Valeria Duran

Math 6388

April 23, 2020

Final Project

Introduction

Tuberculosis meningitis (TBM) is a type of meningitis that is characterized by the inflammation of the membranes around the brain or spinal cord. Tuberculosis meningitis is caused by a bacterium known as *Mycobacterium tuberculosis (M. tuberculosis)*, where the disorder develops gradually. Its treatment consists of antibiotics and other drugs and is usually effective against the infection. However, untreated, TBM can lead to seizures, hydrocephalus, which is the accumulation of fluid in the brain activity, deafness, mental retardation, paralysis of one side of the body (called hemiparesis) and other neurological abnormalities. Diagnosis can be made via the examination of the cerebrospinal fluid. Tuberculosis meningitis is typically found in children aged one to five years old, however it can occur at any age. In developing countries, childhood tuberculosis is a major public health problem with TBM being a serious complication with high mortality and morbidity (Israni, et al. 2016).

Methods

The BioProject, *Tuberculosis Meningitis in Pediatric Patients*, will be used to explore the differences in expression profiling between TBM pediatric cases and non-infection healthy controls. For the experiment, whole blood transcriptional profiles in children with or without tuberculosis meningitis were compared using RNA-Seq and a biomarker signature driven by inflammasome activation and activity was identified. A total of 43 whole blood samples were sequenced, which include 15 tuberculosis meningitis cases, 4 non-TBM cases, and 24 non-infection healthy controls. RNA sequencing is a technology-based sequencing technique that uses next-generation sequencing to show the presence and quantity of RNA in a sample at a given moment. RNA-Seq facilitates the ability to look at differences in gene expression in different groups or treatments (Blog 2015).

Data Preparation

The data was retrieved from the SRA. The SRA stores data from high-throughput sequencing projects. The data can be viewed in the BioProject site under PRJNA437114, and the RNA sequencing data from the project can be tracked via the GEO accession GSE111459. The data for the experiment was collected using an Illumina HiSeq 4000. The data type is transcriptomic paired-end cDNA reads.

RNA-Seq was conducted on whole blood samples. The sample data are stored as fastq files. Since there a total of 43 samples consisting of classes TBM, non-TBM, and non-infection healthy control, the non-TB cases were dropped in the consideration process, resulting in 39 samples to choose from. For easier data analyzation purposes, only 6 total samples will be used for comparison. Because of this, 3 samples chosen consisted of TBM samples and 3 samples consisted of non-infection healthy control. These samples were chosen based on the samples containing the smallest reads sequenced. With this factor in mind, the samples selected were SRR6809855 (base size 5.77 G), SRR6809856 (base size 6.29 G), SRR6809862 (base size 5.67 G) for pediatric tuberculosis meningitis, and SRR6809883 (base size 6.71 G), SRR6809886 (base size 8.52 G), and SRR6809890 (base size 9.15 G) for pediatric non-infected healthy control.

The data was downloaded via the fastq files using the SRA toolkit in the Unix command line using the following batch job:

```
# Use fastq-dump, from the sratoolkit, to download SRA files
fastq-dump --split-files --gzip SRR6809855
fastq-dump --split-files --gzip SRR6809856
fastq-dump --split-files --gzip SRR6809862
fastq-dump --split-files --gzip SRR6809883
fastq-dump --split-files --gzip SRR6809886
fastq-dump --split-files --gzip SRR6809890
```

Figure 1. Batch job for SRA toolkit.

The samples took approximately 24 hours to completely download. The output of each SRA fastq file was a forward run `SRR6809855_1.fastq.gz` and reverse run `SRR6809855_2.fastq.gz` (example given for sample SRR6809855). Once downloaded into the directory, a kallisto template was used to run kallisto on each. The batch file was edited using vi as follows:

```
#!/bin/bash
#SBATCH -J GSE111459               # Name of the job
#SBATCH -N 1                   # Number of nodes
#SBATCH -n 1                   # Number of cores (processors) per node
#SBATCH -t 24:00:00            # Runtime in HH:MM:SS
#SBATCH --mem=4G               # Memory requested in MB (see also --
mem-per-cpu)
#SBATCH -o GSE37704_%j.out     # File to write STDOUT, %j=jobid
#SBATCH -e GSE37704_%j.err     # File to wrote STDERR, %j=jobid
#SBATCH --mail-type=ALL            # Send email when job starts, ends,
fails, etc
#SBATCH --mail-user=vduran4@uh.edu

#Load the "kallisto" module
module load kallisto

# Move to your directory
cd /project/meisel/vduran4/FinalProject

# Use kallisto to quantify gene expression
kallisto quant -i Homo_sapiens.GRCh38.cdna.all.release-94_k21.idx -o
kallisto_sample1/SRR6809890_GRCh38 SRR6809890_1.fastq.gz
SRR6809890_2.fastq.gz
```

