Homework 2: Anchored Global Sequence Alignment

CSCI 5481, Computational Techniques for Genomics
University of Minnesota
Instructor: Dan Knights

Instructions

- Please turn this assignment in on the course web page.
- There are multiple files to turn in. All text and code should be placed into a single folder with a name like *lastname_exerciseXX*. The folder should then be compressed and submitted as a single archive (.zip or .tgz)
- You must do this work on your own, although you are encouraged to have general discussions with other students. The work you turn in must be your own. Your code will be checked for overlap and for surprising idiosyncrasies in common with other submissions.
- Please write the names of all students with whom you discussed the assignment at the top of your code.
- Please include copious comments in your code. Full credit will only be given for code that is fully commented, meaning that every line that is not completely obvious needs a comment. Partial credit may be given for broken/non-functioning code if the code is well-commented.
- You may use any programming language you wish.

Background

This homework assignment is implementation of an anchored version of the standard Needleman-Wunsch algorithm and application of the algorithm to align spike and capsid proteins from SARS-CoV-1, the original “SARS” virus that appeared in 2002, and SARS-CoV-2, the virus that causes COVID-19. The anchored global sequence alignment assumes known matched regions between two sequences and applies Needleman-Wunsch algorithm to align the unaligned regions between the matched ones. Implement the anchored global sequence alignment algorithm and align the given sequences. (Hint: write Needleman-Wunsch first -- then the anchored algorithm is a very simple extension of the Needleman-Wunsch algorithm and you only need to implement a wrapper function that calls your other function).

Dataset

Download and extract the SARS-CoV-1 and SARS-CoV-2 S and N protein sequences from the Homework02 directory in the Files section of the course Canvas page. For the matches files, i.e. "Match_S.txt" and "Match_N.txt", the first 2 columns represent the start and end positions of matched regions for the SARS-CoV-1 sequence, whereas the last 2 columns represent the start and end positions of the matched regions for the SARS-CoV-2 sequence. The positions are 1-indexed, like in R (first position is 1, not 0), and are inclusive of the end position.

Note: The anchored sections still count for the alignment score, even though you don’t have to align them. They may contain a few mismatches.

Input and Output Format:

You can hardcode your substitution matrix and gap penalty values (-3 for mismatches, 1 for a match, -2 for a gap; ignore the affine/linear gap penalty, meaning don't use a linear model. Just count every gapped position as -2, so if you have 3 gaps in a row, that's -6). The command line for calling your program should be of the form: `programname seq1.fasta seq2.fasta [matches.txt]`. Note that [matches.txt] means the third file is optional. If the matches.txt is not provided, your program should run standard Needleman-Wunsch. Output should be both the alignment score for this pair of sequences and the actual alignment itself printed with gaps. You may also use command-line flags to label your parameters, e.g. `python programname.py -q eq1.fasta -r seq2.fasta [-m matches.txt]`.

Treat any special characters the same as the ones in the alphabet, i.e. use the same match and mismatch costs.

Tasks

1. (25 points) : Implement the Needleman-Wunsch algorithm (don't forget your comments).
2. (25 points) : Based on your Needleman-Wunsch algorithm to implement the anchored version (don't forget your comments).
3. (25 points) : Use your algorithm to align the provided two pairs of sequences. Report the alignment and the alignment score.
4. (25 points) : For both pairs of sequences, permute the nucleotides in the sequences (use random library in your chosen language) and repeat the alignment 100 times. Report the distribution with a histogram of the random alignment score and mark the alignment score of the original sequences. Do this only for the non-anchored version.
5. (3 bonus points) : Output a visualization of the anchored S protein alignment with one sequence above the other, with a vertical bar connecting any positions that were aligned (meaning neither sequence has a gap). Find a visual way to show where each codon starts in each sequence. This will help us to see when the frameshifts get out of alignment between the two sequences due to insertions or deletions. Note: you can break the visualization up into pieces if you want to show it in a normal-sized document.

Deliverables

1. Source file (your code). Please note in your code the names of the people who worked on it.
2. Readme file (text). The readme file should contain instructions on how to compile and run the program.
3. Alignment results in a single file (text).
4. A pdf file giving the plots of the permuted and observed alignment scores in task 4.