Introduction to Computing in Biology
Lab Assignment #5

20 points
Assigned: 10/10/19
Due: 10/30/19 11:55pm

Goals of this lab:

- Practice writing programs to accomplish complex tasks.
- Practice defining your own functions.
- Learn about SNP data.

Part I: Practice with Functions

Write functions to check if a string is valid for each of the following cases. Assume that the function would be used in the following context, where we'd like to check if a string has a particular pattern, and if so, execute a set of statements:

```
if  is_valid_humanlocus(string):
     # <do something here>
```

Your function should return a boolean value.

a. Check to see if a string contains a valid *Homo sapiens* locus, e.g. 6p21.3, 11q1.4, 22p11.2. (**Hint**: a human locus name consists of a number between 1 and 22 or an X or Y, p (short) or q (long) denoting a chromosomal arm, a band number, a period, and a sub-band number). Function name: `is_valid_humanlocus`

(1) Add assertion statements to your script to check that your function returns true for the following examples: '6p21.3', '11q1.4', and '22p11.2'. Your statements should look like the following:
`assert is_valid_humanlocus('6p21.3’), “Incorrect output!”`

(2) Add assertions to check that your function returns false for the following examples: 'chr1:1000', 'nonsense', and '2a11p'
`assert not is_valid_humanlocus('nonsense’), “Incorrect output!”`

(3) Write two additional assertion statements that should check invalid examples and explain why you chose them (e.g. is an element out of range, did you expect a number here?, etc)

b. Write a new function, called `is_locus_onshortarm`, that checks if a human locus is encoded on the short arm (p) of a chromosome. This function should first verify that the provided string represents a valid locus (**Hint**: use the function that you wrote for part a), and then check if it represents a local on the short arm of a chromosome.

Part II: Analyzing SNP Data

For this problem, you will analyze data from sets of single nucleotide polymorphisms (SNPs) that commonly vary in the human population. There are two datasets, extracted from http://23andme.com, one from the fictitious male, Greg Mendel, and the other from his wife, Lilly Mendel.

a. The data in these files are poorly formatted; you will need a set of Python string expressions to properly extract all of the information. Parse out the SNP id, chromosome, position and SNPs for each row. For example the first row, `rs3094315chr1-742429(A,G)` could be parsed to:

id	Chr	Position	SNP1	SNP2
rs3094315	1	742429	A	G

Hint: a dictionary for each person, each one containing 4 parallel lists (e.g. key "Chr" is associated with a list with the chromosome values, key "Position" is associated with a list of the position values) is a reasonable data structure for this type of data.

b. Once you've finished part (a), use your code to define a function called `read_SNP_file`, which you then call from your main script to process both Greg and Lilly's data. The function should accept a string with the file name as an argument and return a data structure with all of the individual's SNP information. Also, add an `assert` statement inside this function to guarantee that the chromosome number is valid (we've only given you the data from the autosomes, so all SNPs should be on chromosomes 1-22).

c. On Chromosome 10, find the largest region of shared SNPs between Lilly and Greg. The answer will be in the form of a pair of genomic coordinates (Position1, Position2). Below is an example of a region of shared SNPs (**in bold**). In this case, report the shared region as (31123, 31625).

Chromosome	Position	Lilly	Greg
10	31,000	AA	AT
10	**31,123**	**TT**	**TT**
10	**31,319**	**AT**	**AT**
10	**31,625**	**CC**	**CC**
10	31,779	GA	CC

(Hint: if you've left your SNPs in genome position order in your lists, you can iterate through the list to find stretches of SNPs that are identical)

d. The `SNP_Definitions.txt` file contains information about the effects of various SNPs. Load the SNP definitions into a data structure so that you can lookup a description given a SNP id and the bases. (HINT: use a dictionary with the SNP id as the key)

e. Use the information you read in from `SNP_Definitions.txt` to identify what the region between 22070000 and 22106000 on chromosome 9 suggest about Greg's chance of a heart attack (Note that the medical term used in this file for heart attack is "myocardial infarction"). What about Lilly's chance of a heart attack? (**Hint**: find the SNPs from this region, and use the information from the 'Description' column to guide your reasoning)

(If you feel bold, the use of a list comprehension would be very cool here)

f. Find a SNP locus that interests you at SNPedia.com. Describe what is known about the locus. Also, check what the SNP status is in both Lilly and Greg. What does the SNP suggest about their possible health?

Submit to Canvas

Code that accomplishes all the tasks in Parts I and II. Please ensure that the entire script runs using the big green "play" button or by selecting "Run" from the "Run" menu.

A report in a text file containing:

a. The largest region of shared SNPs between Greg and Lilly
b. Information about Greg and Lilly's respective risks for a heart attack.
c. Answers to the questions in Part II – f