

Assignment

Sequence alignment

- Conditions:
 - At least one algorithm must be implemented.
 - Documentation that covers implementation details and results.
- Award:
15 points + 2 extra points for the gap extension penalty implementation
- Deadline:
6th. lesson (30.3.2017).
- Penalization:
-5 points for each additional week

Needleman-Wunsch & Smith-Waterman algorithm

Implement pairwise Global alignment algorithm (Needleman-Wunch) and pairwise Local alignment algorithm (Smith-Waterman) in an arbitrary programming language (e.g. Python, Perl, Java). Required arguments are two input sequences in a FASTA format, a scoring matrix in CSV format (e.g. blosum62 file), and a gap penalization.

Program 'alignment' should accept following arguments:

- -g/-l global/local pairwise alignment
- -s1 path to the first sequence
- -s2 path to the second sequence
- -e path to the score matrix in CSV format
- -p gap penalization
- -pe gap extension penalization (optional)

E.g.

./alignment -g -s1 A0PQ23.fasta -s2 Q9CD83.fasta -e blosum62.csv -p 4

Run **Needleman-Wunch algorithm** with two input sequences A0PQ23.fasta and Q9CD83.fasta, with the score matrix defined in the blosum62.csv file, and with the gap penalization equal 4.

./alignment -l -s1 EU078679.fasta -s2 CH954156.fasta -e blastmatrix.csv -p 4

Run **Smith-Waterman algorithm** with two input sequences EU078679.fasta and CH954156.fasta, with the score matrix defined in the blastmatrix.csv file, and with the gap penalization equal 4.

The final alignment and the corresponding score print to STDOUT in the following format:

- Alignment of the first sequence
- Alignment of the second sequence
- Score

Output:

TTGACAGTACATAG

TTGA-AGTTTGTAG

30