Program Assignment

due day: 4 April 2023

This assignment requires you to implement an algorithm that find the best global alignment between two sequences S1 and S2 allowing affine gap penalty.

Your program should be flexible, i.e., it should be possible to:

- Align sequences over any alphabet. The alphabet can be {A, C, G, T} for DNA or 20 letters alphabet for amino acids, or any alphabet. The alphabet is specified in the parameter file.
- Use any score matrix. The score matrix is specified in the parameter file.
- Handle **affine** gap penalty.

Your program needs to output the alignment score and the alignment.

The input sequences are in FASTA format. For example,

```
>seq1
tgacaatccc
>seq2
tgaggatggt
```

The score matrix, the alphabet, the gap penalty score are specified in the parameter file.

You need to output the optimal score and an optimal alignment. The computed optimal alignment should be output in FASTA alignment format (also called Pearson format after the creator of the FASTA alignment program). In FASTA alignment format, the two aligned sequences are printed above each other with gaps inserted as described by the computed alignment. For example, assume match scores 2 and mismatch/insert/delete score -1, for aligning the above two sequences, the answer is

```
score=10

>seq1
tga-caat

>seq2
tgagc-at
```

Detail of the programming task

You are required to write one program:

```
   python align_DP.py parameter.txt input.txt output.txt
```

Testing data

You are given three sets of testing data.

The first testing dataset is without affine gap penalty. They are contained in parameter1.txt and input1.txt. The sample output files are output1.txt.

Similar information is provided for the other two testing datasets.

Note that everything after ';' in the dataset is comment. You must make sure your programs can read the parameter files and the input files. Also, your programs must follow the output format as stated in the sample output files.

Written question

You are required to submit a report.

In the report, you need to describe your algorithm for handling affine gap penalty.

Also, you need to perform analysis on the running time for your program.

For your analysis, you need to use some homology sequences. For example, you can get some homology sequences from http://eggnogdb.embl.de/#/app/downloads.

Submission

Please email your python program, a README on how to run your program and your report and analysis to wksung1@qq.com.