

Project #1: DNA

Objectives

- Read in command-line arguments
- Read in data from files
- Write and call user-defined functions
- Write and call a main function

DNA

DNA is the hereditary material in human and other species. Almost every cell in a person's body has the same DNA. All information in a DNA is stored as a code in four chemical bases: adenine (A), guanine (G), cytosine (C) and thymine(T).

Most of our DNA is the same, but there are some spots where they differ. Scientists are trying to figure out what those differences mean, and correlate them to variants such as diseases, physical characteristics, reactions to particular foods, and so on. This is a huge area of research where biologists are working with computer scientists (and fields like computational biology and bioinformatics now exist as interdisciplinary degrees).

People are now able to submit their spit and receive reports about some of their DNA sequencing. 3 such DNA raw data files are provided on Moodle for you to use (nearly 1 million lines of data each!) There are also websites that allow you to view this raw data mapped to research findings, such as the link below. Be aware that these correlations are based on studies that were published, but are not necessarily proven.

http://files.snpedia.com/reports/GNZ/promethease_DGM001_Daniel_MacArthur_pooled.html

For your project, you will work with the raw data files and write a program to simulate parts of the report from the link above.

Specifications

There are several data files you can use to test your program. A couple are on the Moodle, and you can find more DNA data files that people made available at:

<http://www.snpedia.com/index.php/23andMe>

The data files look something like the following, and have nearly one million lines:

```
# This data file generated by 23andMe at: Wed Jan 26 05:37:08 2011
#
# Below is a text version of your data. Fields are TAB-separated
# Each line corresponds to a single SNP. For each SNP, we provide its identifier
# (an rsid or an internal id), its location on the reference human genome, and the
# genotype call oriented with respect to the plus strand on the human reference
# sequence. We are using reference human assembly build 36. Note that it is
# possible
# that data downloaded at different times may be different due to ongoing improvements
# in our ability to call genotypes. More information about these changes can be found
# at:
```



```
# https://www.23andme.com/you/download/revisions/
#
# More information on reference human assembly build 36:
# http://www.ncbi.nlm.nih.gov/projects/mapview/map_search.cgi?taxid=9606&build=36
#
# rsid      chromosome    position    genotype
rs4477212   1          72017      AA
rs3094315   1          742429     AA
rs3131972   1          742584     GG
rs12124819  1          766409     AA
rs11240777  1          788822     GG
rs6681049   1          789870     CC
```

An example of output: (the last line about diet is extra credit)

```
python3 DNAresults.py dna1.txt
```

```
Parsing dna1.txt
Normal risk for Type-2 Diabetes
Probably light-skinned, European ancestry
[88%] Genetic Disprivilege: Only High Intensity Exercise Will Help You Lose Weight
[39%] Genetic Disprivilege: You Will Lose 2.5x As Much Weight on a Low Fat Diet
```

Another example:

```
python3 DNAresults.py dna2.txt
```

```
Parsing dna2.txt
1.3x Increased risk for Type-2 Diabetes
Probably light-skinned, European ancestry
[88%] Genetic Disprivilege: Only High Intensity Exercise Will Help You Lose Weight
[39%] Genetic Disprivilege: You Will Lose 2.5x As Much Weight on a Low Fat Diet
```

Another example:

```
python3 DNAresults.py dna3.txt
```

```
Parsing dna3.txt
No DNA info on Type-2 Diabetes
Probably darker-skinned, Asian or African ancestry
[12%] Genetic Privilege: Any Exercise Works For You
[39%] Genetic Disprivilege: You Will Lose 2.5x As Much Weight on a Low Fat Diet
```

Requirements

1. The name of the file must be called **DNAresults.py**
2. The 1st line of code must be: **#!/usr/bin/python3** (this is called the “shebang”) - it allows the auto-grader to know it should use python3 to grade your assignment.
3. Comments at the top of your program
 - a. Your name
 - b. Date
 - c. Project #1
 - d. List of Collaborators and where you got help
 - e. Brief description of the assignment (one or two lines max)

- In your main function, read in the name of the data file from command-line arguments.
The first line of output should print out the data file name. For example, if the data file name is 'dna1.txt', then you should output:

```
Parsing dna1.txt
```

- Create a function named 'parseFile' that takes the data filename as a parameter, to do the following:
Create a dictionary variable to keep track of the rsid and genotypes where the rsid is the 'key' and the genotype is the 'value'. **Note: You must store all rsid keys and their genotype values into your dictionary.**

Open the file.

For each line in the file,


If the line starts with a '#' then that line is a comment so skip over it

All other lines in the file contain the following 4 items:

rsid, chromosome, position, genotype

For example, the following line:

```
rs4477212      1      72017      AA
```



Add the rsid/genotype key/value pair to your dictionary

Hint: Make sure you get rid of the newline character on each line!

- Create separate function for each of the following checks. Each function takes the genotype as a parameter, and prints out the results shown below based on the genotype.

Topic	RSID	Geno-type	What you should output to the screen
Type-2 Diabetes	rs7754840	CG	1.3x Increased risk for Type-2 Diabetes
		CC	1.3x Increased risk for Type-2 Diabetes
		GG	Normal risk for Type-2 Diabetes
		**	No DNA info on Type-2 Diabetes
Skin Type	rs1426654	AA	Probably light-skinned, European ancestry
		AG	Probably mixed African/European ancestry
		GG	Probably darker-skinned, Asian or African ancestry
		**	No DNA info on Skin Type

**** If the genotype does not match any displayed or if the RSID does not exist in the data file.**

You can see where we got this information from and more details about the rsid/genotype pairs by going to this website (or replace Rs7754840 with whichever rsid you want to look up): <http://www.snpedia.com/index.php/Rs7754840>

7. Implement the **exercise** portion of the flow chart diagram shown at <http://rockstarresearch.com/these-5-genes-predict-what-kind-of-diet-and-exercise-is-best-for-your-body-2/>
This one can either be done in a function or all of it inside the main function.
It cannot be outside of any function (like previous assignments), it has to at least be inside the main function.

For the diagram, assume all rsid values from the diagram exist in all data files that you will test. In other words, you don't have to worry about checking if it is in the dictionary first because it will be there.

For extra credit you can also do the diet portion of the diagram as well.

Hint: to make your life easier with the if statements, first get each of the rsid values that you need from the dictionary. You will need:

```
rs4994 = data.get("rs4994")
rs1042713 = data.get("rs1042713")
rs1801282 = data.get("rs1801282")
rs1042714 = data.get("rs1042714")
rs1799883 = data.get("rs1799883")
```

8. Remember to close your data file when you are finished with it!
9. Submit your assignment to [web-COG](#) as a zip file named **Firstname_Lastname_Project1.zip**. Select the assignment "Project #1 Submit".
Also submit your .zip file to Moodle.
10. You must submit to COG for the 40 points on this project.
11. You must sign up for an interview to get the 60 points for this project. If you do not do an interview for the project, then you will get a zero on the project (regardless of what you got from the COG grade).
12. You must sign up by Wed, Oct 1 by noon for an interview on Moodle. If you miss your appointment (and haven't used up your 3 penalized misses) then you need to email the person you missed an appointment with to set up another one.