Visual Implementation of the Nussinov Algorithm

CS 466 Final Project Report (available at https://github.com/srikur/nussinov-viz)

SRIKUR KANUPARTHY, University of Illinois Urbana-Champaign, USA
ISABELLA LEBOVIC, University of Illinois Urbana-Champaign, USA
NEHA KAKI, University of Illinois Urbana-Champaign, USA

ACM Reference Format:
Srikur Kanuparthy, Isabella Lebovic, and Neha Kaki. 2022. Visual Implementation of the Nussinov Algorithm: CS 466 Final Project Report (available at https://github.com/srikur/nussinov-viz). 1, 1 (December 2022), 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The motivation of this paper is to discuss a web application that provides an interactive implementation of the Nussinov algorithm, which computes a pseudoknot-free secondary structure of a given RNA sequence with the maximum number of complementary base pairings. The application then provides the visualization of the RNA secondary structure in real-time. The user enters a string of nucleotides, and our implementation shows a representation of the structure with the dot-parenthesis format in order to indicate which nucleotides are paired. There is also a table which represents the score for each cell as determined by the Nussinov algorithm, along with the backtrace for the specific inputted sequence, with backpointers in each table cell. Lastly, a weighted graph representation of the determined pseudoknot-free secondary structure aims to assist in the visualization of the determined structure.

The central dogma of molecular biology states that genetic information flows from DNA, which is then transcribed into RNA, which in turn is translated into proteins. RNA is a single-stranded molecule that is made up of a ribose sugar backbone with attached nitrogenous bases that include adenine (A), cytosine (C), uracil (U), and guanine (G). These bases primarily interact with each other by forming hydrogen bonds, and there is nucleotide complementarity between A and U, with two hydrogen bonds–a slightly weaker interaction than between C and G, with three hydrogen bonds. The pairing of G to U is occasionally observed, but is much less common than the others, and is not as stable

Authors' addresses: Srikur Kanuparthy, srikurk2@illinois.edu, University of Illinois Urbana-Champaign, USA; Isabella Lebovic, lebovic2@illinois.edu, University of Illinois Urbana-Champaign, USA; Neha Kaki, kaki2@illinois.edu, University of Illinois Urbana-Champaign, USA.

XXXX-XXXX/2022/12-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

of a pairing. The interactions between the different nucleotides result in the formation of structures within a single molecule of RNA. This forms what is referred to as the secondary structure of the molecule, which can consist of substructures such as stacks, hairpin loops, internal loops, and multi-branched loops.

Scientists used to believe that RNA's sole purpose was as a messenger to be immediately translated into a protein. However, it has been found that RNA performs multiple functions based on its three-dimensional structure, and there are many different types of RNA that are suited to different purposes. For example, the structure of an RNA molecule can influence the susceptibility of its backbone and bases to be modified by certain enzymes that attach methyl groups to molecules. Understanding the secondary structure of an RNA molecule can help to understand what functions it performs, and how.

When determining the pairings that will occur between nucleotides, it is important to note that each nucleotide can only be paired with one other nucleotide, so the secondary structure can be determined by a set of non-overlapping nucleotide pairs. Nested base pairs refer to portions that are bound together and are not overlapping each other. When half of one stem of an RNA loop structure is between two halves of another stem, this is referred to as a pseudoknot. Pseudoknots are less common in RNA, and complicate the determination of which base pairs might be bound to one another.

