A deep dive into Crohn's Disease Genetic Biomarkers

Remi Sampaleanu

A. Background and Problem Statement

Crohn's disease (CD) is a chronic inflammatory bowel disease (IBD) that affects the digestive tract. Its symptoms vary, but commonly include abdominal pain, diarrhea, and weight loss. In serious cases, or if left untreated, the disease can lead to life-threatening complications (Torres, et al., 2017). Diagnosis and treatment for CD is challenging, and a greater understanding of the disease's molecular mechanisms and gene expression patterns in affected individuals can help shed light on its pathogenesis (Feuerstein & Cheifitz, 2017). Conducting genomic data analyses can allow for the development of new potential therapeutic targets, personalized medicines, and the identification of differentially expressed genes which may play a role in disease development (Cosnes, et al., 2002). In this project, we seek to do just that. We will be attempting to conduct our own differential gene expression analysis and compare our findings with that of Sadler, et al., (2016). In this original publication, the authors attempted to determine which DNA methylation and gene expression patterns were characteristic of Crohn's disease-associated fibrosis. One of the experiments conducted by the researchers was a differential gene analysis using Fragments Per Kilobase Million (FPKM) data from six different RNAseq samples and the R "limma/voom" pipeline. The researchers were able to produce two heatmaps (one for genes upregulated in CD fibroblasts, and one for downregulated genes), which can be seen in **Figure 1.** It is this experiment that we will be attempting to recreate. We will however be using the "R/edgeR" pipeline and conducting some additional analysis on enriched pathways.

B. Data

In this section, we will be outlining details about the data used in our analysis. The original raw data used in both the original publication as well as our own analysis is Illumina HiSeq 2500 RNAseq data. There were six single-end read Fastq formatted samples available for download, which is also what was used in the original publication. Three of these samples were controls (prefixed with "NL") from healthy colon fibroblasts, and three were from CD-affected colon fibroblasts (prefixed with "CD"). There were no sample replicates taken or available for download. The average number of sequencing reads across all six samples is 56,680,066. For most of the analysis in this project, we used counts data. This data, before filtering, contained the count data for 23,678 genes. The average size of the samples, when stored on a computer, was 3.49 gigabytes. All samples used in this project can be found with GEO accession number GSE67250, or at the following link: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE67250.

C. Analysis Steps

The first step of our analysis was to try and obtain counts data from the raw RNAseq data (Fastq samples) that we were able to download. This involves a few simple steps, run from the command line and using the STAR alignment package. The first of these steps, after downloading the Fastq samples, was to download the human reference genome (in our case Hg19, as this was the one used by Sadler, et al.,). With this this file ("hg19.fa"), and its

associated annotation file ("hg19.ncbiRefSeq.gtf"), we then attempted to create index files which would speed up our alignment and mapping considerably (Danek, et al., 2014). Unfortunately, at this step, we repeatedly ran into errors involving memory on our server (see **Figure 2** and **Figure 3**). Assuming we were able to generate these index files, the next steps would be to perform quality control and preprocessing on our samples (i.e. with FastQC), run our STAR alignment command to align samples to the reference genome, generate our counts data with the featureCounts function, and then finally load the output from that command into R and convert it into a dataframe for further analysis. In **Figure 4**, we show how these commands would have appeared had we run them to generate our counts data. For our project, however, we were forced to use the counts data (in tab separated .txt format) provided by the researchers. To compensate for this lack of alignment, we have conducted additional pathway enrichment analyses and have built heatmaps showing differential expression.

After downloading the counts data, the first step was to create our "countdata" and "sampleInfo" data frames, from which further analysis could draw on. The "countdata" data frame consists of a list of 23,678 genes with the respective number of counts for each gene in each sample. The "sampleInfo" data frame has six rows, one for each sample, and describes the condition of the sample (i.e. control vs Crohn's Disease). This process is shown in **Figure 5**. The next step was to make bar graphs of the library sizes for each sample. The goal of this step was to determine if there were samples with library sizes under 2,000,000, which would then be removed from the analysis (also shown in **Figure 5**). Next, we performed gene filtering. We loaded the edgeR package and calculated the counts per million (CPM) for each gene, and tried to determine a suitable CPM threshold for filtering. We then identified the genes with CPM values not above the threshold (in our case 0.125) in all six samples, and removed those genes from our countdata (see **Figure 6)**. After filtering, we converted our data to a DGE object and performed some quality control steps to ensure that filtering was successful. Namely, we created boxplots of LogCPM vs Samples, and an MDS plot of the samples colored by their condition (**Figure 7, Figure 8)**. Following filtering, we normalized our data and generated abline plots to show the before and after effects of this normalization (**Figure 9)**. Finally, we could create our model matrix and find the top differentially expressed genes between our control and CD samples. We used a threshold of differentially expressed genes whose P values were less than or equal to 0.01 **(Figure 10)**. Out of interest, and as additional steps – we visualized these differentially expressed genes in a volcano plot and annotated them using the "org.Hs.eg.db" library (**Figure 11,** and **Figure 12)**. Finally, we were able to write the (novel) code that generated the heatmaps for upregulated and downregulated genes in CD fibroblasts. To do so, we first sorted our differentially expressed genes by LogFC, and then selected the 20 highest as our most upregulated (the most positive LogFC genes correspond to the upregulated differentially expressed genes), and the 20 lowest for our downregulated genes. With these two sets of 20 genes, we loaded the "pheatmap" library and constructed our two heatmaps (**Figure 13, Figure 14)**. Finally, we performed our second novel piece of analysis, our WikiPathways enrichment analysis. Since WikiPathways uses the ENTREZIDs of genes for lookup, we first had to convert our differentially expressed gene matrix into one containing their ENTREZ IDs. We could then load the packages (clusterProfiler, org.Hs.eg.db, pathview, and DOSE) that were required, and run our enrichment analysis. We also created a dotplot to view the results of this analysis (**Figure 15)**.

D. Results

Making Countdata. In this step, our output was a data frame with 23,678 genes (see **Figure 16)**. We also generated a bar graph showing the library sizes for each sample. This graph clearly shows that there were no samples with library sizes under 2 million, which was our cut-off (**Figure 17)**.

Filtering Genes. In this step we filtered all low count genes (with CPM < 0.125). We determined this threshold by plotting the counts vs CPM for each gene and finding the point on the line where the CPM corresponded to a threshold value of 10 reads. After removing the genes which had CPM values below the threshold in at least one sample, we reduced the total number of genes by 10,654 to 13,024 (**Figure 18**).

Quality Control. A graph of the LogCPM vs Samples (of the filtered genes) is shown in **Figure 19**. One will immediately notice the even distribution of the LogCPM in each sample, and the lack of any major outlier samples. These are good indicators that our filtering step was successful in removing genes with particularly low reads.

Normalization. This was a relatively simple step. We simply found the log counts of our filtered genes and plotted the distribution before and after normalization. This can be seen in **Figure 20**. While it is very difficult to tell with the naked eye, there was a very slight shift towards an Expression log-ratio of 0, indicating that our data was normalized.

Differentially Expressed Genes. In this section of our analysis, we found the differentially expressed genes between the "control" and "Crohn's disease" conditions, using our dgeObj, model matrix, and the glmFit/glmLRT functions. We noticed that no genes were significant at FDR <= 0.05, so we selected for differentially expressed genes based on their P values. We initially used a threshold P value of 0.05 and obtained 560 downregulated genes and 379 upregulated ones. We determined that this was perhaps too lenient and allowed for too many genes in our analysis, so we decreased the threshold to allow for only genes with P values of <= 0.01. This ultimately yielded 280 differentially expressed genes. We can see this selection of genes in **Figure 21,** and **Figure 22,** which shows a plot of logFC vs average logCPM, in which genes that are differentially expressed are colored red. We can also see these genes in a volcano plot of logFC vs -log(P) (**Figure 23).** As an additional step, we annotated our list of 13,024 to include information about their Genename, EntrezIDs, and whether they were significant (based on the aforementioned selection process). This can be seen in **Figure 24** and was useful in further downstream analysis.

Heatmaps. The main component of our analysis, once we identified our differentially expressed genes, was to try and recreate the heatmaps found in the original publication by Sadler, et al., and see how ours compares. Once we selected our top 20 upregulated and downregulated genes (see section **C.** for details, and **Figures 25, 26** for a list of these genes), we obtained two heatmaps that – if we ignore some minor differences in the ordering of columns – are relatively similar at first glance to those in the publication (see **figures 27, 28**). In the heatmap showing the top 20 downregulated genes, we can see that most of the rectangles corresponding to the Crohn's disease samples ("CDx") are colored blue – indicating negative log fold change or downregulation. The opposite is true for the heatmap showing upregulated genes. One may notice, however, that the list of genes in our heatmaps versus those in the published ones are not completely identical. Some of the most downregulated or upregulated genes, however, are

persistent across both (as an example, see PREX2 or TLR4 for the upregulated genes, or NAP1L2 and SORBS1 for the downregulated genes). Together, these facts seem to indicate that our heatmap generation was mostly successful. Discrepancies between the two are brought up in section **E.**

Pathway Enrichment. In this step, we conducted a WikiPathways enrichment analysis on the genes we discovered as differentially expressed. We found a total of seven enriched pathways (Cell cycle, Retinoblastoma gene in cancer, Adipogenesis, DNA IR-damage and cellular response via ATP, Gastric cancer network 1, Cohesin Complex, and Regulation of sister chromatid separation). Thes pathways were all plotted on a dot plot (see **Figure 29**). Many of these pathways can be linked in some way to Crohn's disease. Adipogenesis, for example, is the process of fat cell formation. In Crohn's disease, patients can experience changes in adipose tissue, such as fat creeping, which is the extension of mesenteric fat onto the bowel wall (Ahmad, et al., 2020). This could potentially lead to the upregulation of adipogenesis-related pathways.

