CSE185-LAB4-README

July 17, 2023

1 CSE185 Lab 4 Report - Code Documentation (10 pts)

- Document any commands used or additional analysis steps below!
- You should include enough detail that the instructors (or your future self) could come back to this several months from now and know exactly what you did and why you did it.
- We will not run this notebook, but will look back to see what you did especially if you end up with different answers.

For grading purposes only - Do not copy or edit this cell!

Question 1: Command Documentation

I ran the following commands which open the .genes.results files and then check for the TPM field (fifth field) which are greater than 0 then output the number of lines that follow that criteria. I made sure I was in the ~/public/lab4 directory and ran the following commands:

[1]:
```
cat Chow_Rep1.genes.results | awk '$5>0' | wc -l
cat Chow_Rep2.genes.results | awk '$5>0' | wc -l
cat Chow_Rep3.genes.results | awk '$5>0' | wc -l
cat HFD_Rep1.genes.results | awk '$5>0' | wc -l
cat HFD_Rep2.genes.results | awk '$5>0' | wc -l
cat HFD_Rep3.genes.results | awk '$5>0' | wc -l
```

```
  File "/tmp/ipykernel_369/583362978.py", line 1
    cat Chow_Rep1.genes.results | awk '$5>0' | wc -l
        ^
SyntaxError: invalid syntax
```

