
End-to-end RNA Secondary Structure Prediction using Deep Neural Network

1 Introduction

Once believed to be an intermediate molecule that serves as a messenger between DNA and protein, RNA is now known to be involved in many aspects in gene regulation and expression. Unlike DNA which typically forms stable double helix between two molecules, RNA is highly flexible such that a single RNA molecule can fold onto itself by forming base pairs via hydrogen bonds, including Watson-Crick pairs A-U, G-C and non-canonical pairs such as G-U. Base pairs form the basis for local structures like stems and loops, which assemble into the global secondary structure.

State-of-the-art RNA secondary structure prediction algorithms, such as ViennaRNA[1] and Mfold[2], are based on the fundamental property of base-pairing. Each type of local structure comes with free energy that is measured experimentally. Total free energy is assumed to be the sum of all local free energy, and can be minimized efficiently using dynamic programming. Although researchers have worked out a large set of local structure types and their associated formulation for free energy calculation, there is no guarantee that it's either accurate or complete. There has been effort to fine tune the free energy parameters by training on experimentally solved structures[3], but it's still limited by the known set of local structure types.

Another limitation of dynamic programming based approaches is that it is extremely inefficient at predicting pseudoknot (TODO citation + existing work e.g. approximation).

Moreover, over the last decade, combination of RNA structure probing and high throughput sequencing has enabled the measurement of genome-wide RNA structural at single nucleotide resolution in multiple organisms and cell types (TODO citation). Due to the level of noise and missing data present in high throughput experiments, we will need a modelling approach that is more flexible and can be trained on different types of data.

In this work, we propose a deep neural network that can be trained end-to-end on sequences and structures. When conditioned on the input RNA sequence, the model can generate a distribution of structures, including ones with pseudoknot.

1.1 Problem Formulation

For a RNA sequence of length L , we are interested in all possible secondary structures it can take on. For a specific structure, there are three common ways to represent it:

- Undirected graph, where each node is a base in the sequence, and each vertex represent base pairing.
- Upper triangular matrix (excluding the diagonal) of size $L \times L$ with binary values, where a value of 1 at (i, j) represents base pairing between sequence position i and j , and 0 represents no base pairing.
- A dot-bracket notation of length L where unpaired bases are denoted as dots, and paired bases are represented as left and right brackets.

As an example, a short RNA sequence GUUGUGAAAU of length 10 (Entry CRW_00083 from [4]) takes a structure that consists of a stem and a loop, as seen in Fig 1(a), represented by an undirected

graph. This structure can also be represented by the upper triangular 10×10 matrix with all 0's, except for positions (1, 10), (2, 9) and (3, 8), all being 1, as shown in Fig 1(b). This contiguous stretch of 1's along the diagonal corresponds to the stem formed by the three base pairs: G-U, U-A and U-A. In Fig 1(c) we show the dot-bracket notation of this structure, where the three pairs of left-right brackets represent the stem.

2 Related Work

There have been a few work that solve partial problem....

Wu[6] presented a convolution neural network to predict the free energy of local structure motifs. Convolution was run on circular matrix representation of the structure motif, to reflect the loop-like nature of local structures. The model was trained on experimentally measured free energy of short structure motifs, as well as random structure motifs whose energy was estimated using existing linear approximation models. Although it shows promising result on modelling free energy of short structure motif from experimental data, and can be used to estimate the free energy for a *given* structure, it does not solve the problem of predicting structure from sequence.

Other work were done to tackle the learning problem end-to-end. Researchers have framed the problem as a sequence to sequence learning task, and the proposed model either predicts the probability of being paired for each base, or some forms of the dot-bracket notation. Willmott et. al[7] proposed a bidirectional LSTM that predicts from RNA sequence the probability for 3 classes: paired, unpaired, and end-of-Sequence. The model was trained on experimentally solved structure of 16S rRNAs. It is unclear whether the model generalize to other types of RNAs. Zhang et. al[8] proposed using convolutional neural network to predict from sequence the dot-bracket notation, where each position is encoded as a 3-class softmax: left bracket, right bracket and dot. In order for a dot-bracket notation to yield valid structure, it has to follow a few syntactic constraints. For example, each left bracket needs to have a corresponding right bracket. As the authors have mentioned in the paper, output produces by the neural network is not guaranteed to be valid, thus needs to be further processed by dynamic programming to yield a valid structure prediction. Wang et. al[9] used bidirectional LSTM and modelled the dot-bracket notation as a 7-class softmax, which also includes additional bracket notations to allow for pseudoknots. Similar to [8], the prediction does not yield a valid structure, and needs to be corrected by additional logic. Furthermore, they used a dataset consisting of only four RNA families, but reported performance based on randomized training and validation set split. Random split is almost certain to result in sequences from the same family to be in both the training and validation set, thus the generalization performance of the model remains unclear.

