RNA Structure Prediction: Nussinov Algorithm

Alejandro Medina Diaz

25/10/2024

1 Objectives

The objective of this project is to predict the secondary structure of RNA molecules using the Nussinov algorithm, which is a dynamic programming approach. The project focuses on identifying stable base pair configurations that maximize the molecule's stability by optimizing for the highest number of compatible base pairs between nucleotides. The Nussinov algorithm is particularly relevant for biological applications where understanding RNA structure helps elucidate its role in various biological functions, such as gene regulation and catalysis.

Problem Context RNA molecules consist of a sequence of nucleotides (A, U, C, G), where certain pairs (A–U, C–G, and G–U) can bond to form stable structures. This pairing creates the RNA's secondary structure, a crucial intermediate state before it folds into its 3D tertiary form. Predicting these pairs is important because RNA's structure is key to its function.

Algorithmic Focus The Nussinov algorithm identifies the optimal pairing pattern by analyzing all possible base pair combinations and determining the configuration with the maximum pairings. It solves this by defining subproblems over smaller subsequences of the RNA string, storing solutions in a table to avoid redundant calculations. This approach allows it to compute solutions in (n^3) O(n^3) time with (^2) O(n^2) space, where n is the sequence length.

2 Experimental Setup

Describe the configuration used in the experiments. This implies the following: (1) indicate what kind of experiments will be conducted (i.e., indicate in which way the algorithm will be run and what will be measured) and what will be the particular parameters that will be used in those experiments (i.e., their numerical values); (2) provide a description of the computational environment in which the experiments are run (see Table 1).

Table 1: Computational environment considered.

CPU	11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz 2.90 GHz
OS	Windows 11 Home version 23H2
Java	Java 22.0.2 2024-07-16

3 Empirical Results

A summary of the experimental results is provided in Tables 2-2 in the Appendix, along with the statistical fitting of the data to different growth models.

Describe the results, in particular Figure 1. The data shows a strong alignment with theoretical predictions, particularly in relation to the algorithm's cubic time complexity. The statistical fitting applied to different growth models confirms this, as the best fit was observed with a power-law relationship close to O(n^3), validating the expected behavior of the algorithm as the RNA sequence length increases.

The curve aligns with the data, especially for larger sequences, suggesting that the algorithm scales predictably in terms of time as the complexity of the input grows. The relatively small error bars on each point further support the conclusion that the variability in the algorithm's performance is minimal and that it performs consistently under the tested conditions.

This is generally the point with the exponential functions, of course the importance of having a good processor is there, but where it takes actual relevance is on the algorithm efficence.

