Lab 4: RNA-seq

Skills: differential expression, basic R, enrichment analysis

For this week you'll need to complete the following, which are described in more detail below:

- `CSE185-LAB4-EXERCISES.ipynb` (20 pts)
- `CSE185-LAB4-REPORT.ipynb` (70 pts) (this notebook)
- `CSE185-LAB4-README.ipynb` (10 pts)

Intro

You would like to study the effects that eating a high fat diet might have on your health. To explore this, you perform an experiment using mice. You feed three mice a standard "chow" diet and another three mice a high fat diet ("HFD"). After 7 weeks, you collect liver biopsies from each of the six mice and perform RNA-sequencing. You'd like to analyze the RNA-seq results to determine which genes changed their expression, and what those genes might be doing.

Our pipeline will consist of:

1. Alignment of RNA-seq data and quantifying expression levels (we've already done that for you)
2. Differential expression analysis
3. "GO" Enrichment analysis to characterize differentially expressed genes.
4. Visualizing expression data with IGV

Note, the data from this lab is from the paper High fat diet-induced changes of mouse hepatic transcription and enhancer activity can be reversed by subsequent weight loss *Scientific Reports* 2017.

Note on new lab notebook format

This week, we'll be transitioning to a new lab format which is no longer autograded and therefore a bit more open-ended. You'll be filling out the following:

- `CSE185-LAB4-REPORT.ipynb` : will provide instructions (and hints) about the analyses you'll be performing that week. It will have prompts for the questions you should answer. This file is graded. The report is worth 70 pts (70%) of the points for this week's lab. From here on out, we will not be asking you to paste specific commands you used to get your answer in the actual report. Instead, we will ask you to document your commands in the following file:

- `CSE185-LAB4-README.ipynb` : We have included a mostly blank document for you. You should use this file to keep track of exactly the commands you used for each analysis, so that you can come back to it next week (or next year!) and remember what you did. This file is graded. Documentation is worth 10 pts (10%) of the points for this week's lab.

There are additional exercises in `CSE185-LAB4-EXERCISES.ipynb` . As usual exercises are worth a total of 20 pts (20%) for this week's lab. You'll also see `CSE185-LAB4-PREPROCESSING.ipynb` with more information for how we performed the alignment and expression quantification steps.

Summary of tools covered

In this lab we'll be using the following tools and resources:

- Gene Expression Omnibus to access publicly available data.
- IGV a genome browser. Used for visualizing RNA-seq alignments (and other types of genomic data)
- DESeq2 for differential expression analysis. This is an R package. It takes in gene expression levels across multiple replicates of multiple conditions and determines which genes are differentially expressed.
- PANTHER (or your tool of choice) for Gene Ontology analysis.

We will also refer to these tools but will not run them directly:

- STAR for aligning RNA-seq reads to a reference transcriptome.
- RSEM for quantifying gene expression. RSEM takes aligned reads as input and outputs expression levels for each gene.

Note: Because DESeq2 is only available as an R package, we'll be writing a small bit of R code. We'll point you to some resources to help you figure this out.

And as usual, we'll do some plotting with the matplotlib Python library (or, whatever method you want to use for plotting. You do not have to use matplotlib. You can also write R code directly in Jupyter notebooks as we'll see below).

Summary of data provided

For this lab, to save you time we have already precomputed part of the results you will need to complete the lab, which can be found in `~/public/lab4/` . You can read about the steps we used for this in `CSE185-LAB4-PREPROCESSING.ipynb` . We will go over how data was obtained from the paper and preprocessed in lecture.

You should see:

- `*.genes.results` : gene-level expression results output by RSEM for each condition/replicate.
- `GRCm38.75.gene_names` contains a mapping between ENSEMBL gene ids (`ENSMUSG...`) and gene names (e.g. Cdc45, Klf6)
- `~/public/cs185-sp22-a00-public/lab3/preprocessing/` : This directory contains fastq files and BAMs used as input to transcriptome quantification with `RSEM` . Processes used to compute these results are described in `CSE185-LAB4-PREPROCESSING.ipynb` . (Note this path is from a previous year. The data is large so we didn't copy it over.)

1. Quantifying gene expression

We have already aligned raw reads to the reference transcriptome using STAR and quantified expression of each gene using RSEM. You can find the results in `~/public/lab4/*.genes.results` .