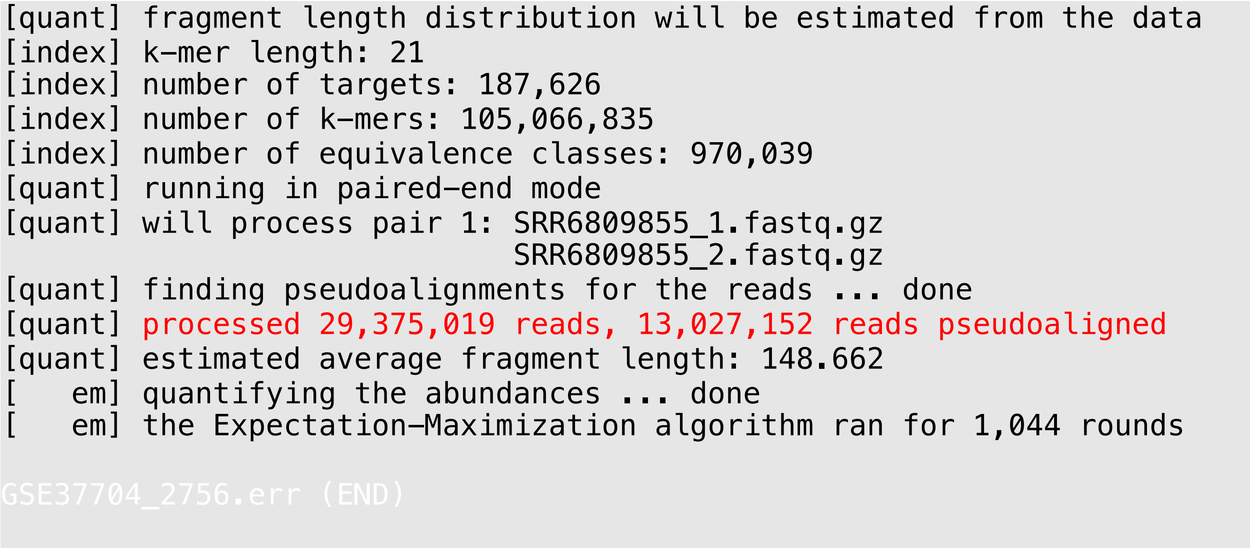
Figure 2. Kallisto batch file for sample SRR6809890.

The job was adjusted for each sample and ran using `sbatch kallisto_final_project.sh.` Kallisto is run using a quantification algorithm. The reference genome was GRC38 and is described as the Genome Reference Consortium Human Build 38. The default *k*-mer size for kallisto is a maximum value of 31, however, for runs used, a *k*-mer size of 21 was used. The *k*-mer size of the reference genome must be less than the length of the RNA-seq reads. The RNA-seq reads in the samples were approximately 100 nucleotides. Each batch job took approximately 40 minutes to process. Each batch job outputted a directory called "SRRXXXXXXX_GRCh38." These directories contained 3 files: an abundance.h5 file (a binary file viewable by some gene expression analysis software), an abundance.tsv file (a table of the length, "effective length," "estimated counts," and TPM for every transcript in the annotation, and finally a run_info.json file which shows information about the kallisto run.

For the focus of the data analysis, the files used will be the abundance.tsv files which will be imported into R and analyzed with the package DESeq2.

Results

Before any analysis done in R, the proportion of reads aligned was checked for each sample. A snippet of the GSE37704_XXXX.err file is shown below in order to help determine the proportion of reads aligned for sample SRR6809855:



```
[quant] fragment length distribution will be estimated from the data
[index] k-mer length: 21
[index] number of targets: 187,626
[index] number of k-mers: 105,066,835
[index] number of equivalence classes: 970,039
[quant] running in paired-end mode
[quant] will process pair 1: SRR6809855_1.fastq.gz
                             SRR6809855_2.fastq.gz
[quant] finding pseudoalignments for the reads ... done
[quant] processed 29,375,019 reads, 13,027,152 reads pseudoaligned
[quant] estimated average fragment length: 148.662
[   em] quantifying the abundances ... done
[   em] the Expectation-Maximization algorithm ran for 1,044 rounds

GSE37704_2756.err (END)
```

Figure 3. GSE37704_XXXX.err file output showing the reads and reads pseudoaligned for sample SRR6809856.

