spatial correlation to each other, but an RNA residue is very correlated to its neighboring residues being bonded physically by strong chemical bonds. This is the reason for a 1D convolution along the features axis followed by 1D convolution along the length. The resultant (1 * length * 96) tensor will undergo inverse SVD across each of the 96 channels to attain a (length * length * 96) tensor.

Secondly, the pair-wise features under go 2D convolution to form a (length * length * 64) tensor and merged with the output from the inverse SVD to form a (length * length * 160) tensor. A 2D CNN is subsequently used with a kernel sizes 3 or 5 and 'same' padding. Three inception3 units were used and the final output (length * length * 224) undergoes a linear combination to output the logits for the final pair-wise distance matrix which will be optimized with a average sigmoid loss over the entire matrix. The diagonals and missing data will be masked with 0 and do not contribute to the loss.

Dropout was defined in model but not used while the learning rate had a cosine decay over each epoch. The cosine learning rate allows a search for both the global and local minima. The initial learning rate was 0.1, 0.01, 0.001 at epoch 0 - 150, 150 – 225, 225 - 245. Layer normalization was also used since batch normalization will prove difficult with different RNA sequence lengths. The code will be shown in an environment.

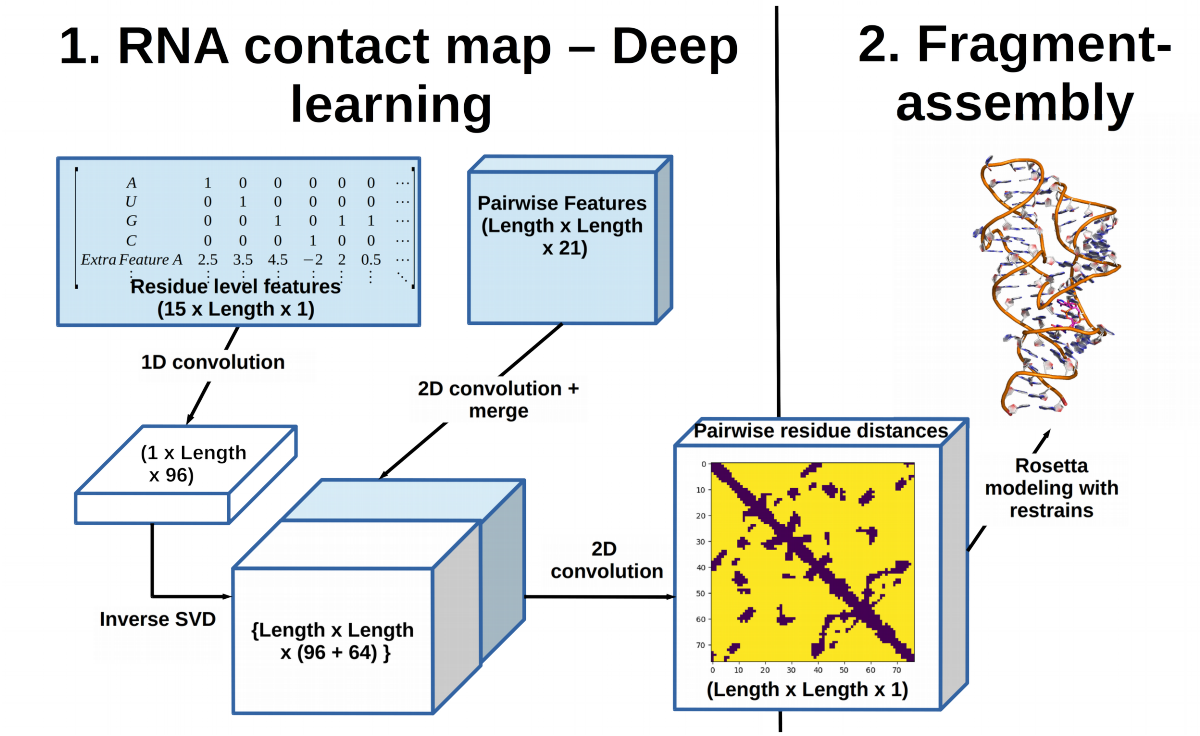


Figure 3 Architecture of model : Inputs are shown in blue, and the output is a pair-wise distance matrix. Deep learning portion only consist of part 1, part 2 is outside the scope of the project but completes modeling of the RNA structure.

f. Refinement

Many initial models failed to converge, and the only one that converged was this initial model. One thing that helped to converge was adding momentum and a cosine decay to the learning rate (result of unconverged models not shown). The initial test balanced accuracy was 0.758, which increased to 0.766 upon masking the diagonal loss. Since the models take 2 days to train on a gpu, extensive parameter tuning is difficult. However, other things to test include different CNN architectures like a wide or deeper network and are underway.

