Programming For Biologists 2022

Midterm

You may use your notes, previous homeworks, and a python interpreter (google colab or jupyter notebook/lab) to answer questions in part1 and part2. You may NOT share ideas and answers with other students.

Part1: Short Answer Questions (50pts)

1) List comprehension. (10 pts)

Convert the following loop into a list comprehension such that the entire code would work in just one line.

```
results=[]
for i in range(10):
  if (i % 2 == 0):
    results.append(i**2)
```

2) Python data (10 pts)

A matrix is a 2-dimensional object, very much like a table, that has rows and columns.

a) Give an example of how you can represent a matrix that has 6 columns and 4 rows using lists.

b) Explain how you would retrieve values from the **three different scenarios (the code/solution does not have to be written in just one line)**:
a) the value in row 2, column 4.
b) all values in row 3
c) all values in column 2

3) Python object (30 pts)

Create a class called **Gene** which stores:

- Name - as a string
- Species - as a string
- Sequence - as a string
- Coordinates - as a list of two values, first is start coordinate and the second is the upto but not including the end coordinate.

The __init__ function should expect the user to provide all the variables requested above when creating a new instance of gene.

Overload the operator __len__ such that it returns the difference between the end . For example, If the coordinates for a gene called BRCA1 are 1200, 1500, then the len(BRCA1) should return 300.

Part2: GFF parser (50pts)

One of the most common tasks in Bioinformatics is to parse a file (which hopefully follows some standard) and retrieve the information you want. The Biopython module makes this task quite easy. Unfortunately, not all formats are currently integrated into the Biopython module so you will have to write your own parsers.

Your task is to write a parser that will find and **output a list of genes** that are annotated in a given region of the genome (on both strands). There are cases when Biologists identify QTL (Quantitative Trait Loci) region which they believe is responsible for a certain phenotype that they interested in. They often want to identify all the genes that are present in that region so they can hypothesize which of the genes is responsible for the phenotype. This script would come in handy so that they can retrieve the genes that are present in that QTL region.

The file format that contains gene locus annotations is called a GFF file. There may be modules written to parse them, so you are more than welcome to use them. However, the format is a simple tab delimited format containing 9 columns, so it may easier to simply write your own parser. The nine columns of a GFF file are :

1) Reference sequence – for example a chromosome name
2) Source – who created the annotations
3) Feature – genomic features such as gene, exons, CDS, etc. Notice they are hierarchical. For our task we are only interested in the "gene" features.
4) Start – Start position of the feature on the Reference Sequence
5) End – Stop position of the feature on the Reference Sequence
6) Score – Such as Blast score, if there isn't one you will see a "."
7) Strand – Positive or Negative
8) Phase – Used for CDS to explain which of the three phases is being translated.
9) Annotation – Details of the feature such as name and parent feature. For this example we are only interested in "Name"

Write a Python function named **gffparser** that accepts four options

- path to the GFF file
- Chromosome name
- Start coordinate of the region

- End coordinate of the region

An example GFF file is provided : **TAIR10_GFF3_genes.gff**
The output should be only the names of the genes, for example:
AT1G01010
AT1G01020

For example:
gffparser(“TAIR10_GFF3_genes.gff”,”Chr1”,1,10000)

You may write your own code or use the python modules listed on this site:
https://biopython.org/wiki/GFF_Parsing