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 forms stable double helix (between two molecules), RNA is highly versatile and? a single RNA moldecule 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. These base pairs act as building blocks, from which local structure forms? global?

State-of-the-art RNA structure prediction algorithms, such as ViennaRNA[1] and Mfold[2], are based on the fundamental property of base-pairing. Each type of base-pairing, as well as each type of local structure, has its own associated 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 people 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.

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, which have only been combined? with dynamic programming using heuristics (ref?). Such data also exhibit a high level of noise and sometimes with missing data (ref, sequencing, DMS), which also calls for new approaches that can be trained end-to-end on different types of data, to model ?differences in cell types and conditions.

In this work, we'll .... (overall summary)

TODO if we mention the above we'll need to talk about how to train on those data.

sequence alone

one sequence, many structures

1.1 Problem Formulation

For a sequence of length L, it is possible? to take on a distribution of strutures?. There are three common ways to represent a specific RNA secondary structure:

- 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 paring.
- A dot-bracket notation of length L where unpaired based 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 $y_{1,10}, y_{2,9}$ and $y_{3,8}$,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. (TODO define x and y first)

2 Related Work

There have been a few work using deep nn for modelling RNA secondary structure.

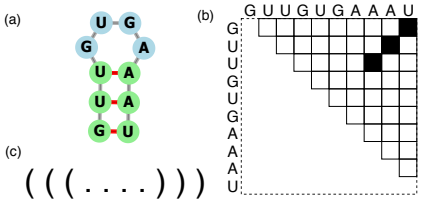

Figure 1: Three different ways to represent secondary structure of the RNA sequence GUUGUGAAAU. (a) undirected graph, generated by [5], (b) upper triangular binary matrix, (c) dot-bracket notation.

Wu[**?**] 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 structure. 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 better 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 end-to-end.

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 predicts some form of the dot-bracket notation. Willmott et. al[**?**] 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. (talk about the cons) Zhang et. al[**?**] 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, 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[6] 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 [**?**], 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 whether such performance generalizes to an unseen RNA sequence is 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.

SCFG: Learning RNA secondary structure (only) from structure probing data not that relevant, maybe we can skip it

TODO talk about what's lacking in all above approaches.

3 Method

In this work, we propose a new framework (deep nn) that can be trained end-to-end on dataset with sequences and structures. The model is capable of generating a distribution of structures, conditioned on the input sequence. (maybe mention this after literature review?)

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

(move the following paragraph to the model section) We factorize the conditional distribution as follows:

$$P(\boldsymbol{Y} \mid \boldsymbol{x}) = P(\{y_{i,i+1}\}_{i=1,2,\ldots L-1} \mid \boldsymbol{x}) P(\{y_{i,i+2}\}_{i=1,2,\ldots L-2} \mid \boldsymbol{x}, \{y_{i,i+1}\}_{i=1,2,\ldots L-1}) \ldots P(y_{i,j} \mid \boldsymbol{x}, \{y_{todo}\})$$

The generative process implied by such formulation is illustrated in Fig 2.

TODO re-generate plot using draw.io, or use the model plot?

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, as shown in yellow on the plot. When generating the second slice (in green), we condition on the input sequence and the generated values in the first slice (in yellow). When generating the third one (in blue), we condition on the input sequence and both the yellow and green slices. This process continues until we fill the upper triangular matrix, where the last entry (in red TODO color plot) is generated conditioned on the input and the entire upper triangular matrix except for itself.

Above paragraph is quite long, cut it?

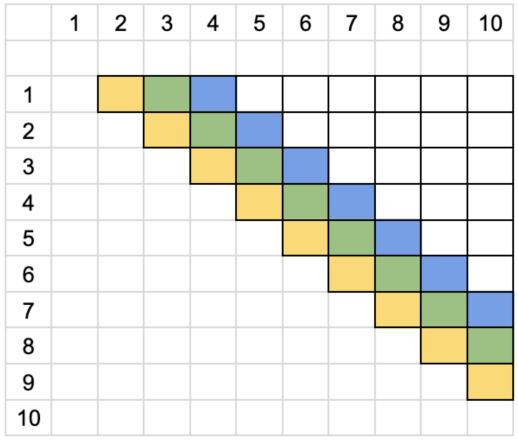

Figure 2: TODO

3.1 Model

(doesn't flow well, combine with previous paragraphs?)

We propose an architecture that encourages? learning basic rules of base pairing and local structures, to construct the global structure, 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:

TODO refer to parts in plot when talking about architecture.

TODO itemize takes a lot of space, trim it? or combine items. gorup things so they bettwen correspond to stuff on the plot.

