6.047 Computational Biology Midcourse Report

Improving The Runtime Of Nussinov's Algorithm By Partitioning

(Previously: Novel RNA Folding Method
Using Classification And Clustering Techniques)

Nicole Power and Lily Seropian

November 25, 2013

Progress

Logistical

We have familiarized ourselves with the RNA strand database (which is sadly lacking in documentation). We used the website's search function to determine the average lengths of different structural elements, which is used in our partitioning algorithm. We figured out what information in the database would be relevant to our project and how to perform some basic queries on the database using MySQL and a Python wrapper for MySQL. We learned how to interpret CT (connect) format, since the database came with CT files for all of our sequences. We also found a mentor, Sean Simmons, who is a graduate student in Bonnie Berger's group.

Coding

All of the coding done so far is hosted at https://github.com/lilyseropian/rna-folding. So far, we have:

- Probability distributions based on structure lengths, which support probabilistic sampling
- A partitioning algorithm, which given a structure length partitions a given sequence into subsequences of that length
- A folding algorithm that partitions a sequence and then calls a scoring algorithm to determine the optimal partitioning and structural classification, then concatenates the results together into a folding for the original sequence
- A scoring algorithm, which given an aligned sequence of bases scores the folding, with higher scores given to better foldings

Theoretical

We did not write a classification or clustering algorithm because during the planning for these algorithms, we realized there are several significant problems with using partitioned classification to compute the folding. These problems are described in the **Preliminary Results** section. Upon this realization, we came up with a new plan for completing the project, which is detailed in the **Revised Aims** section.

Preliminary Results

Previously, we planned to implement a Naive Bayes Classifier to determine the structural element which best characterizes a subsequence of an RNA strand. We successfully implemented a partitioning function for the RNA sequence, but unfortunately determined that the Classifier would not be a suitable means of determining the structure of the subsequence. Because the length of structural elements is fairly small, the partitions were too short to give useful characterising information. We also found we did not have a wealth of data on attributes with which to characterise the subsequence.

There was also an issue that the partitioning function did not consider which bases were capable of pairing, which is key to RNA folding, but non-trivial to add, and we are uncertain of how to move from classification back to pairing in a format which could be compared for accuracy programmatically. Finally, we would need to ignore so many factors of true RNA folding--pseudoknots, free energy, stacking interactions--that we think the accuracy would suffer greatly, and we wouldn't learn much from the output.

Revised Aims

Aim 1: Develop a partitioning algorithm that enables a significant improvement in runtime over existing RNA folding algorithms.

Traditional RNA folding algorithms consider the entire segment of RNA. While these are able to achieve ever increasing success, they also require the time to classify the entire, potentially long, segment, and dynamic programming approaches take $O(n^3)$ time to compute. We will develop a partitioning algorithm that will split the RNA up into constant size segments. The optimal size of these segments will be determined by experimentation, but we estimate that we will use around 10 to 50 base pairs. The first half of a subsequence will be taken from the beginning of the RNA, and for the second half all subsequences of that size in the remainder of the RNA will be considered. The

small segments resulting from the partitioning will be folded individually using Nussinov's algorithm. The best partition, determined by the score outputted by Nussinov's algorithm, will be chosen and removed from the sequence. This will be repeated until the whole sequence is folded. This gives us an $O(n^2)$ algorithm. For every iteration, we partition the sequence into a linear number of partitions, each of constant size We then run Nussinov's algorithm on said constant size partition. We next remove one of these partitions from the sequence, and repeat until the whole sequence is classified. This gives us $O(n)$ iterations with $O(n)$ partitions of constant size per iteration, resulting in an $O(n^2)$ algorithm. The accuracy may suffer because we are computing solutions that are more locally optimal than globally optimal, but because RNA folds into more or less discrete segments anyway (structural elements), we should not suffer a large loss of accuracy.

Aim 2: Implement Nussinov's algorithm for use of folding and scoring.

Nussinov's algorithm can be used both to compute an optimal folding and to score said folding. We will modify the scoring algorithm we did in Pset 2 to compute the folding, and adjust our partitioning algorithm to produce output that can be fed directly into Nussinov's.

Aim 3: Determine parameters and approach that yield the best results, and compare accuracy and runtime with existing algorithms.

There are two aspects of our algorithm that we are interested in evaluating: accuracy and time. In order to evaluate accuracy, we will compare the results of our algorithm, and the results of running Nussinov's on the entire non-partitioned sequence. This comparison will be done via the scoring metric that Nussinov's algorithm uses. To evaluate time, we will use Python's timeit utility.

Adjusted Timeline

Due Date	Task
11/27/13	Write code to parse database for pairing and original sequence Write code for Nussinov's Algorithm Update code to score base pairing Update partitioning function
12/2/13	Analyse results

12/4/13	Final report
12/11/13	Presentation