Mitra Shokat
Bio 131
Spring 2017

Predicting the Secondary Structure of RNA

The structure of RNA differs from that of DNA in a few key ways. First, the nitrogenous base thymine (T) found in DNA is replaced by uracil (U) in RNA. Second, RNA is single-stranded while DNA is double-stranded. This single-strandedness allows for intramolecular base pairings between nitrogenous bases on the same strand of RNA. The pattern in which those bases interact forms the secondary structure of RNA. This structure is crucial to the biological functions of RNA, and thus it is of interest to be able to predict the conversion from primary to secondary structure. This, however, is not a simple task.

Several computational approaches to the problem of predicting RNA secondary structures exist. Many of these algorithms involve dynamic programming, which is a method of approaching a large problem by segmenting it into smaller problems of the same type. The issue that many of these algorithms face, however, is the fact that there are many factors that must be considered when attempting to optimize RNA secondary structures. The factors that determine which secondary structure is most favorable for a given strand of RNA include the number of base pairings and the thermodynamics of any given paired, mismatched, or unpaired base. It is challenging to incorporate all of these factors into a single algorithm.

The Nussinov algorithm simplifies the problem slightly by defining the optimal RNA secondary structure as the one that contains the maximum number of paired bases and no pseudoknots. For my dataset, I chose a short string of RNA (‘GACACGACGA’) which I made up. I chose to test my program on this small sequence because it allowed me to compare the program's output to the "known" secondary structure of the strand of RNA (Figure 1).

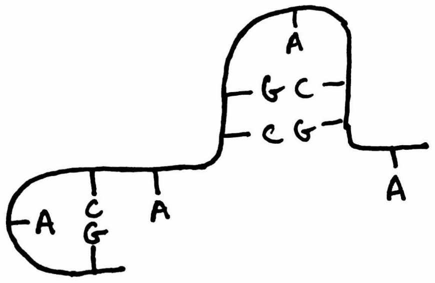

Figure 1. Predicted RNA secondary structure of test sequence.

After confirming that my program worked on a short sequence of RNA, I attempted to implement it on the RNA primary sequence of alanine tRNA ('AGGGAAAAUAGUUUAAUAAAAAUAUUUUACUUGCAGUAAAAAGUUAUUUCUAU AAUUUUUCUUU'). Again, by using a sequence with a known secondary structure, I was able to evaluate the efficacy of my program.

The basic approach for this algorithm is to create a dynamic programming table that compares the string of RNA to itself. The table is initialized with zeros on the main diagonal and on the diagonal to the left of the main one. Each index of the table is then filled in with a score. At each index i,j , there are four possible choices for determining the score, and thus the maximum of these scores is chosen for the table. At an index i,j , the goal is to determine the optimal secondary structure that contains the most base pairings. This is done by breaking down the sequence into subsequences, and breaking down those subsequences further into even smaller subsequences. The first choice for index i,j is to pair rna[i] and rna[j] and attach to best structure for rna[i+1:j-1]. The second choice is to add rna[i] to best structure of rna[i+1:j]. The third choice is to add rna[j] to best structure of rna[i:j-1]. Finally, the fourth choice is to combine two optimal structures for rna[i:k] and rna[k+1:j]. Each time a pairing of bases is added to the structure, the score increases by 1. The score of the optimal alignment is thus located in the upper-right corner of the dynamic programming table. After scoring is completed, a backtracking method is used to translate the filled scoring matrix into the optimal secondary structure of the RNA.

The scoring matrix for my test sequence of RNA is shown in Figure 2 along with the output graphics depicting the secondary structure in Figure 3.

G	**A**	**C**	**A**	**C**	**G**	**A**	**C**	**G**	**A**	
0	0	1	1	1	2	2	2	3	3	**G**
0	0	0	0	0	1	1	1	2	2	**A**
0	0	0	0	0	1	1	1	2	2	**C**
0	0	0	0	0	1	1	1	2	2	**A**
0	0	0	0	0	1	1	1	2	2	**C**
0	0	0	0	0	0	0	1	1	1	**G**
0	0	0	0	0	0	0	0	1	1	**A**
0	0	0	0	0	0	0	0	1	1	**C**
0	0	0	0	0	0	0	0	0	0	**G**
0	0	0	0	0	0	0	0	0	0	**A**

Figure 2. Scoring matrix for test sequence.

