Lab 06 - Identify if a genetic mutation exists.

200pts pts total.

Problem

Some diseases are the result of a single letter genetic mutation. Cystic Fibrosis (CF) is one of these. Usually CF is fatal by age 28. There is a treatment for CF that is very effective. The problem with the treatment is that it is fatal if given to people that do not have the CF genetic mutation.

We have some genetic data (not actually Stephie's) that we are going to search to see if it has a genetic mutation.

The pattern we are looking for is:

```
ACTGGCTGGCCTAGTGACTTCCAGCTGCACAGCTATCGACCCAGGGCTGGACAGCCCCTGCCTGGC
```

If the pattern is found then the mutation exists.

The genetic data is 32 meg out of 2.3 gig of a full genetic sequence. To download the data use the link in the Announcement - from your browser and Right-Click on Windows or Control-Click on a mac and get the context menu, then pick "Save Link" in the menu.

The link is:

http://uw-s20-2015.s3.amazonaws.com/GCF_000001405.39_GRCh38.p41214_genomic.s44a-21.fna

Create a new file to write the program. Your program must print the starting location of the pattern in the data file. If the pattern is not found then print "pattern not found".

You can refer to the first program in the file https://github.com/Univ-Wyo-Education/F21-1010/blob/main/class/lect/Lect-12/Lect-12.html.pdf if you have any questions.

The process is:

1. Create a varible and assign the pattern we are looking for to the variable.
2. Import the file readGeneticData. Use the function readGeneticData in the file `readGeneticData.py` to read the data. https://github.com/Univ-Wyo-Education/F21-1010/blob/main/lab/lab-06/readGeneticData.py. The function takes the filepath as the input and it returns all the lines in the files as a list. For eg: ["line 1", line 2"].

3. Concatenate all the strings in the list to one single string. The python function `"".join(list)` will do this for us.
4. Clean up the data - in this case it is mixed case and we have an all upper case pattern. So convert the string to upper case. `stringname.upper()` will do this for us.
5. Search the data to see if the pattern is in the data. `stringname.index(pattern)` can be used to do this. It returns the location of a pattern in a string. If not found it throws an error. This means we will need to put it in a try/except block to catch the error.

Turn in your python program and the output location of the pattern if it is found.

Copyright