Presently, all attempts to sequence cyclopeptides rely on “branch-and-bound” methods to do so. Amino acids are added to “candidate” peptides that grow with each addition, and after each addition all peptides that do not fit the experimental spectrum are discarded. Sadly, such methods require the bulk of mass spectrometry data to be accurate representations of the cyclopeptide. As this requirement presently cannot be presently met, a new algorithm is required to succeed. Here I will explain the steps and concepts behind the algorithm I have developed.

Step one, predicting the weight of the full cyclopeptide. While difficult, cyclopeptide sequencing has one distinct advantage over linear peptide sequencing. When the weight of any segment of the cyclopeptide is subtracted from the full weight, the result is the value of the “opposite” strand of the cyclopeptide. In other words, one can predict additional fragments of the cyclopeptide. Successfully “drawing” the cyclopeptide circle requires knowing the full length (or in this case, weight) of the circle first.

In my code, I divided this step into two parts. In the first part, I add all data points of the experimental spectrum to every other data point. If the two data points happen to be opposite ends of the cyclopeptide sequence, their addition would equate the size of the full cyclopeptide. Because several of the true data points are so paired, the size of the full cyclopeptide will appear more often than a random value. After choosing frequently appearing segments, this step produces several candidate cyclopeptide weights. The second part uses known amino acid weights to normalize and further process the candidate weights. This part relies on the assumption that several data points in the experimental spectrum are short a single amino acid. Should the assumption prove true, then several data points merely require the addition of the missing amino acid weight to equal the full cyclopeptide weight. All such candidate data points are paired with their respective amino acid for each candidate full weight. By taking the average of the data points with their respective amino acid weights, the candidate full weight can be normalized. All candidate full weights that have few to no data points that are one amino acid short are no longer considered in the final evaluation.

The end result is still an approximation, and steps will be required to account for any inaccuracy. However, one of the end results should be within 0.2 Daltons or so of the actual full weight of the cyclopeptide.

Step two is where things get tricky and will require some background explanations. Visuals will be provided.

As stated previously, all currently proposed algorithms designed to sequence cyclopeptides require the addition of amino acids to growing candidate polypeptide chains. This algorithm requires looking at a cyclopeptide from a slightly different angle. The following is a pie chart representation of Tyrocidine B1, the cyclopeptide for which I have mass spec data. Different colors correspond to different amino acid weights, some of which have duplicates. Weights are represented by length, and the full length of the circle is approximately 1322.7.

\*Note: I didn’t notice the symmetry of this cyclopeptide until I created this visual. It probably relates to the function of the cyclopeptide, but I’m not sure if the pattern would reliably extend to other cyclopeptides in such a way as to use it in sequencing cyclopeptides.

Rather than look at the cyclopeptide as a series of weights, I have found it helpful to place emphasis on the places between amino acids, or the “junctions” of the cyclopeptide. To aid in this consideration, I numbered each junction, as demonstrated in the following visual. In looking at a cyclopeptide from this angle, each true data point represents two junctions. For example, the data point 405.19 (NQY, or 114.04+128.09+163.06) would represent junctions 7 and 10. The data point 99.06 (V) would represent junctions 1 and 10.

Note that, for each 2-junction pairing, there are technically two data points that could match. For the above example of NQY, the opposite strand of NQY (VKLFPWF, or 917.5) would *also* represent junctions 7 and 10. Fortunately, once the full sequence has been predicted, both data points can be represented by a single data point (dataPoint1 = fullWeight-dataPoint2).

From this perspective, several patterns become clear. It is evident that all junctions are touched upon by several data points (the lowest is junction 9, which is touched on by 4 data points in the experimental spectrum, while the highest is 3, which is touched on by 8). In terms of sequencing the cyclopeptide, this information is invaluable.

Sequencing the cyclopeptide requires analyzing and comparing the distances between junctions. Between any two neighboring junctions is a distance corresponding to an amino acid weight. In predicting the full weight of the cyclopeptide earlier, several candidate amino acids were determined. Each of these amino acids, then, represents two neighboring junctions (V represents junctions 1 and 10, P represents junctions 4 and 5 etc.).

