CS230

RNABERT: RNA Family Classification and Secondary Structure Prediction with BERT pretrained on RNA sequences

Colin Hall Kalicki
Department of Biomedical Informatics
Stanford University
ck9898@stanford.edu

Esin Darici Haritaoglu
SCPD AI Graduate Program
Stanford University
dhesin@stanford.edu

1 Introduction

RNA secondary structure prediction is currently one of the main challenges facing RNA structural biologists. A better understanding of RNA folding rules, or creating a structure prediction model, is needed to discover how RNA function is determined by its folding. A better structure prediction model could help in many downstream tasks, such as the designing of RNA-targeting drugs. The classical free-energy minimization approach presents limits, especially regarding complex structures such as pseudo-knots or long sequences (>200 nucleotides (nt)). [8]. Furthermore, current RNA datasets are limited in size and prone to over-fitting. To tackle the problem of RNA structural prediction with limited data, we propose using the Bidirectional Encoder Representations from Transformers, or BERT, model [5] to accomplish secondary structure prediction. First, we pretrained our model, RNABERT, with primary RNA sequences. Then we finetuned the RNABERT with a family classification task to understand the performance of pretrained network. We compared this task to a non-pretrained RNABERT and simpler bidirectional-LSTM (BILSTM).

2 Related Work

Our work resembles the works of DNABERT [6] and E2Efold [4]. We pretrained BERT similar to how DNABERT was pretrained [6], but with RNA sequences from the Rfam database [7]. Similar to [2], we worked with Rfam families of RNA sequences. While we worked on directly classifying Rfam families, [2] worked on RNA family clustering. They used a similarity measure between two RNA sequences with respect to soft symmetric alignment [3] to measure the quality of the embeddings. We also worked on Rfam family classification to measure the effectiveness of the RNABERT embeddings in other downstream tasks while using accuracy and F1 score as a metric.

3 Code

Codebase is available at https://github.com/dhesin/RNABERT-2. We have used example repositories from HuggingFace, namely https://github.com/huggingface/transformers/tree/main/examples/pytorch/language-modeling. We have modified the code to download our datasets and train a custom model from scratch since the example code uses models and datasets at the HuggingFace hub. Additionally, we created a tokenizer module for RNA sequences and finetuning code.

,

4 Dataset and Features

We have downloaded sequences from 29 noncoding RNA (ncRNA) families from RFam database [7] 2 mRNA families from virus and Humans for a total of 31 RNA families. For pretaining we selected sequences with lengths less than 400 basepairs (bp) to train 1-mer sequences. Later, we used 410K sequences for pretaining of 6-mer sequences across all 31 families. Preprocessing of the data included removing edges of sequences with non-canonical bases (A,C,U,G) on the edges of the sequences and removing sequences that were comprised of more than 1% non-canonical bases. We then selected for sequences of lengths <= 512 as this the size of input for RNABERT. Sequences were not truncated or concatenated to avoid our models from learning human-made modifications. We used 3% of them for evaluation during pretraining. Sequences were preprocessd by splitting them into k-mers (1-mer and 6-mer for pre-training task and 6-mer for non pre-training and BILSTM benchmarks). Sequences were then one-hot encoded. The vocabulary size for these sequences is 8277, including special tokens for padding, masking, classification, separator, and unknown. Although there are only 4 nucleotides that make up sequence bases, namely C, A, G, and U, there are sometimes uncertainties in the sequences, and they are represented with additional characters [1] Data was split keeping ratios of number of sequences per family intact.

