RNA 3D Structure Prediction via GNN

Goal: Predict 3D nucleotide coordinates from RNA sequences using a Graph Neural Network (GNN) enhanced by structural and evolutionary features.

MODEL PERF. TM-score ~ 0.8!!! (1 = perfect alignment of the predicted vs true structure)

1. Introduction

Predicting the three-dimensional (3D) structure of RNA from sequence remains a central challenge in structural biology. RNA folds into intricate 3D conformations driven by base-pairing, stacking, and tertiary interactions. Although high-resolution experimental methods (X-ray crystallography, cryo-EM) produce accurate structures, they are laborious and not feasible at genomic scale. Computational approaches ranging from physics-based simulations to comparative modeling struggle with large, flexible RNAs or novel folds. Recently, deep learning techniques, especially Graph Neural Networks (GNNs), have demonstrated promise for modeling biomolecular structures by directly learning from large datasets. In this project, we harness a GNN to predict 3D coordinates of nucleotides in an RNA chain using a combination of sequence-derived and evolutionary features. Our ground-truth data come from a CSV file containing each molecule's name, primary sequence, and experimentally determined target coordinates. We augment this with secondary structure predictions, k-mer composition, thermodynamic MFE, and multiple-sequence-alignment (MSA) derived signals to build a rich feature set. By integrating these modalities into a graph representation, we train a GNN to output per-residue 3D coordinates with a loss function focused on preserving pairwise distances. This report details of data preparation from CSV, feature extraction pipelines, graph construction, GNN architecture and training, evaluation metrics (distance loss, TM-score) and experimental findings and future directions.

2. Background

2.1. RNA Folding Principles

RNA molecules fold hierarchically from a linear sequence of nucleotides (A, U, G, C) into complex three-dimensional structures. The folding process begins with the formation of secondary structures, such as stems, loops, and bulges, stabilized by base pairing (Watson–Crick and wobble pairs). These secondary elements further organize into tertiary structures through long-range interactions and stacking, enabling the RNA to achieve its functional conformation. Understanding these principles is essential for modeling and predicting RNA 3D structures accurately.

2.2 Why Graph Neural Networks?

RNA folds can be naturally represented as graphs: each nucleotide is a node, connected to its sequential neighbors (nucleotide *i* to *i+1*) and to non-local partners (base-paired positions or co-evolving residues). GNNs work on graph-structured data, where nodes (nucleotides) connect via edges (backbone links, base pairs, co-evolutionary contacts). In each message-passing layer, a node gathers feature vectors ("messages") from its immediate neighbors, aggregates them (e.g. by summation), and passes the result through a small neural network to update its own embedding. Stacking L such layers lets information propagate up to L hops away: after one layer, each node knows about its direct neighbors; after two layers, it also knows about neighbors' neighbors, and so on. By the final layer, each node's representation encodes a summary of its entire local subgraph. In our RNA application, that means each nucleotide's embedding reflects its own nucleotide identity plus the identities and structural or evolutionary signals of all bases directly or indirectly connected by edges.

Here, for each RNA molecule, we will build one GNN graph per row (per molecule) by concatenating all features into node embeddings and defining edges from sequence, secondary structure, and MSA that we will feed each graph into the same GNN model (via a DataLoader loop), which predicts a matrix of coordinates. The GNN naturally enforces relational constraints (edges) that help capture conserved tertiary interactions. In other words, it models the relative geometric constraints between residues so that their final coordinates maintain those pairwise relationships.

3. Data and Preprocessing

3.1. Input Data: CSV file with 1 RNA molecule per rows, each containing: name(RNA identifier), sequence: RNA primary sequence (only A, U, G, C retained), target_coordinates: list of (x, y, z) positions per nucleotide and other description that we will not use for features engineering

3.2. We filtered any rows with primary sequences containing non-AUGC residues and NaNs or invalid coordinates

3.3. Standardization: Coordinates are centered at the origin by subtracting the mean and then dividing by std to minimize scale variance.