E. Discussion

In this section, we will mostly discuss the differences between our heatmaps and the published ones, and the potential reasons for non-overlapping differentially expressed genes in our respective analyses. While our heatmaps share some similarities, they are not composed of the exact same gene sets. For one, the researchers found 72 differentially expressed genes (at P values < 0.05, FDR $< 5\%$. And logFC > 1.5. Our selection criteria was even *more* stringent (P values < 0.01), yet we discovered 280 differentially expressed genes (Sadler, et al., 2016). We must conclude, then that the difference lies in the fact that they used normalized FPKM values, rather than raw counts data like in our experiments. Additionally, this fact means that in their heatmaps there was no need to select only a portion of the most highly downregulated or upregulated genes, as they could simply use all of them. It is possible that our process of selecting these most differentially expressed genes – by their logFC values – may have caused some of the difference. It would be interesting to see the result of selecting these genes by their P values instead. Nevertheless, we created a Venn diagram to visualize the differences between the published down or upregulated genes and our own. In **Figures 30** and **31**, we can see that the common downregulated genes are: "NAP1L2", "SORBS1, and "AOC3". These are all in the top 50% lowest logFC genes in our list of top 20 downregulated genes and are also some of the authors most downregulated genes. However, this still means that only 15% (3/20) of our downregulated genes were also found to be downregulated using the FPKM to limma/voom pipeline that the authors used. Common upregulated genes included: "CADM1", "PREX2", "TSPAN11", "ITGA8", "TLR4", and "RAPGEF5". While these are some of our most upregulated genes, there is still only a 30% commonality with published ones. It would be interesting to see if we can increase the number of common genes by changing our selection criteria for up or downregulated genes. It is also possible that by using logFC instead of adjusted P values we have obtained a slightly different set of genes.

F. Appendix

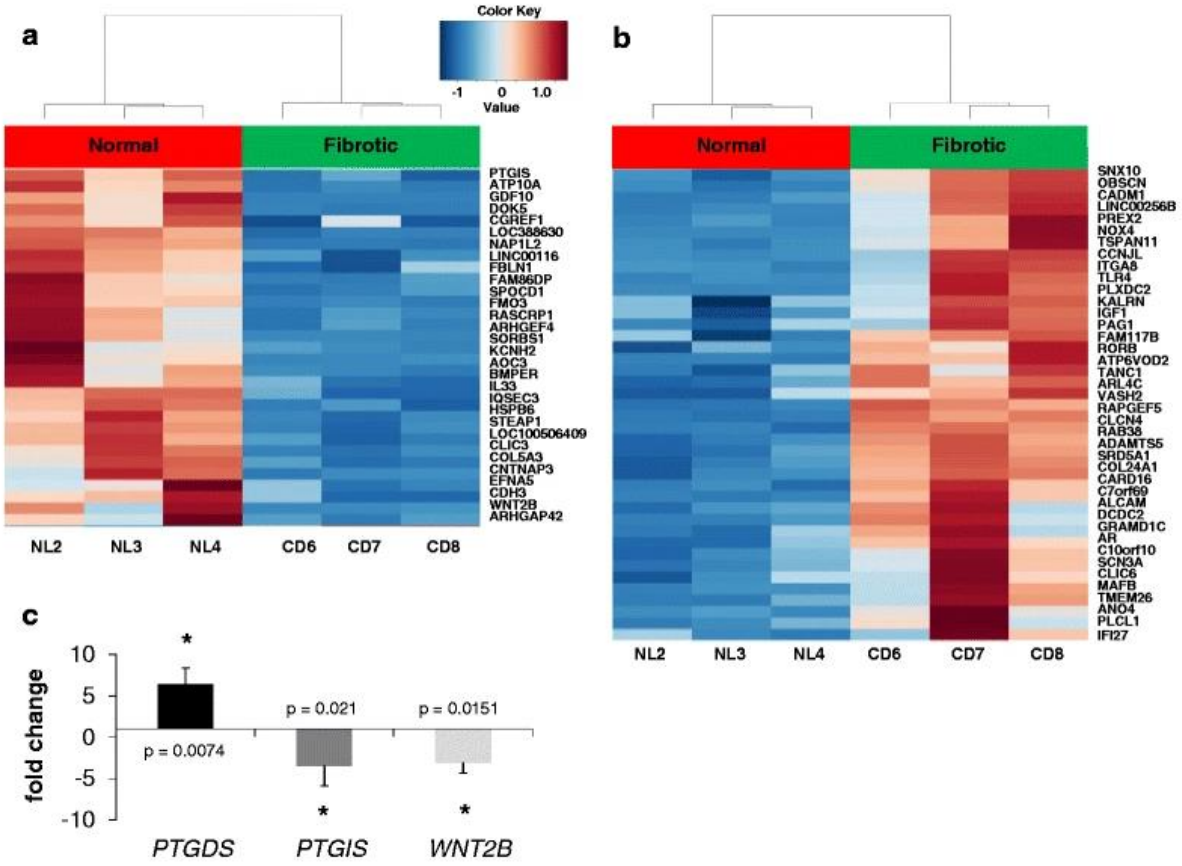


Figure 1. These are the two published heatmaps (Sadler, et al., 2016) showing differentially expressed genes in Crohn's disease fibroblasts. **A.** shows downregulated genes, while **B.** showcases upregulated genes in CD fibroblasts.

```
rsampale@compute:~/MyWorkDir/final2/data$ /home/shared/MBI4850G/STAR-Aligner/STAR-2.7.10b/source/STAR --genomeDir index_directory --runMode genomeGenerat
        /home/shared/MBI4850G/STAR-Aligner/STAR-2.7.10b/source/STAR --genomeDir index_directory --runMode genomeGenerate --runThreadN 12 --genomeFastaFil
        STAR version: 2.7.10b   compiled: 2023-02-09T21:45:12-05:00 :/home/shared/MBI4850G/STAR-Aligner/STAR-2.7.10b/source
Apr 10 15:28:04 ..... started STAR run
Apr 10 15:28:04 ... starting to generate Genome files
Apr 10 15:30:49 ..... processing annotations GTF
Apr 10 15:31:28 ... starting to sort Suffix Array. This may take a long time...
Apr 10 15:31:43 ... sorting Suffix Array chunks and saving them to disk...

Genome_genomeGenerate.cpp:286:genomeGenerate: exiting because of *OUTPUT FILE* error: could not write the output file index_directory//SA_5
fail()=1 ; bad()=1
Error while trying to write chunk # 0; 0 bytes
File size full = 1091721072 bytes
File size on disk = 963772416 bytes
Solution: check that you have enough space on the disk
Empty space on disk = 1515389530112 bytes

Apr 10 15:36:52 ..... FATAL ERROR, exiting
rsampale@compute:~/MyWorkDir/final2/data$
```

Figure 2. Memory error experienced when trying to create index files for the human reference genome.

```
MyWorkDir > final2 > data > $ command_lines.bash
# downloaded 3 control group samples, and 3 Crohn's disease associated fibrosis group samples
~/sratoolkit.3.0.1-ubuntu64/bin/prefetch --option-file SRR_Acc_List.txt
~/sratoolkit.3.0.1-ubuntu64/bin/fasterq-dump --split-files SRR1928190
~/sratoolkit.3.0.1-ubuntu64/bin/fasterq-dump --split-files SRR1928189
~/sratoolkit.3.0.1-ubuntu64/bin/fasterq-dump --split-files SRR1928191
~/sratoolkit.3.0.1-ubuntu64/bin/fasterq-dump --split-files SRR1928192
~/sratoolkit.3.0.1-ubuntu64/bin/fasterq-dump --split-files SRR1928188
~/sratoolkit.3.0.1-ubuntu64/bin/fasterq-dump --split-files SRR1928193
# rename files to control_x or crohns_x

wget https://hgdownload.soe.ucsc.edu/goldenPath/hg19/bigZips/latest/hg19.fa.gz
wget https://hgdownload.soe.ucsc.edu/goldenPath/hg19/bigZips/genes/hg19.ensGene.gtf.gz
wget https://hgdownload.soe.ucsc.edu/goldenPath/hg19/bigZips/genes/hg19.refGene.gtf.gz
wget https://hgdownload.soe.ucsc.edu/goldenPath/hg19/bigZips/genes/hg19.knownGene.gtf.gz
wget https://hgdownload.soe.ucsc.edu/goldenPath/hg19/bigZips/genes/hg19.ncbiRefSeq.gtf.gz
# then gunzip them

#make genome indicies:
mkdir index_directory
/home/shared/MBI4850G/STAR-Aligner/STAR-2.7.10b/source/STAR --genomeDir index_directory --runMode genomeGenerate --runThreadN 12 --genomeFastaFiles hg19.fa --sjdbGTFfile hg19.refGene.gtf
/home/shared/MBI4850G/STAR-Aligner/STAR-2.7.10b/source/STAR --genomeDir index_directory_2 --runMode genomeGenerate --runThreadN 12 --genomeFastaFiles hg19.fa --sjdbGTFfile hg19.knownGene.gtf
/home/shared/MBI4850G/STAR-Aligner/STAR-2.7.10b/source/STAR --genomeDir index_directory --runMode genomeGenerate --runThreadN 12 --genomeFastaFiles hg19.fa --sjdbGTFfile hg19.ncbiRefSeq.gtf
```