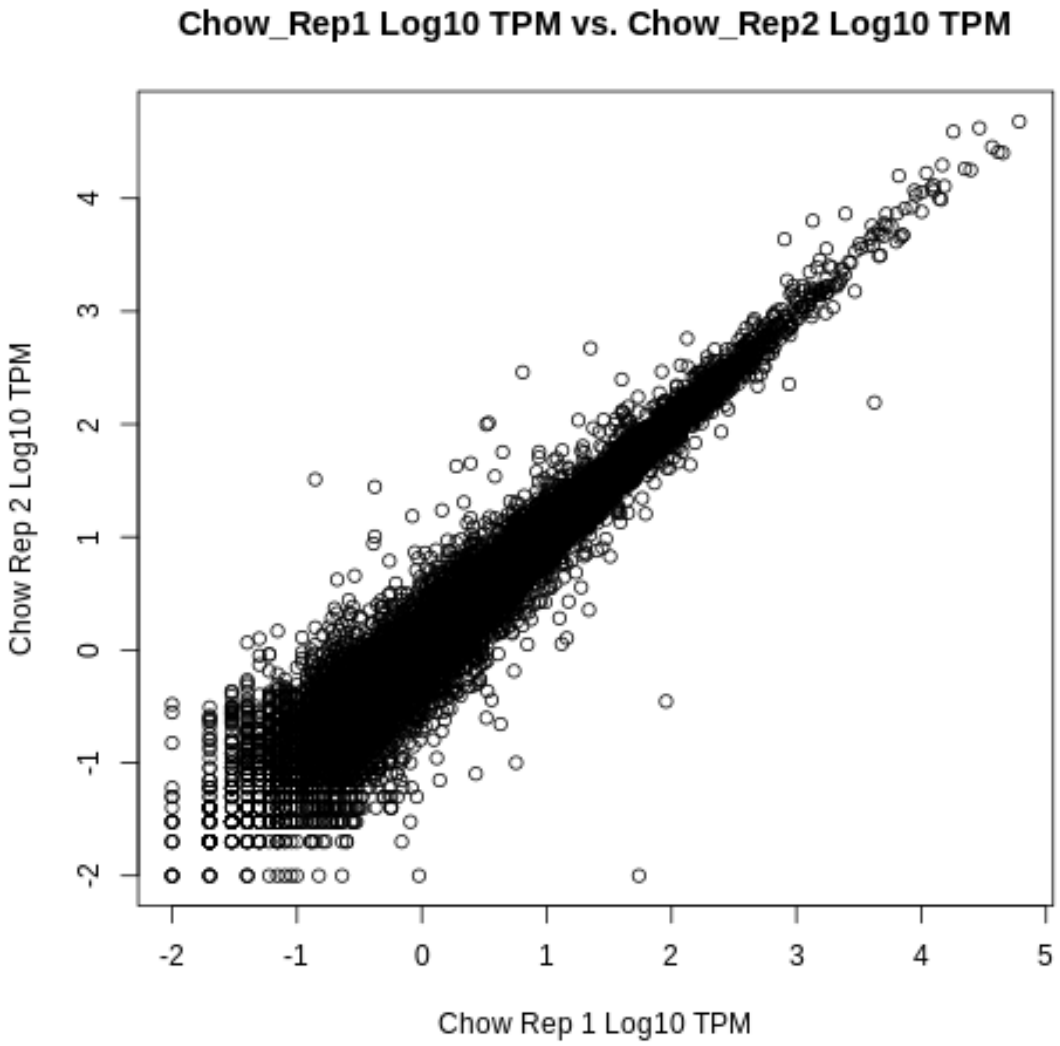
Question 2: R Code

Code for Scatter Plots comparing log10 TPM values for Chow Rep 1 vs Chow Rep 2 & HFD Rep 1 vs HFD Rep 2

[2]:
```
# Run this to allow using the %%R cell magic
%load_ext rpy2.ipython

# Suppress warnings (most notable from rpy2)
import warnings
```

```
warnings.filterwarnings('ignore')
```

[3]:
```
%%R

# Read in Chow Rep 1 and 2 delim files
data1 <- read.delim("Chow_Rep1.genes.results", header = TRUE, sep = "\t")
data2 <- read.delim("Chow_Rep2.genes.results", header = TRUE, sep = "\t")

# Convert into 1og10 TPM values
TPM1 <- log10(data1$TPM)
TPM2 <- log10(data2$TPM)

# Scatter Plot with Chow Rep 1 as x-axis and Chow Rep 2 as y-axis
plot(TPM1, TPM2, main = "Chow_Rep1 Log10 TPM vs. Chow_Rep2 Log10 TPM", xlab =␣
 ↪"Chow Rep 1 Log10 TPM", ylab = "Chow Rep 2 Log10 TPM")

# Read in HFD Rep 1 and 2 delim files
hfd1 <- read.delim("HFD_Rep1.genes.results", header = TRUE, sep = "\t")
hfd2 <- read.delim("HFD_Rep2.genes.results", header = TRUE, sep = "\t")

#Convert into log10 TPM values
tpm1 <- log10(hfd1$TPM)
tpm2 <- log10(hfd2$TPM)

# Scatter plot with HFD Rep1 as x-axis and HFD Rep2 as y-axis
plot(tpm1, tpm2, main = "HFD_Rep1 Log10 TPM vs. HFD_Rep2 Log10 TPM", xlab =␣
 ↪"HFD Rep 1 Log10 TPM", ylab = "HFD Rep 2 Log10 TPM")
```



Part 2: Differential Expression Analysis

[17]:
```
%%R

##### Load the libraries we need #####
library("DESeq2")
library("tximport")

# This cell is an example. We recommend putting the R code you use for this
# in the CSE185-LAB4-README.ipynb notebook

# You might find some of the code below helpful!
# Or, you can ignore what we have below and follow
```

```
# http://bioconductor.org/packages/devel/bioc/vignettes/DESeq2/inst/doc/DESeq2.
 ↪html

