Extra Credit

**PP1 Explanation**

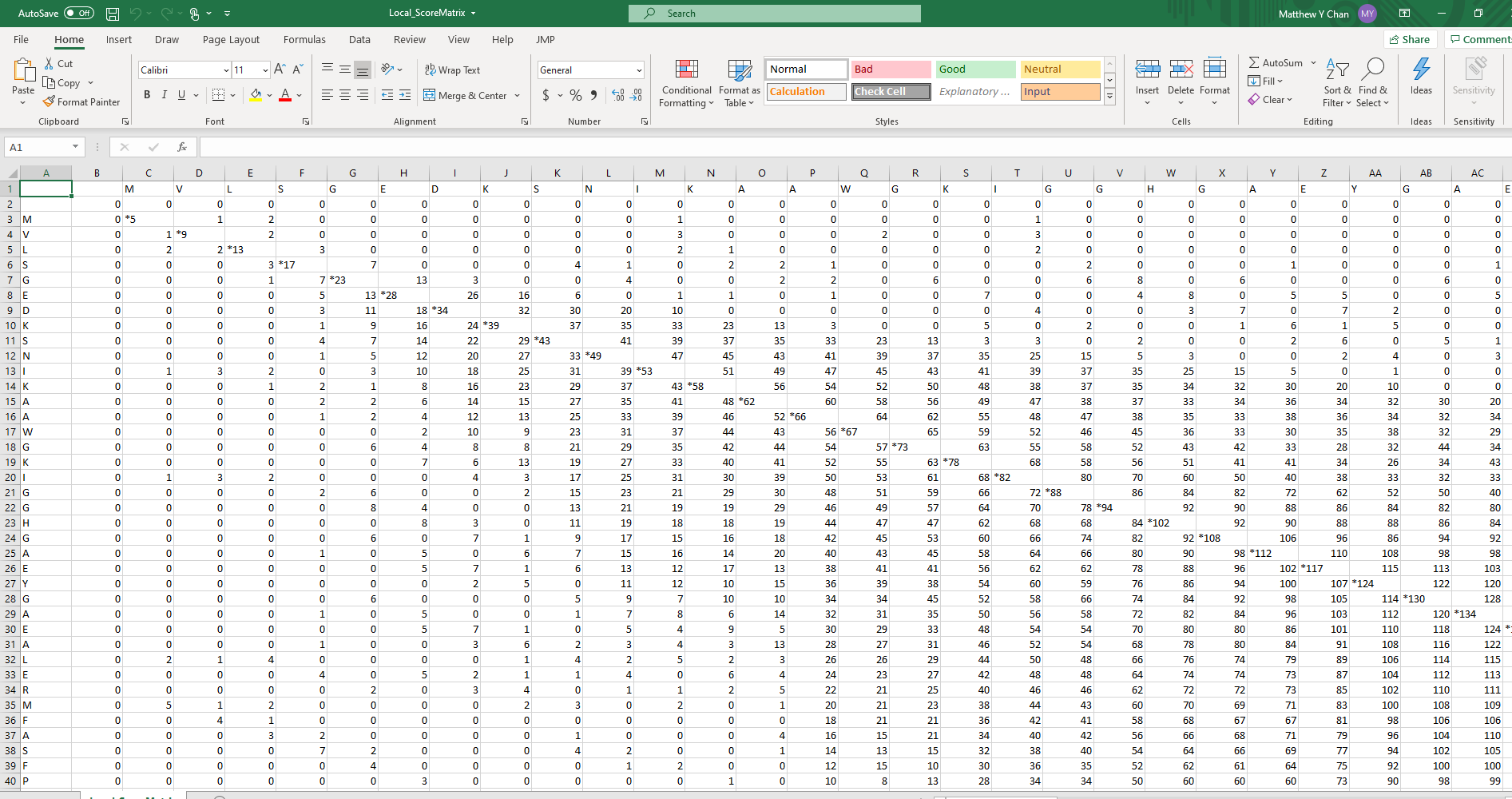
Local and global alignment were implemented in similar manners except for a few key differences in the scoring matrix and traceback. The overall idea for the alignment code was first getting all the necessary sequences and scoring matrices, then translating the DNA sequence, creating an appropriate score matrix, and tracing the score matrix back to get the alignment. This approach was implemented by using Python.

