The goal of this project was to implement the cyclopeptide sequencing for an ideal spectrum. I then also add the goal of reaching and implementing the leadership cyclopeptide sequencing with an experimental spectrum as well.

The data was acquired from the Rosalind website. The data is just in the form of list variable that is defined in my main function. This is the same for the leadership function. The data is just in the form of list variable that is defined in my main function. The date I used for the ideal spectrum was [0, 113, 128, 186, 241, 299, 314, 427]. The data I used as the experimental spectrum was [0, 71, 113, 129, 147, 200, 218, 260, 313, 331, 347, 389, 460].

This is the pseudocode for the project, below this I go into detail of the high level steps I have bolded the parts I talk about in the high level steps.

**CYCLOPEPTIDESEQUENCING(Spectrum)**

Peptides ← a set containing only the empty peptide

while Peptides is nonempty

    Peptides ← Expand(Peptides)

    for each peptide Peptide in Peptides

        if Mass(Peptide) = ParentMass(Spectrum)

            if Consistent(Peptide) = True

                output Peptide

                remove Peptide from Peptides

            else if Peptide is not Consistent with Spectrum

                remove Peptide from Peptides

            else if Peptide is not Consistent with Spectrum

                remove Peptide from Peptides
**LEADERBOARD CYCLOPEPTIDE SEQUENCING**

(Spectrum, N)

Leaderboard ← {0-peptide}
LeaderPeptide ← 0-peptide

while Leaderboard is non-empty

   Leaderboard ← Expand(Leaderboard)

   for each Peptide in Leaderboard

      if Mass(Peptide) = ParentMass(Spectrum)

         if Score(Peptide, Spectrum) > Score(LeaderPeptide, Spectrum)

            LeaderPeptide ← Peptide

         else if Mass(Peptide) > ParentMass(Spectrum)

            remove Peptide from Leaderboard

   Leaderboard ← Cut(Leaderboard, Spectrum, N)

output LeaderPeptide

The high-level steps of my code were as follows:

First, I had my overall function cyclopeptide sequencing spectrum which took as an input an ideal Spectrum (list of integers) and output all possible peptide strings in a list. In order to make this function I defined a couple of smaller functions. A consistent function which checks to see if the peptide is consistent with spectrum. It took as input a peptide (list of mass) and a spectrum. It outputted True or False. A function that finds the total mass of a peptide which takes an input of peptide (list of masses) and outputs total mass. Finally, I defined an expand function which add to the list of total peptides (this accomplishes the branch step). The input is the current peptide list and it doesn’t return anything, but instead modify the peptide list that was inputted.
Second, I defined my overall function for the leadership cyclopeptide sequencing which took as inputs an ideal Spectrum (list of integers) and a N (the highest score, with ties). This function utilizes the subfunctions I defined for the ideal spectrum problem, but also added more. The first I added was a function that defined the ideal spectrum for a given peptide. This took as inputs a peptide (string) and outputs a spectrum (list). I also define a score function which compares the experimental spectrum to the ideal spectrum and gives a score based on number of values that match. The inputs are peptide (list of masses) and spectrum(list). It outputs a score (integer).

Overall, I found that the leadership function was the closest in finding the out what the peptide was, but it still is not perfect. There is some aspects of the branch and bound function where in order to save function time you may actually cut away the correct answer. However, for an estimation it seems to do a good job at finding the spectrum and any other method would be too long and cumbersome to output any form of answer.