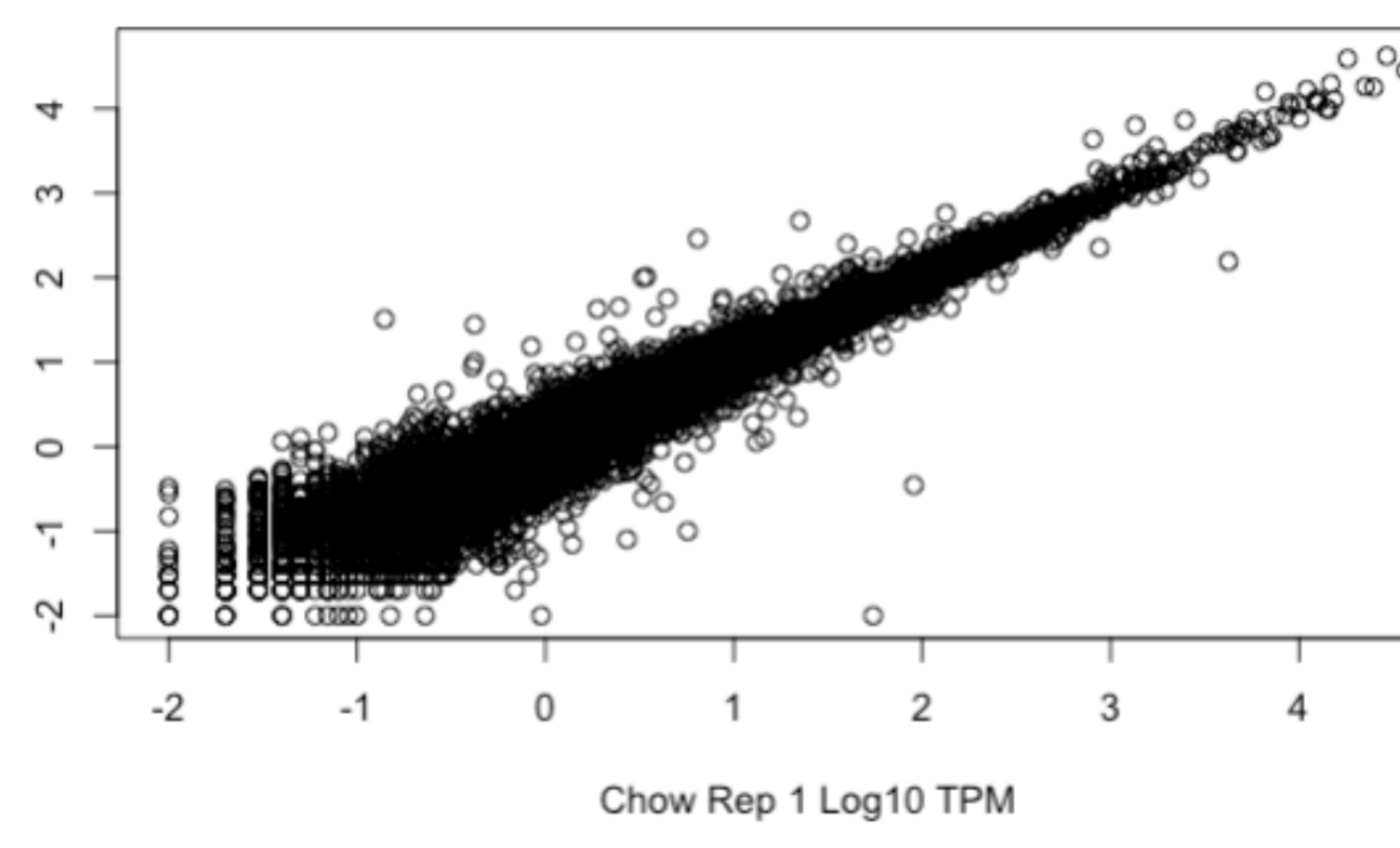
Take a look at the output file. You will see (among other things): gene_id (a funny looking string like ENSMUSG...), a list of transcript ids for that gene, the length of the gene, and the estimated expression level given in both TPM and FPKM. Refer to the exercises and the lecture slides for more info about the difference between these metrics.

Question 1 (5 pts) Summarize the expression results by reporting how many genes were expressed (TPM>0) in each sample. Report your results in a table with columns "Dataset" (e.g. Chow_Rep1, Chow_Rep2, etc.) and "# genes". You should have a total of 6 rows (3 each for Chow and HFD). (helpful Markdown table syntax)