4. Feature Engineering

4.1. L = length of the primary sequence (number of nucleotide for each molecule)

4.2. Primary Sequence Encoding: One-hot vector per base: A, U, G, C → [1,0,0,0], [0,1,0,0], etc.

4.3. K-mer Composition: Trinucleotide frequency vector, tiled across sequence length. K-mer composition captures short, recurring nucleotide patterns that reflect local structural or functional motifs in RNA. These patterns provide context beyond individual bases, helping the model recognize biologically meaningful sequence elements relevant to folding.

4.4. Secondary Structure (via RNAfold):

- Dot-bracket notation → vectorized: this encodes RNA secondary structure by indicating which nucleotides are paired or unpaired, capturing local folding patterns. Vectorizing the dot-bracket format allows the GNN to interpret structural roles (paired left, paired right, or unpaired) as node features.

- MFE (Minimum Free Energy): this scalar represents the predicted thermodynamic stability of the RNA's secondary structure. Lower MFE values indicate more stable folds and provide global context about the molecule’s structural compactness.

- BPPM (Base Pair Probability Matrix): This L × L matrix gives the probability of each nucleotide pair forming a base pair, derived from thermodynamic ensemble predictions. It captures uncertainty and flexibility in folding and is used to add edges between non-adjacent but structurally interacting residues.

4.5. MSA-Derived Features from fasta files:

MSA (Multiple Sequence Alignment) fasta files are created by aligning homologous RNA sequences from different organisms using tools like BLAST, Infernal, or RNAcmap. These tools align sequences to highlight conserved positions and co-evolving regions, producing a matrix where each column represents a nucleotide position across aligned sequences.

- Entropy Vector (L × 1) captures how conserved each position is across the alignment; lower entropy means higher conservation, often indicating structural or functional importance.

- PSSM (L × 5) represents the frequency of each nucleotide (A, U, G, C, gap) at each position, providing a detailed view of allowed base variability.
- MI Matrix (L × L) quantifies the statistical dependency between pairs of positions, identifying co-evolving residues that may form structural contacts in 3D.

5. Graphs construction

For each molecule we will build a graph with:

- Nodes (L): each node in the graph corresponds to a single nucleotide residue with its associated features:
  - one-hot primary sequence
  - K-mer
  - Dot-bracket notation
  - MFE
  - Entropy
  - PSSM

- Edges:
  - Sequential (i ↔ i+1): These edges connect each nucleotide to its immediate neighbors along the RNA backbone. They capture the linear structure of the molecule and ensure continuity in the graph.

  - Base Pair Probability Matrix (BPPM): Edges are added between residues *i* and *j* if the predicted probability of them forming a base pair exceeds 0.5. This encodes likely secondary structure interactions derived from RNAfold.

  - Mutual Information (MI): Edges are included between residues *i* and *j* if their mutual information exceeds 0.1, indicating co-evolutionary relationships. The edge weight reflects the MI value, emphasizing the strength of the correlation.

6. GNN Model

Here, Graph Neural Networks (GNNs) work by treating each RNA molecule as a graph where nodes represent nucleotides and edges represent their structural or evolutionary relationships. At each GNN layer, a node updates its feature vector (a tensor) by aggregating information from its neighbors—this is called message passing. These node features are high-dimensional tensors that can include one-hot encodings, structural data, and MSA-derived metrics.

The model stacks several such layers, allowing information to propagate across the graph. Finally, a fully connected layer maps the final node embeddings to 3D coordinate predictions, represented as tensors of shape (L × 3) where L is the sequence length. This process enables the GNN to predict the relative spatial positions of nucleotides based on both local and long-range interactions encoded in the graph.

Here, the RNAGNN model is a graph neural network designed to predict the 3D coordinates of RNA nucleotides from graph-structured input. It includes two hidden layers implemented using GCNConv, each followed by a ReLU activation to introduce non-linearity and enable the network to learn complex patterns in the data. These hidden layers update each node's embedding by aggregating information from its neighbors based on the input graph's edges (e.g., sequential connections, base pairs). After message passing, a final fully connected layer maps the learned node embeddings to 3D coordinates (x, y, z). The model processes tensor inputs for node features and edge indices, and outputs a tensor representing predicted spatial positions for each residue.

