test

August 29, 2020

1 Introduction

dp: pros: hard-constraint satisfied by design (as reflected by valid recursion steps); models distribution cons: efficient dp recursion based on nesting assumption, cannot model pseudoknot; additive energy terms, hard-coded.

scfg extension: pros: constraints; models distribution trainable param cons: efficient dp recursion based on nesting assumption, cannot model pseudoknot

todo: plot explaining dp scfg approaches

nn: pros: expressive, can be trained on lots of data; cons: too many parameters, risk of overfitting if trained on dataset not diverse enough; hard constraints cannot be captured by default (some work using optimization as unrolled rnn, but no guarantee it'll converge, and even if it converges there's guarantee the output satisfies the constraints, since the original discrete variables need to be relaxed in order to run through the optimization nn, so we're no longer operating in discrete space); does not model distribution

todo: plot explaining nn approaches

In this work we are aiming at developing a model that makes use of the expressiveness of deep nn while respecting the hard constraints.

This report covers the following sections:

- predictive problem formulation
- we review the commonly used data representation for rna ss
- we propose an alternative way to represent/parameterized the ss
- under the alternative parametrization, we propose a 2-stage model, which satisfies the biological hard constraints by design
- describe stage 1 model: data, training result.
- ideas for stage2

1.1 Problem formulation

Given an RNA sequence of length L, we are interested in all possible secondary structures. To represent a specific secondary structure, there are three commonly used conventions: (1) undirected graph, where each node is a base in the sequence, and each edge represents base pairing. (2) 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. (3) 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, for a short RNA sequence GUUGUGAAAU, one possible structure it can take on

consists of a stem and a loop, as seen in Fig **??**(a), represented by an undirected graph. Such structure can also be represented by a 10×10 upper triangular matrix with all 0's, except for positions $(1,10),(2,9)$ and $(3,8)$, all being 1, as shown in Fig **??**(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. The equivalent dot-bracket notation is shown in Fig **??**(c), where the stem is represented by three pairs of left-right brackets.

2 Structural components

Just as sequence motif is defined in the linear sequence space, a similar notion of 'structural motif' also exists. A commonly adopted way is to break down structure into the following components:

- hairpin loop: todo
- stem: todo
- internal loop: todo
- bulge: todo
- external loop: todo
- multibranch loop: todo
- pseudoknot: todo

Such structural motifs can be easily identified on the graph representation of a structure, as shown in Fig 1, where we only annotated one instance for each structural motif class for clarity. (todo add a minimal example for pseudoknot)

