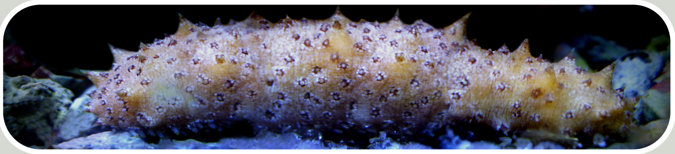
EDAF05 Lab exercise 5

ME: GORILLA OR SEA CUCUMBER?

Thore Husfeldt
27 March 2009 10:50

Description

Implement the dynamic programming sequence alignment algorithm and run it on some realistic protein sequences.



Inputs

There are two input files containing protein sequences. The first, “Toy-FASTAs.in” contains very short proteins from three fictional species that you can use for testing. The second, “HbB_FASTAs.in” contains proper data from a protein sequence database.

A third file, “BLOSUM62.txt” contains a matrix of scores.

Output

You need to compute the pairwise distance between all the given sequences in the “HbB_FASTAs.in” file. (It will correctly determine whether homo sapiens is closer to a gorilla or to a sea cucumber. But I assume you knew the answer to that already.) My output is in “HbB_FASTAs.in”, I hope it’s correct. (Note that there might be several alignments that give the same score.)

Requirements

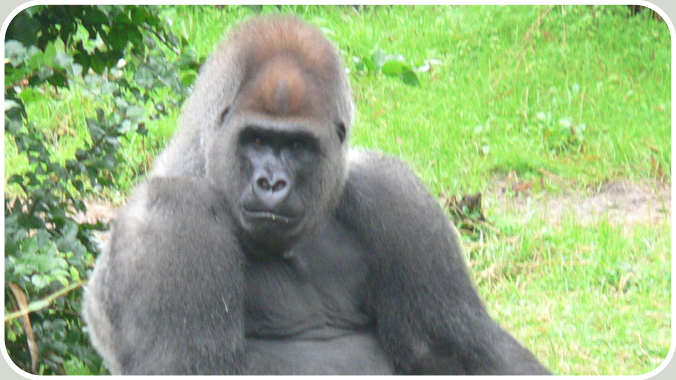
As you can see from the sample output file, you need to compute both the score and the proper alignment. Your code has to use the dynamic programming idea described in the book. Quadratic space is fine.

Your code must read the input “FASTA” file from a file or standard input and must to standard output. The score matrix (in the “BLOSUM” file) you can open as a file, or even hard-code it into your program if you find that easier. *Minimal* requirement is to do this for the Toy-FASTA file. The *good solution* works also for the real-life file. (It’s marginally harder to parse.)

Tips

You need to stare a bit at the BLOSUM matrix in order to understand it — note that the scores in the BLOSUM matrix are large for good alignments, and small for bad ones. That is the opposite convention from what is assumed in the book, so you either need to change signs in the matrix, or use “max” instead of “min”. (I chose the latter.)

You can decide yourself if you want to use a recursive or bottom-up solution. I did the latter, and came to become pretty annoyed with it. Reporting just the score is easy enough in the bottom-up solution. But I think the recursive approach is easier if you actually need to report an optimal alignment. See for yourself.



```
>Sphinx
KQRK
>Bandersnatch
KAK
>Snark
KQRIKAAKABK
```

Toy-FASTA.in

```
Sphinx--Snark: -8
KQR-------K
KQRIKAAKABK
Sphinx--Bandersnatch: 5
KQRK
K-AK
Snark--Bandersnatch: -18
KQRIKAAKABK
-------KA-K
```

Toy-FASTA.out

```
>Human 2144721 HBHU 4HHB
MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAHGKKVLG
AFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVAN
ALAHKYH
>Human-sickle 2392691 2HBS
VHLTPVEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAHGKKVLGA
FSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANA
LAHKYH
...
```

First few lines of HbB_FASTAs.in. The Human protein takes up three lines, “MVHL...KYH”. Ignore things like “2144721 HBHU 4HHB” — I have no idea what that stuff even means.