CISC 471 - Proposal

Andrew Ma (20030440)
Rayan Shaikli (20059806)
Hershil Devnani (20001045)

March 5, 2021

1 Introduction and Motivation

The algorithm that we have chosen to complete is the Assessing Assembly Quality with N50 and N75. We chose to take a look at this algorithm given the fact that, often times, we are given problems to generate contigs, find motifs, score sets of DNA strands, make adjustments for misalignments etc. But we have not come across a method to actually identify if we have a good basis of data to begin with. The Assessing Assembly Quality with N50 and N75 problem takes a look at the quality of DNA contig reconstruction, and these N values give us an indication as to whether we need to revise our data or perform another set contig generation.

In some of our assignments we have already taken a look at reconstructing strands of DNA given contigs, searching for frequently appearing k-mers or patterns and searching for motifs. One thing we have not done, however, is use or develop any type of method that could provide us with a quantitative quality metric. We have implemented some of these alignment and scoring techniques for the assignments, but we have not yet come across any method that could actually help quantify the quality of our approach. We chose this problem so that we can gain a better intuition about some metrics that can help us analyze data sets.

Ideally we want to explore how to statistically quantify if a set of contigs are significant enough to pursue further analyses, such as the frequent k-mer problem of motif finding. More so in this case, if after we have a set of contigs, how a quality metric can give us an idea of whether we have an opportunity worth pursuing for any additional analyses we would like to conduct.

2 Research

1. Briefly explain the inputs and outputs of the algorithm. What is it's space and time complexity and why?

   The input of the algorithm is a collection of DNA strings composed of at most 1000 DNA strings, where the combined length does not exceed 50 kbp (kilo base pairs). The DNA strings are separated by a new line. The output of the algorithm is the N50 and N75 for the given collection of strings. Both space and time complexity of the algorithm are $O(n)$. This is because the algorithm needs to iterate through the collection of DNA strings to determine their length and sort them accordingly which is $O(n)$ time. Space complexity of the algorithm concerns ordering the DNA strings based on their length which is $O(n)$.

2. Explain what kind of experiments you would like to perform and why they would be important to know.

   Some of the experiments we would like to perform on our algorithm include comparing the significance of different scoring types, such as N50 vs U50 vs L50, for a given data set. In addition, we can experiment with data sets of varying contig lengths, i.e. data sets with large contigs, short contigs or a mix, as well as data sets containing varying amounts of repeating patterns. The data sets can also be varied in proportionality ratios such as the proportion short contigs to large contigs, or the proportion common patterns among the contigs and understanding how these factors affect the assembly quality metrics we are measuring.

3. Hypothesize an addition to the algorithm for a specific use-case to improve it. I.e. make an assumption on the data, and show how you could extend the algorithm to perform better with that data.

An addition we can make to the assembly quality algorithm is for the specific use case of when we have a set of contigs where the proportion of very short contigs to long contigs is large. A set of contigs with that contains a greater number short contigs may negatively impact an N score, so a modification to the algorithm we can make here is to systematically remove short sets of contigs that may not significantly continue to the contig assembly. One method this can be achieved is by analyzing the short contig sequences for frequently occurring short sequences, and then disregarding these recurring sequences from the contig assembly score.

In addition, some changes can be made to the algorithm to evaluate the assembly quality using metrics other than N50 or N75. These include the L50 metric, which measures assembly quality as the smallest number of contigs whose lengths makes up half the genome, the U50 metric which removes overlapping contig sequences, or even newer percent based metrics such as UG50%. Modifications to the original assembly quality algorithm can allow us to compare and analyze the difference between the N50, L50 and U50 scores of a given set of contigs. The use of these alternative metrics may provide us with advantages over N50 such as reducing errors due to poor assembly, eliminating inflated scores due to overlapping contigs or eliminating lower N50 scores caused by small contigs. As well, the use of newer percentage based scoring may allow insight into which scoring representations are suitable given a particular context or base set of contigs.

4. Explain how you intend to visualize and summarize your findings? Graphs, tables, etc.. What kind of graphs? What would be important to show to prove your proposed addition is better than the baseline algorithm?

   - Graphs
     - L50, N50, U50 and UG50% score vs proportion of short to large contigs increases
     - L50, N50, U50 and UG50% score vs set of contigs
     - How the score compares as the data set of contigs grows
   - Tables
     - Percentage variance in the scores given by the different scoring methods given the same data set, i.e. the assembly quality spread of each scoring method
     - Analysis of scoring methods discussing if the proposed addition gave better results or more significant results, i.e. If addition helped better identify if the contigs were significant or not
     - Table of runtimes of each of the scoring implementations

3 Resource Allocation

We decided to break down the components of the research project into the following high-level sections:

- Initial Setup + N50 Algorithm
  - Implementing the N50 Sorting Algorithm (Hershil)
  - Implementing the N50 Grouping Algorithm (Rayan)
  - Developing data sets for testing (Everyone)
- Experimental Additions to the Algorithm
  - L50 Algorithm (Rayan)
  - U50 Algorithm (Andrew)
  - UG50% Algorithm (Andrew)
  - Sorting Algorithms (Andrew)
- Data Visualization, Graphs and Tables (Everyone)
- Runtime Analysis (Everyone)
- Report Components (Everyone)