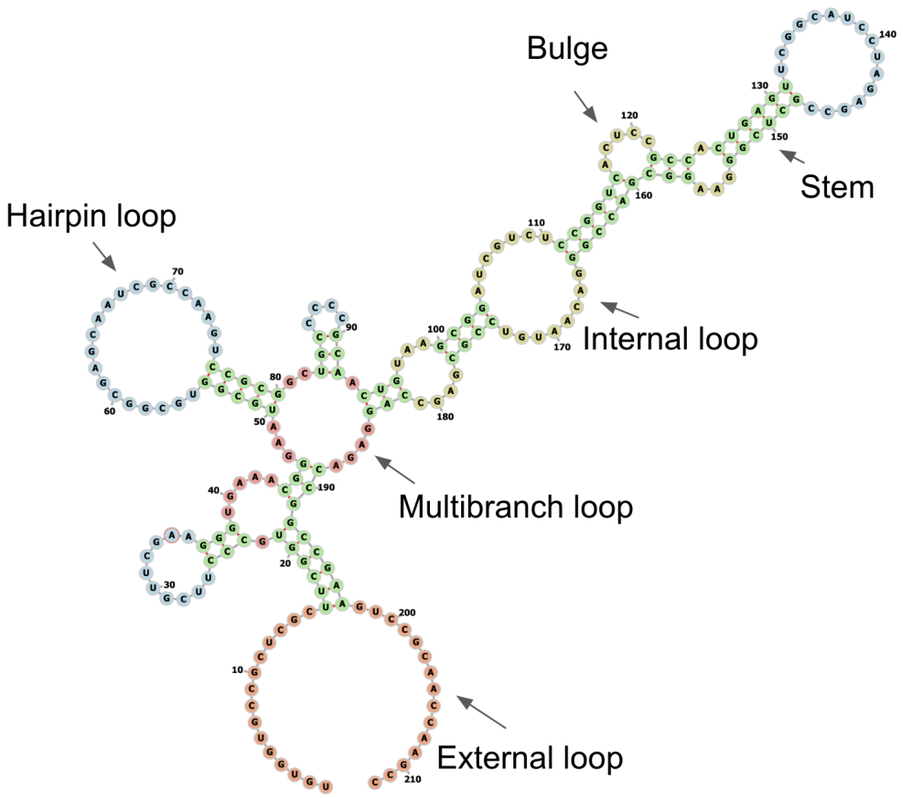
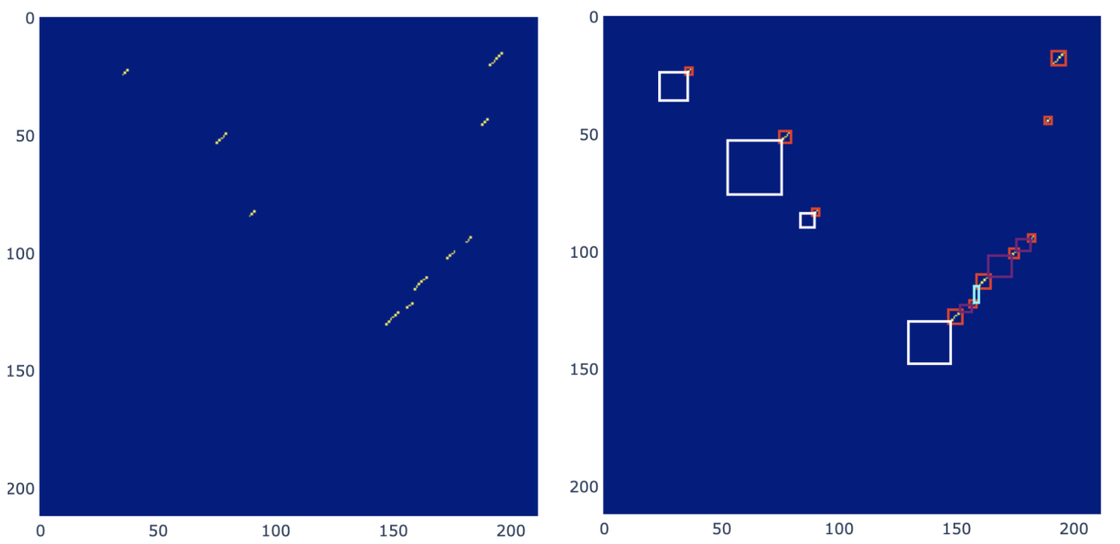


Figure 1: TODO

In this work, we distinguish structural motifs that are 'local' and 'non-local', where local-ness is defined w.r.t. the 2D matrix representation of the structure. The importance of such distinction will become clear in Section (todo). To illustrate this idea, we plot the 2D matrix representation of the above structure in Fig 2(a), where majority of the pixels

are in *navy* (unpaired positions), and a few pixels in *yellow* (paired positions). For each type of structural motif, if it can be fully represented by the interaction between two (contiguous) substrings of the original sequence, it is considered a 'local' structure, and can be fully identified by a bounding box on 2D matrix. Note that interaction does not necessarily mean the substrings bind to each other. On the other hand, certain structure motifs can only be represented by interaction between three or more (contiguous) substrings of the original sequence,which we refer to as non-local structures.

