Project Report : CS 7643
A comparative analysis of RNA secondary structure prediction models

Karen Gaffney
Georgia Institute of Technology
kgaffney3@gatech.edu

Evan Hao
Georgia Institute of Technology
ehao6@gatech.edu

Ian Hsiao
Georgia Institute of Technology
ihsiao9@gatech.edu

Abstract

RNA secondary structure prediction is a crucial area of research in RNA vaccines. However, predicting the 2-D structure is a challenging problem due to the number of possible configurations of RNA bonds and the sequential nature of RNA. Traditional prediction methods like ViennaFold, use thermodynamic principles but struggle with complex structures. Recently, deeplearning has shown promise in advancing the RNA secondary structure prediction field by leveraging transformer and LSTM architectures that are effective for sequence-based tasks while also incorporating thermodynamic principles. In this study, we performed a comparative analysis of RNA secondary structure models; EternaFold, MXFold2, and Vienna ViennaFold as our benchmark. We evaluate these models on the basis of their performance across different sequence lengths and structural motifs. In addition, we performed hyperparameter tuning to optimize accuracy and explored a hyper ensemble model using a consensus structure approach.

1. Introduction/Background/Motivation

In this study, we aimed to capture the performance of new deep learning models in RNA secondary structure prediction in comparison to Vienna RNA fold, a thermodynamic model. And create a hyper ensemble that combines the predictions of ViennaFold, EternaFold, and MXFold2 to provide a consensus prediction. Understanding RNA folding is essential in prediction RNA function, with applications in fields like vaccine development and disease research. Double stranded RNA, a result of RNA with secondary structure, is a known inducer of immune-inflammatory reactions and is the suspected cause of unexplained myocarditis cases in COVID-19 vaccinations [2]. In addition, biological mechanisms such as cellular RNA transcription are regulated by the secondary structure of T7 terminators. Discovering and predicting cryptic T7 terminators are essential to understanding RNA transcription in RNA-based diseases and RNA production for vaccine research [3]. However, predicting RNA structures is challenging due to the vast number of possible ways RNA can form bonds and fold sequentially (Figure 1).

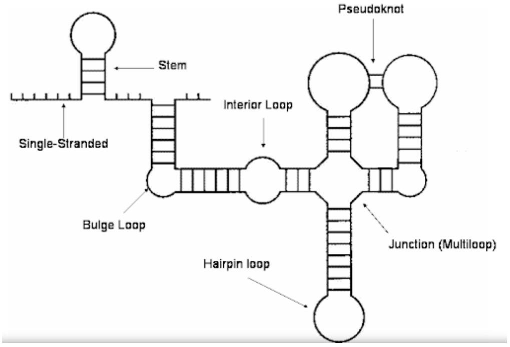

Figure 1. RNA secondary structure categories [1]

Traditional tools, like ViennaFold, use thermodynamic calculations to find the most stable structures. Specifically, ViennaFold predicts the structure that corresponds to the minimum free energy (MFE) configuration — the most stable folding arrangement possible under given conditions. ViennaFold evaluates all potential pairings (or bonds) between nucleotides in an RNA sequence and calculates the free energy for each possible structure. The structure with the lowest free energy is considered the most stable and is predicted as the RNA's likely folding pattern [4]. However, ViennaFold struggles to predict long RNA sequences and may take a long time to run due to the computational limits of dynamic programming. In addition, because ViennaFold relies purely on thermodynamics, it cannot learn from large datasets to capture patterns missed by thermodynamic calculations.

To study the performance of deep learning models compared to ViennaFold, we choose Eternafold and MXFold2 as they are both published deep learning RNA secondary structure prediction libraries that have published papers. EternaFold EternaFold is a deep learning-based RNA secondary structure prediction model that uses a Transformer architecture [5]. Transformers are effective for sequence-based tasks such as RNA secondary structure prediction because they capture long-range dependencies through self-attention mechanisms. EternaFold benefits from learning patterns within large datasets and integrating this knowledge to predict complex structures, such as pseudoknots and intricate motifs, which traditional thermodynamic models struggle with. Additionally, it uses ensemble learning, combining multiple predictions for greater accuracy.

