Integrative Transcriptomics
Prof. K. Nieselt,
Institute for Bioinformatics and Medical Informatics Tübingen
Prof. S. Nahnsen,
Institute for Bioinformatics and Medical Informatics Tübingen

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Lecture: Grundlagen der Bioinformatik SoSe 2023

Assignment 2 (20 points)

Hand out: Thursday, May 4
Hand in due: Thursday, May 11, 18:00
Direct inquiries via the ILIAS forum or to your respective tutor at:
mathias-alexander.witte-paz@uni-tuebingen.de
meret.haeusler@student.uni-tuebingen.de
simon.hackl@uni-tuebingen.de
carolin.schwitalla@qbic.uni-tuebingen.de

In order to take into account the existing expertise of all course participants and to encourage students with already very good knowledge, we have introduced alternative, more difficult sub-tasks (marked by ′) on this assignment sheet on an experimental basis. In the corresponding tasks, only one sub-task, e.g. either **a)** or **a)**′, has to be completed at a time and there are no extra points for the more difficult tasks. The choice of the task version is left to you.

Theoretical Assignments

1. Global and local alignment by hand (6P)

a) For the following two sequences, X = `TCACAACC`, Y = `TCCAC`, compute an optimal global alignment using the Needleman-Wunsch algorithm.

a′) For the following two sequences, X = `TCACAACC`, Y = `TCCAC`, compute an optimal global alignment using an adapted version of the Needleman-Wunsch algorithm that ensures that the two bases at position 6 in sequence X and 4 in sequence Y are aligned. Positions are 1-based. Also denote the adapted recursion for this problem.

b) For the following two sequences, X = `TATGCAGG`, Y = `TGATG`, compute an optimal local alignment using the Smith-Waterman algorithm.

In each sub task, use a linear gap penalty and the following scoring parameters:

$s(a, b) = -2$ if $a \neq b$ and $s(a, a) = +2$ and $d = 3$.

Hand in the DP matrix, as well as one alignment via traceback for each problem.

Hint: You can use `https://www.tablesgenerator.com/` to create a table in LaTeX. If you want to include pictures, please include only **good quality** pictures or scans.

2. Adapting Smith-Waterman (4P)

Adapt the Smith-Waterman Algorithm to align a short sequence (globally) in a long sequence (locally). Specify how to adapt the recursion of the Smith-Waterman algorithm and briefly explain (max. 100 words) why the adjustment achieves the desired effect. Can you think of a biological question to which you can apply this adapted algorithm?

Practical Assignments

3. **Smith-Waterman Algorithm** (10P)

   In this task, you are asked to implement a program that applies the Smith-Waterman algorithm with linear gap penalties to two DNA sequences to compute an optimal local alignment. The input to the program should be read from two FASTA files, each containing one sequence; use your FASTA parser from the previous assignment to read the files. The parameters for the file input, match score, mismatch score, and gap penalty should be parsed from the command line.

   a) Implement your program to the point where the DP matrix is correctly initialized, filled in and the optimal local alignment score is output to the console.

   b) Extend your program such that a trace-back matrix is computed while filling in the DP matrix. Your extended program should (additionally to a)) reconstruct **one** optimal local alignment and write the aligned sequences to a file.

   b′) Extend your program such that a trace-back matrix is computed while filling in the DP matrix. Your extended program should (additionally to a)) reconstruct **all** optimal local alignment and write the aligned sequences to a file.

   Apply your program to the sequences **MTB_target.fasta** and **MTB_query.fasta** (the files are located inside the resources directory of the code skeleton) with the following parameters:

   $$s(a, b) = -3 \text{ if } a \neq b \text{ and } s(a, a) = +3 \text{ and } d = 5.$$

   Also make up a short example for testing purposes in order to see if your implementation works correctly.

4. **Gain Extra Points: Enrich your Output with Additional Information** (+3P)

   Write a more verbose output of your Smith-Waterman algorithm to a file. The output should contain (but is not limited to) the following:

   (a) A visual alignment showing the matches, mismatches and gaps between both sequences.

   (b) The used algorithm parameters.

   (c) The number of matches/mismatches/gaps.

   An example of an output can be found on page three of this assignment sheet.

   Submit your source code as well as either your compiled code or your compiled code packed into a JAR file (we will inform you about how to create a JAR file via the Ilias forum). Explain in the PDF document how to run your code from the command line.