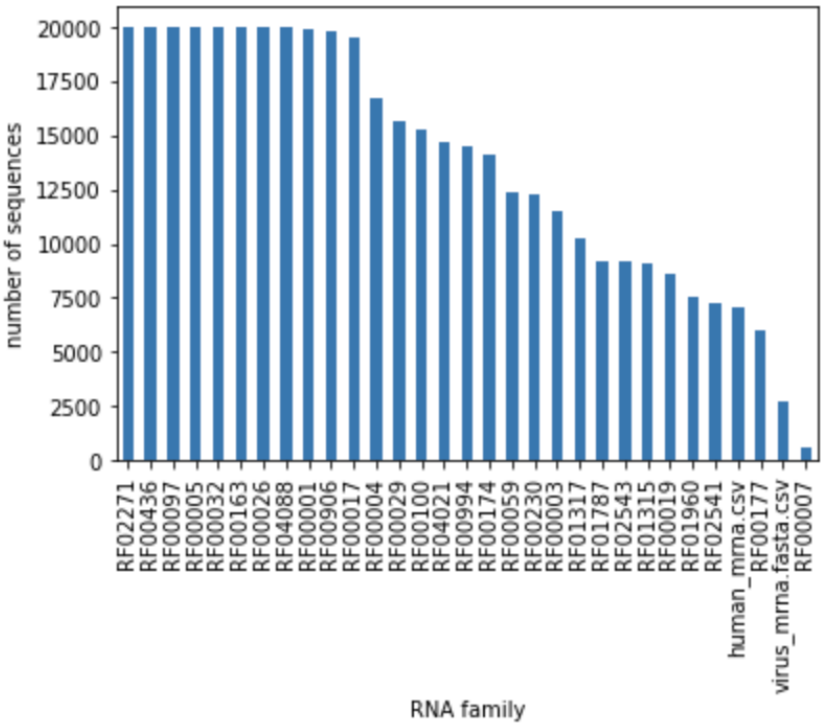


Figure 1: Number of sequences per RNA family

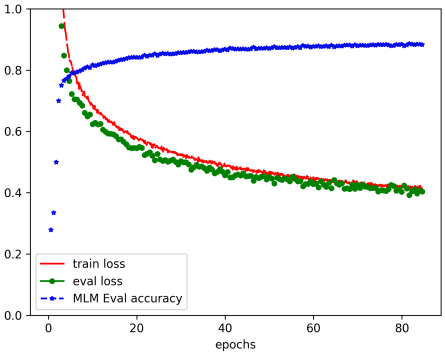
5 Methods

We experimented with 2 methods;

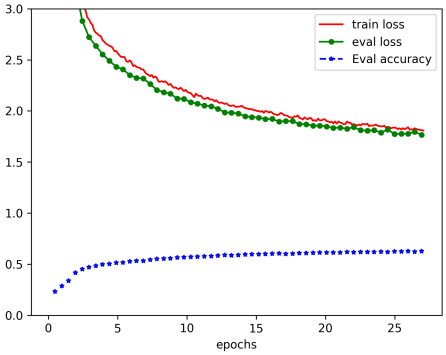
- Pretraining a BERT model with primary sequences, then classifying RNA families with features from BERT
- Training LSTM network with primary sequences to classify RNA families

5.1 RNABERT

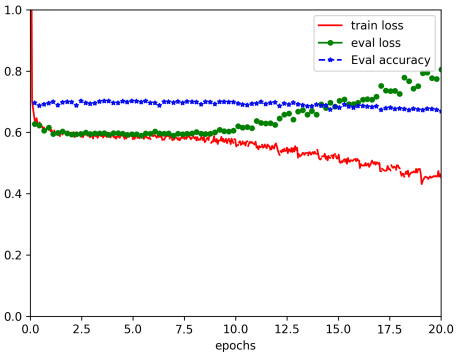
We have pre-trained BERT network with 6-mer sequences with 18% and 33% masking. We have used 6 hidden layers instead of 12 while keeping the other BERT parameters, like hidden and intermediate



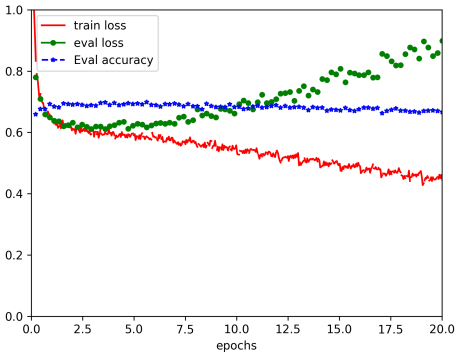
(a) Pretraining with 18% masking



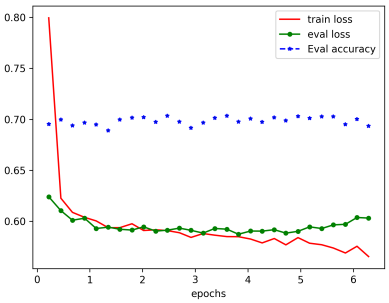
(b) Pretraining with 33% masking



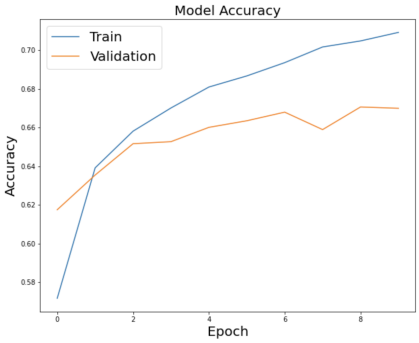
(c) Finetuning with pretraining - 18% mask



(d) Finetuning without pretraining



(e) Finetuning with pretraining - 33% mask



(f) Training statistics for BILSTM

sizes, the same. During pre-training, tokens are selected to be masked initially. Then these selected tokens and the following 5 tokens are also masked to avoid the network from peaking into masked tokens through the following 6-mer for 18% masking. For 33% masking previous 5 6-mer and following 5 6-mer are masked. This approach is also used in [6]. To implement consecutive masking, we have implemented a custom data collator, which takes care of collating sequences into batches and masking them.

