Assignment 1: Counting nucleotides and determining if a sequence is i.i.d.

Due date: Friday, 1/27 10am

The goal of this homework assignment is to get comfortable with handling genomic sequences using simple scripts to test the hypothesis that genomic sequences are “i.i.d.” (independent and identically distributed) an assumption that most statistical models make. We will obtain sequences from human chromosome 20 to calculate the frequencies of nucleotides and dinucleotides and compare these frequencies to those generated with a first-order random background model.

Part 1

Obtain the human chr20 sequence.

Preparatory Steps
ssh to the class server using your username and password
You should see `username@genomic:~$` (or something similar) where `username` is replaced with your username. This is the command prompt.

Setup your assignment 1 directories
Make your assignment1 directory
Note: do not type the ‘`$`’, it’s only included to tell you this is a command prompt.

```
$ mkdir assignment1
```

Change directories to your assignment1 directory

```
$ cd assignment1
```

Make your work and submission directories

```
$ mkdir work
$ mkdir submission
```

Change directories to your work directory

```
$ cd work
```

Download human chromosome 20 from NCBI
Use `wget <link to file>` to download human chromosome 20.

```
$ wget
https://ftp.ncbi.nih.gov/genomes/H_sapiens/CHR_20/hs_ref_GRCh38.p7_chr20.fa.gz
```

(*Alternatively, use FTP directly and connect to NCBI. This connection is currently blocked in our server.*

```
$ ftp ftp.ncbi.nih.gov
```

Follow the instructions to log into ftp anonymously
type “`anonymous`” as your username and “`email`” as your password.

Change directory and get the file. The "`ftp>`" is the FTP prompt (which will show up on the command prompt/terminal). Do not type "`ftp>`".

```
ftp> cd genomes/H_sapiens/CHR_20
ftp> get hs_ref_GRCh38.p7_chr20.fa.gz
```

Terminate the FTP session

```
ftp> bye
```

)

Unzip the file

```
$ gunzip hs_ref_GRCh38.p7_chr20.fa.gz
```

There should be a file called `hs_ref_GRCh38.p7_chr20.fa` now in your directory.

At the end of this process, you will have a sequence file in fasta format.
You can look at the first 10 lines of this file by typing

```
$ head hs_ref_GRCh38.p7_chr20.fa
```

We have created a template `README.txt` file for you to edit and turn in. Please replace '{}' and everything in between with your answers, but keep everything else the same. Copy the template to your working directory

```
$ cp /home/assignments/assignment1/README.txt .
```

← note the period.

Part 2

The script `nuc_count.py` counts the number of As, Cs, Gs, and Ts, and Ns in a fasta sequence file and prints the results. This script only counts one strand and is case-insensitive, e.g., both a and A bases are used to count the number of As in the sequence.

The usage of `nuc_count.py` is:

```
$ python3 nuc_count.py <fasta>
```

Before using this script, copy it to your work directory

```
$ cp /home/assignments/assignment1/nuc_count.py .
```

Question 1

Run `nuc_count.py` on `hs_ref_GRCh38.p7_chr20.fa`. How many times do each of the 4 nucleotides occur in chr20?

Part 3

Modify `nuc_count.py`, so that it also outputs frequencies of A, C, G, T. Ignore N (and any other nonACGT nucleotides) from this point forward.

Question 2

Run your modified `nuc_count.py` on `hs_ref_GRCh38.p7_chr20.fa`. What are the frequencies of the 4 nucleotides on chr20?

Part 4

In this section, you will finish writing a script, `make_seq.py`, that generates a random sequence given a sequence length and nucleotide frequencies.
The usage of `make_seq.py` is:

```
$ python3 make_seq.py <sequence_length> <A_freq> <C_freq> <G_freq> <T_freq>
```

The script prints a random sequence of length <sequence length> to the terminal (stdout). The random sequence should have the same nucleotide frequencies as the input nucleotide frequencies.

First, copy `make_seq.py` to your work directory.

```
$ cp /home/assignments/assignment1/make_seq.py .
```

Finish the script by writing code where it says "TODO". Refer to the assignment 1 presentation for tips on random number generators. Test to see if your code is working by generating files with different nucleotide frequency inputs and then checking the nucleotide frequencies within those files using your modified nuc_count.py script.

Using `make_seq.py`, generate a random sequence with length 1,000,000 using the nucleotide frequencies calculated in part 3 and save it to *random_seq_1M.txt.* Please keep at least two decimal places from the original calculated frequencies when generating your new random sequence file.

To save the sequence to a file, redirect the standard output using ">". Here's an example of this:

```
$ python3 make_seq.py 1000 0.25 0.25 0.25 0.25 > random_seq_1k.txt
```

Part 5

Modify `nuc_count.py` to also output frequencies of all dinucleotides using an overlapping window method. By an overlapping method we mean if the sequence is 'ACGC', then there are 3 di-nucleotides, 'AC', 'CG', and 'GC' instead of 'AC' and 'GC'.

When you run `nuc_count.py` the output should look like this:

```
$ python3 nuc_count.py <fasta>
```

```
Dinucleotide Frequencies
AA:0.063
AC:0.133
AG:0.025
AT:0.065
CA:0.140
CC:0.145
CG:0.028
CT:0.088
GA:0.023
GC:0.043
GG:0.010
GT:0.018
TA:0.060
```

```
TC:0.080
TG:0.030
TT:0.049
```

Question 3
Run the modified `nuc_count.py` for both human chr20 and your generated 'random_seq_1M.txt' from part 4. Compare the two lists of frequencies. What are the differences? Can you provide a biological explanation for these differences?

What to turn in

- **Two** modified scripts `nuc_count.py` and `make_seq.py`.
- A completed `README.txt`.
- The sequence file `random_seq_1M.txt`

These four files should be in your `assignment1/submission` folder.

Note: to copy your work files to your submission folder, type

```
   $ cp <file_name> ~/assignment1/submission/
```

where `<file_name>` is the name of the file you want to copy.