Figure 3. Process of downloading samples and Hg19 reference genome (with annotations). Also shown is the code to generate the index files for the reference genome which gave the error shown in **Figure 2**

```
STAR --genomeDir <path_to_index_directory> --readFilesIn <path_to_fastq_1> [<path_to_fastq_2>] --runThreadN <num_threads> --outFileNamePrefix <output_prefix> --outSAMtype BAM SortedByCoordinate
featureCounts -a <path_to_annotation_gtf> -o <output_file> -T <num_threads> <list_of_sorted_bam_files>
```

Figure 4. Theoretical next steps of generating counts data from the raw rnaSeq data. Placeholder names are used instead of actual paths and file names due to the fact that they were never actually created.

```r
library(dplyr)

# Read the text file into a data frame
file_path <- "~/Documents/r_saves/R_sessions/final/data/GSE67250_RnaSeqCounts.txt" # Replace with your file path
data <- read.table(file_path, header = TRUE)

# Create the seqdata data frame
seqdata <- data

# Extract the sample names from the column names (excluding the GeneID column)
sample_names <- colnames(data)[-1]

# Create the sampleInfo data frame
sampleInfo <- tibble(
  Samples = sample_names,
  Condition = ifelse(grepl("^NL", sample_names), "control", "crohn's disease")
)

# Display the seqdata and sampleInfo data frames
sampleInfo <- as.data.frame(sampleInfo)


#bar plots of library sizes:
library(ggplot2)
library(dplyr)

countdata <- seqdata[,-1]
rownames(countdata) <- seqdata[,1]
Col_counts <- colSums(countdata)

data_for_counts_bar <- cbind(sampleInfo, Col_counts)
counts_bargraph <- barplot(height = data_for_counts_bar$Col_counts, names.arg = data_for_counts_bar$Samples, ylab = "Library Size", xlab = "Sample")
# NO SAMPLES HAVE LIBRARY SIZES OF UNDER 2 MILLION, CAN KEEP ALL

countdata <- rbind(countdata, Col_counts)
```

Figure 5. Shows the steps of creating our countdata and sampleInfo data frames, and the generation of a histogram showing library sizes for each sample.

```r
# FILTERING LOW COUNT GENES:
library("edgeR")
myCPM <- cpm(countdata)
# make a data frame with CPM values added:
my.df <- data.frame()
for (sample.name in colnames(countdata)) {
  mat <- cbind(CPM = myCPM[,sample.name], Counts = countdata[,sample.name], Sample = rep(sample.name, nrow(countdata)))
  my.df <- rbind(my.df, mat)
}
my.df$CPM <- as.numeric(my.df$CPM)
freq_vs_cpm <- ggplot(data=my.df, aes(x=CPM)) +
  geom_histogram(binwidth=0.5) +
  labs(title = "Distribution of CPM",x="CPM",y="Frequency") +
  theme_minimal() +
  xlim(0,600) + ylim(0,100)
freq_vs_cpm

subset.df <- my.df[my.df$Sample == "NL2_RNASEQ",]
subset.df$CPM <- as.numeric(subset.df$CPM)
subset.df$Counts <- as.numeric(subset.df$Counts)
scat1 <- ggplot(data=subset.df,aes(x=CPM,y=Counts)) + geom_point() + xlim(0,2.5) + ylim(0,100)
scat1
# CPM thresh = 0.125
AboveThresh <- (myCPM > 0.125)
num.samples.aboveThresh <- rowSums(AboveThresh)
my.df2 <- as.data.frame(num.samples.aboveThresh)
aboveThresh_histo <- ggplot(data=my.df2,aes(x=num.samples.aboveThresh)) + geom_histogram()
aboveThresh_histo

# filter the genes that are not above the threshold in all 6 samples (10654 genes):
length(which(num.samples.aboveThresh < 6))
keep.genes <- names(which(num.samples.aboveThresh >= 6))
counts_filtered <- countdata[keep.genes,]

# Make filtered data a DGE object:
dgeObj <- DGEList(counts_filtered)
```

Figure 6. Code showing the process of finding the CPM for each gene, and filtering by our CPM threshold of 0.125.

```r
# QUALITY CONTROL:
log.cpm <- cpm(dgeObj, log=TRUE)
log.cpm.df <- data.frame()
for (mysample in colnames(counts_filtered)) {
  Mat <- cbind(Sample = rep(mysample,nrow(log.cpm)), Log.CPM = log.cpm[,mysample])
  log.cpm.df <- rbind(log.cpm.df, Mat)
}
log.cpm.df$Sample <- as.factor(log.cpm.df$Sample)
log.cpm.df$Log.CPM <- as.numeric(log.cpm.df$Log.CPM)
cpm_boxplot <- ggplot(data=log.cpm.df,aes(x=Sample,y=Log.CPM)) + geom_boxplot() +
  labs(x="Samples",y="Log CPM",title="Distribution of Log CPM across samples")
cpm_boxplot
# Conclusion: No real outliers or strange samples

```

Figure 7. Code for the creation of a boxplot showing Samples vs logCPM of the genes within them.

```r
sampleInfo$Condition <- as.factor(sampleInfo$Condition)
col.condition <- c("purple","cyan")[sampleInfo$Condition]
data.frame(sampleInfo$Condition,col.condition)
plotMDS(dgeObj,col=col.condition)
legend("topleft",fill=c("purple","cyan"), legend = levels(sampleInfo$Condition))
title("Condition Type")

```

Figure 8. Code for the creation of an MDS plot of the various samples, colored by condition.

```
# NORMALIZATION:
dgeObj <- calcNormFactors(dgeObj)
par(mfrow=c(1,2))
plotMD(log.cpm,column=2)
abline(h=0,col="blue")
plotMD(log.cpm,column=4)
abline(h=0,col="blue")

logcounts <- cpm(dgeObj,log=TRUE)
plotMD(logcounts,column=2)
abline(h=0,col="red")
plotMD(logcounts,column=4)
abline(h=0,col="red")
# conclusion: minor shift/change after normalization

```

Figure 9. Code for the generation of abline plots showing the before and after effects of normalization.

```
# DIFF EXPRESSION:
mat_design <- as.formula(~ Condition)
model_matrix <- model.matrix(mat_design, data=sampleInfo)

dgeObj <- estimateCommonDisp(dgeObj)
dgeObj <- estimateGLMTrendedDisp(dgeObj)
dgeObj <- estimateTagwiseDisp(dgeObj)
# NOTE: ADD OTHERS HERE?
plotBCV(dgeObj)

fit <- glmFit(y=dgeObj,design = model_matrix)
lrt.CvsC <- glmLRT(fit, coef = 2)
topTags(lrt.CvsC)
# lowest false discovery rate is 40% (all of them are around here), none are significant at FDR <= 0.05

diff_ex_results <- as.data.frame(topTags(lrt.CvsC,n=Inf))
de <- decideTestsDGE(lrt.CvsC,adjust.method = "none",p.value=0.05, lfc=0)
# 560 downregulated, 379 upreg
# Log FC vs log CPM, significant in red
de2 <- cbind(de, diffExp=as.logical(de))
detags <- rownames(dgeObj)[as.logical(de)]
plotSmear(lrt.CvsC,de.tags = detags)
# decreasing p value to 0.01:
de <- decideTestsDGE(lrt.CvsC,adjust.method = "none",p.value=0.01, lfc=0)
detags <- rownames(dgeObj)[as.logical(de)]
plotSmear(lrt.CvsC,de.tags = detags)

```

Figure 10. Shows the main code for the identification and filtering of highly differentially expressed genes between the two conditions (“control” and “Crohn’s disease”), using only genes differentially expressed with P values less than 0.01.

```
# annotating with info from org.Hs.eg.db
library(org.Hs.eg.db)
all.genes <- rownames(diff_ex_results)

status <- all.genes %in% keys(org.Hs.eg.db, keytype="SYMBOL")
annotations <- select(org.Hs.eg.db,keys=rownames(diff_ex_results),keytype="SYMBOL",columns = c("GENENAME","ENTREZID","SYMBOL"))
annotations <- annotations[-12443,]
anno_results <- cbind(diff_ex_results,annotations)
head(anno_results)
```

Figure 11. Code for the annotation of all our differentially expressed genes using the “org.Hs.eg.db” library.

```r
#volcano plot of logP vs log FC
negLogPValue <- -log10(anno_results$PValue)
absLogFC <- abs(anno_results$logFC)
thresh <- -log10(0.01)
Significant <- (negLogPValue >= thresh & absLogFC > 3)
anno_results <- cbind(diff_ex_results,negLogPValue,Significant,annotations)
anno_results$logFC <- as.numeric(anno_results$logFC)
anno_results$Significant <- as.factor(anno_results$Significant)
volc <- ggplot(data=anno_results,aes(logFC,negLogPValue)) +
  geom_point(color=1+Significant) +
  theme_light() +
  labs(x="Log Fold Change",y="-log10(P value)")
volc
```

Figure 12. The creation of a volcano plot showing all genes, with only differentially expressed genes (absolute logFC > 3 and P value < 0.01) colored red. Also adds a column to our annotations data frame which tells if a gene is significant or not.

```r
# HEATMAP CREATION:
library(pheatmap)

# Sort the results by log fold change (logFC) in descending order
sorted_results <- diff_ex_results[order(diff_ex_results$logFC, decreasing = TRUE),]

# Extract the top 10 most upregulated genes
top10_upregulated <- sorted_results[1:10,]

# Sort the results by log fold change (logFC) in ascending order
sorted_results <- diff_ex_results[order(diff_ex_results$logFC),]

# Extract the top 10 most downregulated genes
top10_downregulated <- sorted_results[1:10,]

# Combine the top 10 most upregulated and downregulated genes
top10_genes <- rbind(top10_upregulated, top10_downregulated)

# Print the results
print(top10_genes)

# NEW HEATMAPS BUT WITH ONLY TOP 20 THIS TIME:
# Extract the top 20 most upregulated genes
top20_upregulated <- sorted_results[1:20,]
# Extract the top 20 most downregulated genes
top20_downregulated <- sorted_results[(nrow(sorted_results)-19):nrow(sorted_results),]
# Get the expression values for the top 20 most upregulated genes
upregulated_genes_expr <- logcounts[rownames(logcounts) %in% rownames(top20_upregulated),]
# Get the expression values for the top 20 most downregulated genes
downregulated_genes_expr <- logcounts[rownames(logcounts) %in% rownames(top20_downregulated),]
# Scale the expression values by row (gene)
scaled_upregulated_expr <- t(scale(t(upregulated_genes_expr)))
scaled_downregulated_expr <- t(scale(t(downregulated_genes_expr)))
```

