Algorithms in Computational Biology, 2016
Exercise 1 - Programming - Sequence Alignment

Due date: 04/12/2016

Submission of theoretical questions MUST be computer printed and submitted as PDF.

The solution for the programming part should be submitted as a tar file containing the code (Python or Matlab), the output plots/files and a README file.

1 Alignment Programming

In this question you will implement a sequence aligner, that given two FASTA files, type of alignment, and a scoring matrix, will print the optimal alignment and its score. FASTA is a common format for biological sequence:

```
>name of seq1
ACACGGTGGACCGGAT
AACACGGTAATACCAG
```

- Input:
  - FASTA files - For simplicity we will assume the aligner should handle only FASTA sequences from the nucleotide alphabet Σ, e.g $\Sigma = \{A, T, G, C\}$ (you may assume the input is correct, but we will add bonus for general implementation over any alphabet). You are encouraged to use the provided fastaread (in Python) or the built-in fastaread in MATLAB.
  - Score matrix - You are provided with a scoring matrix S. The first row and first column describe the characters in the alphabet or gap ($-$), and the rest of the cells describe the score for substitution or deletion, e.g $S_{A,A} = \sigma(A, A)$ (the score for match of A), $S_{A,T} = \sigma(A, T)$ (mismatch of aligning A and T), and $S_{A,-} = \sigma(A, -)$ (aligning A to a gap). We provide a sample scoring matrix: 'score_matrix.tsv' (in tab-seperated format) derived from NUC 4.2. You may assume that the matrix is in the expected format, and in a fixed order $\{A, C, G, T\}$ of nucleotides.

- Alignments: Given a pair of input sequences $X = (x_1, ...x_n)$ and $Y = (y_1, ...y_m)$, the aligner should support the following types of alignments:
  - Global alignment ("global") - In this type of alignment, the algorithm seeks for the best match between X and Y, such that all the characters of X and Y are aligned either one to another (e.g $x_i$to y_j ; where $1 \leq i \leq n$;$1 \leq j \leq m$), or to a gap (either x_i or y_j).
  - Local alignment ("local") - In this type of alignment, the algorithm seeks for the best substring match of X and Y, e.g the best alignment of $X_{s1:e1} = (x_{s1}, ..., x_{e1})$ and $Y_{s2:e2} = (y_{s2}, ...y_{e2})$, where $1 \leq s1 \leq e1 \leq n$ and $1 \leq s2 \leq e2 \leq m$.

It should be possible to invoke the aligner from command line using the following format:

- In Python:

```
python seq_align.py a.fasta b.fasta --align_type global
  --score score_matrix.tsv
```

- In MATLAB:

```
matlab -r "seq_align('a.fasta','b.fasta','global',
  'score_matrix.tsv');"
```

The program should print:

1. The optimal alignment (if there are multiple choices it is enougth to print one of them). In the following format:

```
TCGAATCG—CACGCGCGGCTCTCCTTAGAACCGGCCGGCTCCCGAATAA
TTGGGTCGGTTTCACCCGGTCTTCATCCG—CCGACTGTTTAAAAACCAA

TGTTTCAGTGTTTGACAAACTCAATCGGAGGTCTCGGAAGA——AGTATC
CAAGGTAAGAGGAGGGGAGCTTTGTTGTTGTTTTAACGTGTGTTAGTGAC

AAAAAAAAAAAA
AAAAAAAAAAAA
```

   e.g as blocks of 2 lines of length of up to 50 characters (the alignment for each of the sequences) followed by a line break.

2. Score for the best alignment

For testing purpose you are provided with some sample FASTA files that you can test as input.

Enjoy!