The Nussinov algorithm determines a pseudoknot-free secondary structure with the maximum number of complementary base pairings, given an RNA sequence. Mathematically, it can be defined as follows:

$$s[i,j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i+1, j-1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \\ s[i+1, j-1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \\ s[i+1, j], & \text{if } i < j, \\ s[i, j-1], & \text{if } i < j, \\ \max_{i<k<j}\{s[i,k] + s[k+1, j]\}, & \text{if } i < j. \end{cases} \tag{1}$$

The results are stored in a table $s[i, j]$ that denotes the maximum number in a subsequence from v_i to v_j. In considering a sequence from the beginning (at index i) and the end (at index j), the cases that occur include a pairing between the base at index i and j, the base at i could be unpaired, the base at j could be unpaired, or there could be a bifurcation where the base at i is paired to some middle index k, and the cases between i to k, and $k+1$ to j are considered. The pairings that result in the maximum number of complementary base pairings can be determined by performing a backtrace through the table, and it is possible that there might be multiple optimal results. Additionally, the base pairing can be represented by a dot-parenthesis format where, in the RNA sequence, two paired elements have a set of parentheses, while unpaired elements have a dot (or dash). A force directed graph can assist in a visualization of the secondary structure based on the pairings determined from an implementation of the Nussinov algorithm.

The Nussinov algorithm can also be adjusted to address situations that would be more likely to occur in practice, such as adding a minimum length to each hairpin loop. From a structural standpoint, it is unlikely that two bases in an RNA strand that are immediately

next to each other would be paired together, since there might not be enough space between the two bases for them to have hydrogen bonding interactions with each other. Therefore, the Nussinov algorithm can also include a minimum length variable that will provide secondary structures that take this into account.

2 METHODOLOGY

Since we decided to host our application on a static GitHub Pages site, we were confined to a mostly vanilla JavaScript implementation – the only outside libraries we used were jQuery and D3.js. The former offered several quality-of-life functionalities while the latter allowed us to easily create the force-directed graph. Our first step was to implement three functions for the Nussinov algorithm: first, the actual recurrence that generates the dynamic programming table as well as storing backpointers for each cell; second, a function for the traceback, which stores the path of the optimal solution to the algorithm and stores a separate array containing only the matched base pairs; and third, a short function that takes in the sequence and matched base pair array and generates the dot-parenthesis structure for the sequence.

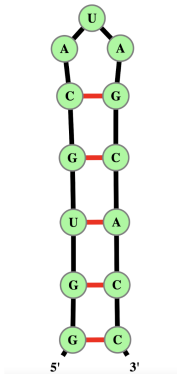
We initially implemented all three functions in a Google Colaboratory notebook in Python, allowing for an easy environment in which to test functionality before taking the time to write everything in JavaScript. In the end, all functionality for the web application was implemented in JavaScript to run client-side in the browser.

The sequence input section of the website includes an input for specifying the minimum length of a hairpin loop, should the user want to do so. Otherwise, this value is set to a default value of zero. We also implemented other features for the application, including a red-highlighted path for the optimal solution calculated by the Nussinov algorithm's backtrace. Hovering over any of the tables in the cell will highlight that cell teal, while also highlighting yellow the cell(s) that point to where the value derived from. A tooltip is also created that hovers next to the cursor.

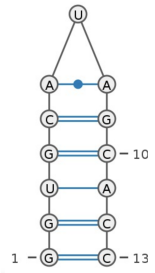
It is important to note that the traceback step introduces ambiguity into the predicted secondary structure, in contrast to the unambiguous nature of the actual Nussinov algorithm. Since the forward progression of the Nussinov algorithm only cares about the value of the number of base pairs, there is never any confusion as to what a cell in the dynamic programming table's value should be. However, this introduces ambiguity when tracing back to recover the optimal path, as there are often multiple options as to which cell to choose. The actual area in the code that influences the traceback's decision is the ordering of the if-statements. Therefore, we chose to follow the ordering presented by the algorithm in Professor El-Kebir's lecture slides.

For the force-directed graph, which serves as the visualization of the predicted RNA secondary structure, we relied on the data science visuals library D3.js, which has standard functions for implementing such a graph. The nodes of the graph are the nucleotides in the sequence, which are connected by black lines if they are adjacent and do not form a base pair, and red lines if they are adjacent and form a base pair. Since the graph is influenced by certain parameters, such as charge (the repulsion between nodes) and the distances between nodes, we allow for the user to set these values, with the graph automatically and immediately updating after such changes. The force-directed graph also allows for

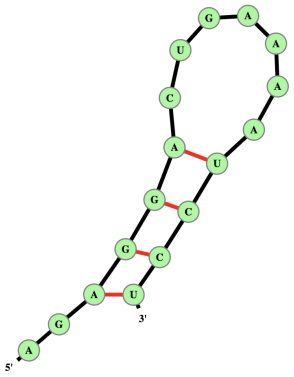
zooming and the manual dragging of nodes, which can help rectify cases of overlapping nodes or links.