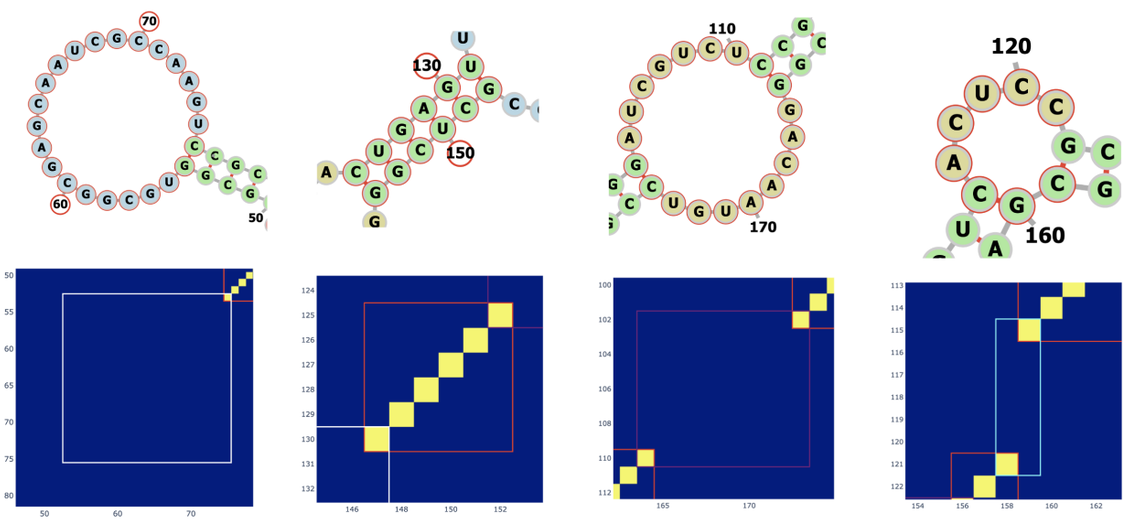


(a) 2D matrix representation (b) with local structure bounding boxes

Figure 2: TODO

2.1 Local structures

Using the above definition, we drew all the local structure bounding boxes in Fig 2(b).



(a) Hairpin loop (b) Stem (c) Internal loop (d) Bulge

Figure 3: TODO

To precisely define what constitutes each type of local structure, we zoom in to each annotated structure in Fig 1, and compare it with the corresponding portion zoomed-in on 2D matrix, as shown in Fig 3. We can see that:

- hairpin loop: Fig 3(a), square bounding box across the diagonal, all pixels have value 0 except the top right corner being 1, where the top right pixel corresponds to the closing base pair, i.e. the base pair that's 'shared' between the hairpin loop and the stem. Note that hairpin loop does not exist in a stand-alone fashion. There is always a stem next to it.
- stem: Fig 3(b), square bounding box with 1's on the off-diagonal and 0's elsewhere.
- internal loop: Fig 3(c), rectangular (or square) bounding box of all 0's except the top right and bottom left corner being 1, where these two pixels correspond to the closing base pairs, one on each side. Note that internal loop does not exist in a stand-alone fashion. There is always two stems next to it, one on each side.
- bulge: Fig 3(d), almost the same as internal loop, except for the bounding box size is constrained to be $2xN$ or $Nx2$, since one of the 'side' only has 2 consecutive base pairs, by definition.

One notable fact is that all pixel values inside the bounding box are deterministic given the box type and shape. This also guarantees that hard constraints are satisfied within each bounding box.

2.2 Non-local structures

We're left with 3 structural motif types that cannot be repsented by a bounding box on 2D matrix. We will show that non of these structures contribute pixel value of 1 in the 2D matrix, and that their existence can be described in an alternative way, which pinpoint how local structures can be assembled into *valid* global structures.