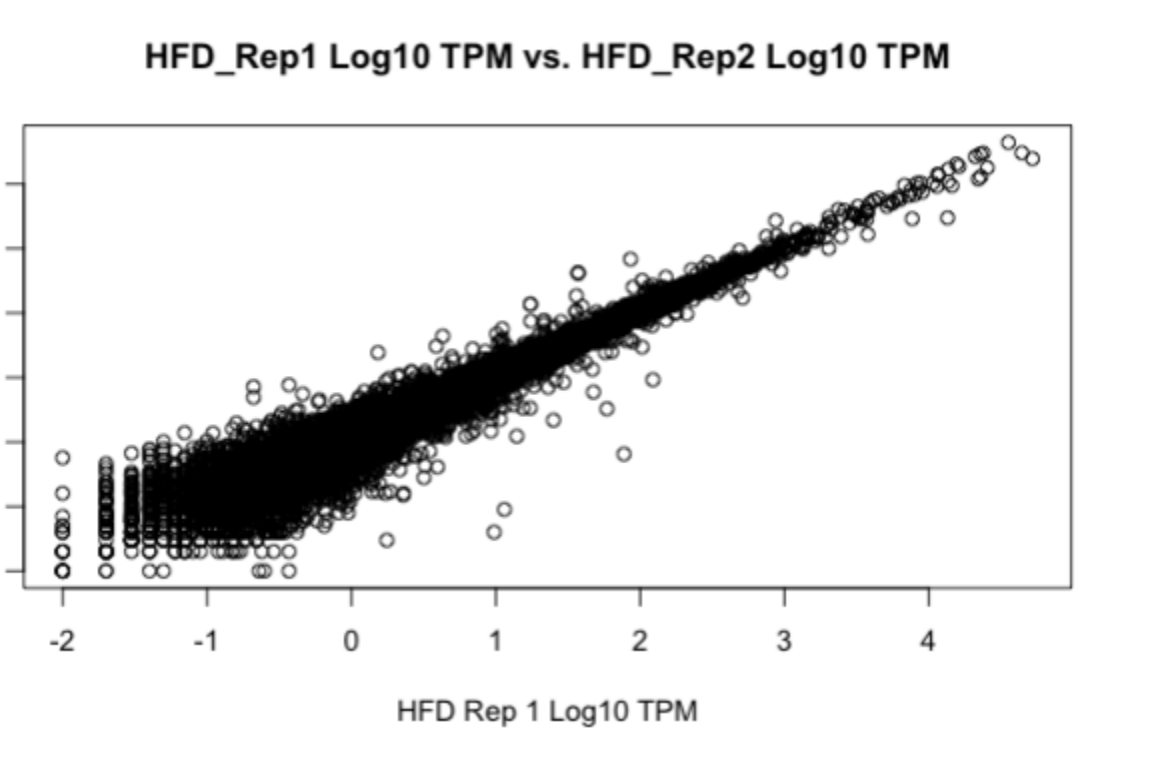
Dataset	# Genes
Chow_Rep1	16919
Chow_Rep2	17096
Chow_Rep3	16795
HFD_Rep1	17078
HFD_Rep2	16847
HFD_Rep3	16734

Question 2 (10 pts) Since we included three replicate samples for each condition, one way we can assess the quality of our results is by comparing replicate samples. Provide scatter plots comparing TPM values in Chow Rep1 vs. Rep2 and HFD Rep1 vs. Rep2. Since expression values have a large range, you should compare log10 TPM values rather than raw values themselves. Make sure to label the axes on your plots. Provide some interpretation of your plots - do the expression values between replicates seem to be well correlated or not?

Note: you can insert figures by saving them to a file and including them here using markdown syntax.



I believe the expression values between the Chow Rep1 and Chow Rep2 seem to be well correlated since I think the plot exhibits a linear relationship and one can easily imagine be a line of best fit since there is basically one on the plot by itself. Additionally, there aren't any extreme outliers



I believe the expression values between the HFD Rep1 and HFD Rep2 seem to be well correlated since I think the scatter plot also exhibits a linear relationship and one can also easily imagine a line of best fit since there is basically one on the plot by itself. Additionally, there aren't any extreme outliers so I think these expression values are well correlated.

2. Differential expression analysis

Now that we have quantified gene expression in each of our samples, we can perform differential expression analysis. For this, we will be using DESeq2, a widely used method that takes in estimated gene expression levels across different samples and outputs a set of differentially expressed genes.

This tool is an R package, and so must be run from R. This means we have to learn at least a bit of a new language. Fortunately, we can write R code directly in Jupyter notebooks using the R "cell magic" (see below). Alternatively, you can open a separate "R" document from JupyterHub or use R at the terminal if that's more comfortable.

The example cells below shows how you can turn a cell into an R cell. It also loads `DESeq2` and `tximport` , two libraries you will need, and provides some helpful comments to get you started.

Use DESeq2 to identify genes that are differentially expressed between Chow and HFD. You'll need to write some R code that does the following:

- Set up the list of files to be used as input to DESeq2. We will input the RSEM results from each of our replicates in each condition. Recall these can be found at `~/public/lab4/*.genes.results` .
- Import the data to DESeq2 using the tximport library.
- Run DESeq2 to find differentially expressed genes between the two conditions.
- Output results to a file `chow_vs_hfd_deseq2.csv` .