MXFold2 combines deep learning with thermodynamic modeling to predict RNA secondary structures. Its architecture integrates Bidirectional Long Short-Term Memory (BiLSTM) networks with an attention mechanism, capturing relationships between distant nucleotides. MXFold2 integrates its classification with established thermodynamic principles for physically accurate predictions [6]. This hybrid approach allows MXFold2 to improve accuracy, especially for challenging structures like pseudoknots.

For our analysis we used the datasets provided on Eterna Fold's github [7]. We chose the EternaFold dataset for our experiments because it provides a diverse and high-quality collection of RNA sequences specifically designed for RNA secondary structure prediction. This dataset includes a wide range of RNA lengths, structural motifs (like hairpins, internal loops, and pseudoknots), and annotations, making it suitable for a comprehensive evaluation of different prediction models. The dataset is curated from community-driven experiments through the Eterna project and contains 1533 structures in training and 652 structures in testing. This diversity allows us to test the robustness of our models, MXFold2 and ViennaFold across various motifs. Additionally, using a single, consistent dataset helps maintain fair comparisons between models and ensures that any differences in performance are due to the models themselves and not inconsistencies in the data.

By capturing the RNA secondary structure prediction performance of deep learning models and traditional thermodynamic energy models over a variety of conditions, we hope to enable researchers to select the secondary structure prediction model that best suits their needs. In addition, by building a hyper ensemble that combines all three predictive models we hope to further increase the accuracy of our predictions.

2. Approach

In this study, we evaluated the performance of EternaFold [7], MXFold2 [9], and ViennaFold [10] under various conditions. After installing each package and confirming functionality, we trained the models on Eternafold's train A data set and took the following approach:

Prediction Evaluation: We assessed the accuracy of each model on predicting RNA structures of different sequence lengths and structural motifs, such as hairpins, pseudoknots, and internal loops, on Eternafold's test A dataset. We also measured the accuracy of each model across a variety of sequence lengths. To ensure consistency, we used the EternaFold dataset, which contains diverse RNA sequences with detailed annotations. To measure performance we used two forms of accuracy. To look at whole sequence accuracy we used F1 accuracy as it is a standard metric to evaluate secondary structure predictive success.The F1 score is the harmonic mean of precision and recall. Where precision is the proportion of correctly predicted positive instances out of all instances that were predicted as positive and recall is the proportion of correctly predicted positive instances out of all actual positive instances in the data (Equations 1). To look at accuracy based on motif we took base pair accuracy as we needed to separate our accuracy across the different regions present in each sequence.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1a}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{1b}$$

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1c}$$

Equations 1. Precision, Recall, and the F1 Score, which measure the balance between precision and recall in classification tasks.

Hyperparameter Tuning: Through iterative testing we optimized the hyperparameters of EternaFold and MXFold2 (e.g., learning rate, batch size, and dropout rate) to improve prediction accuracy.

Ensemble Method: To leverage the combined knowledge of the predictions of ViennaFold, EternaFold, and MXFold2. This ensemble approach creates a consensus structure by taking the mode of all three predicted structures at each base pair position. We hypothesized that taking the most common prediction at each base pair position would improve robustness and accuracy by leveraging the confidence of each individual model. This novel approach of combining the pattern predicting power of Deep Learning models with traditional free energy only models may lead to a predictive model that is well balanced across complex motifs and sequence length (Figure 2). In addition we attempted a second hyperensemble that utilized a transformer model to intake the input sequence, and the model predictions of ViennaFold, EternaFold, and MXFold2 (Figure 3). We hypothesized that this transformer model would further outperform our mode based model as it has a larger attention window to evaluate the sequence and learn from the patterns of the three models predictions.

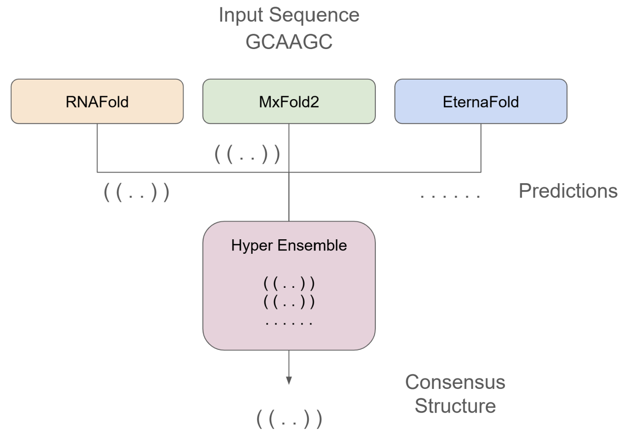

Figure 2. Our proposed mode hyper ensemble takes the mode at each base position and returns a consensus prediction