Figure 13. Code showing how we obtained the top 20 most upregulated and downregulated genes, for use in our heatmaps.

```r
# Create the heatmaps
color_map <- colorRampPalette(c("blue", "white", "red"))(100)
pheatmap(scaled_upregulated_expr,
         color = color_map,
         scale = "none",
         clustering_distance_rows = "euclidean",
         clustering_distance_cols = "euclidean",
         clustering_method = "complete",
         show_rownames = TRUE,
         show_colnames = TRUE,
         annotation_col = sampleInfo_mod)

pheatmap(scaled_downregulated_expr,
         color = color_map,
         scale = "none",
         clustering_distance_rows = "euclidean",
         clustering_distance_cols = "euclidean",
         clustering_method = "complete",
         show_rownames = TRUE,
         show_colnames = TRUE,
         annotation_col = sampleInfo_mod)
```

Figure 14. Code showing the actual generation of our heatmaps, using the top 20 most upregulated and downregulated genes in CD colon fibroblasts.

```r
# ATTEMPT WIKIPATHWAYS ANALYSIS:
# Create a new data frame with rownames as gene symbols
annotations_new <- annotations
rownames(annotations_new) <- annotations$SYMBOL

# Get the Entrez IDs for the differentially expressed genes
de_symbols <- detags
de_annotations <- annotations_new[de_symbols, ]
de_entrez <- de_annotations$ENTREZID

library(clusterProfiler)
library(org.Hs.eg.db)
library(pathview)
library(DOSE)

data(de_entrez, package="DOSE")
enrich_result <- enrichWP(de_entrez, organism = "Homo sapiens")
dotplot(enrich_result)
```

Figure 15. This code shows how we conducted our WikiPathways enrichment analysis, and generated a dotplot to display these results in a visual manner.

	NL2_RNASEQ	NL3_RNASEQ	NL4_RNASEQ	CD6_RNASEQ	CD7_RNASEQ	CD8_RNASEQ
A1BG	418.7535609	255.362994	312.1691779	201.100228	311.806324	366.8102555
A1BG-AS1	56.0552913	65.483764	89.3778791	60.753095	93.330420	49.8516144
A1CF	3.5881515	4.965438	5.3605672	3.255712	3.988129	5.3385992
A2LD1	307.8571390	287.683772	354.8726436	109.673552	277.607157	139.9379073

Showing 1 to 5 of 23,678 entries, 6 total columns

Figure 16. Our countdata data frame, containing 23,678 genes which were used for all later downstream analysis.

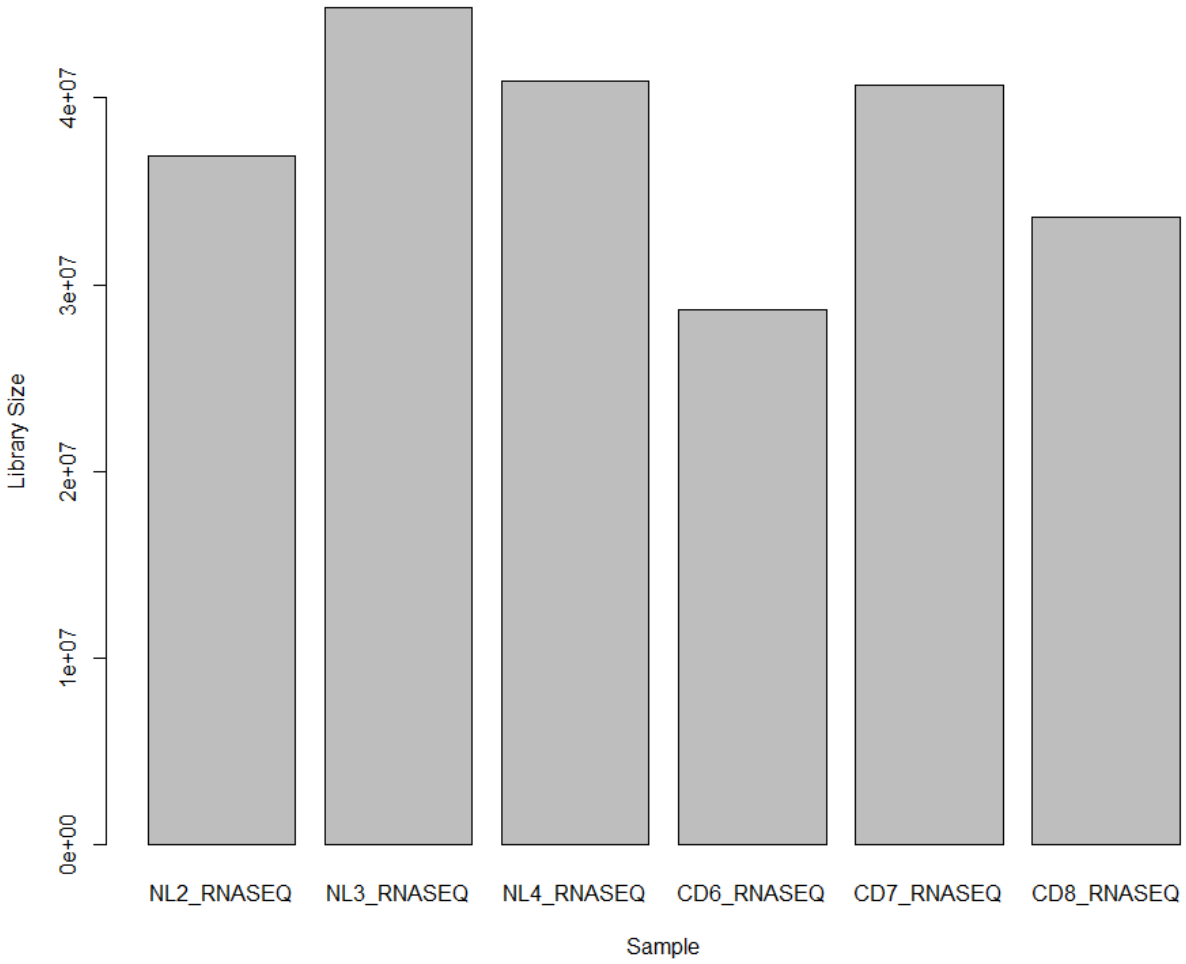


Figure 17. This graph shows the library sizes for each of our six samples. Samples beginning in "NL" represent the control (healthy colon fibroblast) samples, while those beginning in "CD" are the Crohn's disease colon fibroblast samples.

	NL2_RNASEQ	NL3_RNASEQ	NL4_RNASEQ	CD6_RNASEQ	CD7_RNASEQ	CD8_RNASEQ
A1BG	418.75356	255.36299	312.16918	201.10023	311.80632	366.81026
A1BG-AS1	56.05529	65.48376	89.37788	60.75309	93.33042	49.85161
A2LD1	307.85714	287.68377	354.87264	109.67355	277.60716	139.93791
A2M	673.64461	3231.83710	278.17201	1501.03613	4286.94397	3803.52530

Showing 1 to 5 of 13,024 entries, 6 total columns

Figure 18. Our "counts_filtered" data frame, containing 13,024 genes (compared to the original 23,678 present in our original countsdata structure. We can see a clear reduction in genes which did not meet the CPM cut-off and were filtered for having too few reads.