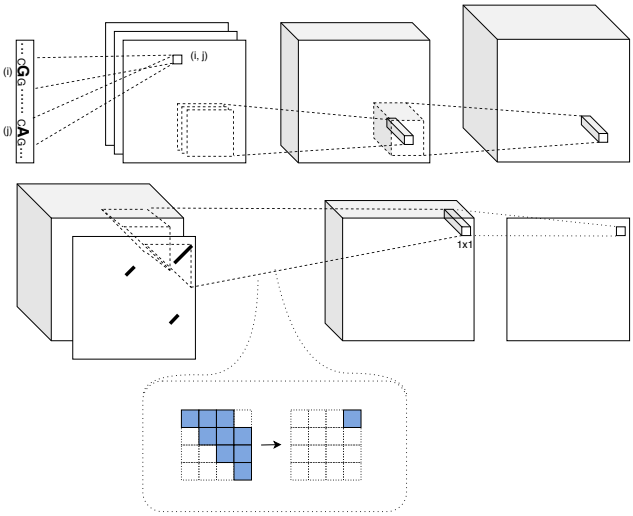

Figure 3: TODO

- 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. Note that although not shown explicitly on the plot (to keep it simple), multiple 1D conv layers...stacked...
- Activations of each 1d conv layer (from both sets) 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? where the ? shows one element at position (i, j) in one 2D feature map is created by the dot-product between the two sets of activation units of a specific 1D convolution layer (in this case the first one) at position i and j. 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. The hope is to learn low level sequence features that affect the 'compatibility' between two stretches of sequences.
- Each 2D feature map corresponds to a receptive field?. Multiple 2D feature maps (formed via multiple layers of 1D conv) are concatenated, followed by a couple of 2D conv layers. Note that although not shown explicitly on the plot (to keep it simple), multiple 2D conv layers...stacked...
- Activation of the last 2D conv layer is concatenated with target from the previous "time-stamp" y^{t-1} (todo define notation), and the output is generated by an upper triangular convolution, which masks "future" timestamps and ensure the output is generated in auto-regressive fashion. (TODO more details on masked conv)
- Finally, there is a fully connected layer along the feature (3rd) dimension, with sigmoid activation to produce an output between 0

and 1, for each position in the upper triangular matrix.

At training time, different timestamps can be trained in parallel. At test time, we need to initialize the output at time $t = -1$, typically with a matrix of all zeros, then sample one slice at a time, until the full upper triangular matrix is filled. For a sequence of length $L - 1$, we need to run $L - 1$ steps sequentially. Note that multiple outputs can be sampled in parallel at test time.

TODO plot for NN architecture

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 (TODO ref) 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.

TODO hyperparameters

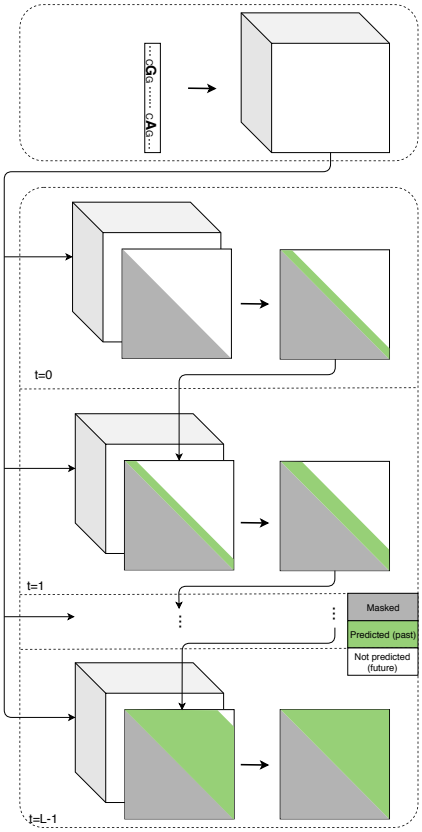

Figure 4: TODO

3.3 Sampling structures

At test time, we can sample structures conditioned on the input sequence. As described in Section ?, 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.

4 Analysis

TODO show that we can predict well-known structure. Any known multi-structure sequences?

4.1 Generate distribution of structures

As an example, for a 323-base long sequence (TODO link to DB), we sampled 20 structure from the model output, as shown in Fig 5. On the left we show the binary upper triangular matrix sampled by the model (TODO it's too tiny to see anything), on the right we show the corresponding secondary structure rendered as a graph (only showing 4 due to space limit).

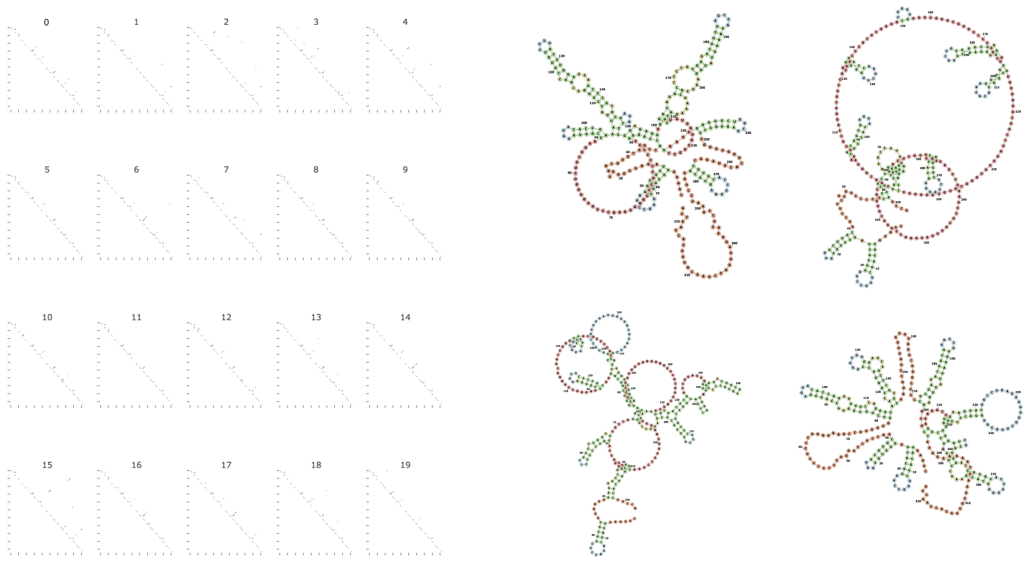

Figure 5: TODO

4.2 Run time comparison

4.3 Performance comparison

cross validation test set (real sequences)

4.4 Interesting cases

4.5 Differentiable model

5 Conclusion

future work: train on real sequences and structure (we imagine the performance will improve)
different AR/sampling setup, different factorization, better performance?
train on high throughput data
graph nn

References

[1] Ronny Lorenz, Stephan H Bernhart, Christian Höner Zu Siederdissen, 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] 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.