The accuracy reported here will differ from the benchmark since there are certain structures in the test set not used in the benchmark.

IV. Analysis

Data Exploration and Exploratory Visualization

Provided in environment.

The inputs and outputs however have no clear correlation, consistent with the difficulty of the problem. I have included some visualization for the inputs, but there are no plots I could find that throughly give and intuition or show any correlation with the output. This further emphasizes the need or deep learning to automate learning of features to solve the problem. However, the output has been extensively explored to find the number of missing outputs and rationale of our 16Å cutoff for the categorical distance matrix.

V. Results and Benchmark

I used RNA modelling competitions and converted the structures into a distance matrix for equal comparison to my deep learning method (Miao et al. 2017). There were 21 RNA competitions, with 16 having released results. Among these 16, one had an erroneous sequence between the predicted and modeled structure (Puzzle 5, 13), one was trivial (puzzle 1), two contained more than one RNA chain (puzzle 2 and 10) and outside the scope of our train set. I was left with 11 structures to benchmark our method against others.

I proceeded to compare the balanced accuracy of deep learning compared to other methods used in competitions. Since RNA structure exist as an ensemble in a cell, the competitions allow 10 models per competitor though the experimental structure only contain one model due to experimental constraints. I would take the mean rank of each method per competition and compare with my deep learning model.

I used ranks instead of balanced accuracy to normalize across competitions. Some competitions are inherently more difficult than others and using just balanced accuracy will be favoring methods which participate the easiest competitions. The average rank of each method will be a good metric to benchmark different methods across competitions with different difficulty levels. Since the other methods were outside machine learning, the details will be left out in favor of a brief description.