Existing work either solves a partial problem, or attempts at modelling the problem end-to-end, but requires complicated post-processing to yield valid structure prediction. In this work, we propose a novel method to model the sequence to structure predictive task end-to-end, with no additional post-processing.

3 Method

In this work, we propose a deep neural network that can be trained end-to-end on dataset with sequences and structures. The model is capable of generating a distribution of structures, including structure with pseudoknot, conditioned on the input sequence.

We formulate the predictive task as a conditional generative process. Specifically, given an input sequence with arbitrary length L : $\mathbf{x} = x_1, x_2, \dots, x_L$, we want to predict a distribution of structures conditioned on the sequence $P(\mathbf{Y} | \mathbf{x})$, where the structure \mathbf{Y} is represented by an upper triangular matrix of size $L \times L$, as defined in Fig 1(b).

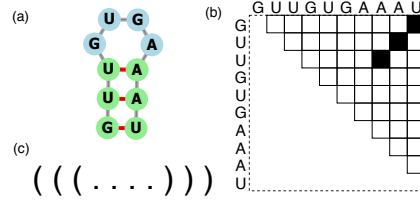


Figure 1: Three different ways to represent secondary structure of the RNA sequence GUUGUGAAAU. (a) undirected graph, generated by [5], (b) upper triangular binary matrix, (c) dot-bracket notation.

We factorize the conditional distribution as follows:

$$P(\mathbf{Y} | \mathbf{x}) = P(\{y_{i,i+1}\}_{i=1,2,\dots,L-1} | \mathbf{x}) P(\{y_{i,i+2}\}_{i=1,2,\dots,L-2} | \mathbf{x}, \{y_{i,i+1}\}_{i=1,2,\dots,L-1}) \dots P(y_{1,L} | \mathbf{x}, \{y_{i,j}\}_{j \neq 1}^{i \neq 1}) \quad (1)$$

The generative process implied by such formulation is illustrated in Fig 2. We generate one off diagonal slice at a time, conditioned on the input sequence and the previous slices, starting from the one adjacent to the diagonal line, shown in light blue with $t = 1$. This process continues until we fill the upper triangular matrix, where the last entry (dark blue, shown with $t = 9$) is generated conditioned on the input and the entire upper triangular matrix except for itself. Intuitively, the distance to the off-diagonal line is the "timestamp" as in traditional autoregressive models.

3.1 Model

We propose an architecture that builds global structure from local interactions between all positions on the sequence, without having any assumptions of the types of local structure and hard-coded parameters, such that the entire model can be trained end-to-end using only sequences and structures. The architecture consists of the following components:

- Two sets of 1D convolution layers on the 1-hot-encoded sequence are run in parallel, each set has multiple convolution layers at different resolutions.
- Activations of each 1d conv layer, one from each set, are used to form a 2D feature map, where the (i, j) -th entry is the dot-product (can be replaced by a fully connected NN) between the activation of first set at position i , and the activation of second set at position j . This is illustrated in Fig 3. The idea of such set up is inspired by the fact that whether two bases form base pair or not is not only affected by the two nucleotides but also surrounding sequences. Such architecture can enable the neural network to learn low level sequence features that affect the 'compatibility' between two stretches of sequences.
- Multiple 2D feature maps (formed via activations of different layers of 1D conv) are concatenated, followed by a couple of 2D conv layers.
- Activation of the last 2D conv layer is concatenated with target from the previous "time-stamp" y^{t-1} , and the output is generated by an upper triangular convolution, as shown in Fig 3, which masks "future" timestamps and ensure the output is generated in auto-regressive fashion.
- Finally, there is a fully connected layer along the feature dimension, with sigmoid activation to produce an output between 0 and 1, for each position in the upper triangular matrix.

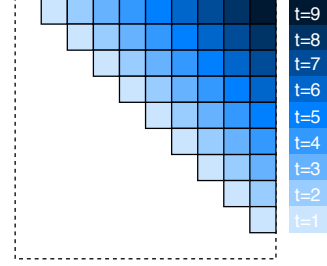


Figure 2: Generative process implied by factorization of the conditional distribution in Equation 1.

At training time, prediction, loss and gradient of all positions in the output can be computed in parallel. At test time, we need to initialize the output at time $t = 0$, typically with a matrix of all zeros, then sample one slice at a time, until the full upper triangular matrix is filled. This is illustrated in Fig 4. For a sequence of length L , we need to run $L - 1$ steps sequentially. Note that multiple outputs can be sampled in parallel at test time.