Please read the questions carefully. If there are any questions, you may ask them during the tutorial session or in the forum of ILIAS. You will usually get an answer in time, but late e-mails (e.g. the evening of the hand-in) might not be answered in time. Please upload all your solutions to ILIAS. Don't forget to put your names on every sheet **and** in your source code files. Please pack both your source code as well as the theoretical part into one single archive file and give it a name using this scheme: `<name1>_<name2>_<Assignment>_<#>.zip`. The program should run without any modification needed.

Example Output of Task 4

```
For the calculation of the optimal global alignment between the two sequences read from
../material/yersenia_1.fasta and
../material/yersenia_2.fasta
the Needleman-Wunsch Algorithm was implemented using the following parameters:
------------------------------------------------------------------------------------------------------------------------
Parameters:
s(a,b) if a == b (match): 2
s(a,b) if a != b (mismatch): -2
d (gap penalty for linear scoring): 4
------------------------------------------------------------------------------------------------------------------------
The selected optimal global alignment has:
------------------------------------------------------------------------------------------------------------------------
106 matches,
102 mismatches and
22 gaps.
------------------------------------------------------------------------------------------------------------------------
and a score of -80
This is the selected optimal global alignment:
(symbol key: '+' = match, '-' in symbol line = mismatch, '-' in sequence line = gap in sequence, ' ' = gap)
------------------------------------------------------------------------------------------------------------------------
position:   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27
f1      :   -   C   C   A   G   T   T   T   C   A   C   G   G   C   T   G   T   G   A   T   T   -   C   A   T   C   -   A
symbol  :       -   -   +   -   -   -   -   +   -   +   -   +   +   -   +   +   -   -   +   +       +   +   +   +       -
f2      :   A   A   A   A   C   C   C   C   C   C   C   C   G   C   C   G   T   A   T   T   T   C   C   A   T   C   G   G
position:  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56
f1      :   G   T   -   C   G   C   G   C   A   -   -   C   C   A   T   C   A   G   G   A   A   G   T   -   A   G   C   C
symbol  :   +   +       +   -   -   +   -   +           +   -   +   +   -   +   -   +   -   -   +   +       +   +   -   +
f2      :   G   T   G   C   A   T   G   A   A   T   G   C   G   A   T   T   A   A   G   C   G   G   T   A   A   G   G   C
position:  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85
f1      :   C   A   G   A   A   G   C   C   C   A   G   A   C   G   G   -   G   T   C   A   G   C   G   T   G   T   C   C
symbol  :   -   +       +   +   -   -   -   -   -   +   -   -   +   +       -   +   -   +   -   -   -   +   +       +   -
f2      :   T   A   -   A   A   T   A   T   T   G   G   G   G   G   G   A   A   T   T   A   C   A   C   T   G   -   C   G
position:  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114
f1      :   T   T   T   C   G   T   A   C   T   T   C   A   G   A   C   G   A   T   C   A   A   C   A   T   A   C   A   C
symbol  :   -   -   -   +   +   -   -   -   +   +   +   +   +   -   -   +   -   +   -   +   +   -   +   +   -   -   -   -
f2      :   C   G   G   C   G   G   T   T   T   T   C   A   G   T   T   G   C   T   G   A   A   G   A   T   T   A   T   T
position: 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
f1      :   C   G   G   T   G   C   T   C   A   G   A   A   G   C   C   G   G   T   G   G   T   G   T   G   T   A   G   C
symbol  :   -   -   +   -   -   -   +   -   +   -   -   -   -   +   +   -   -   +   -   -   +   -   -   -   +   -   -   +
f2      :   A   T   G   A   T   A   T   A   A   C   C   T   C   C   C   C   C   T   A   A   T   C   G   C   T   C   T   C
position: 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172
f1      :   C   A   A   A   C   T   C   T   T   T   G   G   T   A   -   G   T   A   A   G   A   C   G   C   T   T   G   -
symbol  :   +   -   +   +   -   +   -   +   +   +       -   +   +       -   +   +   -   -   +   +   -   -   +   -   +
f2      :   C   T   A   A   T   T   A   T   T   T   -   T   T   A   T   T   T   A   T   T   A   C   A   A   T   G   G   C
position: 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201
f1      :   G   -   C   -   A   G   C   C   C   C   G   C   G   A   C   A   G   C   C   A   G   C   T   T   C   A   T   C
symbol  :   +       +       +   +       -   -   +   -   -   -   +   -   +   +   -   +   +   -   +   -   +   +   +   -   -
f2      :   G   A   C   G   A   G   -   T   T   C   T   T   T   A   T   A   G   G   C   A   T   C   A   T   C   A   A   T
------------------------------------------------------------------------------------------------------------------------
```