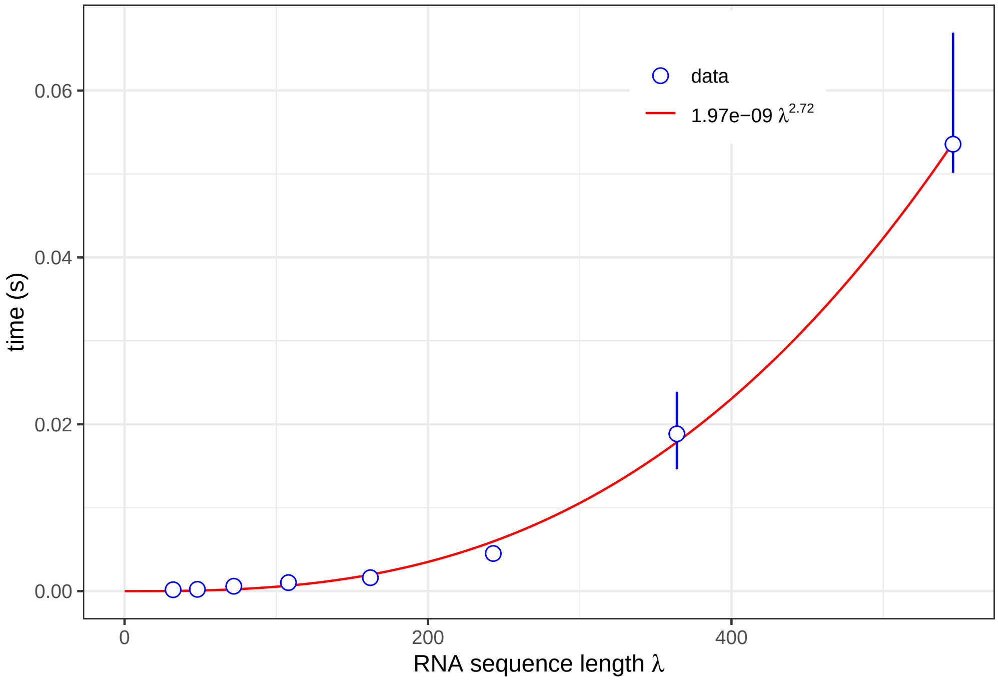

Figure 1: Time required for finding the optimal folding for increasing RNA sequence length

4 Discussion

The results of the experiment offer a clear validation of the Nussinov algorithm's theoretical predictions. In the first plot, we see how the measured execution times for different RNA sequence lengths (represented by blue circles) closely follow the red curve of the power-law fit. This curve is described by the formula

1.97e^−9   ^2.72, where   represents the length of the RNA sequence. The exponent, approximately 2.72, is very close to the expected cubic time complexity of O(n^3) for the Nussinov algorithm. The fact that the red curve aligns so well with the blue data points, especially as the sequences grow longer, shows that the actual performance of the algorithm matches the theoretical expectations. Moreover, the error bars, indicating variability, are relatively small, which means the algorithm's execution times are consistent across different runs for the same sequence length. There's very little unpredictability in how the algorithm performs, particularly for smaller sequences.

Looking at the second visualization, the scatterplot matrix offers additional insight. It confirms the strong correlation between the length of the RNA sequence and the time it takes for the algorithm to process it. Each relationship between these variables reinforces the idea that the Nussinov algorithm handles the increasing complexity of larger sequences in a predictable and controlled manner. The matrix shows that the execution time steadily increases with the sequence length, as expected, with minimal fluctuations between runs, which further underscores the stability of the algorithm's performance.

Overall, these results suggest that the Nussinov algorithm not only behaves as predicted but does so consistently. The fit between the data and the theoretical model is strong, and there are no surprises in the way the algorithm scales with sequence length. This consistency means that, while the cubic time complexity is not optimal for very large sequences, the algorithm is reliable for the lengths tested. If performance issues arise with much larger sequences, further optimizations or alternative algorithms might be considered, but for now, the Nussinov algorithm performs exactly as expected, demonstrating its effectiveness in this context.

A Appendix

A.1 Data Summary

Table 2: Summary of the experimental results. Q1, Q2 and Q3 represent the 1st, 2nd (i.e., median) and 3rd quartile respectively. All times are in seconds.

length λ	**time (Q1)**	**time (Q2)**	**time (Q3)**
32	7.94e-05	1.71e-04	2.62e-04
48	1.85e-04	2.19e-04	2.59e-04
72	5.28e-04	5.92e-04	7.11e-04
108	8.94e-04	1.03e-03	1.24e-03
162	1.47e-03	1.62e-03	1.91e-03
243	4.29e-03	4.52e-03	5.04e-03
364	1.47e-02	1.89e-02	2.39e-02
546	5.01e-02	5.36e-02	6.69e-02

A.2 Model Fitting

A.2.1 Power-Law Fit

##

```
## Formula: time ~ a * length^b
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## a 1.966e-09  1.094e-09   1.796    0.123
## b 2.717e+00  8.901e-02  30.524 8.21e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0007687 on 6 degrees of freedom
##
## Number of iterations to convergence: 13
## Achieved convergence tolerance: 4.282e-06
```