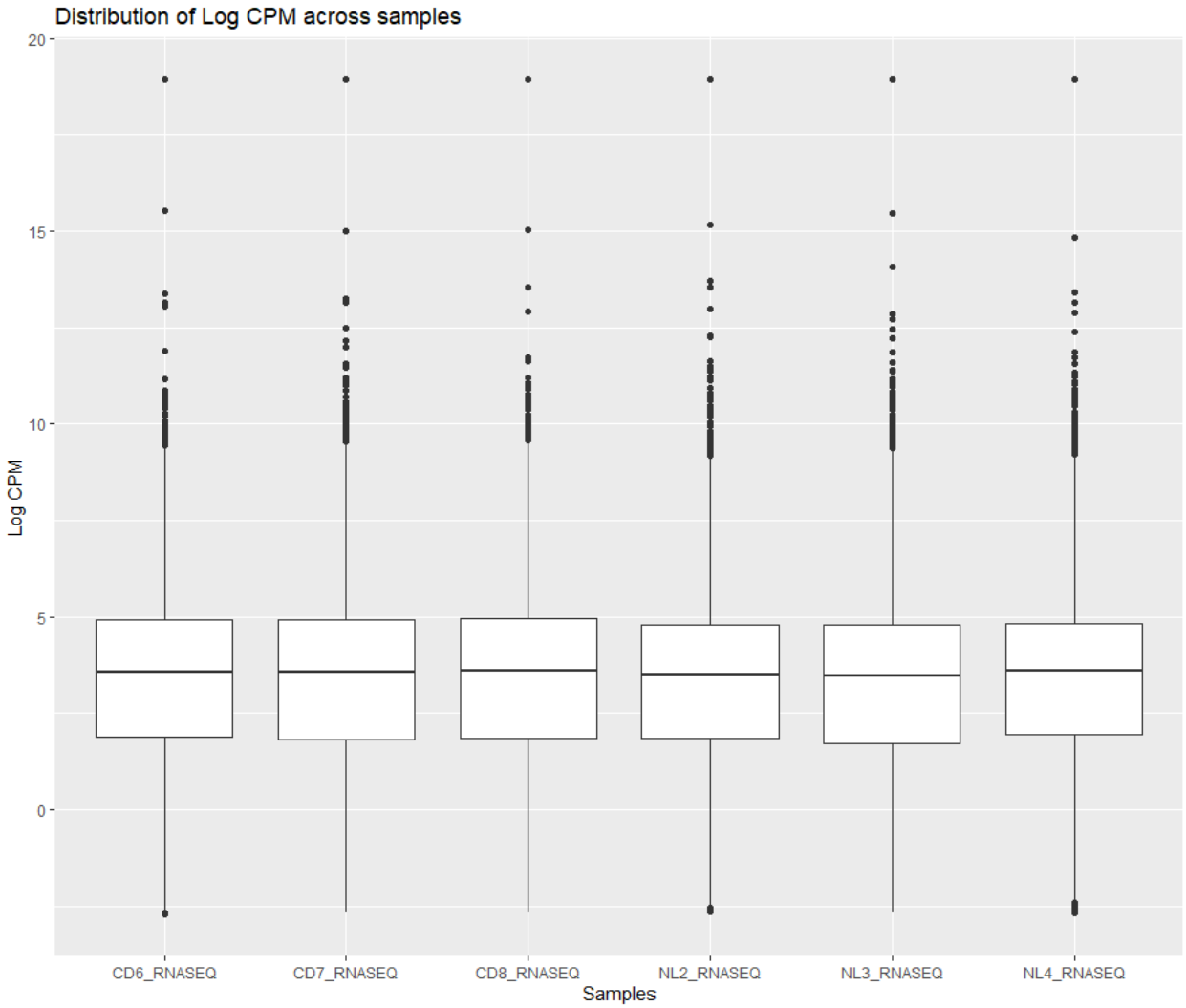
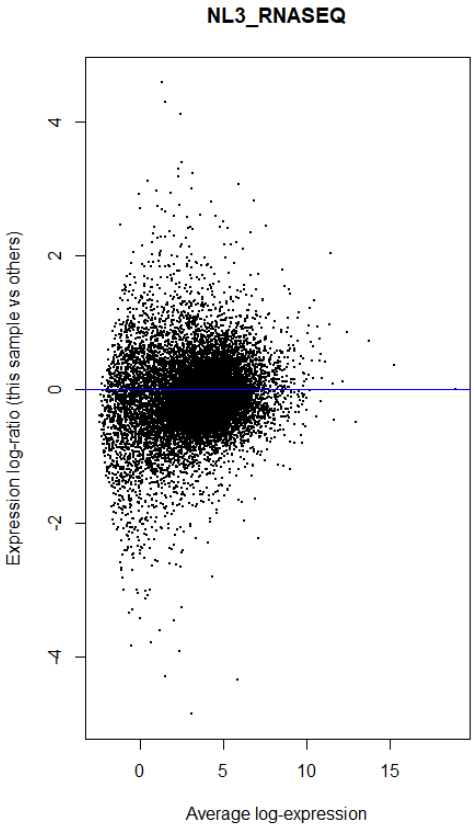
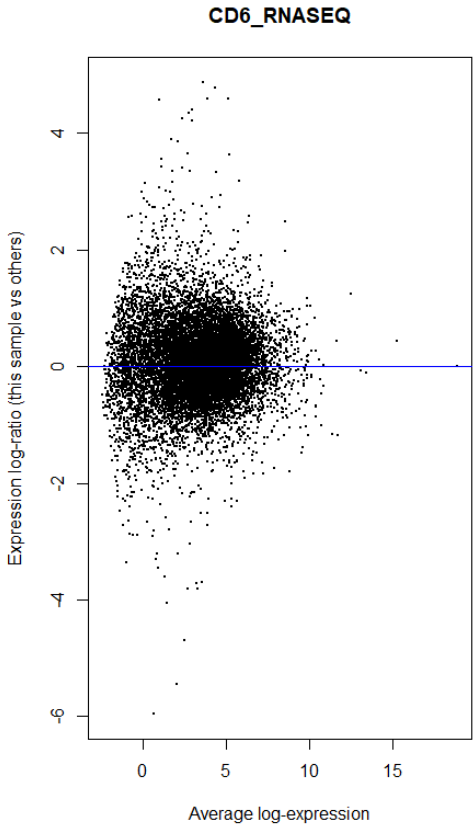
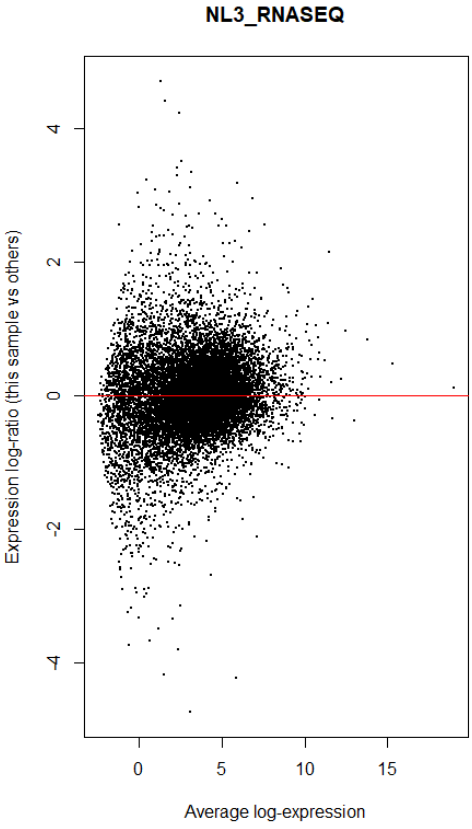


Figure 19. This shows a boxplot of the LogCPM distribution of the genes in each of our six samples. There are no apparent outliers or strange samples.


NL3_RNASEQ
Expression log-ratio (this sample vs others)
Average log-expression
0
5
10
15
-4
-2
0
2
4


CD6_RNASEQ
Expression log-ratio (this sample vs others)
Average log-expression
0
5
10
15
-6
-4
-2
0
2
4


NL3_RNASEQ
Expression log-ratio (this sample vs others)
Average log-expression
0
5
10
15
-4
-2
0
2
4

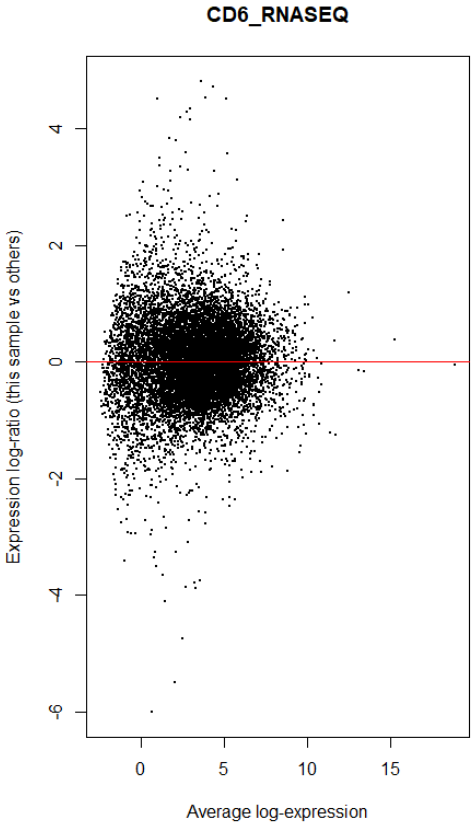

CD6_RNASEQ
Expression log-ratio (this sample vs others)
Average log-expression
0
5
10
15
-6
-4
-2
0
2
4

Figure 20. Before (blue line) and after (red line) abline plots showing the effects of our normalization on the gene data. There is a very slight shift towards an expression log-ratio of zero after normalization that is difficult to see with the naked eye but is apparent when the images are superimposed.

	NL2_RNASEQ	NL3_RNASEQ	NL4_RNASEQ	CD6_RNASEQ	CD7_RNASEQ	CD8_RNASEQ
ACTG2	9.61434152	6.07677257	11.34096905	6.25382456	6.4419663	7.2386833
ADAMDEC1	-2.01747150	-0.85677587	-0.02725680	5.01590808	2.0273992	5.1157354
ADAMTS5	5.23330555	5.93421187	4.64310055	7.81583619	7.5923128	7.6260436
ADH1B	8.47486588	8.87388017	6.02560878	5.08761611	6.4362427	5.5174213

Showing 1 to 5 of 280 entries, 6 total columns

Figure 21. This data frame contains only 280 entries, each representing a gene that was deemed to be differentially expressed between the two sample conditions according to our criteria.

