Prof. Dr. Knut Reinert
Christoph Hartmann
Konrad Rudolph

Institut für Informatik
AG Algorithmische Bioinformatik

Freie Universität Berlin

Algorithmen und Datenstrukturen in der Bioinformatik

Viertes Übungsblatt WS 09/10

Abgabe Montag 16.11.2009 12:00

Name: Übungsgruppe: A☐ B☐ C☐

Matrikelnummer: Ich kann Aufgabe _____ nicht vorrechnen.

Exercise 10: Overlap alignment

Calculate the optimal overlap (end-gaps free) alignment for the following sequences.

$$S_1 = \texttt{CARDQ} \qquad S_2 = \texttt{ACEAN}$$

Use -4 as gap costs, and use the substitution matrix given below.

	A	R	N	D	C	Q	E	...
A	4	-1	-2	-2	0	-1	-1	
R	-1	5	0	-2	-3	1	0	
N	-2	0	6	1	-3	0	0	
D	-2	-2	1	6	-3	0	2	
C	0	-3	-3	-3	9	-3	-4	
Q	-1	1	0	0	-3	5	2	
E	-1	0	0	2	-4	2	5	
⋮								

		C	A	R	D	Q
A						
C						
E						
A						
N						

Programming Exercise 1: Local Alignment

The aim of this programming exercise is to implement the local alignment algorithm by Smith and Waterman. Your result should conform to the following specifications. Everything not mentioned in these specifications is left for you to decide. *Document* these decisions and be able to *explain* them.

The formal regulations of the exercise are laid out as follows:

a) Work in groups of at most three people.

b) You may use any programming language you like – within reason. To make things easier for us, we've restricted it to the following languages: C, C++, one of the .NET languages, Java, Haskell, Python, Ruby or Perl. Also, please make sure that your code runs on Windows, Linux *and* OS X (i.e. don't use system-specific functions)!

c) Briefly describe how to *build* the program in a `README` file.

d) Once you've got your solution, compress it in an archive called `P1.zip` (or `P1.tar.gz` or `P1.tar.bz2` – please *no* `rar` archives and no attachments consisting of multiple files).

e) The archive should only contain source files and any necessary resources, no binary files (`*.exe`, `*.out`, `*.o` etc.).

f) Send the solution to krudolph@mi.fu-berlin.de or christophohnegrund@gmail.com.

g) The email should also contain a document describing all decisions that you've made in the fulfilment of this project. This document should be either a `PDF` document or an `ODT` file (e.g. created in OpenOffice.org).

h) All members of the team must be able to explain the code in an oral presentation after the project.

i) You've got three weeks: the due date is 23.11.2009 at 12:00 sharp.

Now for the **specifications**.

a) Implement the algorithm by Smith and Waterman for local pairwise sequence alignment using fixed gap costs.

b) Your program should be launch-able from the command line and accept four parameters:

```
./align sequence1.txt sequence2.txt scores.mat gap-cost
```

Here, `align` is the name of the program, `sequence1.txt` and `sequence2.txt` are names to sequence files, `scores.mat` is the name of a BLOSUM or PAM substitution matrix file and `gap-cost` is a (positive or negative) integer parameter specifying the cost of a gap.

c) You may use either plain text input files for the sequences, or FASTA files, or accept both. In particular, your program *must* be able to read the sample files from the supplementary material, linked below.

d) The substitution matrix will be given in a common format. Again, your program must be able to correctly parse the file from the supplementary material. *Note:* The usual substitution matrix gives an additional "wildcard" row, denoted by "`*`". You may ignore that row and column.

e) For the example input files, your program should produce an output somewhat resembling that given in the supplementary material. An exact match is not necessary (and, given the information, not really feasible).

f) You are not required to output *all* alignments found by the algorithm. It is sufficient to keep track of any one alignment (but it must have the highest score) – tracking back all possible alignments is a *lot* more complicated.

g) You are free to add additional information to the output, or accept additional information as input.

Supplementary material can be found in the wiki.