CMPT 830 — Assignment #1

2019–2020 T1

Due: *23:55, Sunday, Oct 20, 2019*

Please email me your solutions as attachments to mcquillan@cs.usask.ca. Putting each answer in its own file is a good idea. Input data files mentioned in the exercises are provided in a folder under the label "Assignment 1". As an aid to completing Exercises 2 and 3, some additional python programming examples have been added to the the course website. Remember that there were many program examples in the lecture notes, including examples of using the functionality of BioPython. If you can't devise a script that solves the stated problem completely, try to achieve as much of the functionality as you can. Start by solving one small part of it, then gradually add in additional functionality one step at a time while making sure it works at each stage.

Exercise 1

Find a published article (paper) that presents an important application of bioinformatics. Then compose a short document that contains the following:

1. Full bibliographic information for the article.
2. A description of the key points of the article. This can be in either point form (i.e. as a list of points) or prose form. Do not repeat the Abstract.
3. An argument of why you think this is an important application of bioinformatics.

Use good English in your document with a minimal amount of area-specific technical language. Your document should be at least half a page of single-space text in length, and no more than one full page. Use reasonable margins and a reasonable font size.

Exercise 2

A FASTA file consists of one line of header, the so-called "definition line" (which starts with a '>' symbol), followed by lines consisting only of sequence data (usually 60 characters per line except perhaps the last one). A multi-FASTA contains information regarding multiple sequences. It starts with the header line for one sequence followed by the sequence across multiple lines, followed by the header for the second sequence, followed by the second sequence, and so on. The file "multiprotein.fasta" is such a multi-FASTA file containing the sequences of four proteins.

Write a python script (program) that does the following: it asks the user for the name of an input FASTA file. The script then reads the input file. As it is reading from the file, it displays to the screen information regarding each sequence encountered. For each, it displays the header line, the first 10 characters followed by the number of amino acids in the sequence. The latter two items are on the same line in the output. Basically, the script is a tool to summarize the contents of a multi-FASTA file.

You can assume that the input file is in correct FASTA format and is not empty.

Make sure that your script is commented, and that you test it. (That's what file "multi-protein.fasta" is for.) Submit your script as your solution to this question (exercise).

You can use BioPython if you wish, but it is not necessary.

Exercise 3

Sequence motifs are short and widespread patterns that have associated specific biological function. In this exercise you will write a python program to find occurrences of a given motif in an input .fasta file.

Write a python (script) program that will prompt the user for the name of a file with a single sequence in it in FASTA format. The program will then search for the first occurrence of the given motif, and if it is found, output the (start) position of the motif. In general the program will count the number of times that the motif occurs in the sequence and, just before terminating, output that count. That means that if a an occurrence is found, the program needs to keep looking for additional occurrences. The motif that the program is to look for is "ATATGC". Two sample input files are provided for you for testing your script. "find_motif.fasta" contains at least one instance of the motif, while "no_motif.fasta" contains no instances.

You can assume that the input file is in correct FASTA format and is not empty. Output the nucleotide position, where those positions start at 1.

Again, make sure that your script is commented, and that you test it. Submit your script as your solution to this question (exercise). If you would like to check your answers, consider using the fuzznuc program of the EMBOSS suite.

You can use BioPython if you wish, but it is not necessary. Also, note that the ".find()" method returns -1 if the pattern it is searching for is not found.