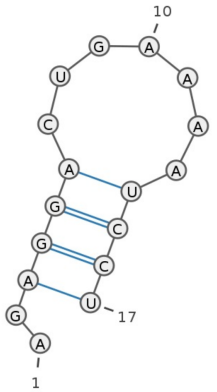
(a) Our Implementation



(b) Real Structure



(c) Our Implementation



(d) Real Structure

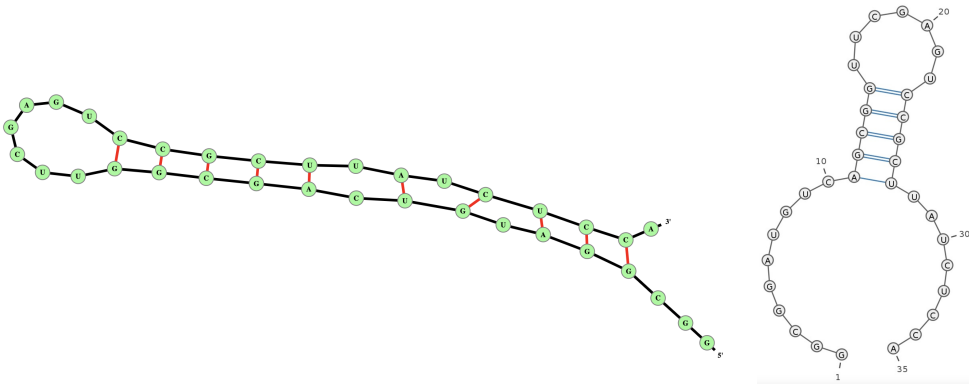
Fig. 1. **Comparison of the Structure Visualization from our Implementation to the Observed RNA Structures.** (a) Our implementation of the RNA structure with reference number 1ESH in the PDB database, using a minimum hairpin loop length of 1. (b) The real structure as shown in the bpRNA-1m database for the same RNA molecule as in part a. (c) Our implementation of an RNA sequence from the Cretava Tapia organism, using a minimum hairpin loop length of 5. (d) The real structure as shown in the bpRNA-1m database for the same molecule as in part c.

3 VALIDATION RESULTS

Our implementation was tested using RNA sequences and their known structures, found in the bpRNA-1m database. We selected sequences that were small enough to reasonably test, and did not contain pseudoknots. We then also adjusted the minimum hairpin loop length to produce a result more similar to what would be seen in real-world scenarios. Adjusting this variable was essential to seeing resulting structures that matched reality, but

the value of the minimum hairpin loop length was not static between all structures, as the environments between the origination of the different RNA structures have an impact on the secondary structure formed, as well as other intrinsic factors that affect the formation in practice.

Many sequences that we tested resulted in the same base pairing and structure visualization as would be expected, once the minimum hairpin loop length was altered (Fig. 1). This was also not accounting for certain structures that included base pairings between nucleotides that are not as commonly seen, like between G and U, or even between two of the same nucleotides. For some sequences where our results did not match the expected results, this was a product of the multiple possible solutions that can be found using the Nussinov algorithm. The produced output still had the same number of maximum complementary base pairs, and the bases were paired correctly–it was just done in a different order because there is no set control on knowing which output of the algorithm will be the one more likely to be observed in real life. There were also other sequences that did not match because there are other factors that affect an RNA secondary structure (Fig. 2). The maximum number of complementary base pairings is not always what occurs in practice, since properties such as free energy and stability can also determine which secondary structure is formed.