We recommend taking a look at the following tutorial: http://bioconductor.org/packages/devel/bioc/vignettes/DESeq2/inst/doc/DESeq2.html. We'll also be going over some hints in lab sessions.

In [5]:
```
# Run this to allow using the %%R cell magic
%load_ext rpy2.ipython

# Suppress warnings (most notable from rpy2)
import warnings
warnings.filterwarnings('ignore')
```

```
The rpy2.ipython extension is already loaded. To reload it, use:
  %reload_ext rpy2.ipython
```

In [6]:
```
%%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

##### Use "tximport" to convert RSEM results to the format needed by DESeq2 #####
# TODO
# 1. Import RSEM results with tximport
# 2. Note, we also used txi$length[txi$length == 0] <- 1
# to add a pseudocount of 1 to fix an error with 0-length transcripts
# 3. Load to a deseq dataset from tximport (see DESeqDataSetFromTximport)

###### Filter things with very low counts so we don't waste time on those #####
# TODO

##### Perform deseq2 #####
# TODO

##### Write results to chow_vs_hfd_deseq2.csv #####
# TODO
```

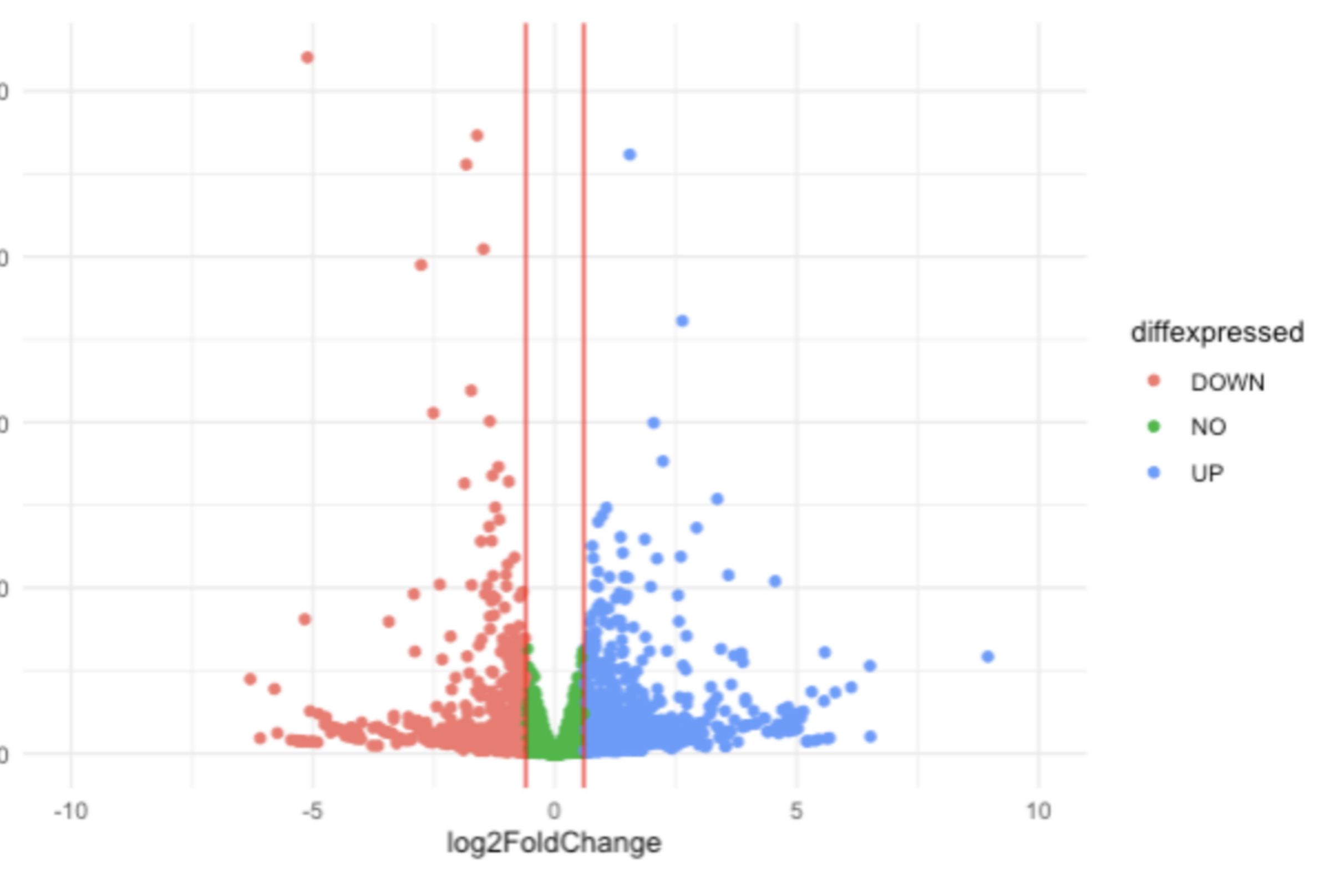
After DESeq2 runs successfully, you should get a csv file with multiple columns:

- Gene id (e.g. ENSMUSG00000000088)
- log2FoldChange: gives the log2 of the fold change in expression of the gene between conditions
- pvalue: gives the nominal p-value for each gene, where here the null hypothesis is that the gene is expressed equally in Chow vs. HFD.
- padj: gives the pvalues "adjusted" for the number of hypotheses being tested (false discovery rate).

Question 3 (5 pts) Describe how you performed differential expression analysis. Which package(s) did you use? Which version(s)? Did you have to do any preprocessing of your data (e.g. removing genes with length 0 or removing genes with very low counts)? Justify any choices you made about which genes to filter or any parameters you had to set.

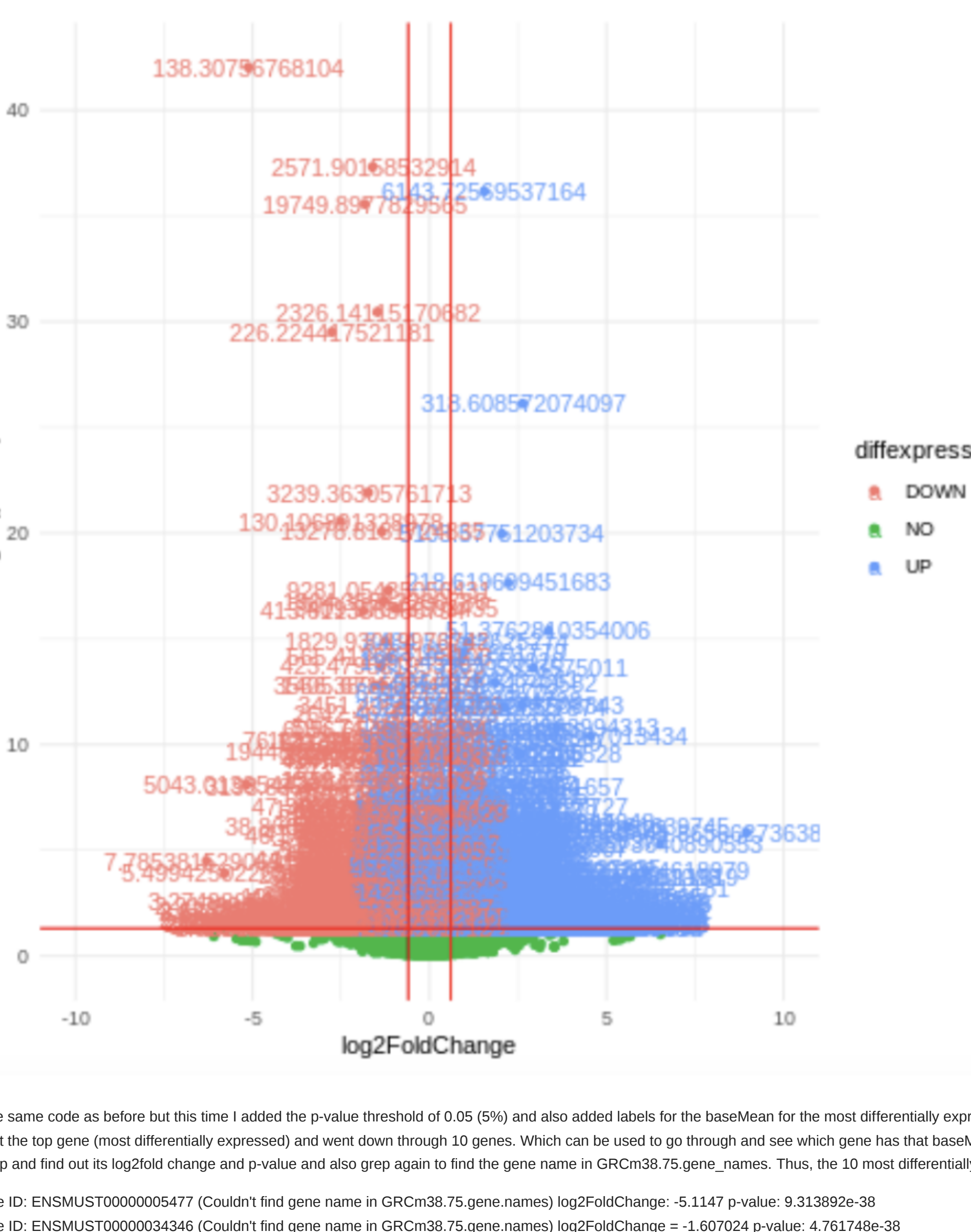
I performed differential expression analysis by first defining the sample names and corresponding file paths. Then, I created a function which reads in the RSEM files for all the samples which is done to extract the transcript counts and IDs from the file. Then, it creates a matrix which consists of the count data from all the samples and also creates a dataframe consisting of the sample names and condition labels. Then, the DESeq2 data object is created from using that matrix and dataframe. Then, I removed genes with low counts and ran DESeq on that filtered data. I did this because it makes the data look better and saves time later on when trying to filter out low counts. Lastly, I wrote the results to the chow_vs_hfd_deseq2.csv file.