After pre-training, we finetuned BERT with classification head on top for family classification of 31 RNA families. Figure2c shows how training, validation loss, and family prediction accuracy evolve over 20 epochs.

We also trained a BERT network with a classification head on top without pre-training to classify 31 RNA families and compare classification results with the pre-trianed network. Figure 2d shows how training, validation loss, and family prediction accuracy evolve over 20 epochs.

5.2 LSTM

We trained a BILSTM using 6-mer, one-hot encoded sequences. The model included an embedding layer, one bidrectional-LSTM unit, two dense layers one with ReLU activation and a softmax output layer. This model was trained for 10 epochs. Evaluation through training can be seen in figure 1f.

5.3 Training Dataset Size

We wanted to also look at the robustness of the pre-trained network in comparison to our benchmarks. To achieve this we created 5 datasets (1 and 1/16 the size of the original dataset) that contained the same distributions of number of sequences per class to test how training-set size impacted our predicitions.

5.4 Controlling Sequence Length Distributions

To understand the effect of the sequence lengths in our classification tasks, we have created multiple datasets with sequence lengths within the specified range. There were 5 sets, each with sequence lengths between 25-75, 75-125, 125-175, 175-225, and 225-275.

New datasets are more unbalanced with the number of sequences representing each family (supplemental figures 5-9)

6 Experiments/Results/Discussion

6.1 RNABERT

Each epoch of pretraining takes 55 minutes with 2 RTX3090 GPU; therefore, it was impractical to experiment with different masking ratios, learning rates, k-mers, etc. We pretrained with 1-mer, 6-mer sequences. We also pretrained 6-mer with 6 and 11 consecutive token masking totaling 18% and 33% masking, respectively (masking is set to 3% but with repetition, effective masking increases). Figure2a and 2b show how training, validation loss, and MLM accuracy evolve for 18% and 33% masking.

With 1-mer pretraining and finetuning, we got an accuracy of 0.67 on the classification task. With 6-mer, our accuracy increased to 0.70 (both 18% masking). We also finetuned after 33% masking, and accuracy was again 0.70. Pretraining through auxiliary tasks without labeled data enables using a limited amount of labeled data for supervised training during finetuning. To test if this holds for our 6-mer pretrained network, we created another dataset with 6% of the original. Accuracy only dropped to 0.67. Note that these results are for train/validation/test sets that include sequence lengths 25 to 512. Finetuning for family classification tasks with (18% masking) and without pretraining results were surprising. They achieved similar accuracy around the same epochs. The only difference for pretrained network was that it took more epochs for the network to start overfitting compared to the network without

	Accuracy %
25-75	92
75-125	86
125-175	87
175-225	91
225-275	95

Table 1: Accuracy for different sequence length bins

pretraining. This result could be explained in several ways. **1.** The network might be learning sequence length differences between families. **2.** 6 consecutive masking was not enough for 6-mer sequences that the network was peeking into surrounding tokens to guess the masked ones.**3.** 0.18 masking was not enough bottleneck for the BERT to learn useful representations.

To test if the model was learning sequence length we checked across 5different length distributions as outlined in our methods. Table 2 shows the weighted accuracy within each dataset. The accuracy and F1 results improved across the high and low-represented families. The results are also mixed since some families with high/low representation also got low F1 and accuracy scores. Appendix figures 5 through 9 show more details about the results on these datasets.

Accuracy results with 18% and 33% masks were also similar, although pretraining plots are quite different. Pretraining plots with 33% mask show that it is harder for BERT to learn representations and predict masked tokens. But finetuning stage was able to obtain same accuracy results which may suggest that network is tuning into sequence lengths for classification rather than other features.

6.2 LSTM

The performance of the LSTM, after training across 10 epochs, achieved an accuracy of 70%. Looking at further breakdown per family we noticed that accuracy was uneven across families. Testing using the training set that was 6% the size of the original yielded an accuracy reduction to 64% for the LSTM.

To test if the model was learning sequence length we checked across the data sets with differing length distributions as outlined in our methods. Table 2 shows the weighted accuracy within each data set. The accuracy and F1 results improved in families with many sequences in that length distribution (supplementary 6-10). The results showed that some families with minimal sequences per family in that length category also achieved high F1 score and accuracy.