Smaller data points, such as amino acids, are more likely to appear at the end of larger data points. Furthermore, each data point must start and end with amino acids that neighbor the junctions represented by that data point. For example, each data point ending at junction 4 must include an amino acid neighboring junction 4. That data point’s opposite must include the amino acid on the other side of junction 4.

Using this logic, the relative distance between further junctions can also be calculated. Take, for example, Valine (V), represented by the length 99.06 (blue) in the above illustration. Valine represents junctions 1 and 10. In our experimental spectrum, we have several cyclopeptide fragments who either neighbor or end in Valine. If a data point ends in Valine, it’s opposite fragment must neighbor Valine. To illustrate, the fragment VKLFPWF (junctions 10,7) has Valine at one end. NQY, the opposite fragment of VKLFPWF (also junctions 10,7), immediately neighbors Valine. As such, if one were to add Valine to NQY, it would result in another true data point, NQYV (junctions 1,7). Thus, by “flipping” a data point (VKLFPWF) and “adding” Valine, one would get another data point (NQYV). The relationship between these two data points is telling – they overlap over Valine (junctions 1,10) *and meet at a completely different junction,* namely junction 7.

This “flip and add” technique provides very useful data. Given an amino acid, you can discover all data points who have a “flip and add” match connected to that amino acid. Some of these are false positives, but of every true positive, each pairing represents another junction not directly connected to those on either side of the amino acid. In the above example, the VKLFPWF-NQYV pairing (data points 917.5- 504, respectively) represents junction 7. It was discovered because each junction on either side of the amino acid Valine (junctions 1 and 10) had another data point that connected them to a mutual third data point (a data point for 1,7 and a data point for 10, 7). For every other junction that can be tied to both junction 1 and 10, the “flip and add” technique will provide a pairing. For the amino acid Valine, the junctions 3, 5, 7, and 8 each have a “flip and add” pair representative. Presumably, if you line up the “flip and add” pairs in the proper orientation, you could recreate a cyclopeptide with junctions 1, 2, 4, 6 7, and 10 all represented. In other words, a 7-segment cyclopeptide chain that represents part of the full 10-segment cyclopeptide.

My present algorithm is an attempt to recreate this scenario for each given amino acid, piece together frequently occurring themes, and account for the variables that make the theory less straightforward. For example, in our previous example the amino acid F appears twice in the cyclopeptide chain. Thus, the “flip and add” technique would work too well, providing matching data points to the two separate amino acids and combining them into one list. Also, lining up the “flip and add” pairings in the proper orientation is no small feat. In summary, this theory provides useful puzzle pieces that can be used to reconstruct a cyclopeptide. The challenge is figuring out how to put them together properly. My code is my present attempt to do so, with fairly successful results that I am trying to improve upon.

My Tyrocidine B1 Data, bolded numbers being real data points:

372.2, 397.2, 402.0, **406.3**, 415.1, **431.2**, **448.3**, 449.3, 452.2, 471.3, **486.3**, **488.2**, 500.5, **505.3**, 516.1, 536.1, **544.2**, **545.3**, 562.5, 571.3, 599.2, 614.4, 615.4, 616.4, 618.2, **632.0**, 655.5, 656.3, **672.5**, **673.3**, 677.3, **691.4**, **692.4**, 712.1, 722.3, **746.5**, 760.4, 761.6, 762.5, **771.6**, 788.4, 802.3, 803.3, 818.5, **819.4**, 831.4, **836.3**, 853.3, 875.5, **876.5**, 901.5, 915.9, 916.5, 917.8, **918.4**, **933.4,** **934.7**, **935.5**, 949.4, **966.2**, 995.4, 1015.6, 1027.5, 1029.5, 1031.5, 1044.5, 1046.5, **1061.5**, **1063.4**, **1079.2**, 1083.7, 1088.4, 1093.5, **1096.5**, 1098.4, 1158.5, 1159.5, **1176.6**, 1177.7, 1178.6, 1192.7, **1195.4**, 1207.5, **1210.4**, **1224.6**, 1252.5, 1270.5, 1271.5, 1278.6, 1279.6, 1295.6, 1305.6, 1306.5, 1307.5, 1309.6