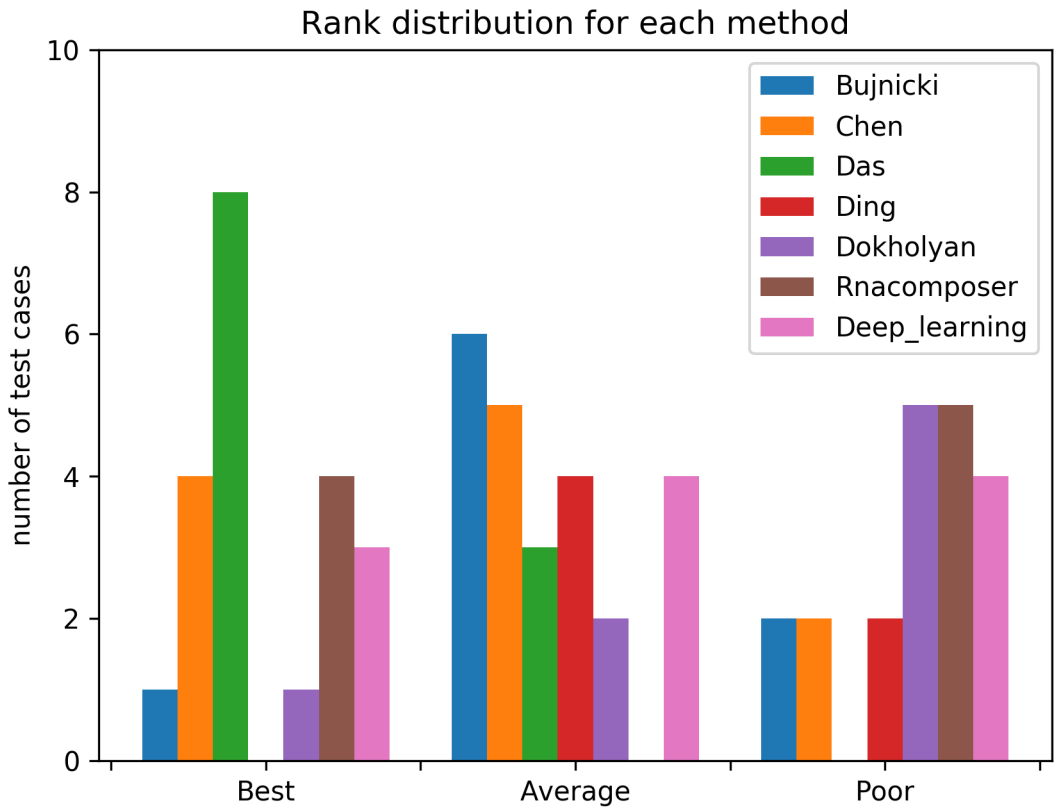


Figure 4 : Ranks of different methods over the 11 competitions. The histogram bins are best (< 0.33 percentile), average (0.33-0.67 percentile) and worst (>0.67 percentile).

Results in figure 4 show the results in three histogram bins, best < 0.33 percentile, average (0.33 - 67 percentile) and worst (> 0.67 percentile). I added a figure with 10 histogram bins at the supplementary for a better view of the data. Deep learning has a validation balanced accuracy of 80.5 percent, and test balanced accuracy of 78.4 percent. The results are also reported in table 2.

Group	Average Rank	Average Balanced Accuracy
Das	0.24	0.833
Chen	0.44	0.8
Deep_learning	0.49	0.784
Bujnicki	0.52	0.795
RNAcomposer	0.55	0.797
Ding	0.61	0.77
Dokholyan	0.699	0.769

Table 2 : The average rank and balanced accuracy. The average balanced accuracy is not the best metric since only Das, Chen and Deep_learning had results for all 11 competitions and difficult competitions tend to reduce the average balanced accuracy but not the average rank.

The Das and Chen, Bujnicki, Ding and Dokholyan group modeled the structures using first principles derived from statistical physics and such methods are known to be computationally expensive (Miao et al. 2017). Meanwhile Deep learning and RNA composer use deep learning and

machine translation methods, though documentation of the latter is extremely sparse (Popenda et al. 2012).

Justification

Deep learning appears to be do relatively well, but not as well as the best method by Das, which is a method from first principles compared to statistical learning. There are possible reasons why the data fails to generalize, and mainly is the the dataset size is relatively small to fully generalize in this case, where there are only 490 data points.

Nevertheless, there is still a space for our method since it can predict contacts relatively fast and accurately compared to Das which requires thousands of CPU hours per model.

Statistics was not used rigourously in the data analysis, since there were only eleven samples. However, the standard deviation of the ranks was 0.2 while that of balanced accuracy was 0.05. This gives a standard error of 0.07 and 0.016, and Das and Dokholyan are probably significantly better and worst from the rest using standard errors.

It is good to note that the competitions are generally RNAs that are unseen, whereby there exit no significantly or remotely similar structure which calls for a competition. However, our validation results show an accuracy of 0.804, and approaches ~90 for certain RNAs. It is not surprising that the model fails to generalize since the number of data is limited. This is not surprising since the problem is very challenging.

VI. Conclusion

Free-Form Visualization

The current model mines RNA information from the structure database. It currently predicts pair-wise distance matrices, but I believe if the deep learning model becomes state of the art, it can be a used for transfer learning, much like that of the inceptionv3 network and it's weights.

