Exercise 3 – Multiple Alignment on your Computer

Objectives:

- perform a multiple alignment of proteins using three different programs.
- prepare the input files, bringing them into the right format.
- check whether the results are identical.

1) Prepare the input files.

In the last two exercises, we've made two files with protein sequences already. One of them came from BLAST, it will form the basis for an 'easy' alignment because the proteins are very similar. The other is from a domain database, this will be the 'hard' alignment because the proteins span a very large area in sequence space. We'll use those input files to create six different multiple alignments; but first we'll have to properly prepare the input. Both input files should be un-aligned, and for one of them we also still need to add our test protein.

Prior to the course, the files "`input_proteins_1.fa`" and "`input_proteins_2.fa`" were unaligned using regular expressions. The character "-" was replaced with empty character "" and empty lines were removed. This removed all the gaps, and hence destroyed the alignment. This was saved into "`unaligned1.fa`" and "`unaligned2.fa`" respectively; the files are available on OLAT.

- download the files "`unaligned1.fa`" and "`unaligned2.fa`" from OLAT and story them in your home directory (you may have to use "right-click" in the browser to indicate where you want the files stored.
- now, to deal with our test protein: in one file, it simply needs to be renamed … and in the other file it is missing, so we need to add it.
- open the File '`unaligned1.fa`' in an editor of your choice, and change the name of the very first protein, to read 'query_protein'. Then save the file. The first lines of the file should now look something like this:

```
>query_protein
GFFGDRVGRKFIIWFSILGTAPFALWL
PYADADTTAILVILIGFIISSAFASILVYSQELLPKKIGMISGVFYGFAFGMGGLASAL
LGKLIDLTDITFVYKVCSFLPLMGLIAYFLPNLRKVKMKE
>gi|488743029|ref|WP_002666381.1| fosmidomycin resistance protein [Capnocytophaga gingivalis]
METKQRTQYLIIIL
ISLSHCLNDLLQGVLPSIYPALQSKFALSMAQIGLITFCYQIAASILQPIVGAYTDKHPK
PYAQVVGMAFSALGIGLLSWVDSYTLVLCSVVFVGIGSSIFHPEASRISFLASGGKRSFA
[…]
```

- open the second File '`unaligned2.fa`' in an editor of your choice, and add the query protein to the beginning of the file (simply copy-and-paste from the example output above).
- *optional extra task for the geeks among you: can you do the above file manipulations using the unix-editor "**`vi`**" as your editor? If you can't, you're not a geek … ☺*

2) Multiple alignment using ClustalX.

- download the ‘ClustalX’ application, from here:
  http://www.clustal.org/download/current/

  (choose the file `clustalx-2.1-macosx.dmg`, mount the disk image, and copy the application to your home directory. Make sure you choose ‘clustalx’, not ‘clustalw’).
- launch clustalx with a double-click on the application icon.
- click ‘File’ -> ‘Load Sequences’, and choose the File ‘`unaligned1.fa`’.
- go to ‘Alignment’ -> ‘Output Format Options’, and set the ‘Output Order’ to ‘Input’. This way, the order of sequences will not be changed. Also, choose the output format ‘FASTA’.
- now, to align, choose ‘Alignment’ -> ‘Do Complete Alignment’. You can keep the suggested filename for the guide-tree, but please change the output filename to ‘`aligned.clustalx.easy.aln`’. Then click OK (the alignment takes a few seconds).
- now let’s make a nice graphical overview (for example to print it out later). Choose ‘File’ -> ‘Write Alignment As Postscript’. On the dialog that comes up, choose the filename ‘`aligned.clustalx.easy.ps`’, set the ‘block length’ to 300, and choose ‘OK’ (you can ignore the warning). This will create a postscript graphics file … open the Mac “Preview” application (“Vorschau” in german), locate the file you just made and open it. This should give you a nice illustration of the alignment.
- repeat the exact same procedure for the second input file. This time, name the outputfile ‘`aligned.clustalx.hard.fa'`. The alignment is more difficult; it should take around five to ten minutes. Again, create a nice graphical overview (‘`aligned.clustalx.hard.ps`’).

3) Multiple Alignment with Muscle on the Command Line

the program ‘muscle’ is somewhat faster and often produces better alignments than ClustalX in benchmarks, so it is often preferred for larger and/or more difficult alignments.

- to install Muscle, proceed to this website:
  http://www.drive5.com/muscle/downloads.htm
  locate the file ‘`muscle3.8.31_i86darwin64.tar.gz`’, and save it to your home directory.

- then, in the Terminal, decompress and unpack the file:

```
cd                                          [to go to your home directory]
gunzip muscle3.8.31_i86darwin64.tar.gz      [decompresses the file, removes ending ‘gz’]
tar xfvp muscle3.8.31_i86darwin64.tar       [unpacks the file]
ls -lart                                    [list files; did everything work as expected?]
```

- now, we can run muscle on both of our input files:

```
./muscle3.8.31_i86darwin64 -in unaligned1.fa -out aligned.muscle.easy.fa
./muscle3.8.31_i86darwin64 -in unaligned2.fa -out aligned.muscle.hard.fa
```

- ok, let's use ClustalX to format this new alignment in a pretty way again: go back to the ClustalX window, and simply overwrite the old sequences with the new alignment, by choosing 'File' -> 'Load Sequences', and selecting the file we just made ('aligned.muscle.easy.fa'). Then, choose 'File' -> 'Write Alignment As Postscript', and proceed as before. Do the same for the other file ('aligned.muscle.hard.fa')

4) Multiple Alignments using HMMAlign

the program HMM-align produces very good alignments, but it can only be used if the proteins to be aligned have a previously known domain.

- to install HMMAlign, download the following file from OLAT: 'hmmer-3.2.macosx.precompiled.tar.gz' and save it into your home directory.

- uncompress and unpack the file, and create links to the executables in your home directory:

```
cd
gunzip hmmer-3.2.macosx.precompiled.tar.gz
tar xfvp hmmer-3.2.macosx.precompiled.tar
rm hmmer-3.2.macosx.precompiled.tar
ln -s hmmer-3.2.macosx.precompiled/bin/hmmsearch .
ln -s hmmer-3.2.macosx.precompiled/bin/hmmalign .
```

  - next, we need a so-called HMM-file, which describes all the knowledge that has been assembled for a given domain. In our case, follow the steps below:
  - Go to this link http://pfam.xfam.org/family/PF07690
  - Next, on the left, find Curation&Model tab, then go towards the bottom of that section and download the raw HMM file and store it into your home directory. Give it the filename "MFS_1.hmm"

- now, we can use the hmm-file to produce the alignments:

```
./hmmalign --outformat A2M MFS_1.hmm unaligned1.fa > aligned.hmmalign.easy.fa
./hmmalign --outformat A2M MFS_1.hmm unaligned2.fa > aligned.hmmalign.hard.fa
```

- as before, open the alignments you just created in ClustalX, and create nice visual representations for them.

5) compare the alignments

You should now have six different alignments: two each from the three different algorithms (one easy, one hard). Compare them side-by-side … are there differences? If so, which of these alignments looks ‘better’? What criteria might be useful for deciding that?
As expected, the ‘easy’ alignments overall appear to be more similar to each other. With one exception: the hmmalign may put our query protein in the first half of the alignment, whereas the others usually put it in the second half (!). Any idea what this might mean?