Midterm Fall 2019

N.B.

Make sure your code is well documented with comments. Clearly mark different sections of your code corresponding to the questions.

Make sure your code is formatted nicely.

Remember to document your code. For example:

```
def my_function(arg):
    """
    This is a docstring.

    Surrounded by triple double-quotes. Description of the whole function goes here.
    """
    # This is a comment, provides concise description of following code.
    pass
```

Your best resource for help is the official python documentation.

Attached Files

- celegans.drosophila.xml
- drosophila.celegans.xml
- celegans.fa
- drosophila.fa

Questions

Identifying Homologs (60 points)

- Python script name: `orthologs_netid_1.py`
- Output file name: `orthologs_netid_1.txt`

BLAST is a standard tool to identify putative orthologs based on sequence similarity. Write a Python script that takes as input two different BLAST outputs in xml format, identifies all pairs of putative orthologs from the two different species and outputs it in a text file. For this script, only consider e-values less than 10^{-20}.

The idea is simple, let's say in Species A the proteins are noted A1, A2, A3 and in Species B they are B1, B2, B3. Let's say A1 matches B1, B2, and B3 (i.e. "matches" means there is a HSP between them). Now let's assume that B2 and B3 match A1. Pairs of putative orthologs would be:

A1 B2

A1 B3

Note that A1 and B1 are not homologous pairs because A1 is not reciprocally a hit for B1. Now let's assume that protein B1 from species B matches A3 from species A, then A3 is a homologous pair with B1. The output should be a simple tab separated file with protein from one species in the first column and its ortholog pair from the other species in the second column. You can use the definitions (*e.g.* 'gi|

671162305|ref|NP_001188986.2| CG30271, isoform I [Drosophila melanogaster]') to label your proteins. Make sure that every protein in one column is from the same species i.e. don't mix the columns up in different rows. If a protein in species 1 does not have a pair in species 2, do not print it out.

(20 points) Also include an optional final argument to your script, such that if "FALSE" is passed as the last argument then the definition of a reciprocal match is no longer both a HSP from species A to B and species B to A, but is defined as a HSP from species A to B **or** species B to A.

Output file name: `orthologs_netid_1_FALSE.txt`

(10 points) Upload an alternative version of your solution that uses a different data structure that has a different computational complexity, and comment, using big Oh notation, on the computational complexity of the corresponding data structures.

Python script name: `orthologs_netid_2.py`

The python script should run without error when executed as follows on the command line

```
python orthologs_netid_1.py example_1.xml example_2.xml orthologs_netid.txt [FALSE]
```

HINT: The data files are large so it is best to use a smaller subset of the data to first test if your script/functions are working or not. For example, instead of looping over the full dataset simply loop over only the first, say 1000, elements in each set.

HINT: You can use functions from biopython as required.

HINT: When your python script needs to accept commandline arguments (e.g. FALSE in the take-home exam), you can use the sys.argv function. For example, see this tutorial.
https://www.tutorialspoint.com/python3/python_command_line_arguments.htm

Code Quality Assessment (10 points)

For code readability, defining variables appropriately, calling functions correctly, running the script without any error and sufficient commenting.

Submission

Warning!

Late submissions that are less than 24 hours late will receive a penalty of 20% of the total grade in the midterm. Submissions later than 24 hours after the deadline **will not** be graded, *i.e.* you will receive a grade of zero, unless prior permission has been obtained.

The output files that you have to submit **will** be graded, *unlike* homework assignments. Please ensure the output files contain only the content specified by each question.

Note: the submitted assignment must be the individual student's work and fully commented. Cut and pasted code from internet sites or colleagues is of course not acceptable! For example, while internet sites can be used for research, simply changing variable names etc from cut and pasted code is not acceptable and all code must be written from scratch.

Files to be submitted

1. orthologs_netid_1.py
2. orthologs_netid_1.txt
3. orthologs_netid_1_FALSE.txt
4. orthologs_netid_2.py

If you used JupyterLab then also upload the generated html files.