(a) Our Implementation (b) Real Structure

Fig. 2. **Inexact Comparison Between Structures**. (a) Our implementation of an RNA sequence from the Alloispermum Scabrifolium organism, using a minimum hairpin loop length of 7. (b) The real structure as shown in the bpRNA-1m database for the same RNA molecule as in part a. It can be seen that, despite the hairpin loop and base pairings underneath being the same between both structures, our implementation calculated further pairings down the rest of the structure. This is because the Nussinov algorithm maximizes the number of complementary base pairings, despite this not being what is always seen in reality.

4 CONCLUSION

The Nussinov algorithm, while not able to perfectly model the secondary structure of an RNA molecule in real scenarios due to the assumptions of a pseudoknot-free structure that

maximizes complementary base pairing, gives a good approximation in polynomial time. This can be used to better understand and predict the functionalities of the molecule, as RNA can perform multiple roles, and its three-dimensional structure can heavily impact its capabilities. We completed an interactive implementation of the Nussinov algorithm that can be used to understand how the complementary base pairings were chosen, as well as visualize the resulting secondary structure.

5 FUTURE WORK

Since there are multiple possible optimal results of implementing the Nussinov algorithm on the same sequence, we can in the future include backtraces and the resulting secondary structure for all possible solutions, or for all of the most likely possible solutions. We can also update the possible pairings to include some that are less common, like between U and G, to be considered as a possibility in the conditions under which such pairings are more likely to occur.

Some future work could also seek to address limitations and potential inaccuracies of the Nussinov algorithm as it is currently being implemented. There is an algorithm that is commonly used in practice to determine pseudoknot-free secondary structure, called Zuker's algorithm. This seeks to minimize the free energy of the RNA molecule and therefore maximize its stability instead of maximizing the number of complementary base pairs. This is done using a "nearest neighbor" model, as well as estimates of thermodynamic parameters for the interactions between bases and loops to compute a score for all possible structures. The results of implementing this algorithm could be compared with our current Nussinov implementation to visualize the differences that result in these two strategies, as well as to determine what might be a more likely secondary structure. Considering pseudoknots in the structure as well would likely be more accurate, but an algorithm that minimizes free energy while also taking into account pseudoknots has been determined to be NP-hard, which makes it much less time efficient in practice.

6 REFERENCES

[1] Y. Wan, "RNA," Encyclopædia Britannica, 10-Nov-2022. [Online].
Available: https://www.britannica.com/science/RNA. [Accessed: 18-Dec-2022].

[2] "bprna-1m," bpRNA, 25-Mar-2018. [Online].
Available: http://bprna.cgrb.oregonstate.edu/about.php. [Accessed: 18-Dec-2022].

[3] R. C. S. B. P. D. Bank, "RCSB Protein Data Bank," RCSB PDB. [Online]. Available: https://www.rcsb.org/. [Accessed: 18-Dec-2022].

[4] G. Lei et al., "CPU-GPU hybrid accelerating the Zuker algorithm for RNA secondary structure prediction applications," BMC Genomics, vol. 13, no. Suppl 1. Springer Science and Business Media LLC, p. S14, 2012. doi: 10.1186/1471-2164-13-s1-s14.

[5] m, “Nussinov algorithm to predict secondary RNA fold structures,” Bayesian Neuron, 06-Dec-2020. [Online]. Available: https://bayesianneuron.com/2019/02/nussinov-predict-2nd-rna-fold-structure-algorithm/. [Accessed: 18-Dec-2022].

[6] S. Hammer and P. Kerpedjiev, “Viennarna Web Services,” RNA/DNA secondary structure prediction for single sequences, sequence alignments and RNA-RNA interactions. [Online]. Available: http://rna.tbi.univie.ac.at/forna/. [Accessed: 18-Dec-2022].

[7] M. Bostock, “Force-directed graph,” Observable, 16-May-2022. [Online]. Available: https://observablehq.com/@d3/force-directed-graph. [Accessed: 18-Dec-2022].