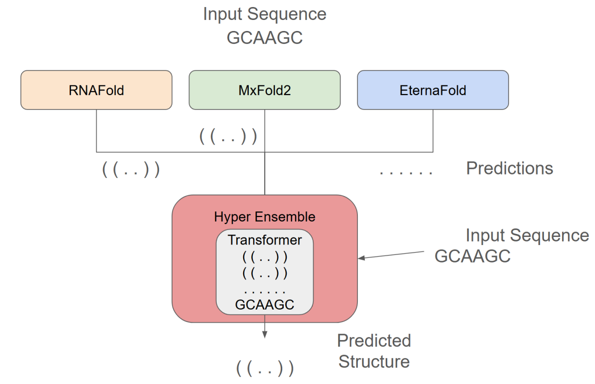

Figure 3. Transformer hyper ensemble takes in all three model predictions and the input sequence

Anticipated Problems and Challenges: We anticipated challenges in optimizing the deep learning models due to their sensitivity to hyperparameters and need for large data sets. In comparison to other structure prediction fields such as protein folding, RNA structure prediction is an emerging field that does not have as many annotated datasets as protein folding [8]. Additionally, we expected ViennaFold to struggle with longer sequences and complex motifs due to its reliance on free energy calculations.

We anticipated that our hyper ensemble may encounter issues in generating a physically possible structure each time. Because we are taking the mode at each base position, two out of three model predictions must agree to determine the structure at each position. For example, in cases where two models predict the start of a loop at the same time, but disagree where it closes, a consensus structure could be formed where there are open loops. Because both ViennaFold and MXFold2 both utilize thermodynamic principles, we hypothesized that this outcome would be unlikely and can be avoided. In addition, our transformer based hyper ensemble model could overcome this limitation by learning patterns from correctly predicted structures and avoid open loops.

3. Experiments and Results

3.0.1 Experiment 1: Base pair accuracy across sequence length

To capture the performance of the three models across sequence length we predicted secondary structure for the whole EternaFold test set which encompassed sequences of length 5 to 670 bases. Then we calculated the F1 accuracy for each sequence as the harmonic mean of the precision and recall for each classification. For each model, F1 accuracy trended lower as sequence length increased. This was expected as longer sequence lengths can lead to more complex patterns that become difficult to classify (Figure 4). In addition, we saw that MXFold2 consistently outperformed the other two models across all sequence lengths. Surprisingly, EternaFold and Vienna fold both underperformed compared to MXFold2 although EternaFold incorporated a deep learning architecture and was testing on it's own published dataset. On average MXFold2 performed better than the other two models across each bin of sequences lengths, scoring the highest accuracy in the shortest 50-100 base pair batch (Table 1).

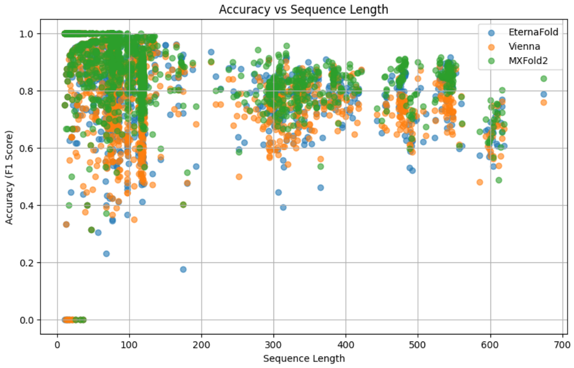

Figure 4. The F1 accuracy of each model across sequence length

3.0.2 Experiment 2: Base pair accuracy across motifs

To test the base pair accuracy across motifs we first batched our test set groups based on their structures. Dot Bracket notation is commonly used to represent RNA secondary

