University of Pennsylvania

BIOL4536 Fall 2023

HW#5

(Alignment)

Assigned October 4th
Due October 11th, 3:30pm

We're finally getting to some real bioinformatics. In this assignment you will:

(1) Learn how to do Smith-Waterman alignment at the Unix command line.
(2) See alignments change as you vary the alignment parameters.
(3) Practice wrapping commands in a python script.
(4) Practice parsing files with python.
(5) Lern how to run ClustalW multiple sequence alignment.

Log into your UNIX account. Create a subdirectory of your home directory called `HW5` and move down into that directory.

We have installed a program called "`water`" which performs Smith-Waterman alignments. Execute the `water` command with the `--help` option to see its usage page:

```
> water --help
```

The option `-h` also works. These options don't work for every command, but most command authors will implement them. Sometimes all you have to do is run a command with no options in order to see its help page. And of course there's also `man water`.

You do not need to read the man or help pages, I'm going to explain how to run them. But just be aware that those pages are there.

We are going to use `water` to see how sensitive the alignment results are to the scoring scheme.

Make a fastA file called `seq1.fa` with the following sequence. You should be able to copy/paste from the PDF.

```
>NP_005359.1 myoglobin isoform 1 [Homo sapiens]
MGLSDGEWQLVLNVWGKVEADIPGHGQEVLIRLFKGHPETLEKFDKFKHLKSEDEMKASEDLKKHGATVL
TALGGILKKKGHHEAEIKPLAQSHATKHKIPVKYLEFISECIIQVLQSKHPGDFGADAQGAMNKALELFR
KDMASNYKELGFQG
```

Next make a fastA file called `seq2.fa` with the following sequence.

```
>NP_000509.1 hemoglobin subunit beta [Homo sapiens]
MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAHGKKVLG
AFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVAN
ALAHKYH
```

The substitution matrices this program uses are in the /usr/share/EMBOSS/data directory. Not everything in this directory is a substitution matrix, but do an ls on this directory to see what substitution matrices are available. *Note: they call BLOSUM and PAM EBLOSUM and EPAM for some reason.*

(1) Pipe the ls command to the proper grep command in order to count how many EBLOSUM and EPAM matrices there are. Give your commands and the answer.

Next we are going to fix the gap penalties and see how the alignment varies depending on the substitution matrix. Execute the following command:

```
water -gapopen 10 -gapextend .5 -outfile alignment1.txt -datafile EPAM10 seq1.fa seq2.fa
```

Examine the command carefully, this will align the two sequences with the PAM10 matrix.

(2) Report the alignment. What is its length and score?

(3) Now do the same but using the EPAM100 matrix. Output this time to a file called alignment2.txt. Report the alignment, length and score.

(4) Now do the same but using the EPAM250 matrix. Output this time to a file called alignment3.txt. Report the alignment. What is its length and score?

Now let's change the gap penalties.

(5) Align them again using the EPAM250 but this time change the gap open penalty to 1. Report the alignment. What's its score? How many more "gaps" are there than there were with the gap open penalty of 10 with EPAM250?

(6) You'll note from the previous exercise that identities and gaps went way up. Which alignment has a higher score?

(7) Is the alignment from question 6 with higher score better? Could we somehow know which to prefer? Discuss.

(8) Create a directory in HW5 called alignments. Write a python or a shell script (you choose) called align.py to execute the water command on every available EPAM substitution matrix (with the default gap penalties: gap-open = 10, gap-extend = 0.5). Output the results for matrix EPAM*N* to a file called alignment_EPAM*N* in the alignments directory (*N* will take the values $10, 20, 30, ..., 500$ and consequently your script will output 50 files). Your answer to this exercise is your code, include it as an attachment. We may also check your directory for the 50 files, so don't delete them.

(9) Write a python program called parse_alignments.py to parse the output files from the previous question to extract the percent identities from each one. The program should output tab-delimited text to the screen with two columns, the name of the matrix and the percent identity achieved with that matrix. Then it should output at the end some sort of horizontal line (several dashes will do) followed by which matrix achieved the highest percent identity, and what that highest percent identity is. Run the program and redirect the output to "hw5_prob9.txt" Include as attachments the hw5_prob9.txt file and your code.

Next we're going to perform a multiple sequence alignment using ClustalW. To prepare, make a fastA file called `hw5.fa` with both of the sequences from the previous exercises. We are going to add three more for a total of five. You already have the first two, we'll get the other three from GenBank.

The following link is to the hemoglobin gene in *Stegostoma fasciatum* (Zebra shark).

`https://www.ncbi.nlm.nih.gov/protein/XP_048408641.1`
Click the "fastA" link to get the protein sequence in fastA format.

Change the ID in the URL to get the following two protein sequences as well:

`XP_033016679.1`
`XP_020922233.1`.

You should now have five sequences in the file `hw5.fa`. Make sure your file is in fastA format, in particular each sequence should have its ID line that starts with ">".

Download the `hw5.fa` file to your local computer. You will need it there to upload to the ClustalW server.

Run ClustalW using the following online server:
`https://www.ebi.ac.uk/Tools/msa/clustalo/`

Under "OUTPUT FORMAT" select option "ClustalW" (*not* "ClustalW with character counts").

Click on "More options" and set both the following to "no":
`MBED-LIKE CLUSTERING GUIDE-TREE`
`MBED-LIKE CLUSTERING ITERATION`
And set `DISTANCE MATRIX` to "yes".

Point it to your fastA file and hit Submit. It will take a minute to run, wait for it.

(10) Include a screenshot of the alignment and the Guide Tree. Click on "Submission Details" and scroll down to "Command". You'll notice this web page is just running a Unix command for you behind the scenes. It won't run on our machine because the `singularity` command is not installed. But copy this command to your write-up (or give a screen shot).