##### List the files and set up metadata #####
# Note, you should change this to use the files in your home directory
#files <- c("Chow_Rep1.genes.results",
#         "Chow_Rep2.genes.results",
#         "Chow_Rep3.genes.results",
#         "HFD_Rep1.genes.results",
#         "HFD_Rep2.genes.results",
#         "HFD_Rep3.genes.results")
#conditions <- c(rep("Chow", 3), rep("HFD", 3))
#samples <- data.frame("run"=c("Chow_Rep1", "Chow_Rep2", "Chow_Rep3",␣
 ↪"HFD_Rep1", "HFD_Rep2", "HFD_Rep3"),
#                     "condition"=conditions)
#names(files) = samples$run

sampleFiles <- c("Chow_Rep1.genes.results", "Chow_Rep2.genes.results",␣
 ↪"Chow_Rep3.genes.results",
                 "HFD_Rep1.genes.results", "HFD_Rep2.genes.results", "HFD_Rep3.
 ↪genes.results")
sampleNames <- c("Chow_Rep1", "Chow_Rep2", "Chow_Rep3", "HFD_Rep1", "HFD_Rep2",␣
 ↪"HFD_Rep3")
samplePaths <- file.path(getwd(), sampleFiles)

# Define a function to read in RSEM files and extract the transcript IDs and␣
 ↪counts
read_rsem_file <- function(filepath) {
  rsem_data <- read.table(filepath, header=TRUE, stringsAsFactors=FALSE)
  transcript_ids <- rsem_data$transcript_id
  counts <- rsem_data$expected_count
  return(list(transcript_ids=transcript_ids, counts=counts))
}

# Read in the RSEM data for all samples and combine into a single matrix
sample_data <- lapply(samplePaths, read_rsem_file)
transcript_ids <- sample_data[[1]]$transcript_ids
counts_matrix <- sapply(sample_data, function(x) round(x$counts))
colnames(counts_matrix) <- sampleNames
rownames(counts_matrix) <- transcript_ids

# Create a data frame with the sample names and condition labels
sample_info <- data.frame(sampleName=sampleNames,␣
 ↪condition=rep(c("Chow","HFD"), each=3))

# Create the DESeq2 data object
```

