

Summer Review 3

Solving the RNA design problem with reinforcement learning

Peter Eastman¹, Jade Shi², Bharath Ramsundar³, Vijay S. Pande¹

1: Department of Bioengineering, Stanford University, Stanford, CA

2: Department of Chemistry, Stanford University, Stanford, CA

3: Department of Computer Science, Stanford University, Stanford, CA

December 5, 2019

The Task

- RNA alphabet: A,G,C,U
- RNA function depends on structure it has folded into.
- folding structure depends on nucleotide sequences
- Inverse Folding Problem: Find nucleotide or base sequence that folds to that structure.
- Predicting Structure from Nucleotide Sequence: ViennaRNA

Inverse Folding as Reinforcement Learning

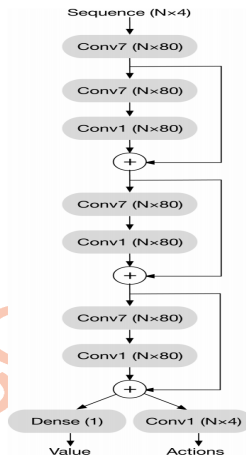
- Current State: candidate sequence
- action: modify type of single base, or two in some cases.
- when target structure is achieved, reward else zero.
- train a policy network to select actions that lead to sequence with desired secondary structure.

- input: $N \times 4$
- output: $N \times 4$: probability of changing one of the N into one of the base
- 25% of all actions simply say that a base should have the same type it already has. To avoid wasting time on unproductive actions, the probabilities of these actions are forced to be 0
- If an action modifies a base that is supposed to be paired in the target structure, the type of its desired partner is checked and, if necessary, modified as well to ensure the two bases are capable of forming a pair

- conv7: The output for each base is a linear combination of the inputs for seven bases, followed by a ReLU activation. The seven bases include:
 - The five consecutive bases centered at that base
 - The base with which that base should be paired in the target structure
 - The base with which that base is actually paired in the structure formed by the current sequence
- The output is computed independently for every base. Each output channel is a linear combination of the input channels for that same base.

$$y = x + \text{conv1}(\text{conv7}(x)) \quad (1)$$

The Model



Training Procedure

- Randomly generated 100000 RNA sequences of 32 length each.
- Compute structure
- Only structures used for training
- Use Asynchronous Advantage Actor-Critic algorithm

Results

- 60/100 puzzles
- second best: Modena (54/100)
- Sequence length is not the only factor of difficulty
 - solves very long but fails to solve some short
 - same length with different solving times
- performance matches how difficult for human players

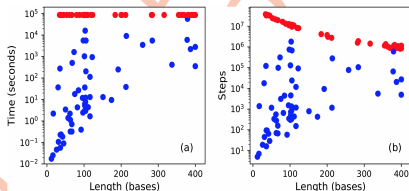








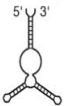







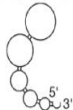















Fig. 2. Difficulty of solving puzzles as measured by (a) clock time or (b) steps, versus sequence length. Blue dots represent the 58 puzzles that were successfully solved on the first attempt. Red dots represent the 42 puzzles on which it gave up after running for 24 hours. (Two more of them were eventually solved on later attempts.) Longer sequences take more time per step, so fewer steps can be completed in 24 hours.

Figure: Length of bases vs time/steps

Difficult Structures

<p>A. Shortie 4</p>  <p>5' 3'</p> 	<p>Shortie 6</p>  <p>5' 3'</p> 	<p>B. Kyurem 5</p>  <p>5' 3'</p> 	<p>Kyurem 7</p>  <p>5' 3'</p> 	<p>C. Loop next to a Multiloop</p>  <p>5' 3'</p> 
<p>D. Just down to 1 bulge</p>  <p>5' 3'</p> 	<p>1,2,3and4bulges</p>  <p>5' 3'</p> 	<p>E. Mat - Lot 2-2 B</p>  <p>5' 3'</p> 	<p>Crop Circle 2</p>  <p>5' 3'</p> 	<p>F. Hard Y</p>  <p>5' 3'</p> 
<p>G. This is actually small and easy 6</p>  <p>5' 3'</p> 	<p>H. Quasispecies 2-2 Loop Challenge</p>  <p>5' 3'</p> 	<p>Water Strider</p>  <p>5' 3'</p> 	<p>The Fractal</p>  <p>5' 3'</p> 	<p>Mutated Chicken Feet</p>  <p>5' 3'</p> 

Bulges and internal loops

- destabilize structures: smaller base pairing and encourage mispairing to form stems
- Similar performance for RL:
 - can solve the relatively easy "Just down to 1 bulge" extremely quickly in only 11.9 seconds
 - fails for "1,2,3and4bulges". "1,2,3and4bulges" was only solvable by MODENA, whereas "Just down to 1 bulge" was solvable by 4 of 6 algorithms.
 - succeeds in solving "Loop next to a multiloop", which consists of two internal loops stabilized by a short stem, in about 25 minutes, but fails for structures such as "Crop circle 2", which consists of five internal loops stabilized by short stems.

Symmetric Structure Elements

- Symmetric lead to opportunities for mispairing
- Shortie 6 and Fractal 2,
- fails for Mutated Chicken Feet: three symmetric branches of consecutive short stems linked by bulges and multiloops, a combination of several factors that make design difficult.