Assignment 4
Gene Expression
Overview

- Due date: 2/16/18
- Start early, this assignment may take some time

Look for TODO comments if you forget what things we want you to do or comment
Premise

- Have 6 subjects who are at a high risk for developing type 2 diabetes
  - You put them on a year-long exercise regimen
- You have muscle biopsies from each subject before beginning the exercise regime and after one year
- You want to characterize their skeletal muscle transcriptomes and hope to better understand the genetic underpinnings of insulin signaling and insulin sensitivity after exercise in human skeletal muscle.
Important Files

- **raw_counts.txt**
  - RNA-Seq count data from the muscle biopsies before and after the exercise regime for the 6 individuals

- **gene_expression.py**
  - Python script that you will need to complete

- **README.txt**
  - README to answer all your questions
Part 0 - Setting up for success

- Fill in the code for the `translate_dictionary` function
  - Translate a dictionary (as input) from a dictionary of counts by sample for genes to a dictionary of counts of genes for each sample
  - `{gene:[list of counts by sample]} to {sample:[list of counts by gene]}`.
- Comment the `upper_quartile_norm` function to explain what each line does
- Create a function `fishers_linear_discriminant`
Part 1- Data filtering

1. Remove genes that have zero counts in all samples
   a. Create a dictionary with genes that pass your filter
   b. TIP: create a new dictionary for each filtering step DO NOT ALTER THE ORIGINAL DATA FILE

2. Calculate the counts per million (cpm) of each gene left in your data
   a. Use the `counts_per_million` function to your data that passed the filter

3. Create a dictionary of genes that pass your second filter
   a. Remove genes that have 6 or more samples have cpm < 1
   b. NOTE: this dictionary should be a dictionary of raw counts, not cpm
Part 2 - Data visualization

1. Plot the library sizes (total counts) for each sample using the genes that passed your filters in Part 1
2. Use the matplotlib library in python to plot things
3. Make sure your plot has labeled axes!
4. Save this plot as library_size.png
Part 3 - Data normalization

1. Use the `upper_quartile_norm` function to normalize the data you have from Part 1

2. Plot the normalized library sizes (total counts)
   a. TIP: Look at your code from Part 2 to make this easier!
   b. TIP: Make a function to plot your data
Part 4 - Data exploration

1. Use Fisher’s Linear Discriminant (FLD) to identify genes that are differently expressed between the Before and After groups
   a. Use the FLD function you wrote in Part 0
   b. Output the genes with the ten highest FLD values (include gene name and FLD values)

NOTE: when calculating FLD for each gene remember to split the expression values into group 1 (Before) and group 2 (After)
Tips and Tricks: FUNCTIONS ARE YOUR FRIENDS

● Functions allow you to re-run the same code over and over again
  ○ Computer scientists are lazy and you should be too with your code!
  ○ Why copy paste if you can just call a function?
● You can provide functions with arguments to make them versatile
  ○ If only one small thing (like the data) is changing, use function arguments to run the same code on different objects
● When debugging functions help in determining which part of your code is breaking

As a general rule: Functions are really good at one thing, break up your steps into functions

TIP: You can write more functions than we specifically ask for if it is helpful to you
What to turn in

● Edited script: gene_expression.py
● Output files
  ○ library_size.png
  ○ library_size_normalized.png
  ○ mean_expression.png
● Your README.txt with the answers to the questions and the commands you used to answer the questions
● Extra credit only: dendrogram.png

REMEMBER TO COMMENT YOUR CODE
I turn
into Code.