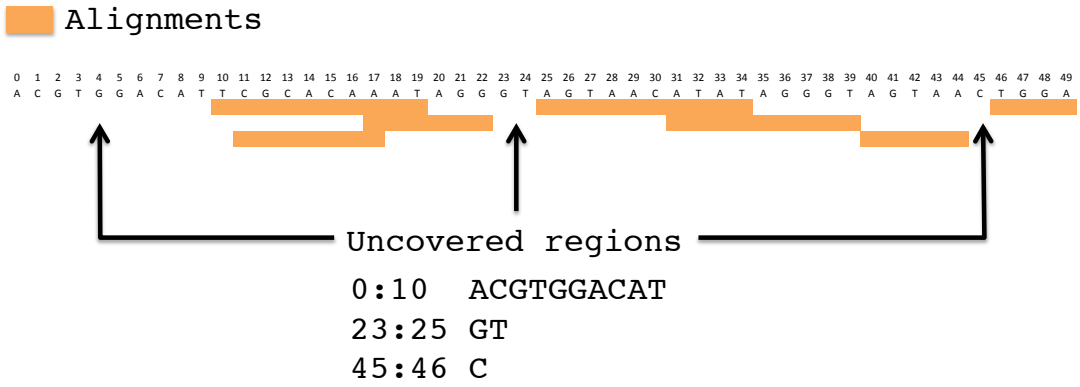
PYTHON EXAM **19 NOV 2015**

Description

Comparing sequences is relevant for many applications, for example comparative genomics and assembly validation. There are many tools available for this task, such as MUMmer and LASTZ.

Today you receive an assembly of yeast chromosome 3 (made using Velvet), and you need to compare it to the sequence of chromosome 3 in the reference genome, using **LASTZ**. You need to report the regions in the reference genome that are not covered by the Velvet assembly. Also, you need to compute and output some statistics on both the reference and Velvet assemblies.



Input

- An assembly of chromosome 3 of the yeast strain CEN.PK 113-7D (velvet_15.fa)
- The reference sequence of chromosome 3 of yeast (chr3.fa).

Login to **`altschul.bioinformatics.nl`** and, on the command line, use the command **`wget`** to download the files from this location:
`http://www.bioinformatics.nl/courses/BIF-30806/docs/chr3.fa`
`http://www.bioinformatics.nl/courses/BIF-30806/docs/velvet_15.fa`

Assignment

Write a python script that performs the following tasks:

1. report the assembly size, N50 size, and N50 index for each assembly
2. compare the two assemblies using **`lastz`**
3. report the regions from the reference genome that are not covered by the Velvet assembly

You could consider building the following functionality:

- Read filenames for both the reference file and the Velvet assembly from the command line (using argv)
- For each assembly (Velvet and reference) calculate and report the total assembly size, N50 size, and N50 index. Definition: the N50 size is the size of sequence X, such that 50% of the assembly is contained in pieces of this size or larger; the N50 index is the number of sequences you need to reach the 50% point.
- In your python script, run the program **`lastz`** to align the Velvet assembly to the reference genome (chr3.fa).
    - Set the output format to **`general`**
    - Call the output file **`outlastz.txt`**
- Parse the resulting output file to extract all alignments. Pay attention to the meaning of the reported coordinates!

- Calculate and report the UNCOVERED regions in the reference genome and the sequences of those regions. Indices in the region should be 0-based and usable to retrieve the uncovered sequence (see Output specification below).
- Also, report the number of uncovered regions and the total number of uncovered bases.

Output
The output of your script should look like this: (Note that the numbers in this output are made up. The numbers in your output will be different!)

```
velvet_15.fa: TOTAL=231177; N50 SIZE=16458; N50 INDEX=14
chr3.fa: TOTAL=411622; N50 SIZE=314410; N50 INDEX=2

Uncovered regions:
0:315 CCCACACACCACACCCACACCAC....
1433:1436 TAG
4011:4444 AATGATTTACAATGGCATACACTTAGAAAAGTGCTATGAC
...
...

Number of uncovered regions: 16
Number of uncovered bases: 12333
```

Development
For development/understanding purposes, you can download this file:
`http://www.bioinformatics.nl/courses/BIF-30806/docs/small_lastz.zip`

Environment
You should work on the **altschul.bioinformatics.nl** server.
The program **`lastz`** is installed there. Try it by typing **`lastz`** or **`lastz`** --help on the command line. You should see information on the usage and options.
More documentation can be found on:
http://www.bx.psu.edu/miller_lab/dist/README.lastz-1.02.00/README.lastz-1.02.00a.html

Additional notes
- Put your full name and student number as a comment in your script and put your username in the file name of your script.
- You may use the slides and the code from your exercises from the first course weeks. You cannot use BioPython or comparable packages, you should write your own parsers. You may not directly copy code from the Internet, but you may use it for inspiration.
- Running **`lastz`** takes some time. To avoid running it over and over again, make sure your code checks whether the output file exists.
- Think about your code organization. Use subroutines.
- Document your code and make sure it has good style and readability.
- Make sure you hand in a working script. If it is unfinished, you can leave the unfinished part in comments.

How to hand it in?
On Thursday 19-11-2015 before 12.15, submit your Python script in BlackBoard under “20151119_exam”

Assessment
We will run your script on the input, check the output, and assess the quality of your code. The grade for this assignment will be 40% of your course grade.