	Accuracy %
25-75	0.85
75-125	0.84
125-175	0.89
175-225	0.92
225-275	0.93

Table 2: Accuracy for different sequence length bins using BILSTM

7 Conclusion/Future Work

Overall results show that RNABERT pretrained with 18% masking has some advantages over the LSTM but not as much as expected. RNABERT has a 1-2 point accuracy advantage on the full set of sequence lengths between 25 and 512 and has a 3-4 point higher accuracy on the 6% of the full set. The results are very similar for RNABERT pretrained with 33% masking. Accuracy on datasets with more uniform sequence lengths are much higher but with mixed results across since RNA families with large and small number of sequences. This suggests that that both models can learn family idenities well with even length distribution but the current BERT model still has limited advantage Future work may include experimenting with a larger RNABERT model to see if low-performing classes would improve. Additionally, analyzing attention maps would help to understand what RNABERT is looking into for each family. The final goal of our project was to predict the secondary structure of the RNA sequences. Due to unexpected results in Rfam family classification, we took time for a more detailed analysis of the classification task. Therefore other future work includes secondary structure prediction.

8 Contributions

Esin downloaded the pre-training dataset, built the RNABERT pretraining/finetuning model, and experimented with it. Colin constructed all curated datasets and built the BILSTM model and experimented with it.

References

[1] Standard iub/iupac amino and nucleic acid codes.

[2] Sakakibara Y. Akiyama M. Informative rna base embedding for rna structural alignment and clustering by deep representation learning. *Genomics and Bioinformatics*, 4, 2022.

[3] Tristan Bepler and Bonnie Berger. Learning protein sequence embeddings using information from structure. *arXiv preprint arXiv:1902.08661*, 2019.

[4] Xinshi Chen, Yu Li, Ramzan Umarov, Xin Gao, and Le Song. Rna secondary structure prediction by learning unrolled algorithms. 2020.

[5] Chang M.-W. Lee K. Toutanova K. Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, 2018.

[6] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, 02 2021.

[7] Ioanna Kalvari, Eric Nawrocki, Nancy Ontiveros-Palacios, Joanna Argasinska, Kevin Lamkiewicz, Manja Marz, Sam Griffiths-Jones, Claire Toffano-Nioche, Daniel Gautheret, Zasha Weinberg, Elena Rivas, Sean Eddy, Robert Finn, Alex Bateman, and Anton Petrov. Rfam 14: Expanded coverage of metagenomic, viral and microrna families. 11 2020.

[8] David Mathews and Douglas Turner. Mathews dh, turner dh. prediction of rna secondary structure by free energy minimization. curr opin struct biol 16: 270-278. *Current opinion in structural biology*, 16:270–8, 07 2006.

9 APPENDIX

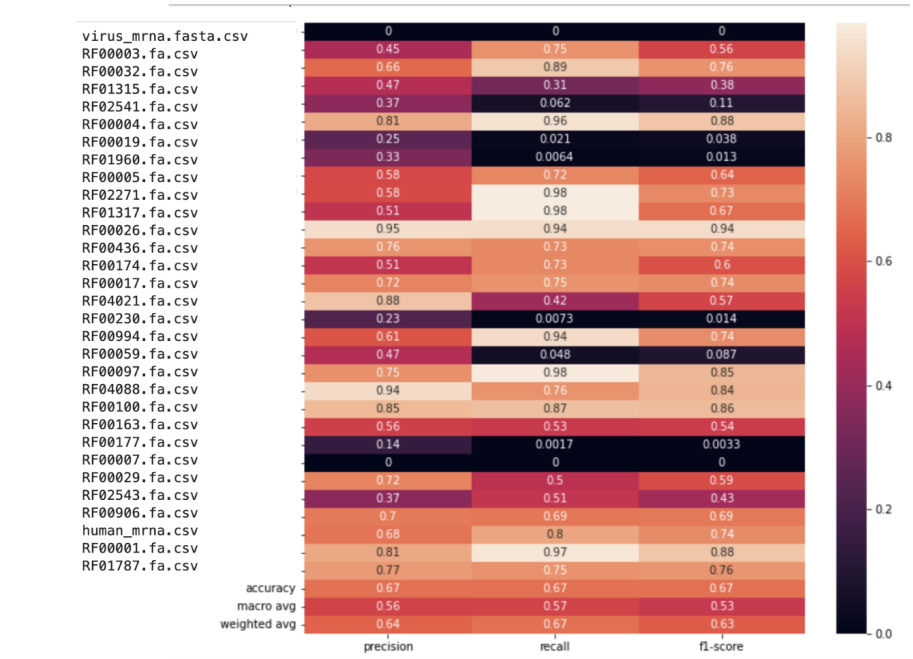
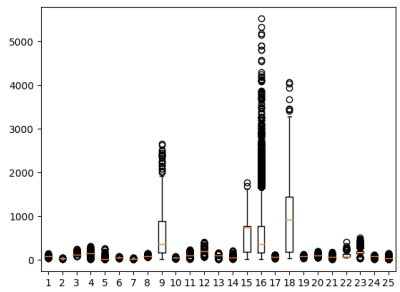
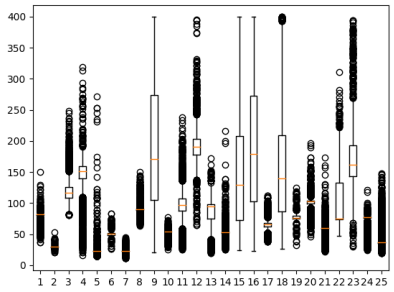