Question 4 (15 pts) Visualize the differential expression results using a "volcano plot", which plots the log2 fold change vs. the -log10 p-value. Try to make your volcano plot more informative by: (1) coloring genes that are significantly differentially expressed in a different color and (2) annotating the names of the top differentially expressed genes.



Question 5 (5 pts) At an adjusted p-value threshold (FDR) of 5%, how many genes were differentially expressed across the two conditions? Report the top 10 most differentially expressed genes, their log2 fold change, and the p-value. Report the gene names (e.g. Fads2) rather than gene ids.

You'll notice all the gene ids are pretty cryptic looking (e.g. ENSMUSG00000000088). We have provided a file, `GRCm38.75.gene_names` in the `~/public/lab4` directory that can be used to convert these to gene names.



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. Thus, the 10 most differentially expressed genes are:

1. Gene ID: ENSMUST00000005477 (Couldn't find gene name in GRCm38.75.gene.names) log2FoldChange: -5.1147 p-value: 9.313892e-38
2. Gene ID: ENSMUST00000034346 (Couldn't find gene name in GRCm38.75.gene.names) log2FoldChange = -1.607024 p-value: 4.761748e-38
3. Gene ID: ENSMUST00000025567 (Couldn't find gene name in GRCm38.75.gene.names) log2FoldChange = 1.548887 p-value: 6.89562e-37
4. Gene ID: ENSMUST00000003137 (Couldn't find gene name in GRCm38.75.gene.names) log2FoldChange = -1.830496 p-value: 2.708261e-36
5. Gene ID: ENSMUST00000021231 (Couldn't find gene name in GRCm38.75.gene.names) log2FoldChange = -1.475336 p-value: 3.47376e-31
6. Gene Name: Pip5k1c log2FoldChange = -2.764005 p-value: 3.153614e-30
7. Gene Name: Rpl10-ps6 log2FoldChange = 2.636098 p-value: 1.518622e-23
8. Gene ID: ENSMUST00000048959 (Couldn't find gene name in GRCm38.75.gene.names) log2FoldChange = -1.729287 p-value: 2.145644e-22
9. Gene ID: ENSMUST00000068150 (Couldn't find gene name in GRCm38.75.gene.names) log2FoldChange = -2.51172 p-value: 2.749133e-21
10. Gene ID: ENSMUST00000004140 (Couldn't find gene name in GRCm38.75.gene.names) log2FoldChange = -1.345106 p-value: 8.73088e-21

3. Gene ontology enrichment analysis

Finally, use gene ontology (GO) enrichment analysis to characterize the genes that were differentially expressed in Chow vs. HFD. We recommend you use PANTHER or DAVID, which are web tools, for doing this. However you are welcome to try out other GO analysis tools instead. You can find a helpful tutorial on DAVID here: https://david.ncifcrf.gov/helps/tutorial.pdf.

For enrichment analysis, you will need to define our "foreground" and "background". Here:

- "Background" consists of all genes that we analyzed (i.e., all genes with any output in the DESeq2 csv file).
- "Foreground" consists of all genes that were differentially expressed (padj<0.05). You should create separate foreground sets for up- and down-regulated genes. Recall up-regulated genes will have log2 fold change > 0, and down-regulated genes will have log2 fold change < 0.

You can upload lists of genes to use as foreground or background. In our solution, we uploaded lists of the gene IDs (ENSG...) to DAVID. Then DAVID will perform GO enrichment analysis for you.

Perform GO enrichment by comparing your differentially expressed genes (adjusted p<0.05) to all genes analyzed. You should perform a separate analaysis for down-regulated (log2 fold change <0) and up-regulated (log2fold change>0) genes.

Question 6 (10 pts) Describe any methods for how you performed gene ontology enrichment analysis. Which tool did you use? How did you define your foreground and background? Were there any extra options you had to set? Justify what you used for each of those.