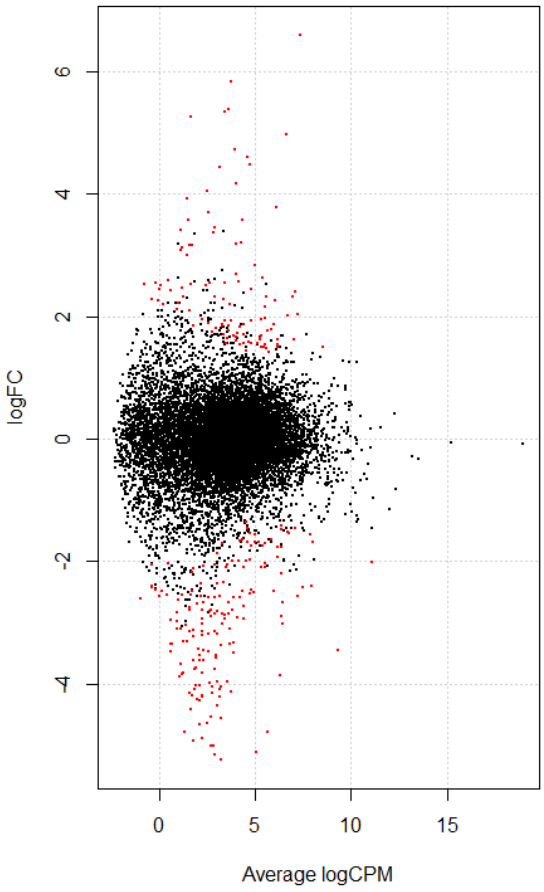


Figure 22. A plot of the logFC vs average logCPM for all of our genes. Genes that were deemed to be differentially expressed according to our criteria are colored red.

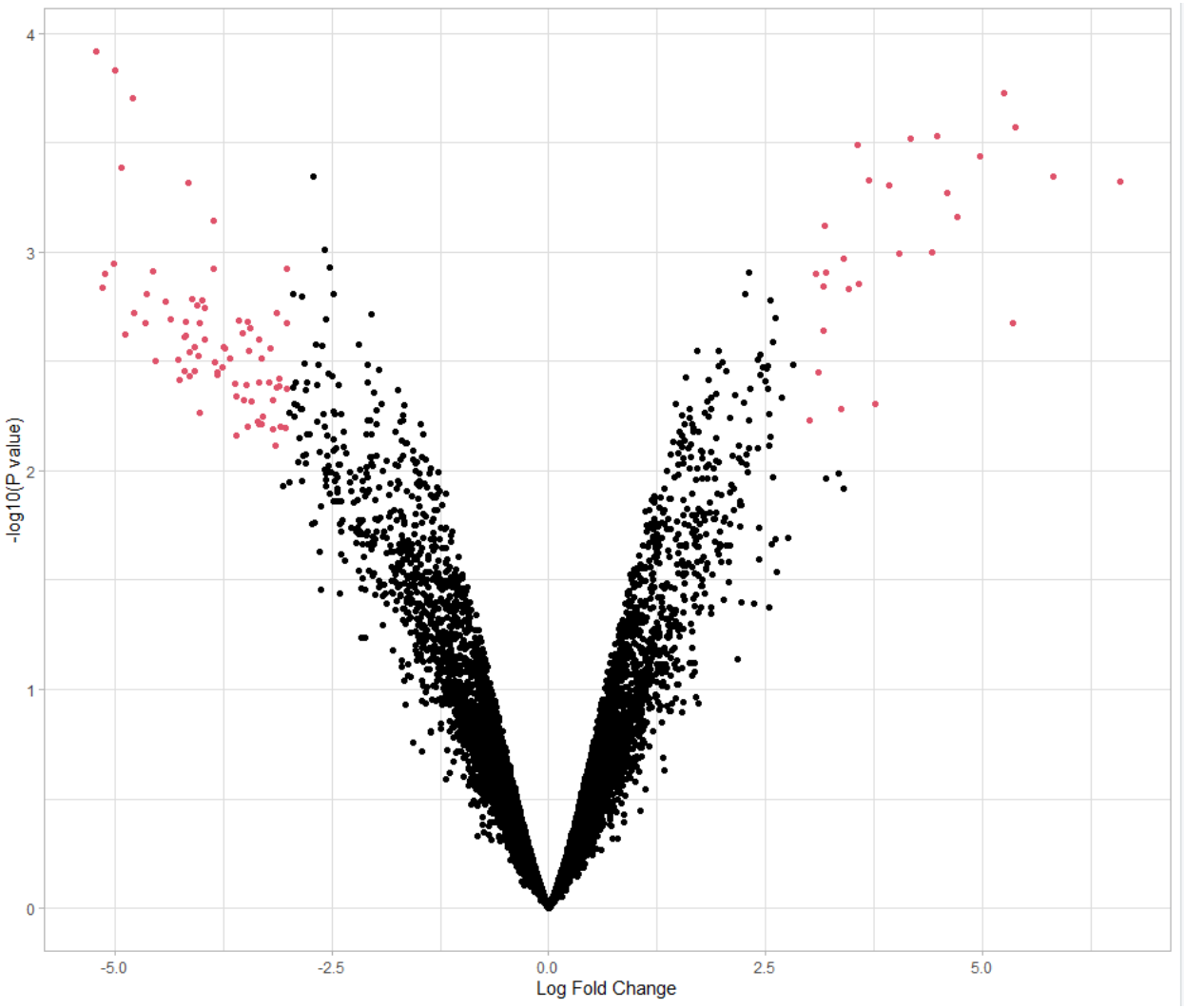


Figure 23. A volcano plot of the LogFC vs -log(P) of our genes, once again differentially expressed genes are colored red.

	logFC	logCPM	LR	PValue	FDR	negLogPValue	Significant	SYMBOL	GENENAME	ENTREZID
SORBS1	-5.226772	3.2464672	14.785720	0.0001204441	0.4017235	3.919214	TRUE	SORBS1	sorbin and SH3 domain containing 1	10580
NAP1L2	-5.003377	2.6769732	14.415009	0.0001466290	0.4017235	3.833780	TRUE	NAP1L2	nucleosome assembly protein 1 like 2	4674
PREX2	5.255626	1.6121478	13.940591	0.0001886796	0.4017235	3.724275	TRUE	PREX2	phosphatidylinositol-3,4,5-trisphosphate dependent Rac exc...	80243
AOC3	-4.794069	5.6439421	13.840086	0.0001990441	0.4017235	3.701051	TRUE	AOC3	amine oxidase copper containing 3	8639

Showing 1 to 5 of 13,024 entries, 10 total columns

Figure 24. Our annotation results data frame, showing how all of our genes were annotated with the "org.Hs.eg.db" library, and genes that were considered significant were given values of TRUE in their "Significant" column.

	logFC	logCPM	LR	PValue	FDR
SORBS1	-5.226772	3.246467	14.785720	0.0001204441	0.4017235
MYBL2	-5.150893	2.909311	10.130840	0.0014580896	0.4189227
RRM2	-5.121245	5.049248	10.409494	0.0012536910	0.4189227
UBE2C	-5.014808	2.852056	10.601670	0.0011298554	0.4189227
NAP1L2	-5.003377	2.676973	14.415009	0.0001466290	0.4017235
PLXNA4	-4.927075	1.775090	12.488774	0.0004094047	0.4017235
GINS2	-4.887939	2.242738	9.224622	0.0023878193	0.4189227
AOC3	-4.794069	5.643942	13.840086	0.0001990441	0.4017235
NUF2	-4.786443	1.317262	9.656125	0.0018872130	0.4189227
ASF1B	-4.651455	2.118071	9.439162	0.0021240054	0.4189227
HJURP	-4.632497	2.493441	10.000460	0.0015650116	0.4189227
BIRC5	-4.567747	3.247883	10.462004	0.0012185504	0.4189227
KRTAP1-5	-4.533521	2.695092	8.720177	0.0031470746	0.4189227
KIF18B	-4.415408	1.659389	9.853259	0.0016953119	0.4189227
KIFC1	-4.354907	3.007073	9.512214	0.0020410877	0.4189227
CENPM	-4.274879	2.089486	8.738533	0.0031155514	0.4189227
AURKB	-4.252549	1.953041	8.366522	0.0038219637	0.4189227
CDT1	-4.204535	2.239557	8.522944	0.0035069676	0.4189227
CDC6	-4.203095	3.176199	9.177517	0.0024500615	0.4189227
FAM64A	-4.189840	1.709507	9.203834	0.0024150878	0.4189227

Figure 25. Our top 20 downregulated genes (according to their logFC values).

	logFC	logCPM	LR	PValue	FDR
DSG2	3.397706	3.328712	6.304076	0.0120460668	0.4797797
PIK3AP1	3.402154	1.130226	10.703263	0.0010694673	0.4189227
LAMA1	3.457142	2.860788	10.103839	0.0014796080	0.4189227
TSPAN11	3.564714	4.322172	12.918680	0.0003253192	0.4017235
PDGFD	3.580641	1.515251	10.205258	0.0014004078	0.4189227
RAPGEF5	3.687116	2.556067	12.243417	0.0004669024	0.4017235
ITIH5	3.769839	6.128548	7.889033	0.0049735460	0.4331267
TLR4	3.918870	1.428613	12.140000	0.0004935178	0.4017235
RNF144A	4.045527	2.522488	10.793151	0.0010187631	0.4189227
CADM1	4.177654	3.976803	13.063154	0.0003011619	0.4017235
KIAA0040	4.425297	3.129402	10.829656	0.0009988719	0.4189227
COLEC10	4.481557	4.706223	13.107096	0.0002941794	0.4017235
ITGA8	4.599264	4.619840	11.983482	0.0005367421	0.4112076
RELN	4.716328	3.968389	11.502082	0.0006951827	0.4189227
CXCL14	4.968512	6.649858	12.715032	0.0003627281	0.4017235
PREX2	5.255626	1.612148	13.940591	0.0001886796	0.4017235
GNAO1	5.346637	3.405181	9.451137	0.0021101829	0.4189227
ADAMDEC1	5.379509	3.601463	13.286194	0.0002673677	0.4017235
CPXM2	5.825887	3.739676	12.302972	0.0004522376	0.4017235
CHI3L1	6.585584	7.322533	12.219874	0.0004728312	0.4017235

Figure 26. Our top 20 upregulated genes (according to their logFC values).