Figure 3: Results from LSTM on test data

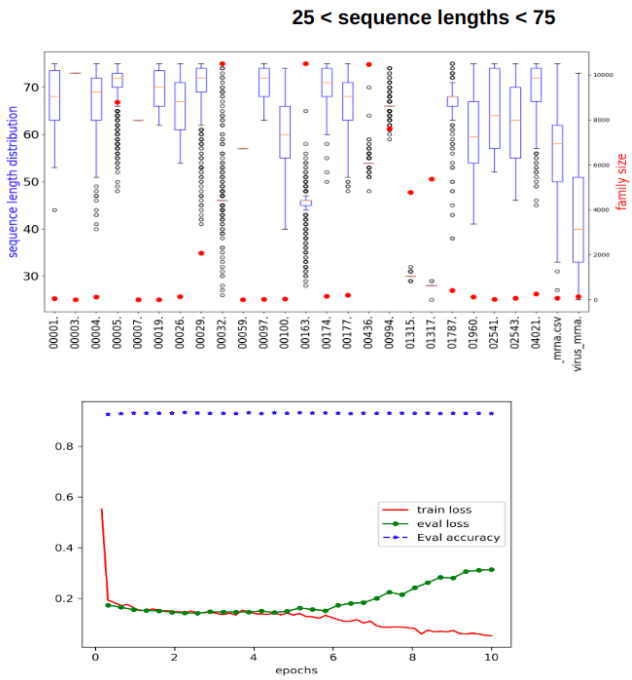


(a) All Sequences of 25 Rfam Family



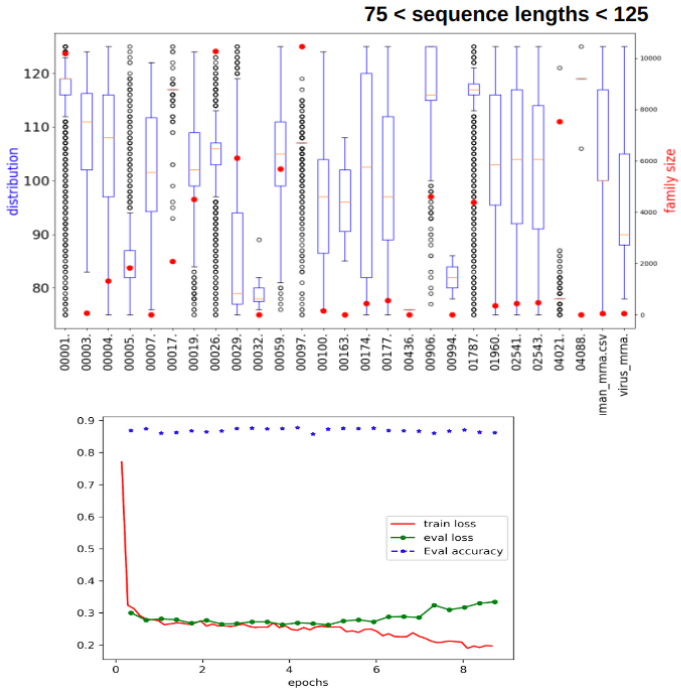
(b) Sequences with length < 400 of 25 Rfam Family