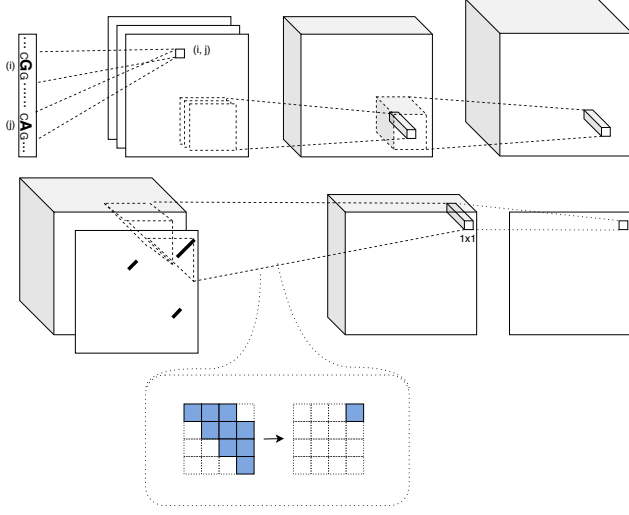


Figure 3: Proposed model architecture. Top: sequence encoding. Bottom: generate structure.

3.2 Training

We trained the model using a synthetic dataset, constructed by sampling 50000 random sequences with length between 10 and 100. For each sequence, we ran RNAfold[1] with the default parameters and record the minimum free energy structure.

For each minibatch, we zero-pad the sequence array and structure matrix to the maximum length in the minibatch. When computing the loss and gradient, entries in structure matrix that were padded are being masked, in addition to the lower triangular entries (since we’re only predicting the upper triangular matrix).

Note that although we present a single output structure for each input sequence at training time, the model is capable of generating a distribution of structures at test time, by generalizing across different examples.

3.3 Inference

TODO this is repeated from the model section, maybe combine?

At test time, we can sample structures conditioned on the input sequence. As shown in Fig 4, we initialize the output structure with a matrix of all zeros, then sample one slice at a time until the upper triangular matrix is filled with sampled values. At each step, we sample a binary label for each position in the current slice based on the Bernoulli probability predicted by the model. To ensure the sampled structure is valid, when sampling the label for location (i, j), if i-th or j-th position is already paired with another position (from samples in the previous timestamps), then we set $y_{i,j}$ to 0 without sampling from the model output.

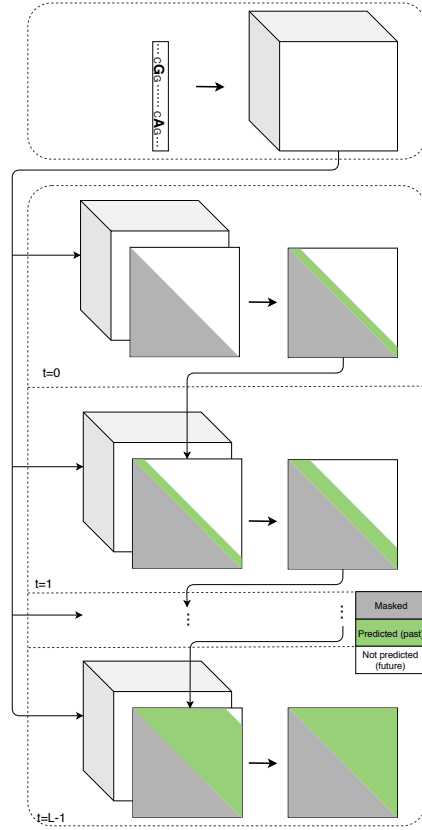


Figure 4: Sample structures conditioned on the input sequence.

4 Result

4.1 Test set performance

To check the model performance, we generated a test set with 100 sequence, using a process similar to the training set. Sequences are sampled randomly using A,C,G,U and lengths are between 10 and 100. For each sequence in the test set, we ran RNAfold (TODO more details) and sampled 100 structures from the ensemble, and used our model to also sample 100 from the output distribution conditioned on the input sequence. To avoid evaluating on low probability structures, we only kept structures that occur more than 10 times. Then, for each unique structure produced by RNAfold, we computed sensitivity (TODO definition), positive predictive value (TODO definition) and f-measure (TODO definition) against all unique structures generated by our model, and recorded the best one, where best is defined by largest f-measure. This represent? how well the model recovers structures produced by RNAfold. Histogram showing the distribution of these performance metric across all structures in all sequences is shown in Figure ??.