I used DAVID for my gene ontology enrichment analysis. To define my background, I used a txt file which contains all the genes. To define my foreground, I used a txt file which contained the significantly up-regulated genes and another txt file which contained the significantly down-regulated genes. I used ENSEMBL_GENE_ID as the identifier since it was able to correctly go through the format of the txt files.

Question 7 (5 pts) Were any gene ontology categories enriched in differentially expressed genes? For each enriched category, provide at least the name of the category and its enrichment p-value.

Yes there were gene ontology categories enriched in differentially expressed genes. Here is a table:

Up-Regulated:

Term	enrichment p-value
KW-0443-Liquid Metabolism	2.5646151e-14
KW-0444-Liquid Biosynthesis	1.469007e-19
KW-0153-Cholesterol Metabolism	4.2743586448e-13
KW-1207-Sterol Metabolism	1.7268408925-13
GO:0006695-cholesterol biosynthetic process	1.27932599ee-15
GO:0006629-liquid metabolic process	1.015854067532e-12
GO:0016126-sterol biosynthetic process	3.724798247e-11
GO:0008203-cholesterol metabolic process	6.281364317e-11
KW-0152-Cholesterol biosynthesis	1.0908163e-11
KW-0752-steroid biosynthesis	2.830480294e-11
KW-0753-steroid metabolism	5.6789128e-11
KW-0756-sterol biosynthesis	1.095774582e-11
GO:0006631-fatty acid metabolic process	9.904032161e-07
KW-0275-Fatty acid biosynthesis	2.633482200087e-07
KW-0276-Fatty acid metabolism	2.15760778e-05
KW-0812-Transmembrane	0.000187617876
GO:0005789-endoplasmic reticulum membrane	0.0030651549380
KW-0325-Glycoprotein	0.00029365546987991653
GO:0016491-oxidoreductase activity	0.01234244176512933
GO:0005615-extracellular space	0.020280975389424638
KW-1133-Transmembrane helix	0.0015644379
mmu01100 Metabolic pathways	0.021567057701669334

Down-Regulated:

Term	enrichment p-value
KW-0408-Iron	3.797884895462289e-09
DOMAIN GST N-terminal	2.1830489738139391e-07
KW-0256-Endoplasmic reticulum	1.3536556764393026e-08
GO:0006629-liquid metabolic process	7.8976599399764e-07
DOMAIN GST C-terminal	4.8253858790e-06
GO:0008905-steroid hydroxylase activity	5.7825933000854e-06
GO:0005783-endoplasmic reticulum	4.015870114e-06
GO:0016491-oxidoreductase activity	1.144361603311e-05
KW-0560-Oxidoreductase	2.417870224e-06
mmu00071:Fatty acid degredation	1.484006e-05
mmu01524:Platinum drug resistance	0.00011863320810334699
GO:0005396-steroid binding	0.0014038203738392374
GO:0043295-glutathione binding	0.0015798031050028305
mmu04750:inflammatory mediator regulation of TRP	0.0014253677258
mmu04976: Bile secretion	0.0016783078
mmu5418: Fluid Sheat Stress	0.003017402377985978
GO:0019899-enzyme binding	0.007158255819032067
GO:0006210-estrogen metabolic process	0.0294473948
mmu95208 Chemical carcinogenesis - reactive oxygen species	0.0099645599813133
mmu05225 Hepatocellular carcinogenesis	0.04334513893587162

4. Discussion questions

Answer the following discussion questions below:

Question 8 (5 pts) Summarize the characteristics of genes that are up- and down-regulated in HFD compared to Chow fed mice. Do the GO categories enriched in those gene sets make sense? Do a little exploring of the top differentially expressed genes to find out what they might be doing.

The up-regulated genes in HFD compared to Chow are enriched in categories such as transport and fatty acid. While, the down-regulated genes in HFD to Chow are also enriched in categories such as fatty acids. These GO categories enriched in these gene sets do make sense because HFD literally means "High Fat Diet" so it would make sense that HFD is more enriched in the fatty categories compared to Chow which is a normal mouse chow diet.

Question 9 (5 pts) You'll likely notice from your volcano plot that there are many genes with very high fold changes (e.g. log2 fold change > 6) but that are not called as significant. Hypothesize why. it could be helpful to dig in and look at the raw data (e.g. expression levels from RSEM) for some of those genes.

I think it could be helpful to dig in and look at the raw data for some of those genes because just looking at the log2fold change by itself is not enough. There are other external factors that cause a gene to be significant or not, not by just looking at the log2fold change. Thus, I think it could be helpful to look at the raw data and do tests on the expression levels itself as opposed to just the log2fold change since it doesn't accurately solely represent significance.

Question 10 (5 pts) Multiple factors contribute to our statistical power to detect differentially expressed genes. (Power is the probability that we reject the null hypothesis, given that there is really an effect). How might our power be affected by the number of replicates? (here we had 3 replicates each). What about gene length?

