SCHRÖDINGER®

Innovations in Computational Drug Design

Software Engineer, Backend Problem 1

Our science team has come up with a method to predict binding affinity (BA) for a given protein and drug. They ran off to a Star Trek convention at the last minute, leaving our team to determine how best to design drugs using this information. We're busy reviewing lolcat pictures, so we're putting it all on your shoulders.

As it turns out, the only data they decided to use is the name of the protein and the name of the drug. The super-secret algorithm is:

- Take the length of the protein's name; if it's even, the starting BA is the count of vowels in the drug's name multiplied by 2 (The set of valid vowels is a, e, i, o, u)
- Take the length of the protein's name; if it's odd, the starting BA is the count of consonants in the drug's name multiplied by 2.5
- If any common factors (excluding 1) of the length of the protein's name are found as common factors with the length of the drug's name, the predicted BA is increased by 25% above the starting BA.

  Sample scores: (Protein,Drug,Score): (TPHE39A,Lipitor,12.5), (ZO-1,Actos,4), (ACC2,Zosyn,2)

Write a program that pairs each protein with a drug such that the sum of all binding affinities across all pairs is maximized. Each protein can only be paired with one drug and each drug can only be paired with one protein. Don't enforce uniqueness: if you see a duplicate protein/drug, treat it as another, separate protein/drug that you can match. Your application should be runnable from the command line and take two newline delimited files as input, the first holding the names of the proteins and the second holding the names of the drugs. The output should be the sum of all BA values and a list of (protein, drug) pairs. Don't worry about incorrect input but take care of both upper and lower case names. Any language is allowed except VB. Your solution should be able to solve 150 protein & 150 drug size input files within 60 seconds on a single core 1 ghz processor.

You may use any libraries you find online to help you solve the problem.

Please include at least the following:

- Application Program Files
- Output from using sample input files below
- If you use non-standard libraries, please include installation instructions for those libraries.

Sample input files:
http://schrodinger-quiz.s3.amazonaws.com/drugs.txt
http://schrodinger-quiz.s3.amazonaws.com/proteins.txt

Hint: This is a specific subset of matching problem: http://en.wikipedia.org/wiki/Matching. It has been solved using slower dynamically typed scripting languages.

Note: On average, this problem takes at least 3 hours with outside research.