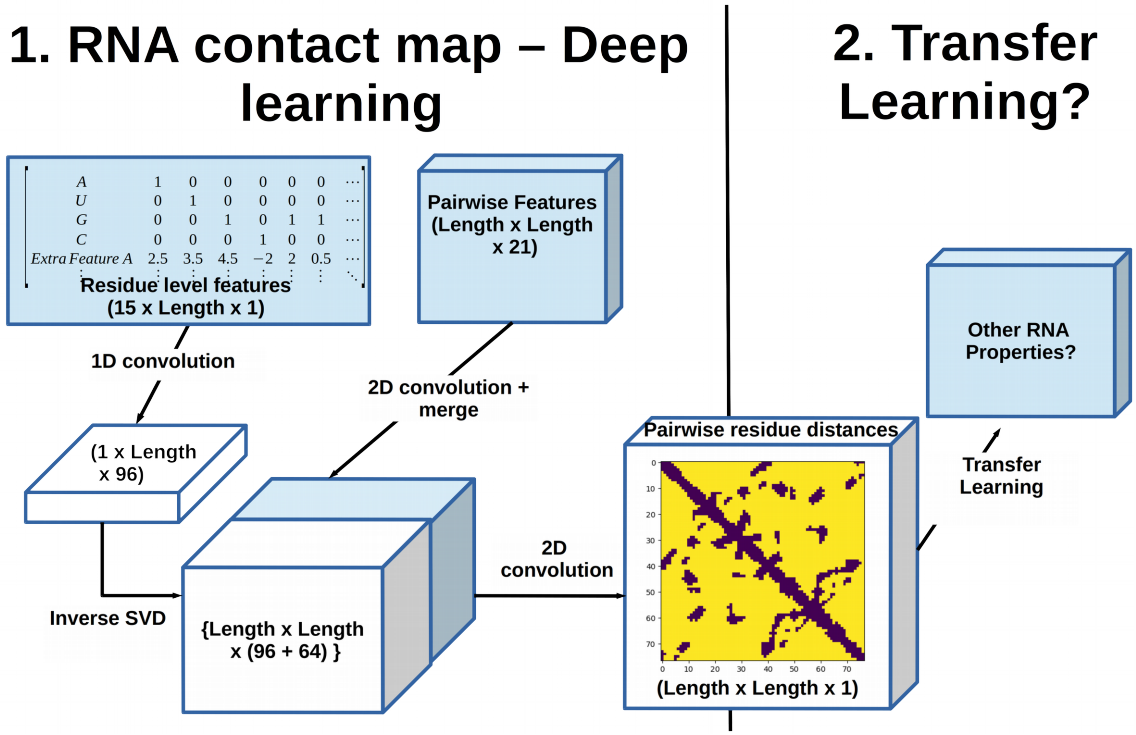


Figure 5 : Mining RNA structural information for various purposes.

Reflection

I have made a deep learning model to predict RNA distance matrix which can be used to help RNA modeling. I used a variety of inputs from sequence features to pair-wise features to produce a generative model that outputs the distance matrix. The most interesting aspect, which is the most difficult is hyper-parameter tuning. I believe and hope that the model can still be tuned better with time. At current, the model is able to predict RNA distances at a very quick pace, though the method is slightly poorer compared to other longer methods. Like all other methods, it is not always generalizable due to the inherent difficulty of RNA structure prediction.

Improvements

The dataset is relatively small and a follow-up of my project is to use the predictions to model a structure using fragment assembly (figure 3 part 2), and proceed to benchmark a larger dataset. This was not included here since fragment assembly is outside the scope of the project. Currently there are also some bottlenecks in the model pipeline, namely the (15 * 1) 1D convolution and also the inverse SVD which very suddenly compresses or expands the data. Alternative methods to prevent hese bottleneck will likely help the model.

Another interesting follow-up is to apply transfer learning on my model to predict another RNA related problem, eg. Secondary structure or other properties.

Reference

Das, Rhiju, and David Baker. 2007. “Automated de Novo Prediction of Native-like RNA Tertiary Structures.” *Proceedings of the National Academy of Sciences* 104 (37):14664–14669.

Dodge, Samuel, and Lina Karam. 2017. “A Study and Comparison of Human and Deep Learning Recognition Performance Under Visual Distortions.” *arXiv Preprint arXiv:1705.02498*.

Ke, A. 2004. “Crystallization of RNA and RNA?protein Complexes.” *Methods* 34 (3):408–14. https://doi.org/10.1016/j.ymeth.2004.03.027.

Lorenz, Ronny, Stephan H. Bernhart, Christian Hoener Zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F. Stadler, and Ivo L. Hofacker. 2011. “ViennaRNA Package 2.0.” *Algorithms for Molecular Biology* 6 (1):26.