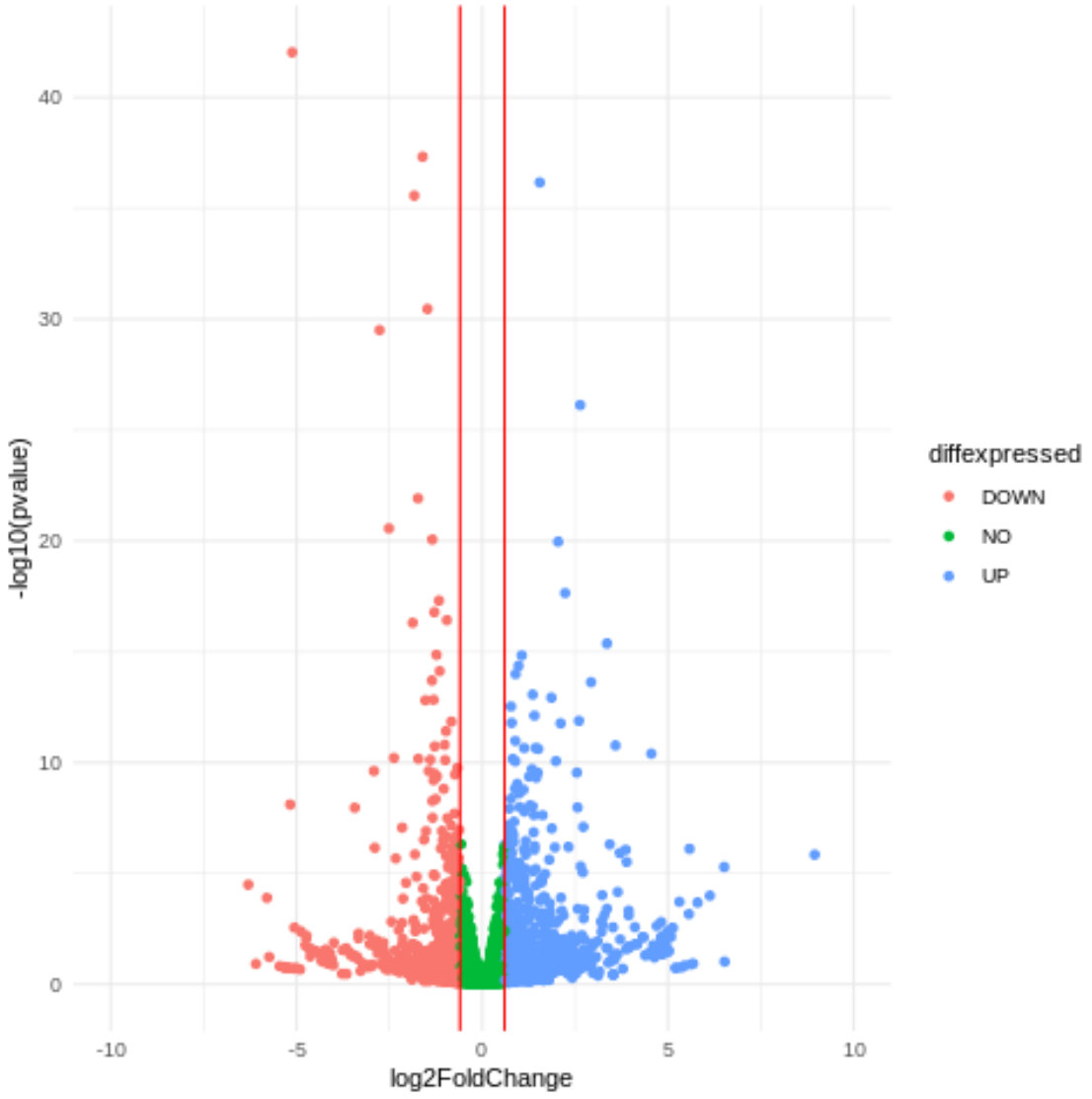
```
dds <- DESeqDataSetFromMatrix(countData = counts_matrix, colData = sample_info,
                              design = ~ condition)

# Filter out genes with low counts
dds <- dds[ rowSums(counts(dds)) > 10, ]

# Run DESeq2 analysis
dds <- DESeq(dds)

# Extract the differential expression results
results <- results(dds)

# Write the results to a file
write.csv(as.data.frame(results), file = "chow_vs_hfd_deseq2.csv")
```

```
R[write to console]: converting counts to integer mode

R[write to console]: estimating size factors

R[write to console]: estimating dispersions

R[write to console]: gene-wise dispersion estimates

R[write to console]: mean-dispersion relationship

R[write to console]: final dispersion estimates

R[write to console]: fitting model and testing
```