7. Loss & Evaluation

The training function uses a distance-based loss to supervise the learning of RNA 3D coordinates. Specifically:

- The distance_loss compares predicted and true pairwise distances between residues using their coordinate differences over the graph edges. For each edge (i, j), it computes the Euclidean distance between pred_i and pred_j, and compares it to the ground truth distance true_i to true_j using Mean Squared Error (MSE). This loss is invariant to global rotations and translations, encouraging the model to learn the relative geometry of the RNA structure rather than absolute positions.

- The model is optimized using the Adam optimizer, and TM-score is tracked to assess global structural accuracy.

This setup ensures the GNN focuses on maintaining realistic spatial relationships between connected residues during training.

8. Results

8.1. Training set analysis:

- General Trend: There is a clear downward trend in loss and a steady increase in TM-score across epochs, indicating that the model is learning meaningful representations of RNA structure over time.

- Early Training Instability: The training loss fluctuates notably in the first few epochs (e.g., high loss at Epoch 2 and 5), which may suggest sensitivity to learning rate or data complexity. Despite this, TM-score consistently improves, showing that structural alignment is becoming more accurate.

- Strong Convergence: By Epoch 10, the model starts to stabilize. From Epoch 15 onward, both the loss drops below 200 and the TM-score exceeds 0.81, which reflects strong structural predictions.

- Final Epochs: In Epochs 18–20, the model achieves its best performance:
    - Loss: ~150
    - TM-score: ~0.83

These results suggest the model is well-fitted on the training set without significant overfitting indicators.

8.2. Validation dataset

Average TM-score: 0.8159:

- This is very close to the final training TM-score (0.8316), indicating that the model generalizes well to unseen data.

- The small gap between training and validation TM-scores suggests minimal overfitting and effective regularization.

9. Conclusion

Model is successfully learning structural representations of RNA sequences and is capable of generalizing to validation data.

The loss decrease and TM-score increase confirm effective optimization. A few spikes in loss (Epochs 2, 5, 10) could be smoothed with a more refined learning rate schedule or additional regularization.

A TM-score > 0.8 is generally considered excellent in structure prediction tasks, implying that the model's coordinate predictions are structurally aligned with the true 3D shapes.

10. How to improve my model?

10.1. Dataset size issue:

Empirical observations and benchmarks across the literature and competitions:

- RNA structure prediction papers (e.g. SPOT-RNA, RGN, E2Efold-3D) typically use >1,000 structures to train deep models for 3D tasks.

- In the previous Stanford RNA 3D folding Kaggle competition, dataset size ranged from 1.4K to 140K molecules.
- QM9 molecule datasets (for 3D prediction) use ~100,000 examples for generalization.

- Smaller GNN tasks (e.g., citation networks like Cora) work with hundreds of graphs, but these are not structure regression tasks.

Larger dataset option:

- Train with a larger dataset (provided by the kaggle competition) with 3154 molecules. This will be very time consuming as MSA features extraction for 500 molecules takes up to 48H.

10.2. Rethink TM-score calculation:

Explicit 3D alignment of predicted and true RNA structures is crucial before computing TM-score because the TM-score evaluates global structural similarity, which is sensitive to spatial orientation and position. Without alignment, even correctly shaped structures may appear dissimilar due to arbitrary rotations or translations. By aligning the predicted structure to the reference (e.g., using Kabsch or US-align), we remove these rigid-body differences, ensuring that TM-score reflects true topological accuracy rather than misalignment artifacts.

We can code a new function that will use Kabsch or US-align to align the new predicted structures to the true structures. The predicted coordinates are rotated and translated to best match the true coordinates. Once aligned, metrics like TM-score can accurately assess how well the predicted structure captures the true 3D topology, independent of global position or orientation.

10.3. Enhancing 3D Prediction with Structure-Aware Embeddings

Incorporating structure-aware embeddings such as those from RiboBERT, ESM, or RNA-MSM can significantly enhance RNA 3D structure prediction by providing rich, contextual representations of nucleotide sequences. These models are pre trained on large corpora of RNA or protein sequences and capture both local sequence motifs and global structural dependencies. By embedding each residue with features informed by evolutionary patterns, base-pairing tendencies, and long-range interactions, these embeddings offer the GNN a deeper understanding of folding constraints that cannot be inferred from raw sequence alone. Integrating them into the node features can improve generalization, especially for novel or structurally diverse RNAs.