Figure 4: Sequence Length Distribution for each of 25 Rfam family



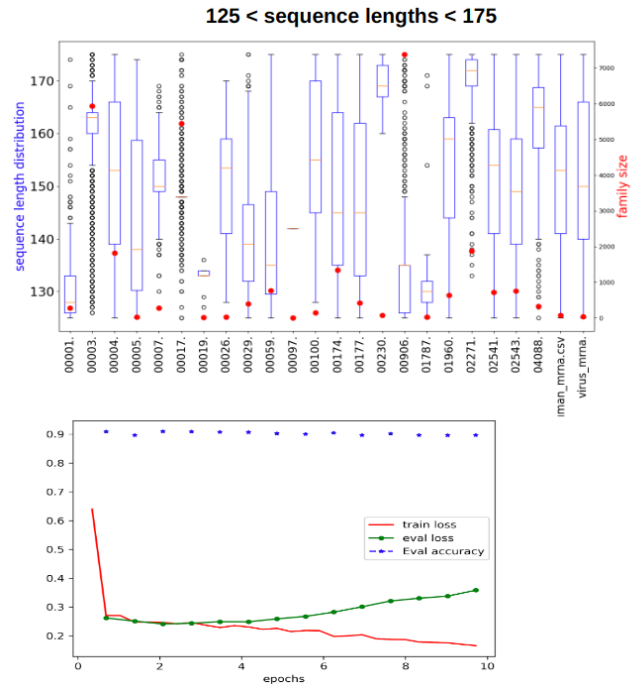
	precision	recall	f1-score	support
RF00001.fa.csv	0.90	1.00	0.95	9
RF00003.fa.csv	0.00	0.00	0.00	0
RF00004.fa.csv	0.68	0.85	0.76	20
RF00005.fa.csv	0.97	0.99	0.98	1087
RF00007.fa.csv	0.00	0.00	0.00	0
RF00017.fa.csv	0.00	0.00	0.00	0
RF00019.fa.csv	1.00	0.67	0.80	3
RF00026.fa.csv	0.94	0.94	0.94	18
RF00029.fa.csv	1.00	1.00	1.00	231
RF00032.fa.csv	0.85	0.83	0.84	1298
RF00059.fa.csv	0.00	0.00	0.00	1
RF00097.fa.csv	1.00	1.00	1.00	1
RF00100.fa.csv	0.00	0.00	0.00	6
RF00163.fa.csv	0.99	1.00	0.99	1296
RF00174.fa.csv	0.00	0.00	0.00	16
RF00177.fa.csv	0.00	0.00	0.00	21
RF00230.fa.csv	0.00	0.00	0.00	0
RF00436.fa.csv	0.84	0.85	0.85	1292
RF00906.fa.csv	0.00	0.00	0.00	0
RF00994.fa.csv	0.97	0.99	0.98	939
RF01315.fa.csv	0.98	1.00	0.99	590
RF01317.fa.csv	0.97	1.00	0.99	663
RF01787.fa.csv	0.87	0.78	0.82	50
RF01960.fa.csv	0.00	0.00	0.00	19
RF02271.fa.csv	0.00	0.00	0.00	0
RF02541.fa.csv	1.00	1.00	1.00	2
RF02543.fa.csv	0.00	0.00	0.00	8
RF04021.fa.csv	0.62	0.64	0.63	33
RF04088.fa.csv	0.00	0.00	0.00	0
human_mrna.csv	0.90	1.00	0.95	9
virus_mrna.fasta.csv	0.00	0.00	0.00	23
micro avg	0.93	0.93	0.93	7635
macro avg	0.50	0.50	0.50	7635
weighted avg	0.92	0.93	0.92	7635

Figure 5: Results for sequence of length > 25 and < 75



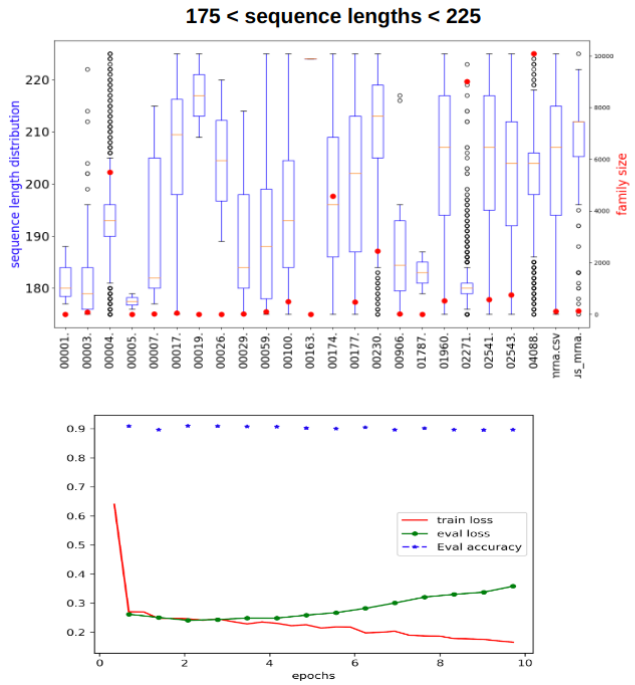
	precision	recall	f1-score	support
RF00001.fa.csv	0.97	1.00	0.99	1132
RF00003.fa.csv	0.00	0.00	0.00	7
RF00004.fa.csv	0.00	0.00	0.00	143
RF00005.fa.csv	0.42	0.70	0.52	184
RF00007.fa.csv	0.00	0.00	0.00	4
RF00017.fa.csv	0.77	0.80	0.79	230
RF00019.fa.csv	0.95	0.40	0.56	500
RF00026.fa.csv	1.00	0.99	1.00	1137
RF00029.fa.csv	1.00	1.00	1.00	677
RF00032.fa.csv	0.00	0.00	0.00	0
RF00059.fa.csv	0.94	1.00	0.97	635
RF00097.fa.csv	0.88	1.00	0.94	1163
RF00100.fa.csv	0.00	0.00	0.00	24
RF00163.fa.csv	0.00	0.00	0.00	0
RF00174.fa.csv	1.00	0.73	0.84	37
RF00177.fa.csv	0.00	0.00	0.00	63
RF00230.fa.csv	0.00	0.00	0.00	0
RF00436.fa.csv	0.00	0.00	0.00	0
RF00906.fa.csv	0.56	0.70	0.63	549
RF00994.fa.csv	0.00	0.00	0.00	0
RF01315.fa.csv	0.00	0.00	0.00	0
RF01317.fa.csv	0.00	0.00	0.00	0
RF01787.fa.csv	0.96	0.80	0.87	485
RF01960.fa.csv	0.82	1.00	0.90	42
RF02271.fa.csv	0.00	0.00	0.00	0
RF02541.fa.csv	0.00	0.00	0.00	37
RF02543.fa.csv	0.80	0.72	0.76	54
RF04021.fa.csv	0.84	1.00	0.91	837
RF04088.fa.csv	1.00	1.00	1.00	1
human_mrna.csv	0.67	0.67	0.67	3
virus_mrna.fasta.csv	1.00	1.00	1.00	9
micro avg	0.88	0.88	0.88	7953
macro avg	0.47	0.47	0.46	7953
weighted avg	0.86	0.88	0.86	7953