Question 4: R Code for Volcano Plot

[25]:
```
%%R

library(ggplot2)
# Read in CSV file
data <- read.csv("chow_vs_hfd_deseq2.csv")

# add column of NAs
data$diffexpressed <- "NO"

# if log2FoldChange > 0.6, set as "UP"
data$diffexpressed[data$log2FoldChange > 0.6] <- "UP"

#if log2FoldChange < -0.6, set as "DOWN"
data$diffexpressed[data$log2FoldChange < -0.6] <- "DOWN"

ggplot(data, aes(x = log2FoldChange, y = -log10(pvalue), col = diffexpressed)) +
```

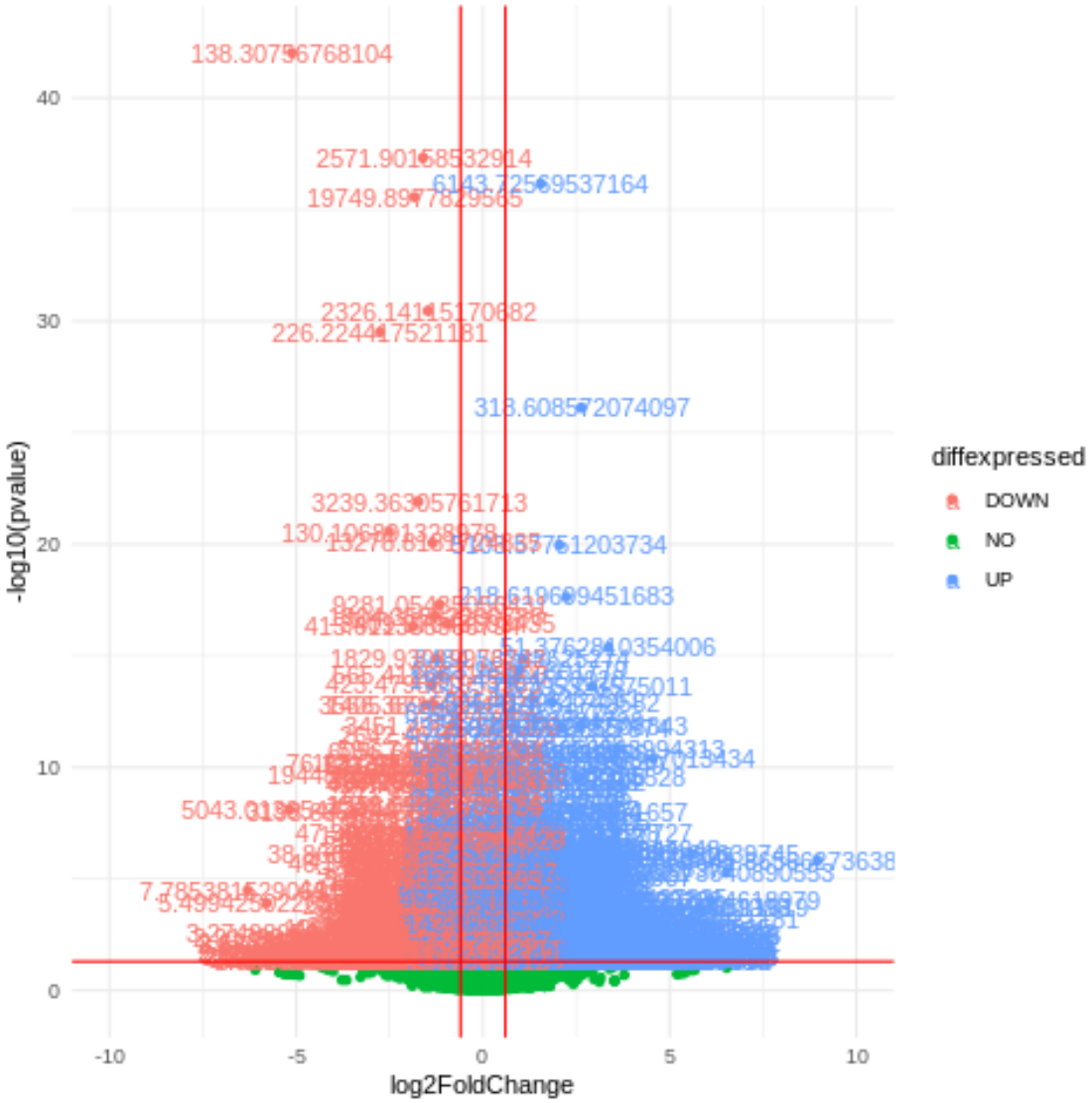
```
geom_point() +
theme_minimal() +
geom_vline(xintercept = c(-0.6, 0.6), col='red') +
xlim(-10, 10)
```



Question 5:

I used the same code as before but this time I added the p-value threshold of 0.05 (5%) and also added labels for the baseMean for the most differentially expressed genes. Then, I went through the plot starting at the top gene (most differentially expressed) and went down through 10 genes. Which can be used to go through and see which gene has that baseMean. Then, I can use this to search for that gene using grep and find out its log2fold change and p-value and also grep again to find the gene name in GRCm38.75.gene_names.

[26]:
```
%%R

library(ggplot2)
# Read in CSV file
data <- read.csv("chow_vs_hfd_deseq2.csv")

# add column of NAs
data$diffexpressed <- "NO"

# if log2FoldChange > 0.6 and p-value < 0.05, set as "UP"
data$diffexpressed[data$log2FoldChange > 0.6 & data$pvalue < 0.05] <- "UP"

#if log2FoldChange < -0.6 and pvalue < 0.05, set as "DOWN"
data$diffexpressed[data$log2FoldChange < -0.6 & data$pvalue < 0.05] <- "DOWN"

data$dlabel <- NA
data$dlabel[data$diffexpressed != "NO"] <- data$baseMean[data$diffexpressed !=␣
 ↪"NO"]

ggplot(data, aes(x = log2FoldChange, y = -log10(pvalue), col = diffexpressed,␣
 ↪label = dlabel)) +
  geom_point() +
  theme_minimal() +
  geom_text() +
  geom_vline(xintercept = c(-0.6, 0.6), col='red') +
  geom_hline(yintercept = -log10(0.05), col = 'red') +
  xlim(-10, 10)
```



Question 6: GO Analysis I went to the GO Analysis Tool DAVID which can be found here: https://david.ncifcrf.gov/tools.jsp Then, I used the following tutorial to perform GO analysis via DAVID and just followed the directions by each slide: https://david.ncifcrf.gov/helps/tutorial.pdf

Question 7: For Question 7, I just made a table of the returned enriched categories and added their corresponding p-value