The text in red indicates the total number of reads as well as the pseudoaligned reads. For this specific sample, proportion read was 44.35%. For sample SRR6809856, the proportion read was 42.13%; sample SRR6809862 was 44.97%; sample SRR6809883 was 44.98%; sample

SRR6809886 was 40.57%; sample SRR6809890 was 39.11%. This shows a very low percentage of reads pseudoaligned, which could be due to multiple factors such as using a proper reference transcriptome, the way the libraries were prepared, etc.

Since each sample comes from a different individual, the libraries were plotted against each other using their transcripts per million in order to verify this fact. Below is an example of a plot between the libraries SRR6809855 and SRR6809856, two pediatric TBM cases, as well as another plot between the libraries SRR6809855 and SRR6809883, one pediatric TBM case and one non-infection healthy control case.

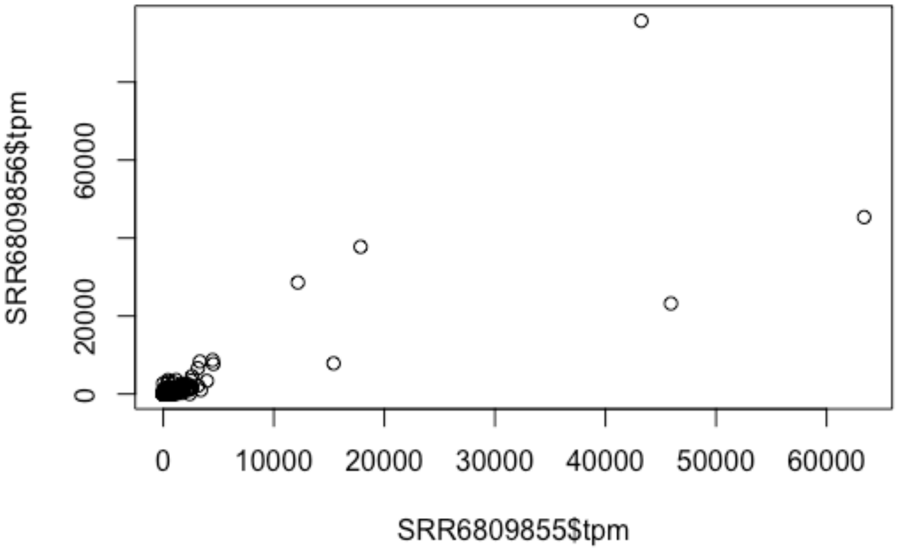


Figure 4. Plot showing tpm for SRR6809856 vs SRR6809883.