Figure 6: Results for sequence of length > 75 and < 125



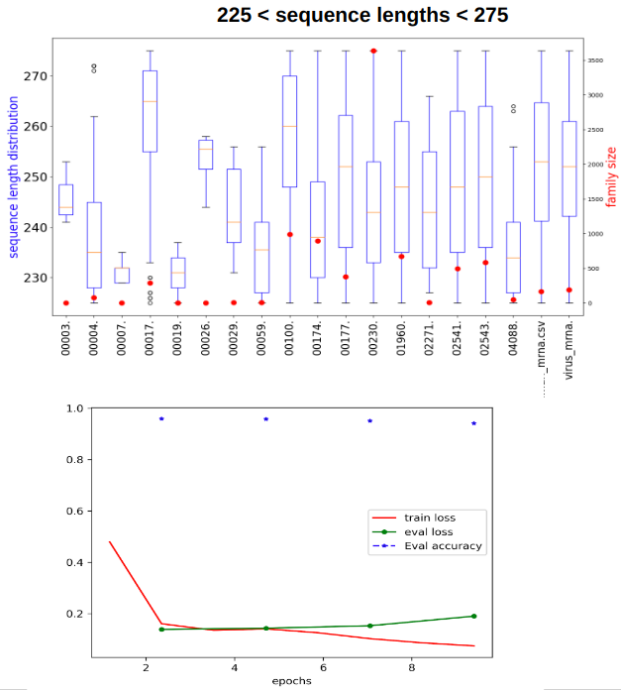
	precision	recall	f1-score	support
RF00001.fa.csv	0.00	0.00	0.00	36
RF00003.fa.csv	0.77	1.00	0.87	734
RF00004.fa.csv	1.00	1.00	1.00	219
RF00005.fa.csv	0.00	0.00	0.00	0
RF00007.fa.csv	0.71	1.00	0.83	32
RF00017.fa.csv	0.96	1.00	0.98	682
RF00019.fa.csv	0.00	0.00	0.00	6
RF00026.fa.csv	1.00	0.75	0.86	4
RF00029.fa.csv	0.85	0.46	0.60	50
RF00032.fa.csv	0.00	0.00	0.00	0
RF00059.fa.csv	1.00	0.71	0.83	95
RF00097.fa.csv	0.00	0.00	0.00	0
RF00100.fa.csv	1.00	0.94	0.97	16
RF00163.fa.csv	0.00	0.00	0.00	0
RF00174.fa.csv	0.99	0.70	0.82	158
RF00177.fa.csv	1.00	1.00	1.00	49
RF00230.fa.csv	0.43	0.43	0.43	7
RF00436.fa.csv	0.00	0.00	0.00	0
RF00906.fa.csv	0.99	1.00	1.00	917
RF00994.fa.csv	0.00	0.00	0.00	0
RF01315.fa.csv	0.00	0.00	0.00	0
RF01317.fa.csv	0.00	0.00	0.00	0
RF01787.fa.csv	0.00	0.00	0.00	2
RF01960.fa.csv	0.50	0.01	0.02	94
RF02271.fa.csv	0.88	1.00	0.94	241
RF02541.fa.csv	0.00	0.00	0.00	96
RF02543.fa.csv	0.67	0.99	0.80	99
RF04021.fa.csv	0.00	0.00	0.00	0
RF04088.fa.csv	1.00	0.89	0.94	38
human_mrna.csv	1.00	1.00	1.00	13
virus_mrna.fasta.csv	0.00	0.00	0.00	4
micro avg	0.90	0.90	0.90	3592
macro avg	0.48	0.45	0.45	3592
weighted avg	0.87	0.90	0.87	3592