Model	0-50	50-100	100-200	200-400	400-500	500+
EternaFold	0.8897	0.7289	0.7696	0.6340	0.6009	0.6175
ViennaFold	0.8692	0.7208	0.6984	0.5859	0.4870	0.5072
MXFold2	0.9337	0.9027	0.8347	0.7076	0.6908	0.6626

Table 1. Average accuracy of different models across sequence length bins.

structure. A ”.” is an unpaired base, ”(” and ”)” are paired bases that open and close looped areas. We classified three basic groups (hairpin loop, unpaired, combination) in our test data according to the most common RNA secondary structure motifs [11] as described in Table 2. Then we ran each group through each model and computed their base pair accuracy. The total number of correctly classified bases divide by the total sequence length. We found that once again Mxfold performed above all other models in all three categories (Figure 5). Overall the models predicted the best across unpaired regions, and Eterna fold had difficulties predicting in hairpin loop regions.

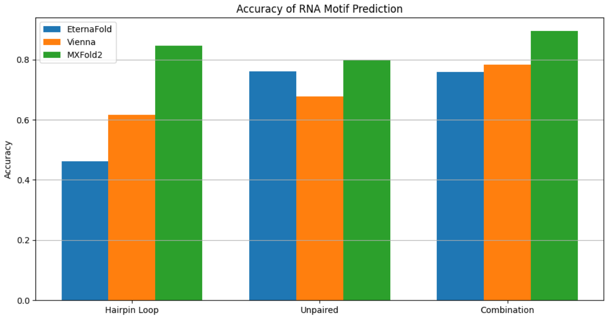

Figure 5. Base accuracy across different motifs

Motif	Dotbracket Notation
Hairpin Loop	(((...)))
Unpaired	.....
Combination	any combination of the above

Table 2. Dotbracket Notation of common RNA secondary structures

3.0.3 Experiment 3: Improved Accuracy after hyper parameter tuning

After experimentation we found that the default hyper parameters found by Steele et al [5] were already the optimal hyper parameters for the dataset. In MXFold2, a grid search was used with the following hyperparameters to find the lowest train loss on Eternafold's training set. The following hyperparameters were adjusted:

Max Helix Length: Defines the maximum length of a helix (a contiguous stack of base pairs), this was increased to allow flexibility for longer sequences.

Embedding Size: The dimensionality of the input sequence embeddings. Larger embedding sizes allow for more complex features but possibly overfit on smaller data sets. Setting this to 0, relying solely on the architecture was beneficial for Eternafold's dataset.

Number of Filters/Filter Size: Convolution layers were used to find local patterns in the RNA sequence. Increasing the number of filters to 96 while also increasing the filter size to 5 allowed the model to discover relevant motifs.

Pooling Size: Maintaining a pooling size of 1 was able to retain the maximum spatial resolution.

Transformer Layers: Used to capture between distant nucleotides that are needed for RNA secondary structure prediction.

Transformer Hidden Units/Attention Head: Transformer hidden units define the size of internal presentations and attentions heads allow the model to focus on multiple interactions simultaneously. 2048 hidden units with 8 attention heads was the best for capturing complexity with the RNA sequences.

LSTM Layers/Units: LSTM layers model sequential dependencies in RNA sequences. Optimally, both were set to 0.

Dropout Rates: Dropout rate regularization used to deactivate a portion of neurons during training to prevent overfitting. Setting both the dropout and connected dropout to 0.0 yielded the best as the data set was diverse enough that overfitting was not a concern with the dataset.

Pair Joining Method: Specifies how features from paired nucleotides are combined. Using the concatenation method (*cat*) preserved maximal information from paired interactions, leading to improved accuracy.

3.0.4 Experiment 4: Accuracy of hyper ensemble predictions

For the hyper-ensemble model, we aimed to evaluate whether base-pairing accuracy could be enhanced by aggregating the outputs of three individual models: Vienna, MXFold2, and Eterna. Our initial approach involved taking the mode of the predictions at each index, or selecting randomly from the models in cases where no mode was present. This approach yielded an accuracy of 0.8269 which performed above EternaFold and Viennafold but was outpeformed by MXFold2.

