CSci 3003: Introduction to Computing in Biology
Lab Assignment #4

30 points

Assigned: 2/14/19

Due: 2/28/19 (before midnight)

Goals of this lab:

- Practice manipulating lists.
- Become familiar with using conditional statements and loops.
- Practice reading in data from files, doing analysis, producing output.
- Practice using dictionaries.

Note on optional parts of the assignment:

Several parts of future lab assignments will include optional parts. The text "**OPTIONAL for CSCI 3003, required for CSCI 5465**" indicates that Csci 3003 students are not required to complete that part of the assignment, but will receive 1-2 points of extra credit for completing it. CSCI 5465 students are required to complete these problems. The text "**OPTIONAL for all students**" will indicate that it is optional for all students, and any student completing that part will receive 1-2 extra credit points depending on the scale of the problem.

With next-generation sequencing technology, researchers are now collecting whole-genome sequences for tens of thousands of people. These data are revealing genome-wide views of which parts of our genome vary and provide the basis for linking specific genetic variants to diseases and other traits. The DNA sequence between any two humans is typically more than 99.9% similar, but even small differences in the nucleotide sequence can have tremendous effects on an organism's phenotype and are responsible for many of the traits that make us unique (e.g. hair/eye color). One specific way in which DNA sequence can vary is via a single nucleotide polymorphism (called SNP for short). Just as it sounds, a SNP is a single-base change between two otherwise identical sequences.

In this lab, we will analyze sequence data from the 1000 Genomes Project, which was one of the first efforts to successfully sequence a large population of human individuals and make the data public. Visit the project's website (http://www.1000genomes.org/) or read this paper (An integrated map of genetic variation from 1,092 human genomes. Nature (2012)) to read more about this project. Since most of the information in the genome is identical from individual to individual, catalogs of variation are usually kept as lists of genomic coordinates at which a SNP has been found. We will give you a catalog of known variants, and your task will be to check several of the individuals sequenced in the 1000 Genomes Project for these variations in their genomes.

Background on the input files provided:

Chromosome 2 is one of the largest human chromosomes, spanning over 242 million nucleotides. In order to keep file sizes manageable, we have slimmed down chromosome 2 for 90 individuals and stored these sequences in *slim_chr2_seq.fasta* (the entire fasta file for Chr2 is over 200Mb!). Note that although a normal human genome has two possible SNPs at each genome location (one for each homologous chromosome), we have only provided you with a single SNP at each locus for each individual for simplicity. Additionally, we have included a file (*slim_chr2_SNPS.vcf*) that contains a table of previously characterized SNPs and some information about them (genome location, reference SNP, variant SNP, disease information). We would like to read these large data files into Python data structures that we can easily manipulate and handle.

Part I: Analysis of single-nucleotide variations in a population

FASTA-formatted files:

Large DNA sequences are often stored in a text file in FASTA format, a standardized format for storing sequences in a text file (you saw this already in Lab 2). FASTA files follow a basic format: Lines starting with ">" are sequence headers, which identify all following lines of sequence until the next ">" header line; FASTA files can contain one or many sequences, each identified by its own ">" header line.

Your Task:

Write a script that reads the provided FASTA file (*slim_chr2_seq.fasta*) as well as the provided SNP table (*slim_chr2_SNPS.vcf*) into lists that we can analyze. For each variant found in the SNP table, calculate the frequency at which the variant (SNP) occurs in the set of 90 individuals we have provided to you. Additionally, we want to find the variants with the maximum and minimum variant frequency in this population.

Your script should do the following:

1. Open and read in chromosomal sequences in *slim_chr2_seq.fasta* for each of the 90 individuals. Store identifier and sequence information in separate lists, one element in each list per individual.
2. Open and read in SNP information stored in *slim_chr2_SNPS.vcf* (which can be opened as a text file). Store all information related to chromosome 2 using seven different lists (open the file in a text editor before writing your code to understand its format). Remember to close any files you open!

   Note: Use the "slim_pos" column to access the base corresponding to the SNP in the fasta sequences we provided!
   PLEASE NOTE: While Python performs indexing from 0, biologists sometimes prefer to index from 1. We see this in the *slim_chr2_SNPS.vcf* file. For example, if the value in the "slim_pos" column is 2000, it refers to the 2000th base in the sequence. If you have one of your sequences in a variable named "`seq`," which element do you access with the code: ``seq[2000]``? Hint: it's not the 2000th base!