Figure 7: Results for sequence of length > 125 and < 175



	precision	recall	f1-score	support
RF00001.fa.csv	0.00	0.00	0.00	0
RF00003.fa.csv	0.00	0.00	0.00	9
RF00004.fa.csv	1.00	1.00	1.00	687
RF00005.fa.csv	0.00	0.00	0.00	0
RF00007.fa.csv	0.00	0.00	0.00	4
RF00017.fa.csv	1.00	1.00	1.00	9
RF00019.fa.csv	0.00	0.00	0.00	0
RF00026.fa.csv	0.00	0.00	0.00	2
RF00029.fa.csv	0.00	0.00	0.00	2
RF00032.fa.csv	0.00	0.00	0.00	0
RF00059.fa.csv	1.00	0.73	0.84	11
RF00097.fa.csv	0.00	0.00	0.00	0
RF00100.fa.csv	0.93	0.98	0.95	51
RF00163.fa.csv	0.00	0.00	0.00	0
RF00174.fa.csv	0.89	1.00	0.94	599
RF00177.fa.csv	1.00	1.00	1.00	54
RF00230.fa.csv	0.85	0.40	0.55	303
RF00436.fa.csv	0.00	0.00	0.00	0
RF00906.fa.csv	1.00	1.00	1.00	1
RF00994.fa.csv	0.00	0.00	0.00	0
RF01315.fa.csv	0.00	0.00	0.00	0
RF01317.fa.csv	0.00	0.00	0.00	0
RF01787.fa.csv	0.00	0.00	0.00	0
RF01960.fa.csv	0.97	1.00	0.98	64
RF02271.fa.csv	0.86	1.00	0.92	1101
RF02541.fa.csv	0.87	1.00	0.93	68
RF02543.fa.csv	1.00	0.27	0.42	104
RF04021.fa.csv	0.00	0.00	0.00	0
RF04088.fa.csv	0.99	1.00	1.00	1248
human_mrna.csv	1.00	0.88	0.93	16
virus_mrna.fasta.csv	0.00	0.00	0.00	21
micro avg	0.93	0.93	0.93	4354
macro avg	0.43	0.40	0.40	4354
weighted avg	0.92	0.93	0.91	4354

Figure 8: Results for sequence of length > 175 and < 225



	precision	recall	f1-score	support
RF00001.fa.csv	0.00	0.00	0.00	0
RF00003.fa.csv	0.00	0.00	0.00	0
RF00004.fa.csv	1.00	0.89	0.94	9
RF00005.fa.csv	0.00	0.00	0.00	0
RF00007.fa.csv	0.00	0.00	0.00	1
RF00017.fa.csv	1.00	1.00	1.00	30
RF00019.fa.csv	0.00	0.00	0.00	0
RF00026.fa.csv	1.00	1.00	1.00	1
RF00029.fa.csv	0.00	0.00	0.00	0
RF00032.fa.csv	0.00	0.00	0.00	0
RF00059.fa.csv	0.00	0.00	0.00	0
RF00097.fa.csv	0.00	0.00	0.00	0
RF00100.fa.csv	0.99	1.00	1.00	115
RF00163.fa.csv	0.00	0.00	0.00	0
RF00174.fa.csv	0.99	0.80	0.89	101
RF00177.fa.csv	1.00	0.98	0.99	52
RF00230.fa.csv	0.97	1.00	0.98	445
RF00436.fa.csv	0.00	0.00	0.00	0
RF00906.fa.csv	0.00	0.00	0.00	0
RF00994.fa.csv	0.00	0.00	0.00	0
RF01315.fa.csv	0.00	0.00	0.00	0
RF01317.fa.csv	0.00	0.00	0.00	0
RF01787.fa.csv	0.00	0.00	0.00	0
RF01960.fa.csv	1.00	1.00	1.00	71
RF02271.fa.csv	0.00	0.00	0.00	0
RF02541.fa.csv	1.00	1.00	1.00	61
RF02543.fa.csv	0.79	0.99	0.88	74
RF04021.fa.csv	0.00	0.00	0.00	0
RF04088.fa.csv	0.83	1.00	0.91	5
human_mrna.csv	0.87	1.00	0.93	20
virus_mrna.fasta.csv	0.00	0.00	0.00	15
micro avg	0.96	0.96	0.96	1000
macro avg	0.37	0.38	0.37	1000
weighted avg	0.95	0.96	0.95	1000

Figure 9: Results for sequence of length > 225 and < 275