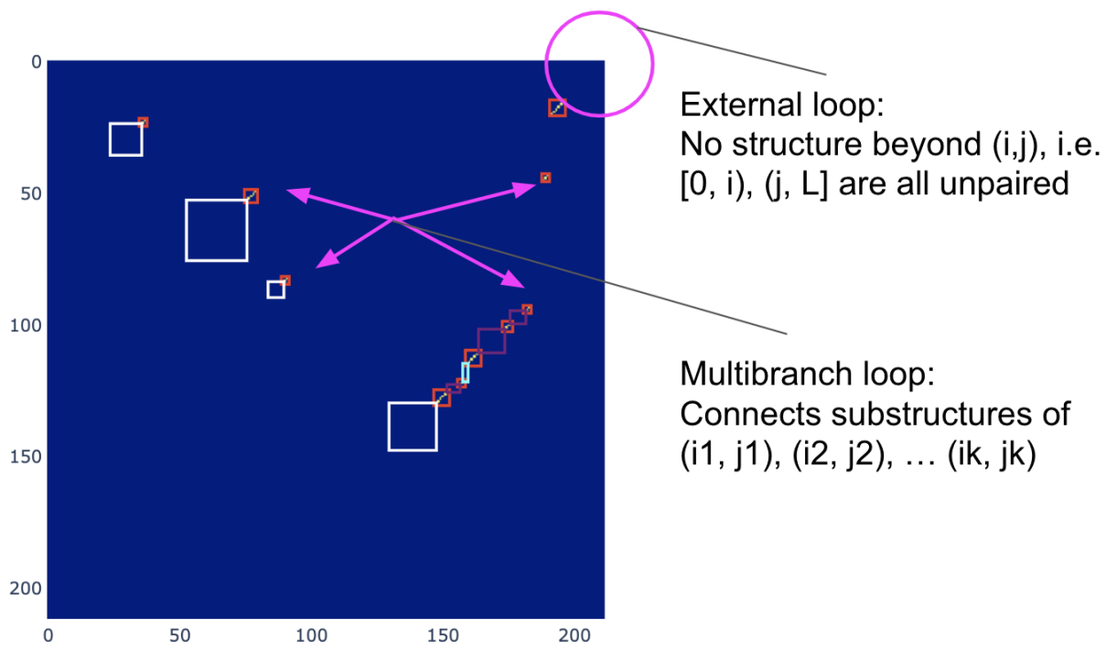


Figure 4: TODO

- external loop: Fig 4, the presence of external loop beyond (i, j) indicates that bases $0 \dots i$ and $j \dots L$ are all unpaired in the global structure, while i and j are paired (not necessarily to each other). One might claim that in this particular example it can be fully characterized by a 2D bounding box with botton left pixel (i, j) being 1 since i

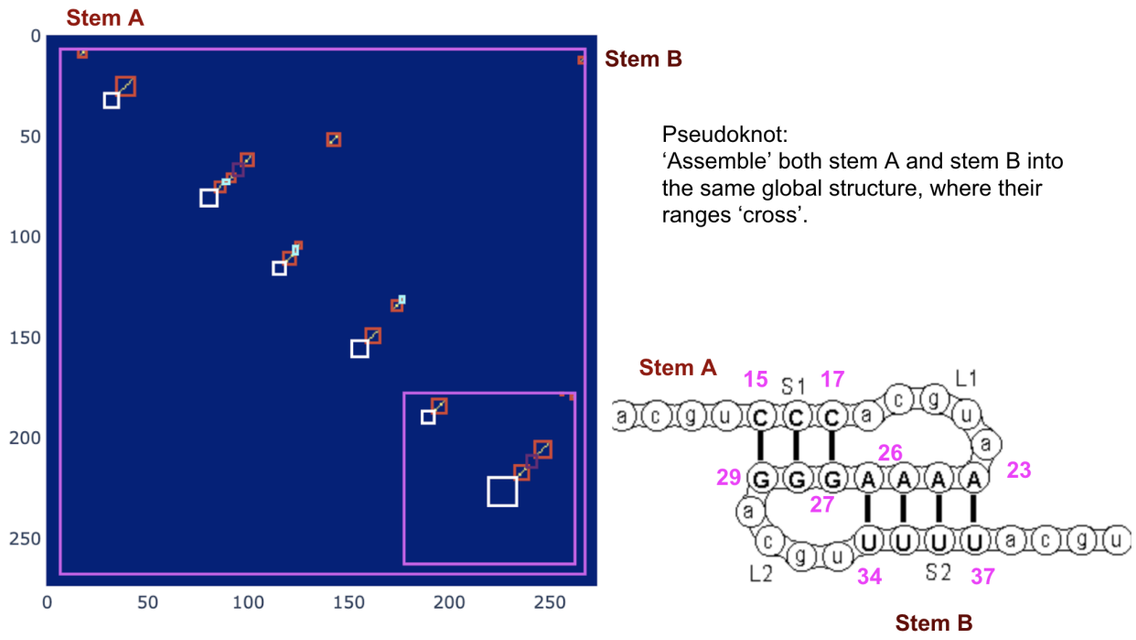


Figure 5: TODO

and j are paired, but in practise external loop can also be multi-branch, i.e. joining multiple stems, in which case i pairs with some other k and j pair with some other u.

- multibranch loop: Fig 4, the presence of multibranch loop implies that multiple substructures get "assembled" into the same global structure, where each substructure is lead by a stem whose closing base pair is (i_k, j_k).