Due to multiple factors contributing to our statistical power to detect differentially expressed genes, I think the number of replicates affects our power by increasing it. For the gene length, I think it will also affect our power by increasing it as well.

Optional: Visualizing alignments in IGV

It is a good practice to visualize our alignments, both to see that the alignment worked well, and also to see whether things we report as differentially expressed make sense. We'll use a genome browser called the Integrative Genomics Viewer, or IGV. In the process, we'll also see how to access data published with the original paper.

Navigate to the web version of IGV: https://igv.org/app/ (or alternatively, download the desktop version which we actually prefer to use). Set the genome to (mm9), since according to the paper they had aligned reads to the mm9 reference genome (see page 11, "Analysis of RNA, ChIP and DNase sequencing"). Note, in our lab we aligned the data to the newer mm10/GRCm38 reference.

Take a moment to orient yourself with IGV. It is basically like a Google Maps for genomes! The top gives the names of each chromosome. The bottom track, labeled "Refseq genes" gives the names and coordinates for all annotated genes. Let's choose one to look at. Type "Fads2" in the search box at the top. This will zoom the view in on this gene. Notice how in the gene you can see the exon and intron structure. The little arrows in the introns point to the left, which means this gene is on the reverse strand of the reference. Take a look at another gene (e.g. Fads3) to see a gene on the forward strand. Zoom in further (using the "+" at the top right) until you can see actual DNA sequence at the top.

Now we'd like to load our sequence alignments. While IGV can directly visualize BAM files, we'll instead look at "counts" files (".bedgraph" format. see http://genome.ucsc.edu/goldenPath/help/bedgraph.html) which are much smaller. These give read counts per position (i.e. coverage) which can give us an idea of the abundance of each gene.

The authors of the paper have already provided data in bedgraph format. Head to the paper where we got the data: High fat diet-induced changes of mouse hepatic transcription and enhancer activity can be reversed by subsequent weight loss *Scientific Reports* 2017. On page 12, you'll see a section "Additional Information" which contains information about where to access the raw data. If a paper generates new sequencing or other data types, they are usually required to make this information available. It will often be in a section labeled "Data availability" or something similar. We used the polyA RNA-seq, which you should find has been deposited in the Gene Expression Omnibus under accession GSE87565.

Go to the Gene Expression Omnibus and search using the accession GSE87565. Scroll all the way to the bottom to find a section labeled "Supplementary file". If you click "custom" under the "Download" column of that table, you'll be able to select individual datasets to download. Check the boxes for:

- GSM2333833_SM1959_RNA_Chow_rep1.bedgraph.gz
- GSM2333834_SM1960_RNA_Chow_rep2.bedgraph.gz
- GSM2333835_SM1961_RNA_Chow_rep3.bedgraph.gz
- GSM2333839_SM1965_RNA_HFD_rep1.bedgraph.gz
- GSM2333840_SM1966_RNA_HFD_rep2.bedgraph.gz
- GSM2333841_SM1967_RNA_HFD_rep3.bedgraph.gz

After checking those, click "Download" to download to your local computer. Once the downloaded is complete, decompress the .tar file (it will be named something like `GSE87565_RAW.tar` .

Now, go back to IGV in your web browser. Click "Tracks" -> "Local File", then navigate to the bedgraph.gz files you had downloaded in order to upload them to IGV.

Navigate again to Fads2 (This should be one of your differentially expressed genes!). Note that the RNA-seq tracks have very "spiky" coverage. Some regions have tons of reads and others are flat. Note how that compares to the structure of the gene annotated on the bottom. As expected, the "spikes" correspond to reads from exons, since intron and intergenic sequences generally aren't sequenced in our RNA-seq experiment.

Take note of the y-axis range for each dataset (after you initially load them, it should range from 0 to a couple hundred for this gene). To make the tracks more comparable, you can set them all to have the same data range by using the "settings" icon to the left of each track and clicking "set data range". Try setting them all to 500. (the appropriate range will be different for different genes, since genes are really highly expressed whereas others are very lowly expressed. See the huge range on your scatterplot!) You should see that the gene is much more highly expressed in the HFD samples.

It is also helpful to color tracks. For instance, you can color tracks by condition (see options in settings to the right of each track). Navigate to different differentially expressed genes and see if the data matches up with your results from DESeq2.

Note: the bedgraph files from GEO look a little weird (the coverage peaks are too wide), which may be the result of an extension step being applied. We also provided tdfs (here) that we created from the aligned BAM files after we ran STAR. You can also visualize these using IGV, just be sure to change to the mm10 genome. Visualizations in the slides are based on these tdfs.

We will continue using IGV to explore ChIP-seq datasets next week.