Python programming for big data and the scientist

PhD course in Uppsala

11th of October 2016

Guidelines

The answers to these exercises are to be written in Python v2.7. Due to the time limits of the teachers, there will be no hand-ins or correction. The answers however will be heavily detailed and published on Wednesday, October 12th. We will also go through a selection of the problems in the lectures on Wednesday!

0.1 Credits

This exercise was originally designed by Aline Dousse and Jia Li while working in Zdobnov's group. It was then adapted by Charles E. Vejnar and further developed by Lucas Sinclair while working at the Bioinformatics Core facility at the EPFL and in Alexander Eiler's group in Uppsala.

1 Opening/reading/writing files

Along with the material for this first exercise you will find a file named `file1.txt`. The script you should write has to open the file and create a new file containing the information formatted in this way:

```
Entry 1 P05784
Entry 2 Q5XKE5
Entry 3 P02533
...
```

2 Filtering by line contents

Along with the material for this second exercise there is a BED formatted file `genome.bed`, containing chromosome coordinates of intervals. Write a python script that ask the user to enter a chromosome name (e.g. chr1) and then opens and reads the BED file. The program must display all the lines from the file corresponding to the chosen chromosome name. A file is written containing the following information for the chosen chromosome:

```
chr20 21283941 21370463 fragment length: 86522
chr20 21284431 fragment length: 86032
chr20 37101485 fragment length: 106017
chr20 37101485 fragment length: 106017
4 entries corresponding to chr20
```

3 Parsing and data structures

A file `distance.dat` is given. Each of its line contains the name of two cities and the distance between them; these quantities are separated by spaces or tabs. For instance one entry could be:

```
New-York Boston 450
```

Write a python program that parses this file and builds a data structure in memory to contain the result. It's important to chose the right data structure for the task at hand. Would you use a list ? A tuple ? A list of lists ? A dictionary ? A set ? A tuple indexed dictionary ? A dictionary of dictionaries ?

Using this data structure, compute the total distance of a given tour, specified in a tuple `tour` containing, in order, the names of the cities to visit. For instance, `tour` could be ("city1", "city2", "city3", "city4")

4 Functions and reusage

4.0.1 A

Write a python function called `is_repetition()` that takes two strings s1 and s2 as arguments and that returns true if s1 is solely composed of the motif s2 which can be repeated as many times as needed. For instance, s1 = "abcabcabcabc" is made of the repetition of the motif s2 = "abc", whereas s1 = "abcab" does not have any repeated motif, except itself.

4.0.2 B

Write a python function called `find_motif(s1)` that returns s2, the shortest motif that builds s1. If s1 has no motif other than itself, the function returns s1.

5 Combining files

A collaborator gives you 3 different files. One with the chromosome names (`chrom.txt`), one with the start and end of fragments (`start_end.txt`) and one with the genes corresponding to it (`gene.txt`). Write a python script that will reconstruct one BED formatted file with these 3 files. It would look like this:

```
chr5 54315961 54317773 CR389614
chr5 54316004 54317773 CR523708
chr5 54340556 54341287 CR385844
...
```