3. For each provided variant, calculate the frequency of that SNP in the population of 90 individuals. Store the variant frequencies in a separate list.
4. Find and print the ids of the SNPs in the population that occur with the highest and lowest frequency.

5. Open a new output file called *slim_chr2_SNPs_withfrequencies.vcf* and print the following information for each SNP:
    a. Chromosome Number
    b. SNP Name
    c. SNP "Slim" Position
    d. SNP Reference SNP
    e. SNP Alternative SNP
    f. SNP Frequency
    g. Gene Disease Information

    Your file should be tab delimited, with the information for one SNP on each line. Please include column titles, as you did in Lab 3! You are provided with a sample output file called '*slim_chr2_SNPs_withfrequencies_small.vcf'* to show the expected format.

HINT: Separate lists of the same length are often good ways of storing related information! The enumerate() or range() functions easily allow you to access the same indices of separate lists!

OPTIONAL for all students: Write your own function that calculates the mean of a list (input: a list, output: that list's mean). Calculate the mean across all SNP frequencies in the population.

Part II: Characterizing SNPs

1. **OPTIONAL for Csci 3003, but REQUIRED for Csci 5465 students.** We've included information about the gene associated with each of these SNPs, as well as any disease associations with each gene. In this part of the lab, we are interested in genetic variants that might be associated with the disease "Non-Hodgkin lymphoma". Specifically, we would like to collect a list of individuals that have variants that occur in genes associated with this disease (i.e. search for the string in the disease association field). Add to your script from Part I new code that that will open a new file called *lymphoma_variants.txt* and print a list of individuals that have variants in any gene associated with Non-Hodgkin lymphoma. Print the results in the following, tab-delimited format (refer to *lymphoma_variants_small.txt*):

    IndividualID   variantID   GeneDiseaseInfo

2. **For Csci 5465 students:** Cross-reference the file you created that contains SNP frequencies (*slim_chr2_SNPs_withfrequencies.vcf*) and compare frequencies of synonymous versus nonsynonymous SNPs.
3. **For Csci 3003 students:** Select any disease you might be interested in that is listed in the *slim_chr2_SNPS.vcf file*, and find a SNP that is in a gene associated with that disease. What kind of SNP is it (i.e. how does it affect the protein)? What's known about the function of the gene it is in and its association with disease? What is the SNP frequency? NOTE: You can answer all of these questions without completing step (1).

Part III: Discovering new variants

While it is important to characterize and catalog known SNPs, one of the important contributions of large scale sequencing projects like the 1000 Genomes Project is the discovery of new variants. Often SNPs can be specific to certain populations and are only found when new individuals are sequenced. The SNPs we have seen so far were previously known before the sequencing of these 90 individuals and were assigned reference SNP ids, which can be found in dbSNP at http://www.ncbi.nlm.nih.gov/SNP/. In this part of the lab, we will scan the sequenced chromosomes of the population and compare them against the reference sequence in order to discover new SNPs.

Your Task:
Write a script that scans over each individual's chromosome sequence. Compare each nucleotide against the reference sequence and keep track of newly discovered variants and the alternative base call in an appropriate data structure. Compose a SNP table listing the newly discovered SNPs.

HINT: Dictionaries will be a very useful data structure to use for this part.

Your script should do the following:

1. For each chromosome in the population, scan the nucleotides, comparing each base against the reference sequence.
2. If you find a SNP (index where the individual's sequence differs from the reference), store its position in the sequence as well as the variant base.
3. Iterate over your newly discovered SNPs and print out the following information to a file named *novel_variants.txt* (consult to *novel_variants_small.txt* for the formatting):
    a. The SNP's position in the sequence
    b. The reference SNP
    c. The alternative SNP
    d. The frequency of the alternative SNP in this population

Submit to Moodle:

When you're finished with the lab, make a report of any questions you answered plus any requested output, and gather the scripts that you modified. Place them all in a folder, and create a single archive file called lab4.zip. Remember, you'll need to use a command like this (if you are working in a UNIX terminal):

```
zip –r lab4.zip <yourFolder>
```

Submit your homework file using the online Moodle submit page.