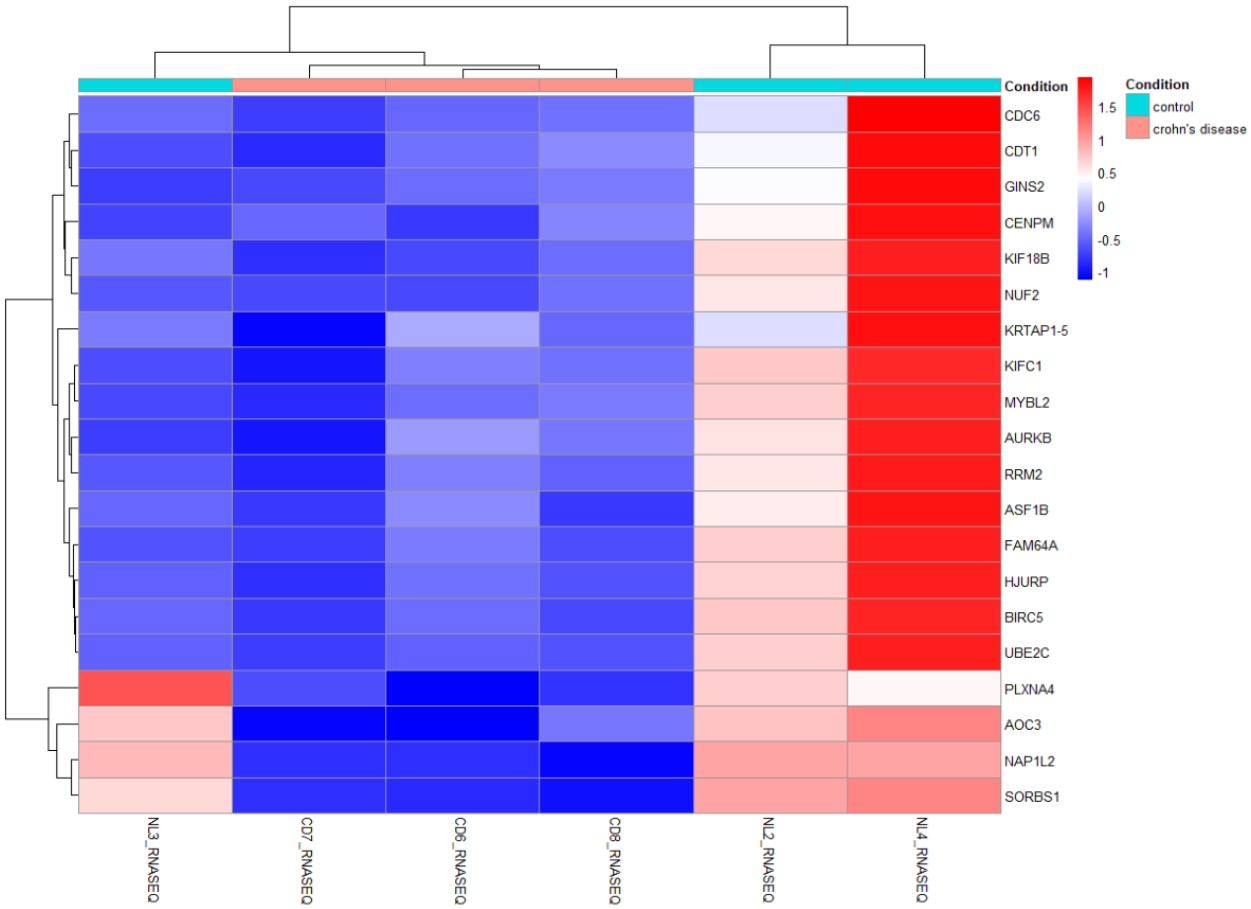


Figure 27. Our heatmap and hierarchical clustering dendrogram showing our top 20 most downregulated genes in CD fibroblasts (samples starting with “CD”), and their relative expression levels. Blue is more downregulated, red means more upregulated.

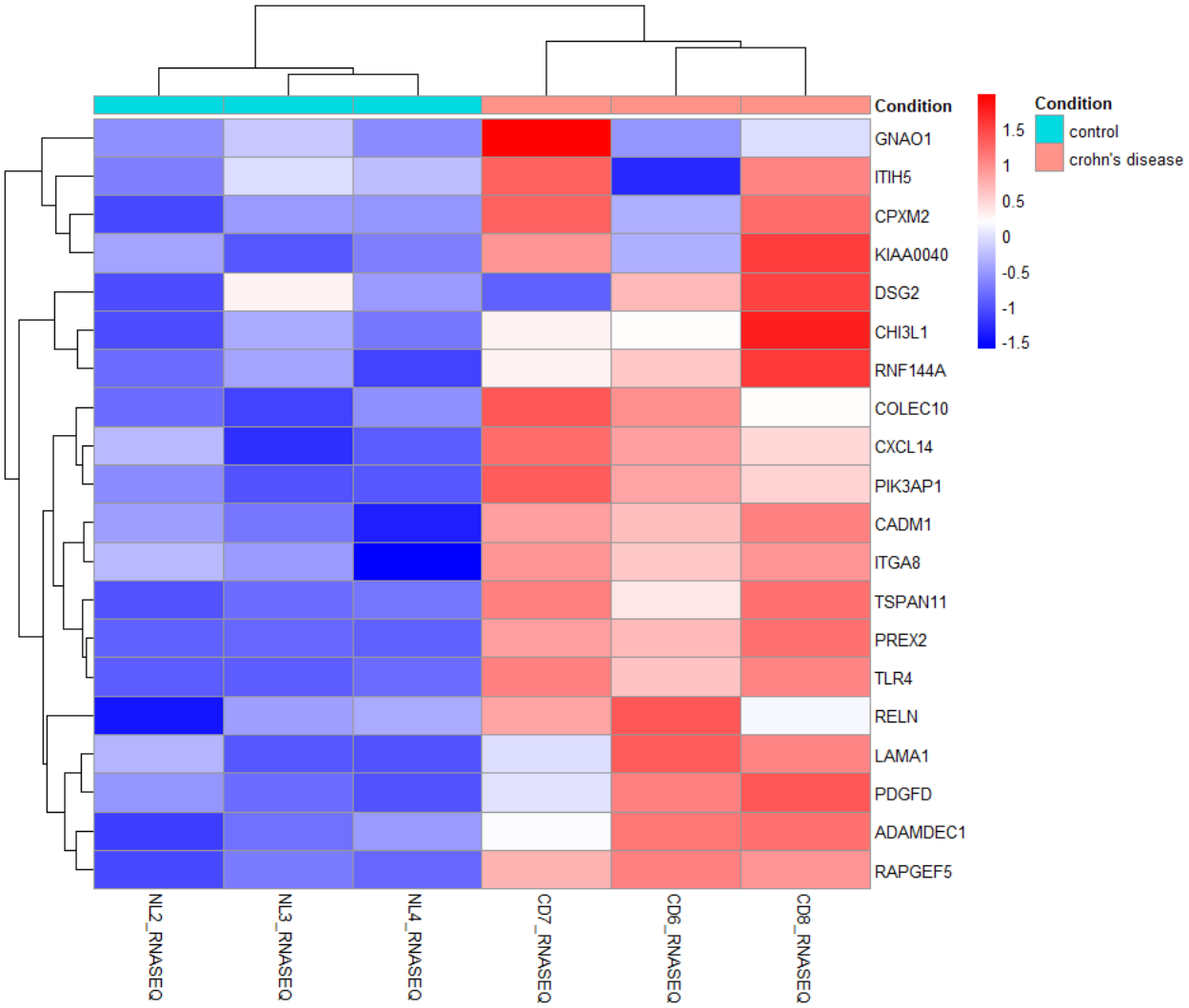


Figure 28. Our heatmap and hierarchical clustering dendrogram showing our top 20 most upregulated genes in CD fibroblasts (samples starting with "CD"), and their relative expression levels. Blue is more downregulated, red means more upregulated.

