Assignment 4: Gene Expression

Due date: ~~Wednesday, 2/17 10am~~ ~~Friday, 2/19 10am.~~ Note: no late assignments will be accepted. Friday, 2/26 10am

There is evidence that skeletal muscle insulin resistance is an initial defect in type 2 diabetes. Short term exercise has been shown to increase skeletal muscle insulin sensitivity. You are interested in whether or not a long-term exercise program will have a pronounced effect on skeletal muscle physiology leading to positive changes in systemic insulin and glucose levels in individuals at risk of developing type 2 diabetes.

In this experiment you recruited 20 subjects whose insulin and glucose levels put them at high risk of developing type 2 diabetes. You put them on a year-long exercise regimen. All of their glucose and insulin levels were in the normal range after one year of exercise. You obtained muscle biopsies from each subject at two time points: 1) before beginning the exercise regime; and 2) 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.

The data file (`raw_counts.txt`) contains RNA-seq count data generated from human skeletal muscle biopsy samples taken from the 20 individuals before and after the exercise regimen. The two groups you are comparing are “Before” and “After”, and every “Before” sample has a matched “After” sample from the same individual.

Somebody in your lab has done this kind of work before, but they did not document their work very well and cannot find the final version of the script they used to analyze the data. Curiously, they are only able to locate a partially completed version of the script you will need to complete your tasks, meaning you will have to fill in some important details to make the script work. The script includes partially completed functions for filtering and normalizing your data.

Copy the incomplete script `gene_expression.py` and `README.txt` from the `/home/assignments/assignment4/` folder to your `assignment4/work` folder. Instead of copying the data for this assignment to your folder, use the absolute path to the data when you specify the filename in your command line arguments (`/home/assignments/assignment4/raw_counts.txt`).

There are prompts in `gene_expression.py` for where you need to do work. (Look for TODO: comments in the script that will tell you what you need to do.)

Inside `gene_expression.py`, your data will be stored in a series of dictionaries as you filter and normalize. The keys of these dictionaries will be gene names. The value associated with each key (gene) will be a list containing the expression levels of that gene found in each sample in your study.

Open up `gene_expression.py` and read through to get a sense of what all you will need to do.

- In the doc string, write a general description of what the script does.
- Make the script exit if the wrong number of input parameters is used.

Part 0 — Functions

Read the `counts_per_million` and `library_sizes` functions to understand what they do. Next,

- Fill in the code for the `translate_dictionary` function. This function will translate your data from having one key per gene to one key per sample. In the translated dictionary, the value associated with each key (sample) will be that sample's gene expression levels at each gene.
- Add comments to the code in the `upper_quartile_norm` function to explain what each line does.
- Write a function to calculate Fisher's Linear Discriminant (and add comments, too!) for all genes in your count dictionary. Call your function `fishers_linear_discriminant`. Remember that functions should be able to work using different data sets, so make sure that your function would work with data from a different experiment with different numbers of before and after samples. For a definition of FLD, see Part 4 below.

Your labmate was at least able to write some code to read the data as input and put it in the correct format for the rest of your script. However, you will have to write the remaining steps using the functions you completed in Part 0. Follow the steps shown below and in the script to make sure you complete all parts of the assignment.

Part 1 — Data filtering

To filter the data, you will need to to remove genes that provide little to no information about the amount of gene expression.

Remove genes that have zero counts in all samples. In other words, create a new dictionary with genes that pass your filter. As part of this filtering step and each filter step to follow, you should save the newly filtered data in a new dictionary. (Note: you should not alter the original data file.)

Question 1
How many genes are left after removing genes with zero expression in all samples?

Next, you want to calculate the counts per million (cpm) for each gene left in your data. The `counts_per_million` function returns a dictionary similar to the raw counts dictionary but with cpm values instead. The formula for calculating counts per million is:

$$cpm_{ij} = \frac{x_{ij}}{N_i} \times 10^6$$

Where:
j = gene
i = sample
x_{ij} = raw count of the *j*th gene in the *i*th sample
N_i = total counts of all genes in the *i*th sample

Apply the `counts_per_million` function to your data that passed the first filter. Now create a new dictionary of raw counts by including genes that pass your second filter. The second filter should not let a gene through if 20 or more samples have cpm < 1. (Note: the end product of this filtering step should be a dictionary with raw counts, not cpm values.)