To further refine the hyper-ensemble model, we employed a transformer architecture, leveraging its capability to process variable-length inputs and its multi-head attention mechanism. This design choice was motivated by the need for the model to capture long-range dependencies within sequences. Accurate base pairing requires that if a base at one index is paired, its corresponding index must also be paired to maintain structural validity. Consequently, the model must consider multiple parts of the sequence simultaneously when predicting folding patterns. This approach produced an accuracy of 0.7569 which underperformed all other models. We believe the model's performance was suboptimal due to the data-intensive nature of transformers, and the limited size of our training dataset likely constrained its effectiveness.

Model	Test Accuracy
EternaFold	0.8030
ViennaFold	0.7778
MXFold2	0.8667
Mode Hyper Ensemble	0.8294
Transformer Hyper Ensemble	0.7569

Table 3. Average accuracy of different models across sequence length bins.

3.1. Conclusions

In conclusion, MXFold2 demonstrated the most consistent and accurate performance in RNA secondary structure prediction across sequence lengths and structural motifs, outperforming both EternaFold and ViennaFold. While EternaFold's deep learning architecture showed potential, it struggled with hairpin loop regions, emphasizing the need for further refinement. Hyperparameter tuning revealed that MXFold2's default settings were nearly optimal, with specific adjustments enhancing its performance. The mode based hyper ensemble approach outperformed both EternaFold and ViennaFold but underperformed compared to MXFold2, hinting that the non MXFold2 models are dragging down the accuracy. The transformer hyper-ensemble approach, which aggregated predictions from all three models, showed promise but was limited by the data-intensive nature of transformer architectures and the small dataset size. These findings underscore the challenges of RNA structure prediction, particularly compared to more mature fields like protein folding, where annotated datasets are abundant. Future efforts should focus on expanding high-quality annotated datasets and exploring more data-efficient transformer architectures to better capture long-range dependencies in RNA sequences. These advancements have the potential to significantly impact the field of medicine by improving the understanding and design of RNA-based therapeutics, vaccines, and diagnostics, accelerating innovation in RNA-focused research and applications.

4. Explanation of Models

4.1. Overview of ViennaFold, EternaFold, and MXFold2

ViennaFold employs a physics-based, thermodynamic approach combined with dynamic programming to predict RNA structures that minimize free energy. This model explicitly encodes the free-energy rules and constraints of RNA folding. Unlike modern approaches, ViennaFold does not utilize deep learning and serves as a baseline for evaluating models that incorporate neural network architectures.

EternaFold and MXFold2 integrate deep learning architectures, employing neural networks to analyze RNA sequences and extract features indicative of pairwise motifs. These models leverage large datasets of known RNA structures, enabling them to capture intricate features and correlations that are challenging for traditional thermodynamic models.

4.2. Model Parameters

ViennaFold operates without learned parameters; all parameters are fixed and determined by thermodynamic rules. Post-processing steps in ViennaFold rely on deterministic algorithms.

In contrast, EternaFold and MXFold2 utilize learned parameters embedded within the neural network layers. These parameters include weights in convolutional and recurrent layers, attention mechanisms, and fully connected layers that map intermediate representations to predicted pairing probabilities. Parameter optimization is achieved through training to enhance prediction accuracy.

For EternaFold, non-learned components involve converting predicted probabilities into final RNA structures and performing post-processing to derive discrete secondary structures. Similarly, MXFold2 employs post-processing steps that optimize folding predictions using probabilities generated by the model.

4.3. Input and Output Representations

EternaFold, ViennaFold, and MXFold2 require distinct input formats for training and prediction, necessitating preprocessing to ensure dataset compatibility across models. Specifically, data transformations between .bpseq, .fasta, and .dbn file formats were essential to maintain consistent workflows.

EternaFold and MXFold2 require .bpseq files for training. For predictions, MXFold2 accepts .fasta files, while EternaFold also relies on .bpseq files. ViennaFold, as a non-machine learning model, does not require training and directly accepts plain RNA sequences for predictions.

These preprocessing steps were crucial for standardizing input formats and ensuring seamless integration across models.

4.4. Overfitting and Generalization

EternaFold and MXFold2 employ hyperparameter tuning and regularization techniques to achieve robust generalization while mitigating overfitting.

