CSE 470: Lab 16
Lab: April 29, 2013
Due: May 3 11:59 PM

For the last lab we are going to implement a speedup of the search tool from lab 15, using the basic principles of BLAST. In your directory you have a new query file, **query.fa** and a new database file, **database.fa**. The database contains fourteen sequences homologous to the single sequence in **query.fa**, and 1000 sequences that are not. Your **homology_search.py** script takes far too long to handle this, but there are two tricks you can use to speed it up:

1) Instead of computing a significance cutoff for each pair, look at the score distribution achieved by picking a random sequence from the database and preforming a shuffle-align. In otherwords:
    - Pick a random sequence from the database.
    - Shuffle it.
    - Align it to the shuffled query.

    Repeat this *t* times. Find the *(1-p)*t* largest score calculated, and use that as the cutoff for *all* alignments.
2) Use the BLAST filtering trick: align query sequence *q* with database sequence *s* only if they share a common substring of length *w*. If you do this correctly, you should need to do *O(|q|)* work setup (once) plus *O(|s|)* work for each sequence *s*. Which is considerably faster than the *O(|sq|)* time required for each alignment.

Your command-line interface should be identical to that of last week, with two modifications:

1) Add an optional switch *–w* to allow the user to specify the length of the required common substring (with default = 12).
2) In addition to creating the file, your program should print (to the output stream) the cutoff score and the number of elements retained after the common-substring filter. In order to ensure that my auto-grader can parse it, the format should look like this:

    CUTOFF: <int>
    RETAINED: <int>