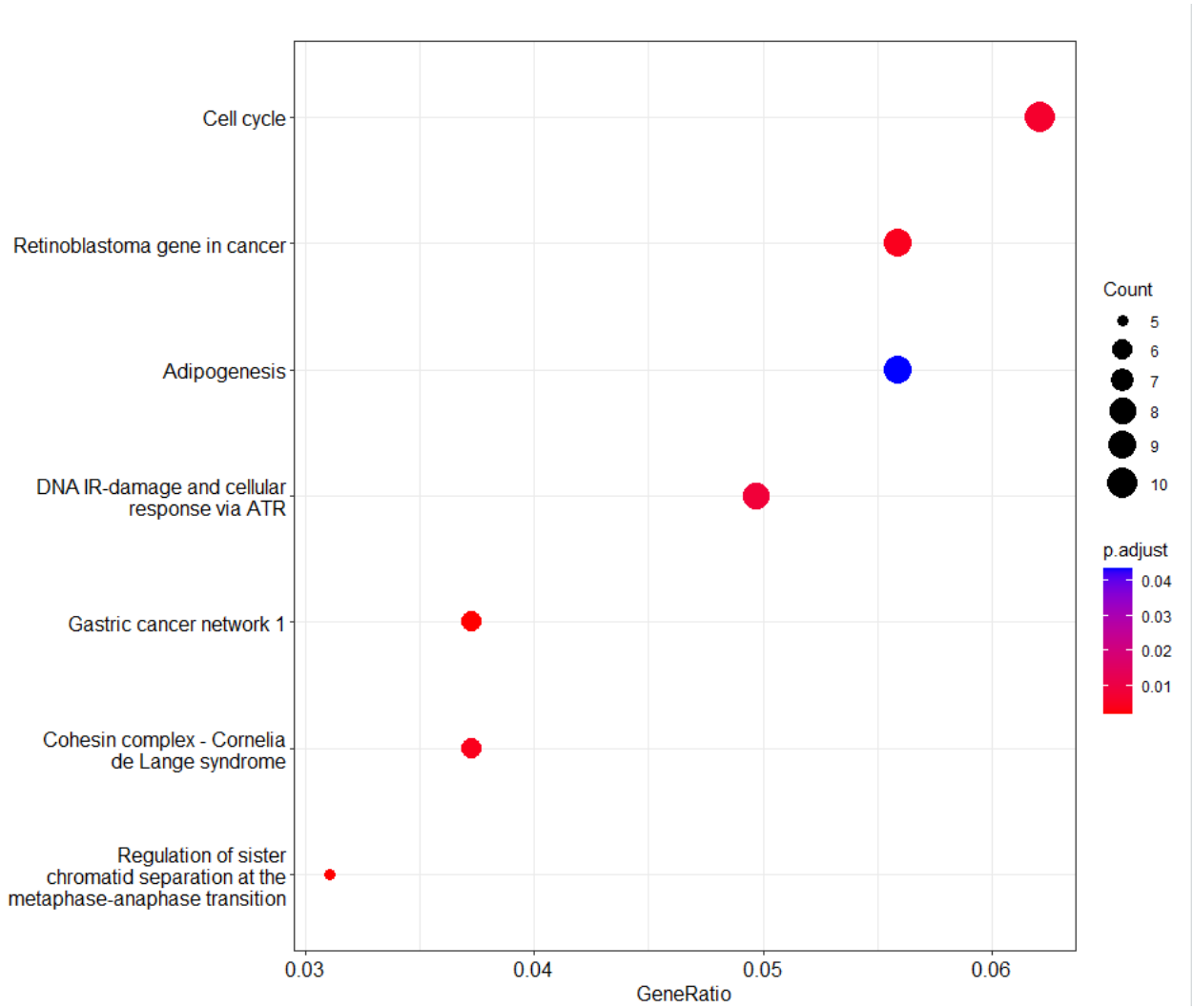


Figure 29. Dotplot showing the results of our wikiPathways enrichment analysis, and the seven most enriched pathways in CD colon fibroblasts.

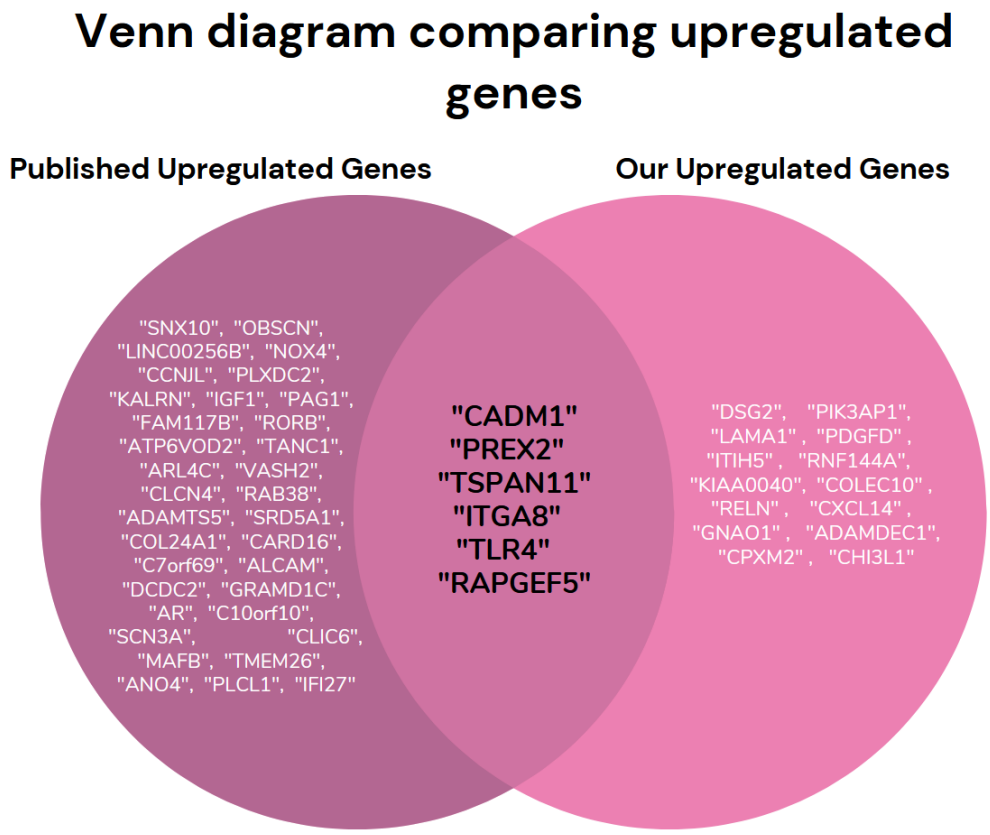


Figure 30. A Venn diagram of common upregulated genes between our generated heatmaps and those found in the original publication (Sadler, et al., 2016)

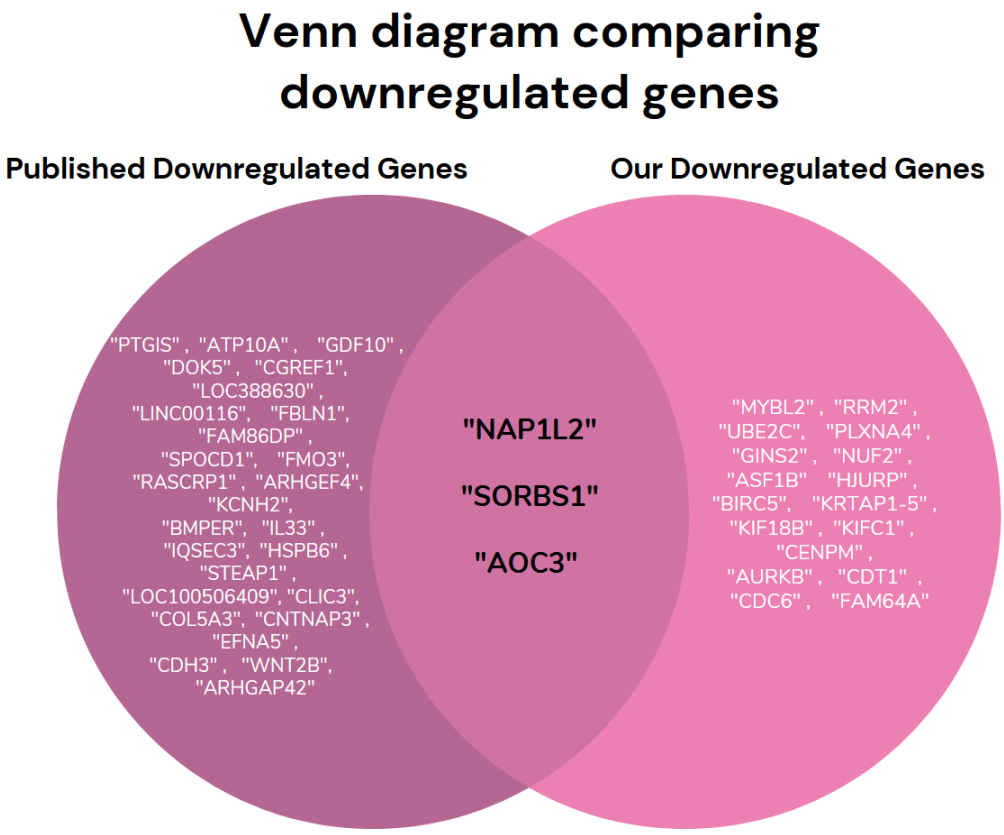


Figure 31. A Venn diagram of common downregulated genes between our generated heatmaps and those found in the original publication (Sadler, et al., 2016)

G. References

Ahmad, B., Serpell, C. J., Fong, I. L., & Wong, E. H. (2020). Molecular mechanisms of adipogenesis: The anti-adipogenic role of AMP-activated protein kinase. *Frontiers in Molecular Biosciences, 7*. https://doi.org/10.3389/fmolb.2020.00076

Cosnes, J., Cattan, S., Blain, A., Beaugerie, L., Carbonnel, F., Parc, R., & Gendre, J.-P. (2002). Long-term evolution of disease behavior of crohn's disease. *Inflammatory Bowel Diseases, 8*(4), 244–250. https://doi.org/10.1097/00054725-200207000-00002

Danek, A., Deorowicz, S., & Grabowski, S. (2014). Indexes of large genome collections on a PC. *PLoS ONE, 9*(10). https://doi.org/10.1371/journal.pone.0109384

Feuerstein, J. D., & Cheifetz, A. S. (2017). Crohn disease: Epidemiology, diagnosis, and Management. *Mayo Clinic Proceedings, 92*(7), 1088–1103. https://doi.org/10.1016/j.mayocp.2017.04.010

Sadler, T., Bhasin, J. M., Xu, Y., Barnholz-Sloan, J., Chen, Y., Ting, A. H., & Stylianou, E. (2016). Genome-wide analysis of DNA methylation and gene expression defines molecular characteristics of crohn's disease-associated fibrosis. *Clinical Epigenetics, 8*(1). https://doi.org/10.1186/s13148-016-0193-6

Torres, J., Mehandru, S., Colombel, J.-F., & Peyrin-Biroulet, L. (2017). Crohn's disease. *The Lancet, 389*(10080), 1741–1755. https://doi.org/10.1016/s0140-6736(16)31711-1