4.5. Hyperparameters

EternaFold and MXFold2 utilize distinct sets of hyperparameters to optimize their performance:

- **EternaFold Hyperparameters:** holdout cross-validation, regularization coefficient, maximum number of iterations for regularization coefficient optimization, initial weight settings for regularization, batch size, and step size for stochastic gradient descent (SGD).
- **MXFold2 Hyperparameters:** epochs, optimizers, L1 and L2 regularization weights, learning rate, maximum helix length, embedding size, pooling size, dilation rate, number of filters, filter size, transformer layers, transformer hidden units, attention heads, LSTM layers, LSTM hidden units, dropout rates, and pair joining methods.

4.6. Deep Learning Frameworks

EternaFold and MXFold2 were implemented using modern deep learning frameworks, with TensorFlow and PyTorch serving as the foundational libraries for their development.

5. Work Division

See Table 4.

6. Supplementary Information

The code used to train and test these models and generate the figures is available at https://github.com/ianjjhsiao/RNA-Secondary-Structure-Prediction.git . The datasets used in this paper are available at https://github.com/eternagame/EternaFold .

7. References

References

[1] RpredictorDB Team. (n.d.). *Structural features of RNA secondary structure*. Retrieved from http://rpredictordb.elixir-czech.cz/documentation/rDoc-User/structural-features.html

[2] Milano, G., Gal, J., Creisson, A., & Chamorey, E. (2021). Myocarditis and COVID-19 mRNA vaccines: A mechanistic hypothesis involving dsRNA. *Future Virology*, *10*, 2217. https://doi.org/10.2217/fvl-2021-0280

[3] Macdonald, L. E., Zhou, Y., & McAllister, W. T. (1993). Termination and slippage by bacteriophage T7 RNA polymerase. *Journal of Molecular Biology*, *232*(4), 1030–1047. https://doi.org/10.1006/jmbi.1993.1458

[4] Lorenz, R., Bernhart, S. H., Höner zu Siederdissen, C., & Hofacker, I. L. (2011). ViennaRNA Package 2.0. *Algorithms for Molecular Biology*, *6*, 26. https://doi.org/10.1186/1748-7188-6-26

[5] Wayment-Steele, H. K., Kladwang, W., Strom, A. I., & Das, R. (2022). RNA secondary structure packages evaluated and improved by high-throughput experiments. *Nature Methods*, *19*, 1234–1242. https://doi.org/10.1038/s41592-022-01605-0

[6] Sato, K., Akiyama, M., & Sakakibara, Y. (2021). RNA secondary structure prediction using deep learning with thermodynamic integration. *Nature Communications*, *12*, 941. https://doi.org/10.1038/s41467-021-21194-4

[7] EternaFold Team. (n.d.). *EternaFold: RNA secondary structure prediction*. GitHub. Retrieved from https://github.com/eternagame/EternaFold

[8] Zhang, J., Fei, Y., Sun, L., & Das, R. (2022). Advances and opportunities in RNA structure experimental determination and computational modeling. *Nature Methods*, *19*, 1193–1207. https://doi.org/10.1038/s41592-022-01623-y

[9] Suzuki, J., & Yamaguchi, K. (2020). *MXFold2: RNA secondary structure prediction using deep learning with thermodynamic integration*. GitHub. Retrieved from https://github.com/mxfold/MXFold2

[10] Lorenz, R., & Hofacker, I. (2011). *ViennaRNA Package: RNA secondary structure prediction and comparison*. GitHub. Retrieved from https://github.com/ViennaRNA/ViennaRNA

[11] Achar, A., & Sætrom, P. (2015). RNA motif discovery: A computational overview. *Biology Direct*, *10*, 61. https://doi.org/10.1186/s13062-015-0090-5

Student Name	Contributed Aspects	Details
Karen Gaffney	Approach, Experimentation, Introduction, ViennaFold Analysis	Contributed to the experimentation section by analyzing results and creating graphs. Wrote the introduction, setting the context and goals of the project. Trained the ViennaFold and analyzed results.
Evan Hao	Experimentation, Explanation of Models, MXFold2 Training and testing	Wrote the Explanation of models section and contributed the experimentation section. Trained MXFold2, analyzing its performance, and contributed to the graphs.
Ian Hsiao	Experimentation, Abstract, data prep, EternaFold Training and testing	Wrote the abstract and approach section. Prepped data for training and trained EternaFold and contributed to the experiments section, creating graphs to compare the models' performances.

Table 4. Contributions of team members.