Prof. Dr. Daniel Huson
Algorithmen der Bioinformatik
Fachbereich Informatik
Mathematisch-Naturwissenschaftliche Fakultät

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Sequence Bioinformatics **WS 2021/22**

Assignment 2 **Due: Nov-3, 10 am**

In this assignment, please implement a Python program `global_aligner_YOUR_NAME.py` that reads as input a file containing two DNA sequences in FastA format and performs optimal global alignment.

The program should use a match score of 1, mismatch score of -1 and a linear gap penalty of $d = 1$.

Provide a command-line option `-m`, or `--mode`, that accepts as value 0 (default), 1 or 2.

1 Needleman-Wunsch basic implementation (3 points)

When launched using the option `--mode 0`, the program will run your own implementation of the Needleman-Wunsch algorithm and traceback, using the basic *quadratic space* formulation. The program should then print out the optimal score and an alignment that achieves the score.

2 Needleman-Wunsch with linear space (3 points)

When launched using the option `--mode 1`, the program will run your own implementation of the Needleman-Wunsch algorithm and traceback, using the *linear-space* divide-and-conquer modification. The program should then print out the optimal score and an alignment that achieves the score.

3 Needleman-Wunsch, no table (2 points)

When launched using the option `--mode 2`, the program will use a *recursive* implementation of the Needleman-Wunsch algorithm, based on the following pseudo-code formulation (which is not discussed in the script):

```
Function computeF($i, j$):
    if $i = 0$ and $j = 0$ then return $0$
    else if $i = 0$ then return $-j \times d$
    else if $j = 0$ then return $-i \times d$
    else return $\max \begin{cases} \mathsf{computeF}(i-1, j-1) + s(x_i, y_j) \\ \mathsf{computeF}(i-1, j) - d \\ \mathsf{computeF}(i, j-1) - d \end{cases}$
```

Here, $s(a, b) = 1$, if $a = b$ and -1, else.

Note that this implementation *does not* use a table and is thus not considered *dynamic programming*. What will be the significant drawback of this implementation?

4 Comparison (2 points)

Add code to estimate the amount of memory and time used. Run all three modes on the three files `short.fasta`, `medium.fasta` and `long.fasta`, which are provided as `data-02.zip` on Ilias) and produce a table or plot that illustrates how the run-time and memory usage compare between the three different modes. What do you think?

Example data

The file `medium.fasta` contains these two sequences:

```
>A.andrenof
GATGCAGTTCCAGGACGAATTAATCAATTGAATTTAACGACCTGGAATTTTTTTTGGTCAATGTTCTGAAATTTGTGGAATAAATCATAG
ATTTATACCAATTATAGTTGAATCAACATCATTTTAAATTGAATTTATAAAATAAATTA
>A.mellifer
TATTAAAGTTGATGCAGTTCCAGGACGAATTAATCAATTAAATTTAATTAGAAAACGTCCAGGAATTTTTTTTGGTCAATGTTCAGAAAT
TTGTGGTATAATTATACCAATTATAATTGAATCAACTTCATTTCAATATTTTATTGAGTAAA
```

Here is an optimal alignment for the two sequences (computed using `https://www.ezbiocloud.net/tools/pairAlign`):

```
Sequence 1: ----------GATGCAGTTCCAGGACGAATTAATCAATTGAATTT--------AACGACCTGGAATTTTTTTTGGTCAATGTTCTG
Sequence 2: TATTAAAGTTGATGCAGTTCCAGGACGAATTAATCAATTAAATTTAATTAGAAAACGTCCAGGAATTTTTTTTGGTCAATGTTCAG

Sequence 1: AAATTTGTGGAATAAATCATAGATTTATACCAATTATAGTTGAATCAACATCATTTTAAATTGAATTTATAAAATAAATTA
Sequence 2: AAATTTGTGGTATAA---------TTATACCAATTATAATTGAATCAACTTCATTTCAA---TATTTTATTGAGTAAA---
```