**CS 476 Bioinformatics Algorithms FA22**

**Project I: Sequence Alignment and Substitution Matrices (UG&G /G)**

1. **Description**

In project 1, you need to implement the following sequence alignment algorithms you’ve learned in class.

1. Global Alignment
2. Local Alignment
3. Semi-Global Alignment
4. Global Alignment with Affine Gap Penalty (for graduate students only)
5. **Program Specification**

Data:

1. You can find three pairs of amino acid sequences in the **P1AASeqs** folder. Your program needs to be able to read and align each pair of sequences with each alignment algorithm.
2. The substitution scores matrices are in the **P1SubMatrices** folder. The scores matrix listed different scores for the match or mismatch of two amino acid. Notice that the scores matrix is symmetrical, and the amino acid order in all matrices:

A,R,N,D,C,Q,E,G,H,I,L,K,M,F,P,S,T,W,Y,V

What to be input:

1. Prompt the user to enter the name of two amino acid sequences files.
2. Prompt the user to enter the name of the substitution matrix.
3. Prompt the user to select the alignment type (global, local, semi-global, and affine).
4. Prompt the user to set a gap penalty (or gap-open & gap-extend for affine).

What to be output to console:

1. The alignment of two sequences (Use ‘-’ to represent gap characters).
2. The OPT matrix. (M, Ix , Iy  for affine)
3. The optimal alignment score.
4. **Extra Credit** **(Optional)**
5. PAM mutation: As you learned in the class, align sequences using PAM(n) mutation probability matrix given n units of evolutionary divergence. Under this setting, users need **to input n as evolutionary divergence, and output both your PAM-n mutation probability matrix and your PAM-n substitution scores matrix to the console** (You can first prompt to ask mutation or not, if yes, then jump to this function).
6. Nucleotide sequence translation: Type in a nucleotide sequence and translate it into an amino acid sequence. (Assuming coding, 5’-3’). Output your translation to the console (Similarly, before typing in sequences, you can prompt as to whether the sequence is nucleotide or peptide, if nucleotide, then jump to this function).
7. **What to turn in**
8. **Code:** You need to submit your complete code with comments for each function cell and point out some key sentences (OPT matrix update, backtrack, etc.) for each alignment algorithm. Add a comment block on the top of your code that tells users how to compile/execute your program on home server [home.cs.siue.edu](http://home.cs.siue.edu).
9. **Demo:** Record a video to demo your program referring to the guidelines for the video demos given in “P1DemoGuidelines.docx”.
10. Compress code and demo into a .zip file following the naming convention <FirstNameLastName>P1.zip and submit. For example, if my name is Bonan Yang, then I would be submitting the zipped file BonanYangP1.zip .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*Any bioinformatic-related toolbox, libs are *PROHIBITED,* to complete projects, which means you need to write your own source code, but you can use any math/matrix libs.

\*You must be responsible for your own submission; the grader will test your code with various input combination on SIUE home server. A failed run (no output, exception thrown, etc.) will result in a score of 0 for the part to which it belongs.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*