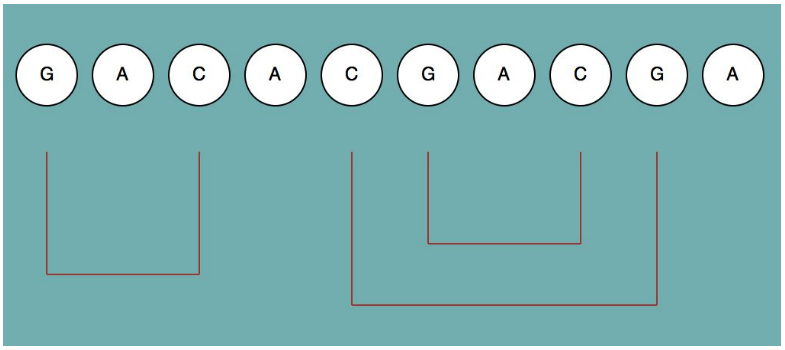

Figure 3. Output graphics of algorithm using test sequence as input. Red lines indicate base pairings.

Although the program ran well with the small test sequence, I ran into trouble when I tested it on the tRNA sequence. The actual secondary structure of alanine tRNA is shown in Figure 4 along with the output given by my implementation of a variation of the Nussinov algorithm shown in Figure 5.

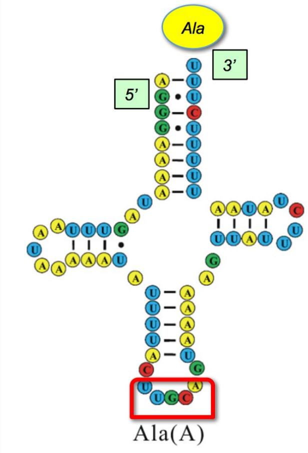

Figure 4. Alanine tRNA secondary structure (image from Alberts et. al., Molecular Biology of the Cell: Fourth Edition)

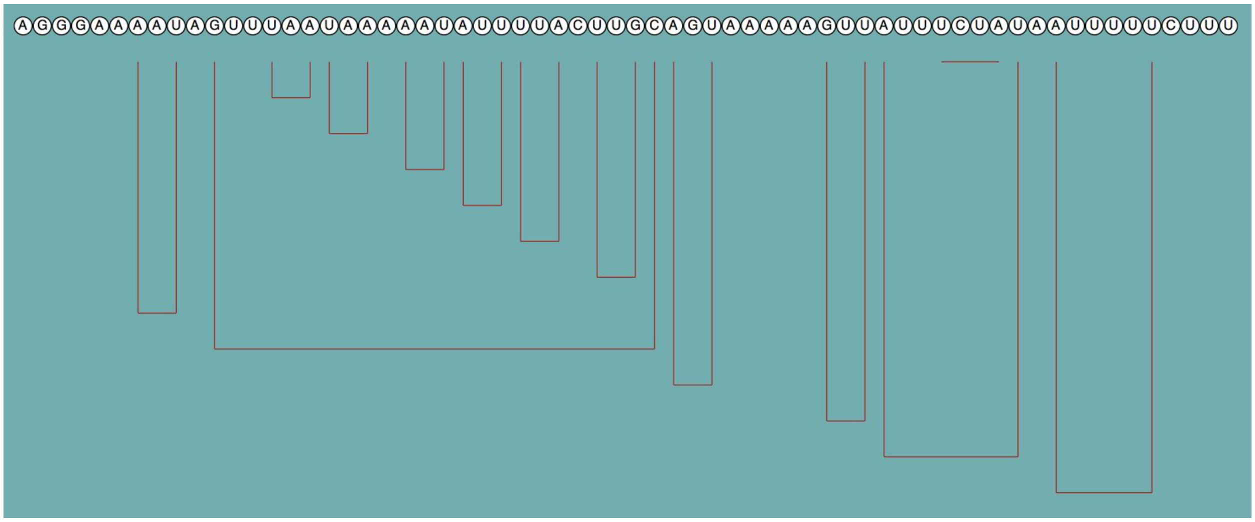
AGGGAAAAUAGUUUAAUAAAAAUAUUUUACUUGCAGUAAAAAGUUAUUUCUAUAAUUUUUCUUU

Figure 5. Output graphics of algorithm using alanine tRNA sequence as input. Red lines indicate base pairings.

Thus, it is evident that my implementation of the Nussinov algorithm, which is actually a slight variation of the original algorithm, still needs to be improved. One possible option for improving the scoring conditions would be to somehow weight consecutive base pairings, which are more favorable in RNA secondary structures.

References:

S. Will. “RNA Structure and RNA Structure Prediction.” MIT Department of Mathematics. Fall 2011.

Mathews, David H., Walter N. Moss, and Douglas H. Turner. “Folding and Finding RNA Secondary Structure.” *Cold Spring Harbor Perspectives in Biology* 2.12 (2010): a003665. *PMC*. Web. 10 May 2017.