Question 2
How many genes are left after removing genes where 20 or more samples have cpm < 1?

Part 2 — Data visualization

Plot the library sizes (total counts) for each sample using the raw counts for the genes that are left after the filtering steps. For full credit, make sure the x-axis, y-axis, and the plot itself have informative titles. The y-axis should have library size reported as millions of counts. Save your plot as `library_size.png`.

Question 3
What is the range of library sizes (min, max)?

Part 3 — Data normalization

To normalize your data, your labmate suggests an upper quartile normalization of the raw counts left after your filtering steps. Use the `upper_quartile_norm` function to normalize the data you have left after the filtering steps. Write additional code to redo the total counts plot from Part 2 using the normalized count data. Save your new plot as `library_size_normalized.png`. The formula for calculating the upper quartile normalization is:

$$UQnorm_{ij} = \frac{x_{ij}}{D_i} \times \overline{D}$$

Where:
j = gene
i = sample
x_{ij} = raw count of the *j*th gene in the *i*th sample
D_i = 75th percentile of raw count expression values in the *i*th sample

$\overline{D}$ = mean value of *D* across all samples

Question 4
What is the range of library sizes (min, max) after normalization?

Question 5
Compare the two library size bar charts you made. How did the distribution of library sizes change after normalization? Briefly discuss why it is important to normalize your RNA-seq data.

Part 4 — Data exploration

Now that you have filtered and normalized your data, you are ready to compare the Before and After samples. (Go back and remind yourself what this experiment was about in the first place!)

You heard a talk once about using Fisher's Linear Discriminant (FLD) to compare two groups and think this would be a great way to identify genes that are differently expressed between the Before and After groups. Calculate FLD for each gene and output the genes with the ten highest FLD values (output should include the gene names and FLD values). Make sure your comments tell your future self exactly what each step does. For a gene *j*, split the expression values into group 1 (Before) and group 2 (After). The formula for calculating FLD for gene *j* is:

$$FLD(j) = \frac{(m_1 - m_2)^2}{(s_1)^2 + (s_2)^2}$$

Where:
m_1 = the mean value of group 1
m_2 = the mean value of group 2
s_1 = the standard deviation of group 1
s_2 = the standard deviation of group 2

Question 6
What are the top ten differentially expressed genes according to your FLD analysis? (Copy and paste your function's output.) Do these genes make sense given the tissue and groups in the experiment?

Question 7
Does your result point toward one gene with large effect or many genes with small effects? Does RNA-seq expression data always give researchers a clear answer?

Question 8
How does the study design of this experiment relate to the assumptions made when studying gene expression data?

Question 9
If you were going to spend time and money following up on one of these top ten genes, what

would be your candidate and why? (There could be many correct answers.)

Generate a bar graph of the upper quartile normalized mean expression level in each group for the gene you want to follow up on. Include standard error (SEM) bars to give a sense of how variable your data is. Of course, label your x and y axes and title your graph. Save your plot as `mean_expression.png`.

The standard error of the mean (SEM) of a group is calculated as the standard deviation of the group divided by the square root of the number of samples in the group.

Extra Credit — Going the distance

Calculate the Euclidean distance matrix for your samples using the normalized data and/or plot a dendrogram showing the relationship between samples, labeling each leaf with the appropriate sample name. Save your dendrogram as `dendrogram.png`.

Question EC.1
What do you expect to see?

Question EC.2
What did you actually see? If you did not find what you expected, what sorts of variation could account for this?

What to turn in (submission folder)

- Edited script: `gene_expression.py`
- Output files
    a. `library_size.png`
    b. `library_size_normalized.png`
    c. `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`

Changes

V3 (2/17 1pm): Due date extended
V4 (2/18 12pm): Due date extended, again :)