Homework 1
CS1820 Spring 2018

Out: 2/14
Due: 10pm, 2/21

Homework Handin

Please turn in a hard copy of your homework with your Banner ID only in the CS182 bin located at the top of the stairs on the 2nd floor of the CIT.

1 de Bruijn Graphs

Recall the concept of de Bruijn graphs presented in class, and please remember that they are directed; you will lose points if your graphs are not directed. Answer the following questions about the following target DNA sequence:

ATTATTCTTG

1. Write down all of the 3-mers of this sequence (with repeats if they exist).

2. Determine all the distinct 2-mers necessary to create all the 3-mers you found in 1.1. As an example, for the 3-mer XYZ, you would need the 2-mers XY and YZ in order to create XYZ.

3. Write down all the 4-mers of this sequence (with repeats if they exist).

4. Determine all the distinct 3-mers necessary to create all the 4-mers found in 1.3.

Now, suppose that we carried out a perfect sequencing of our target sequence, meaning that we have complete coverage, all our reads are of length $k$, there are no sequencing errors, and the overlapping region between adjacent reads is $k - 1$ bases long. After carrying out a perfect sequencing with $k = 3$ of our target DNA sequence, we end up with a set of reads equivalent to your answer from 1.1. After carrying out a perfect sequencing with $k = 4$ of our target DNA sequence, we end up with a set of reads equivalent to your answer from 1.3.

5. For each of the 3-mers determined in 1.1, think of it as an edge. For each of 2-mers determined in 1.2, think of it as a node. Construct a de Bruijn graph using these nodes and edges. Remember that each $k$-mer should only have one edge corresponding to it, but because there may be repeated $k$-mers, you could end up with multiple identical edges. You can take a picture of your graph, or find a way to represent it in Latex or some other software.

6. Carry out a process similar to the one in 1.5 and construct a de Bruijn graph, except this time use the set of $k$-mers from 1.3, and the set of $(k - 1)$-mers from 1.4.

Now, you should have two different de Bruijn graphs representing the results of our perfect sequencing for $k = 3$ and $k = 4$.

7. Does the graph from 1.5 have a Eulerian path? If so, how many Eulerian paths are there? For each of these paths, write down the DNA sequence the path corresponds to.

8. Does the graph from 1.6 have a Eulerian path? If so, how many Eulerian paths are there? For each of these paths, write down the DNA sequence the path corresponds to.
9. Take a look at your answers for the previous two questions. What may be possible drawbacks in using the de Bruijn Graph to reconstruct DNA? Is there any information lost? In what types of cases does a de Bruijn graph fail to help us get the original target DNA sequence?

The following questions are not related to the DNA sequence for the previous questions.

10. Consider the following set of reads: ATC, CTG, TCC, GGA, CCT, GAT, TGG. Construct a de Bruijn graph based on these reads, and determine the target DNA sequence they came from assuming perfect sequencing. If more than one DNA reconstruction is possible, please write down all of them.

11. Suppose that in addition to the reads from the previous question, we also have the following reads: TAT, ATA. Construct a de Bruijn graph based on all of the reads from this and the previous question, and determine the target DNA sequence they came from assuming perfect sequencing. If more than one DNA reconstruction is possible, please write down all of them.

2 Maximal Scoring Subsequences

Problem Description and Definitions

The Maximal Scoring Subsequence problem takes as input a sequence \( x = (x_1, x_2, \ldots, x_n) \) of (not necessarily positive) real numbers, called “scores”. The goal of the problem is to find those contiguous subsequences of \( x \) that have the greatest total score, where the score of a subsequence is determined by summing up the scores of its elements. Namely, the score, \( S_{i,j} \), of a subsequence, \((x_i, x_{i+1}, \ldots, x_j)\), is equal to:

\[
S_{i,j} = \sum_{i \leq k \leq j} x_k
\]

It is important to develop some rigorous definitions while solving this problem. Consider the sequence \( x = (x_1, x_2, \ldots, x_n) \), with length \( n \), and consider \( i, j \) such that \( 1 \leq i \leq j \leq n \). Define the highest scoring subsequence to be the one that maximizes \( S_{i,j} \), as described above.

Next, we will define the \( k \)th best subsequence as the one that maximizes \( S_{i,j} \) among all subsequences of \( x \) that are disjoint (meaning contain no overlapping regions) from all \( k - 1 \) best subsequences. What this means is that our second best subsequence, for example, will not share any elements with the (first) best subsequence. We will also not allow for zero-scoring prefixes or suffixes in our maximal sequences. As an example, the best scoring subsequence in (1, -1, 5, 6) will be (5, 6) and not (1, -1, 5, 6).

Questions

1. Why do you think we only consider disjoint subsequences when looking for “successive best” scoring subsequences?

2. Why is it a good idea to exclude the zero-scoring prefixes and suffixes in our definition?

3. Can you devise a naive algorithm that would be able to find the maximal scoring subsequence? Describe the steps of the algorithm and explain what its runtime will approximately be (constant, linear, quadratic, exponential, etc.).

4. Bonus: One might think that there would be a linear time algorithm to solve this problem, visiting each \( x_i \) once. Consider the following alternate definition for a maximal scoring subsequence. Let \( K \) be a non empty score sequence. We say that \( I \) is a maximal scoring subsequence of \( K \) if and only if:

   - all proper subsequences of \( I \) have a lower score, and
   - no proper supersequence of \( I \) in \( K \) satisfies the previous condition.

Can you think of a way to use this definition and come up with a linear time solution to the maximal scoring subsequence problem?