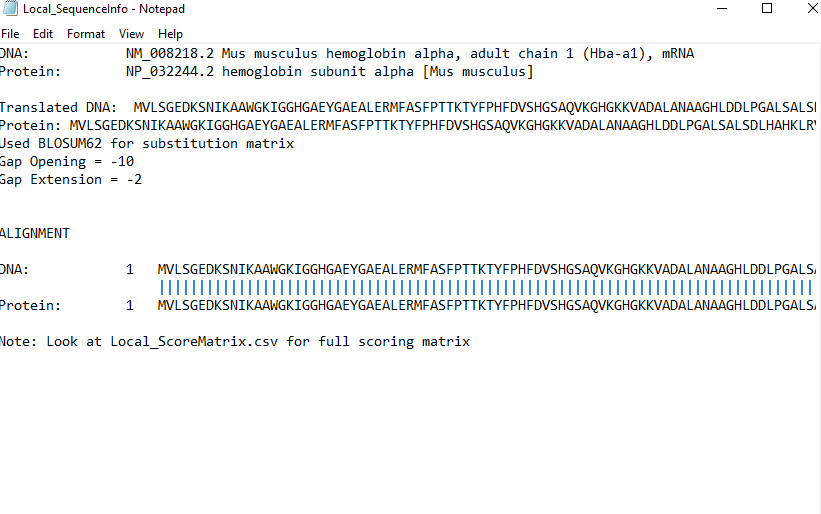
First, obtaining the test sequences was mouse hemoglobin which was obtained on NCBI, where the protein sequence was found and its corresponding coding DNA. The code was written so that a dialog opens, and you choose the protein and DNA sequences. The protein and DNA sequences have to be name prot\_seq and DNA\_seq respectively so the code will run correctly. The DNA sequence is then translated into its protein sequence. This is done by finding all the indexes of the start codon, ATG, and then translating until a stop coding is reached. There may be multiple start and stop codons depending on the reading frame and the longest translated protein was assumed to be the correct one due to the DNA sequence being the coding sequence. The BLOSUM62 scoring matrix was obtained from NCBI and created into a text file in which it is extracted.

Creating the score matrix was done by making a matrix with all values. The first row was initialized. The first row was all zeroes for local alignment and global alignment started with zero, then gap opening, and the rest were gap extension. The rest of the rows were initialized with the first number containing zero (for local alignment) or gap penalties (for global alignment), and the rest of the rows containing nothing used as placeholders for the actual score calculated later. The score was then calculated by going down the rows. First the amino acids at the corresponding position were determined and the score from the BLOSUM62 matrix was obtain depending on the two amino acids. Next, the three possible values were calculated depending on whether it was a gap or match. To determine if the gap opening or extension value should be used, a variable that for the previous score is used, 0 for match and 1 for gap. If previous was match then gap opening value was used, and if gap was previous then gap extension was used. These three values were then compared for the max value which was used as the score and added to the matrix. In addition, the max value and its position was tracked since it would be used in local alignment traceback.

Traceback for both global and local alignment have the same idea except the starting position is different. For local alignment the traceback started at the max value and was traced until zeroes are hit. Global alignment starts at the end of the sequences and finishes at the beginning of the sequences where the score is 0. The traceback works by determining the three possible numbers that it can go to: up, left, and diagonal. Whichever value is greater is what determines the next position of the traceback. Matches or diagonal movements are prioritized and each iteration the amino acids are added to a string which will be used as the alignment. If it moves up or left that means, there is a gap which is denoted using a dash. The path is modified so that the number starts with an asterisk.

Finally, the results are outputted into a csv file and a text file. The csv file is the scoring matrix which can be opened in excel to show the path for the alignment. If conditional formatting is used then the cells with an asterisk can be highlighted showing the path. The text file contains information about the alignment such as the sequences used for the protein as well as the translated DNA sequence. Also, information like the gaps and the type of scoring matrix is included in the text file. Some sample results are included in the submission as well as example screenshots below showing the local alignment.





**PP2 Explanation**

Suboptimal alignment is like local alignment as described in PP1. The approach is the same as local alignment of making a scoring matrix from two sequences then doing traceback to get the alignment, but suboptimal alignment finds more alignments that may not be the best. The whole idea of suboptimal alignment is that once a letter pair has been used, it cannot be used again, which is done by setting all used scores to zero. Therefore, to implement this idea the score matrix was calculated and then multiple tracebacks were done. Score matrix calculation was calculated the same as the regular local alignment.

A function was written to find the max score and its position in the score matrix. A variable that defines how many iterations are necessary is also preset. Each traceback, a new max value is found and this is where the traceback starts. Once a score is used, the score at that position becomes zero so that it will not be repeated. Alignments are further calculated until the number of alignments specified earlier has been reached. A csv file and text file containing alignment information is outputted. The csv file contains the score matrix where the paths are denoted using the alignment number followed by the score (ex. (1)258). Therefore, the path can be shown using conditional formatting in excel. The text file contains information about the alignment such as the sequences used and the translated protein sequence. It then has all the alignments and where each alignment after 1 is considered suboptimal, while alignment 1 is the optimal alignment. The csv file and text file examples are shown below and included in the submission.