4.2 Structure with pseudoknot

Although trained on synthetic dataset with no pseudoknot structures, since our model doesn't incorporate hard-wired rules on how local structures assemble into global structure, it is actually capable of generating structure with pseudoknots. As an example, we use the sequence of mouse mammary tumor virus (MMTV), whose secondary structure contains pseudoknot as measure by nuclear magnetic resonance (NMR), as shown in Fig 6(a). Minimum free energy structure predicted by RNAfold takes a quite different form, which is shown in Fig 6(b). In contrast, when we sample 100 structures from our model, it shows a diverse set of possible structures, including the one predicted by RNAfold, as shown in Fig 6(c3), and more interestingly, the pseudoknot structure from NMR, as shown in Fig 6(c1).

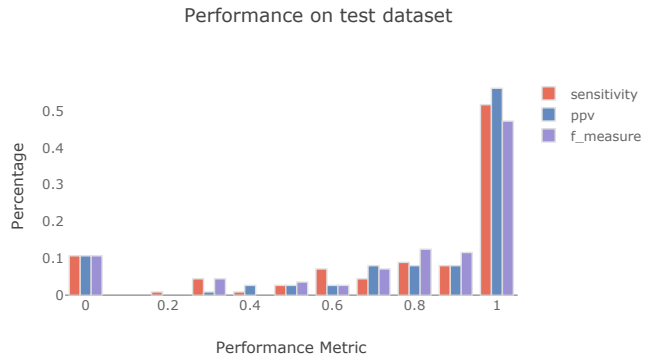


Figure 5: TODO

4.3 Differentiable model

5 Conclusion

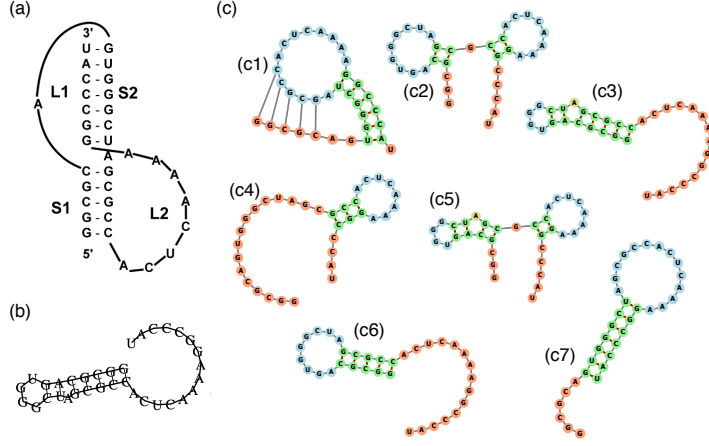


Figure 6: Secondary structure of mouse mammary tumor virus (MMTV): (a) measured by nuclear magnetic resonance (NMR), plot from [10] (TODO plot was a screenshot from the paper, any problems?), (b) predicted by RNAfold web server, (c) Structures generated by our model, specifically, (c1) is a structure with pseudoknot and is identical to (a), (c3) is identical to (b).

References

- [1] Ronny Lorenz, Stephan H Bernhart, Christian Höner Zu Siederdisen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. Viennarna package 2.0. *Algorithms for molecular biology*, 6(1):26, 2011.
- [2] Michael Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic acids research*, 31(13):3406–3415, 2003.
- [3] Mirela Andronescu, Anne Condon, Holger H Hoos, David H Mathews, and Kevin P Murphy. Efficient parameter estimation for rna secondary structure prediction. *Bioinformatics*, 23(13):i19–i28, 2007.
- [4] Mirela Andronescu, Vera Bereg, Holger H Hoos, and Anne Condon. Rna strand: the rna secondary structure and statistical analysis database. *BMC bioinformatics*, 9(1):340, 2008.
- [5] Peter Kerpedjiev, Stefan Hammer, and Ivo L Hofacker. Forna (force-directed rna): simple and effective online rna secondary structure diagrams. *Bioinformatics*, 31(20):3377–3379, 2015.
- [6] Michelle J Wu. Convolutional models of rna energetics. *bioRxiv*, page 470740, 2018.
- [7] Devin Willmott, David Murrugarra, and Qiang Ye. State inference of rna secondary structures with deep recurrent neural networks.
- [8] Hao Zhang, Chunhe Zhang, Zhi Li, Cong Li, Xu Wei, Borui Zhang, and Yuanning Liu. A new method of rna secondary structure prediction based on convolutional neural network and dynamic programming. *Frontiers in genetics*, 10, 2019.
- [9] Linyu Wang, Yuanning Liu, Xiaodan Zhong, Haiming Liu, Chao Lu, Cong Li, and Hao Zhang. Dmfold: A novel method to predict rna secondary structure with pseudoknots based on deep learning and improved base pair maximization principle. *Frontiers in genetics*, 10:143, 2019.
- [10] David W Staple and Samuel E Butcher. Pseudoknots: Rna structures with diverse functions. *PLoS biology*, 3(6):e213, 2005.