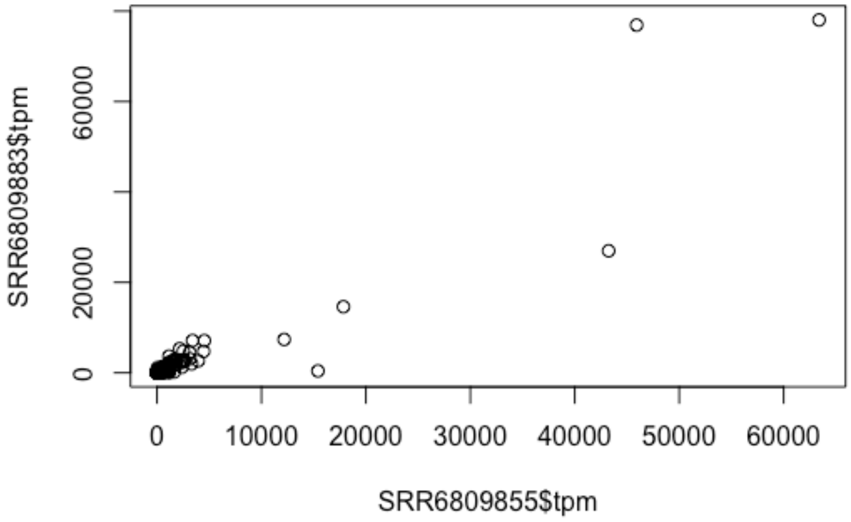
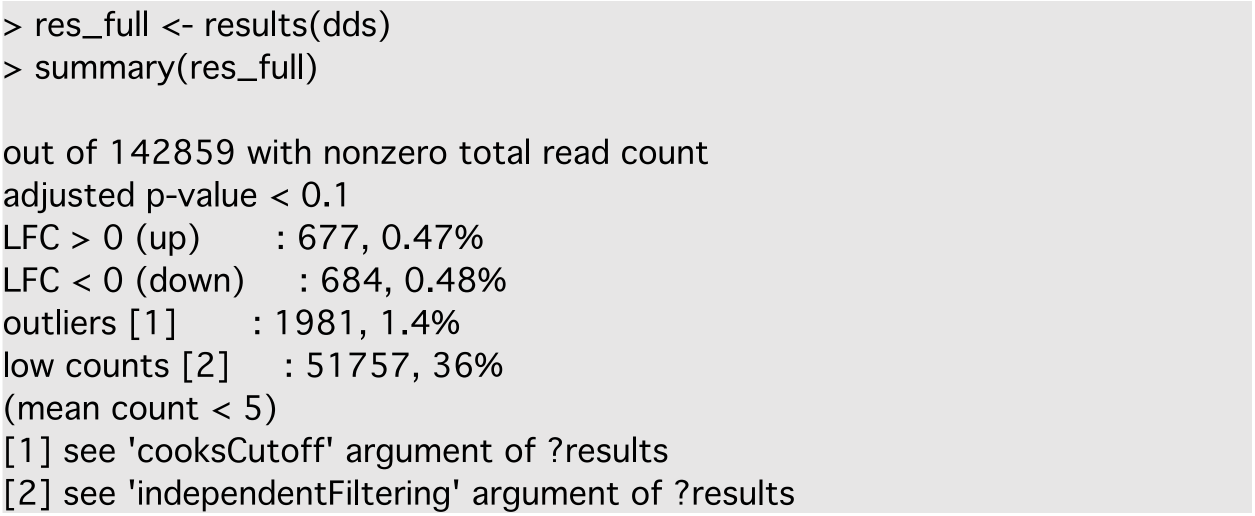


Figure 5. Plot showing tpm for SRR6809856 vs SRR6809883.

These plots indicate that the libraries are different, which is expected since they were not prepared from the same RNA sample. Following this, the data was analyzed using DESeq2. The following output in R shows the summary of the results table from the unfiltered data differential expression analysis conducted:

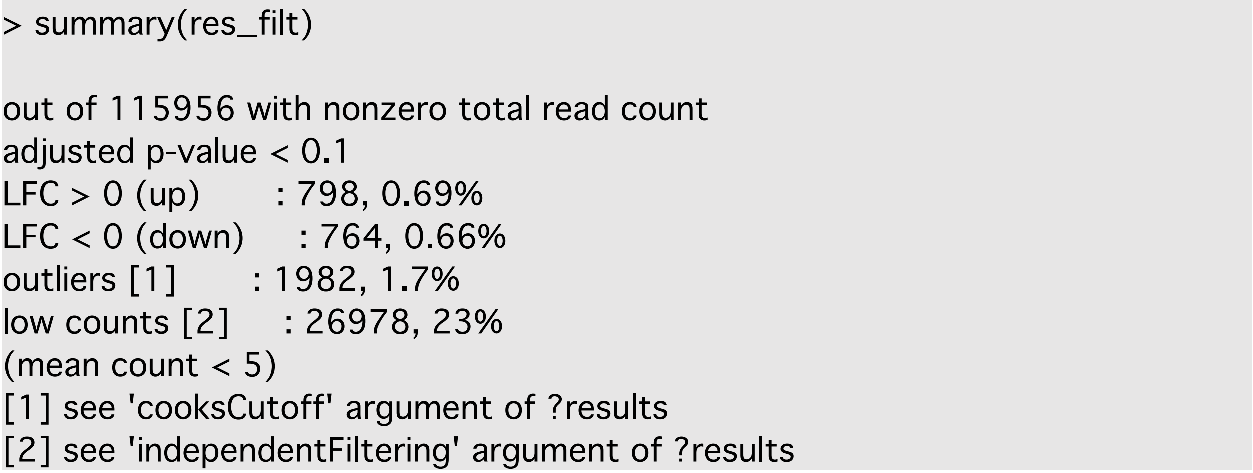


```
> res_full <- results(dds)
> summary(res_full)

out of 142859 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)       : 677, 0.47%
LFC < 0 (down)     : 684, 0.48%
outliers [1]       : 1981, 1.4%
low counts [2]     : 51757, 36%
(mean count < 5)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

Figure 6. Summary of results table for unfiltered differential expression analysis.

In the unfiltered data, there are 145,859 genes with nonzero read count. For this data, an LFC > 0 signifies expression is higher for “pred_TBM”, in other words the pediatric tuberculosis meningitis cases, while an LFC < 0 signifies expression is higher for “HC,” or non-infected healthy control cases. For the unfiltered data, 677 genes are expressed higher in the pediatric TBM cases, and 684 genes are expressed higher in the healthy control cases. The next following output in R shows the summary of the results table from the filtered (keep rows that have at least 10 reads total) differential expression analysis conducted:



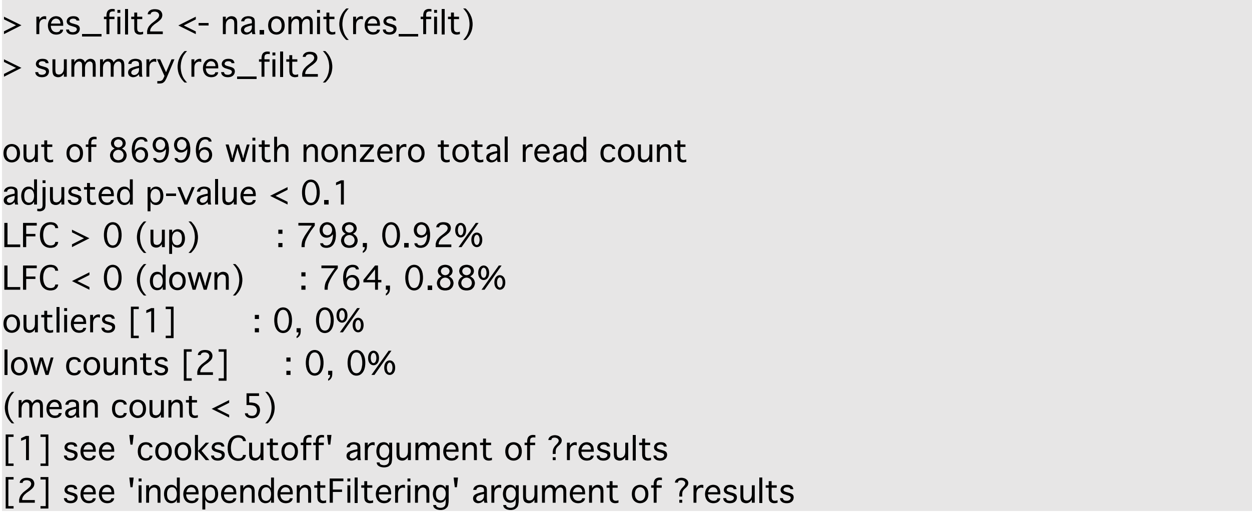
```
> summary(res_filt)

out of 115956 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)       : 798, 0.69%
LFC < 0 (down)     : 764, 0.66%
outliers [1]       : 1982, 1.7%
low counts [2]     : 26978, 23%
(mean count < 5)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

Figure 7. Summary of results table for filtered differential expression analysis.

In the filtered data, there are 115,956 genes with nonzero read count. For the filtered data, 798 genes are expressed higher in the pediatric TBM cases, and 764 genes are expressed higher in the healthy control cases. This signifies a difference of 201 genes between the filtered and unfiltered data, and a 0.4% difference.

