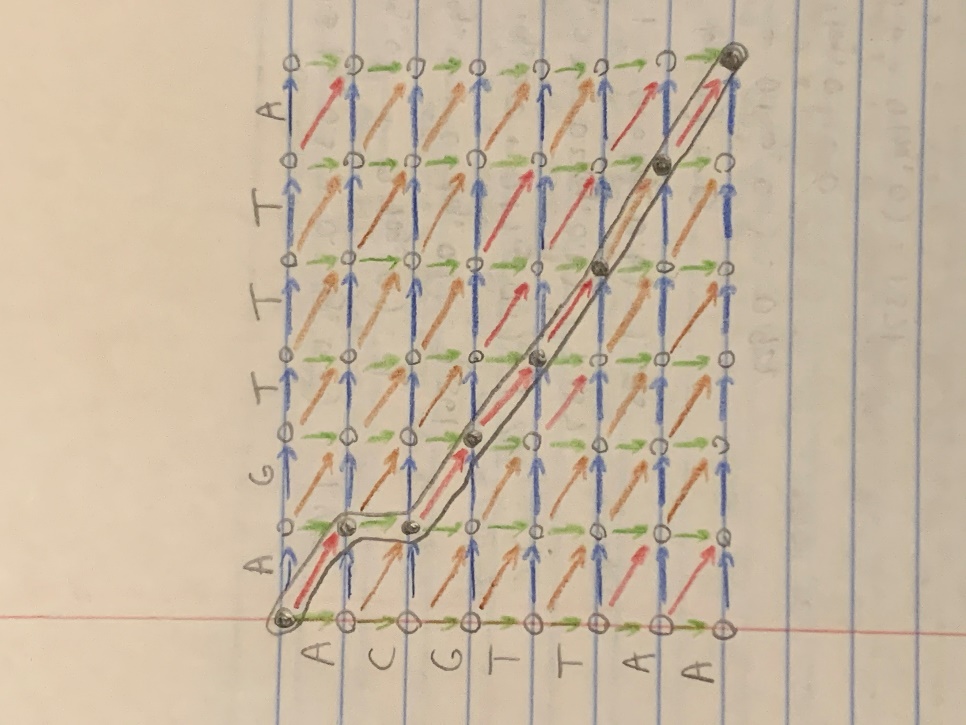
Aman Patel  
CSCI-B 363

October 12, 2020

Assignment 3

1. Complete spectrum:
2. Mr. Study’s algorithm
   1. The algorithm is mostly correct, but it may output values that do not correspond to amino acids. For example, while convoluting a spectrum, you may come across many 15s. While this may be the most common number found, it is far less than the lowest mass of an amino acid and should thus be ignored.
   2. The runtime of the algorithm is due to the double-nested for-loops.
   3. Instead of using an array of size MaxMass, Mr. Study could use a dictionary with the key representing the mass and the value representing the count. This would remove the MaxMass operations to initialize and add to *ConvolutionSpectrum*.
3. Mr. Fuzzy’s claim
   1. Mr. Fuzzy is correct in claiming that the theoretical spectrum of a cyclopeptide is a superset of the theoretical spectrum of any corresponding linear peptide because cyclopeptide spectrums include all elements of linear spectrums along with wrapping peptides.
   2. He is also correct in claiming that a linear peptide can always be formed from a cyclopeptide spectrum because the cyclopeptide spectrum contains all the pieces to reconstruct the corresponding linear peptide along with others that can be treated as noise and ignored.
4. DP algorithm for counting peptides problem
   1. Start with a list of all amino-acid masses
   2. Using a top-down approach, subtract the masses in the list from the given mass
   3. Repeat this process to form a recursion tree
   4. If at any point the mass becomes 0, set the value of that subproblem to 1.
   5. If the mass becomes negative, set the value of that subproblem to 0.
   6. Store the value of each subproblem in a table in case the same subproblem is encountered again later
   7. Trace back from each leaf, adding the values of all leaves beneath it, until you reach the central node, which is the original mass
   8. Output the value for the central node
5. Parent mass from noisy spectrum
   1. First, remove all masses smaller than the mass of the smallest amino acid, as these masses cannot represent amino acids.
   2. Then, perform a spectral convolution to determine which amino acids are most likely to be components of the cyclopeptide
      1. This can be done by using the algorithm shown in Problem 2, with the adjustments given in my answer.
   3. Output the sum of the amino acids as the parent mass
6. and alignment graph