- pseudoknot: We use a different structure to illustrate pseudoknot since the structure we used above does not contain any, see Fig 5. In this example, stem A defines bases $15 \ldots 17$ pair with $27 \ldots 29$, and stem B defines bases $23 \ldots 26$ pair with $34 \ldots 37$. They form a pseudoknot since their ranges cross each other, i.e. non-nested (which is required by most thermodynamic folding).

3 Method

We propose a 2-stage approach to predict secondary structure from sequence. In stage 1, we predict all plausible local structures, represented as various types of bounding boxes. In stage 2, we evaluate all *valid* combinations of local structures from stage 1, and predict the most likely global structure. Non-local structure is being modeled implicitly in stage 2 by defining what combinations are valid.

Benefit of such model:

- Output is guaranteed to satisfy all hard constraints. This is a direct result of the 2-stage approach: local structures predicted from stage 1 is parametrized as bounding boxes whose underlying pixels satisfy constraints by design. Combination of local structure in stage 2 needs to be valid, where validity is precisely defined by hard constraints.

- Possibility of predicting sub optimal structure and structure ensemble. Since stage 1 predicts all plausible local structures, and stage 2 evaluates the likelihood/score of all valid combination, we can predict sub optimal structure, and even the entire ensemble.

- Support prediction of pseudoknot by design. Historically. pseudoknot is difficult to predict since the traditional methods were based on recursion defined at a base-pair level, thus the nested structure assumption has to be made to result in efficient dynamic programming algorithm. Our approach evaluates valid combinations of local structure (rather than at the base-pair level), whose search space is way smaller than evaluating all valid combination of base pairs, so we no longer require the basic components (local structures in our case) to be nested, thus enabling pseudoknot prediction.
- Can be potentially fine-tuned by probing data. Since stage 1 model predicts plausible local structure, and each type of local structure implies certain base-pairing or unpairing for the underlying bases, we can use probing data to fine-tune the prediction. For example, cross-link based RNA probing method detects single and double stranded regions of a RNA, and for the double stranded region, although it does not detect the exact base pairing, it is possible to infer roughly which region binds to which other region. Such information can be incorporated while training the stage 1 model.

This report focus on stage 1 model.

4 Stage 1: Predict local structure

4.1 Problem formulation

Given a input sequence of length L, we would like to predict all plausible local structure bounding boxes in the 2D LxL matrix. Such problem is well studied in computer vision under the topic of object detection, from which we got a lot of inspiration. We notice the following difference compare to computer vision applications:

- In computer vision, the exact pixel location of the bounding box does not matter, as long as the bounding box captures the object of interest. In our case, since a different pixel correspond to different nucleotides, and most of our bounding boxes are small in pixel size, we cannot tolerate even a single pixel shift.
- In computer vision, two bounding boxes can overlap if the underlying objects overlap on the image. In our case, for a set of bounding boxes to be part of the same global structure, they cannot overlap (except for corner pixels), since the location and size of each bounding box fully parameterize the pixel values it encapsulates.
- todo

For each pixel in the output LxL matrix, we predict the presence of 3 types of bounding boxes:

todo: copy from meeting notes, discretized target

4.2 Architecture

4.3 Training

todo: soft mask (due to potential false negative)

4.4 Inference

todo: copy from meeting notes, plot

5 Result

6 Ideas for Stage 2: global structure assembly

7 Future work

satisfy constraints, all possible 2D binary matrix -> huge space, majority does not satisfy biology constraints. bounding box: parameterized by a few parameters, pixel value implied given parameterized, satisfy (local) constraints.

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] Stefanie A Mortimer, Mary Anne Kidwell, and Jennifer A Doudna. Insights into rna structure and function from genome-wide studies. *Nature reviews Genetics*, 15(7):469, 2014.

[5] Philip C Bevilacqua, Laura E Ritchey, Zhao Su, and Sarah M Assmann. Genome-wide analysis of rna secondary structure. *Annual review of genetics*, 50:235–266, 2016.

[6] 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.

[7] Michelle J Wu. Convolutional models of rna energetics. *bioRxiv*, page 470740, 2018.

[8] Devin Willmott, David Murrugarra, and Qiang Ye. State inference of rna secondary structures with deep recurrent neural networks.

[9] 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.

[10] 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.

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] David W Staple and Samuel E Butcher. Pseudoknots: Rna structures with diverse functions. *PLoS biology*, 3(6):e213, 2005.