Week 10

Write a program to measure how long it takes to run a set of 4 jobs using the thread module and then using the multiprocess module, and depending on when you call join.

The program will take 2 command line arguments, one that specifies which module to use and one that specifies when to join.

Each job involves:

Reading in a fasta file and making a list of SeqRecords (look into the SeqIO or SeqParse module of BioPython)

Calculate the number of records in each list and the total sequence length for each file and print these values to the screen after the jobs have finished

Calculate the total run time from start to finish and print results to a file or to the screen.

The four files are: Myotis_aligned_small.fasta; opuntia.fasta; porcelaincrab_aligned.fasta; pyrus.fasta

Run your code from the command line for all 4 sets of starting conditions.

Turn in:

- Your code
- A text file with the timing results for 4 conditions:
  - Thread module, call join() after each job is started
  - Thread module, call join() after all the jobs started
  - Multiprocess module, call join() after each job is started
  - Multiprocess module, call join() after all the jobs started

Remember to run multiprocess within the condition “if __name__ == '__main__': “

To run from the command line you can use something like this. The ar fguments will be in sys.argv:

```
import sys
if __name__ == '__main__':
      “””
            3 arguments:
                  1st is “thread” or “multiprocess” to pick the module
                  2nd is “onebyone” or “afterall”  to join each one after it
                  is started,  or join after all are started.
      “””
```