Red: Match

Brown: Mismatch

Green: Insertion

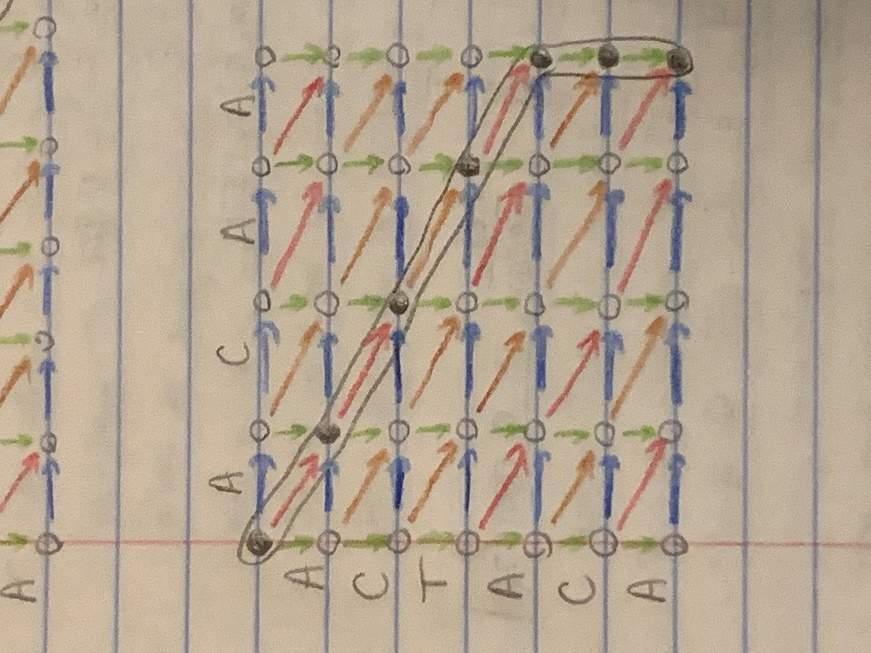
Blue: Deletion

* 1. Optimal alignment:
  2. Score:

1. Global Alignment Overlap Alignment
   1. The global alignment algorithm works very similarly to the overlap alignment algorithm, but it is finding the optimal alignment for whole strings, while overlap alignment finds the optimal alignment between the suffix of one string and the prefix of another.
   2. Assuming the first string will always be suffixed, and the second will be prefixed, run global alignment on each combination of prefix and suffix. For example, if the two input strings are and , each possible suffix of the first string should be globally aligned with each possible prefix of the second.
      1. with , with , with , etc.
   3. Keep track of the score of each global alignment and output the suffix and prefix index with the minimum score
2. Multiple sequence alignment
   1. No, it is not guaranteed that correct multiple sequence alignment will yield correct pairwise alignment.
   2. For example, given the three strings the following alignment is optimal:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | - | T | G | C |
| A | A | T | - | C |
|  | A | T | G | C |

* 1. With this multiple sequence alignment, the pairwise alignment of and is:
  2. This is not optimal, as there are two gaps present that would not be present in a correct pairwise alignment.

1.  and alignment graph

Red: Match

Brown: Mismatch

Green: Insertion

Blue: Deletion

* 1. Optimal alignment:
  2. Score:

1. Modifying the Divide-and-Conquer algorithm
   1. The divide-and-conquer algorithm assumes the mid-node will be in the middle column, which can lead to significant problems in runtime evenness, especially when the mid-node is near the top or bottom of the middle column.
   2. The shortest possible path from the source to the sink is the larger of or . The longest possible path from the source to the sink is . Let us assume .
   3. Averaging these two lengths yields , the average path length from the source to the sink. In practice, the average will be lower, as matches and mismatches are often preferred over insertions and deletions. However, for the purpose of this assignment, I will not estimate the actual average.
   4. Dividing the average in half once again will yield the average distance from the source to the mid-node: .
   5. Now, instead of having the mid-node in a specific column, it can be on a circular radius nodes away from the source and ideally the sink as well.