Bilkent University
CS481: Bioinformatics Algorithms
Homework Assignment #5
Fall 2020

INSTRUCTIONS

- Solve the following problems.
- You must write your code yourself. Sufficient evidence of plagiarism will be treated the same as for plagiarism or cheating.
- Non-compiling submissions will not be evaluated.
- Your code must be complete.
- Do not submit the program binary. You must submit the following items:
  - All of the source files
  - A script to compile the source code and produce the binary (`Makefile`).
  - A `README.txt` file that describes how the compilation process works.
- Submit your answers **ONLY** through the Moodle page.
- **Zip** your files and send them in only one zipped file. File name format `surname_name_hw#.zip`
- C / C++, Python 3, Java will be used as programming language. STL is allowed. The use of `getopt` function is **compulsory** for C/C++ programs. Python programs **MUST** use `argparse` module. Java programs **MUST** use an argument parser such as `ArgParser`
- All submissions will be compiled and tested on **Dijkstra server**.
- All submissions must be made strictly before the stipulated deadline.
- The overall fastest implementation wins. **Bonus** will be given for the fastest code.

1) SEQUENCE TO PROFILE ALIGNMENT

Aim: In this assignment, given **n** DNA sequences, $2 \leq n \leq 10$, in a **single aln-formatted** (multiple alignment formatted) file, we ask to implement **sequence to profile alignment**.
You may utilize the **naïve** implementation of Needleman-Wunsch from the previous assignment or you can write from the scratch. You will take gap penalty, match score, and mismatch penalty *via parameters*.
The length of the **aligned sequences** might be **at most 500 characters**. For the sake of simplicity, we give the alignment file (.aln) described below.

Parameters:

- −−**aln**: Only one aln-formatted file containing all given alignments "aligned_sequences.aln", which contains n DNA sequences line-by-line.
- −−**fasta**: Sequence fasta file to be aligned to the given profile.
- −−**gap**: gap penalty score.
- −−**match**: matching score.
- −−**mismatch**: mismatch penalty score.

Output:

- −−**out:** sequence.aln

2) EXAMPLE

Command line examples: Be sure that your code works using the following command (just one line):

```
alignSeqToProfile \
    --fasta seq.fasta \
    --aln aligned_sequences.aln \
    --out seq.aln \
    --gap ${gap_penalty} \
    --match ${match_score} \
    --mismatch ${mismatch_penalty}
```

seq.fasta

```
> sequence
CTAGATAATTGGAGATGATCAAATTTATAT
```

CTAGATAATTGGAGATGATCAAATTTATAT

aligned_sequences.aln

```
sequence1 ATAC---CTAATTGGAGATGATCAAATTTATAAT
sequence2 TTAT---CTAATTGGCGACGATCAAATTTATAAT
sequence3 ATAT---CTAATTGGTGATGATCAAATTTATAAT
sequence4 ATCA---TTAATTGGAGATGATCAATCCTAATGA
sequence5 CTGTACTTTTATTGGTGATAGTCAAATCTATAAT
```

seq.aln

```
sequence1 ATAC---CTAATTGGAGATGATCAAATTTATAAT
sequence2 TTAT---CTAATTGGCGACGATCAAATTTATAAT
sequence3 ATAT---CTAATTGGTGATGATCAAATTTATAAT
sequence4 ATCA---TTAATTGGAGATGATCAATCCTAATGA
sequence5 CTGTACTTTTATTGGTGATAGTCAAATCTATAAT
sequence  CTAG---ATAATTGGAGATGATCAAATTTATAAT
```

ATAC---CTAATTGGAGATGATCAAATTTATAAT
TTAT---CTAATTGGCGACGATCAAATTTATAAT
ATAT---CTAATTGGTGATGATCAAATTTATAAT
ATCA---TTAATTGGAGATGATCAATCCTAATGA
CTGTACTTTTATTGGTGATAGTCAAATCTATAAT