Lounnas, Valère, Tina Ritschel, Jan Kelder, Ross McGuire, Robert P. Bywater, and Nicolas Foloppe. 2013. “CURRENT PROGRESS IN STRUCTURE-BASED RATIONAL DRUG DESIGN MARKS A NEW MINDSET IN DRUG DISCOVERY.” *Computational and Structural Biotechnology Journal* 5 (6):1–14. https://doi.org/10.5936/csbj.201302011.

Miao, Zhichao, Ryszard W. Adamiak, Maciej Antczak, Robert T. Batey, Alexander J. Becka, Marcin Biesiada, Micha\l J. Boniecki, Janusz M. Bujnicki, Shi-Jie Chen, and Clarence Yu Cheng. 2017. “RNA-Puzzles Round III: 3D RNA Structure Prediction of Five Riboswitches and One Ribozyme.” *RNA* 23 (5):655–672.

Nawrocki, E. P., D. L. Kolbe, and S. R. Eddy. 2009. “Infernal 1.0: Inference of RNA Alignments.” *Bioinformatics* 25 (10):1335–37. https://doi.org/10.1093/bioinformatics/btp157.

Popenda, M., M. Szachniuk, M. Antczak, K. J. Purzycka, P. Lukasiak, N. Bartol, J. Blazewicz, and R. W. Adamiak. 2012. “Automated 3D Structure Composition for Large RNAs.” *Nucleic Acids Research* 40 (14):e112–e112. https://doi.org/10.1093/nar/gks339.

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. “Rethinking the Inception Architecture for Computer Vision.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2818–2826.

Wan, Yue, Michael Kertesz, Robert C. Spitale, Eran Segal, and Howard Y. Chang. 2011. “Understanding the Transcriptome through RNA Structure.” *Nature Reviews Genetics* 12 (9):641–55. https://doi.org/10.1038/nrg3049.

Wang, Sheng, Siqi Sun, Zhen Li, Renyu Zhang, and Jinbo Xu. 2017. “Accurate de Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model.” *PLoS Computational Biology* 13 (1):e1005324.

Supplementary

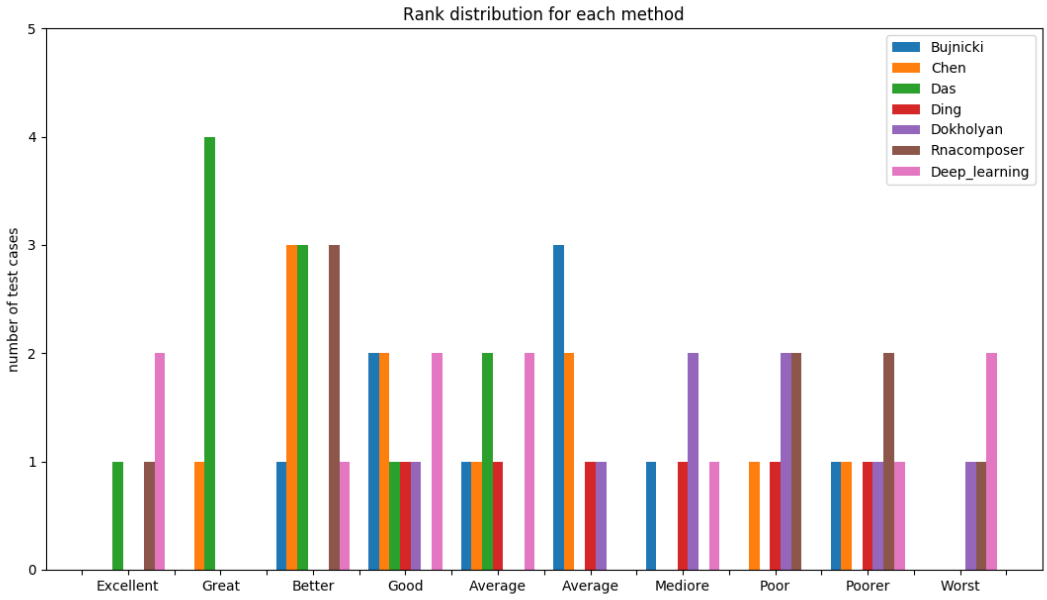


Figure 5 : Results of different methods with 10 histogram bins.