Afterwards, any genes containing NA values were further filtered out for analysis; the following summary shows the results table for the NA filtered data using “res_filt.”



```
> res_filt2 <- na.omit(res_filt)
> summary(res_filt2)

out of 86996 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)       : 798, 0.92%
LFC < 0 (down)     : 764, 0.88%
outliers [1]       : 0, 0%
low counts [2]     : 0, 0%
(mean count < 5)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

Figure 8. Summary of results table for filtered data excluding NA values.

After reordering the results object based on LFC, the gene with the highest LFC of 21.0383867114077 is gene ENST00000452392.2. The FDR p-adjusted value is 4.33893634473689e-05. On the other hand, the gene with the lowest p-value is gene ENST00000424832.6 with a p-value of 1.82391757390609e-14. The following plot shows the plot counts of gene ENST00000424832.6:

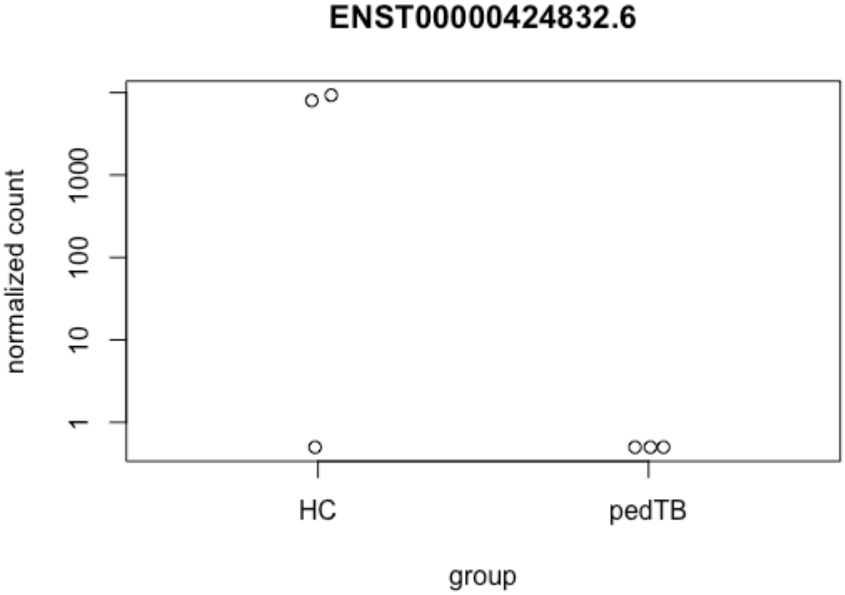


Figure 9. Plot of normalized counts for gene ENST00000424832.6.

The following plot shows the plot counts of gene ENST00000452392.2:

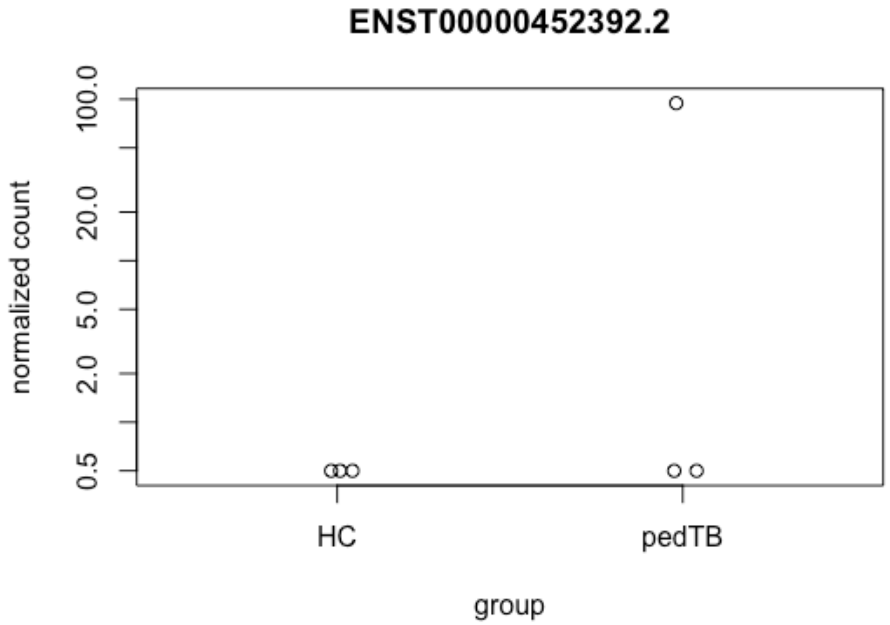


Figure 10. Plot of normalized counts for gene ENST00000452392.2.

When comparing both genes, gene ENST00000424832.6 with the lowest p-value has a higher count for HC, while gene ENST00000452392.2 with the highest LFC has a higher count for pedTB. A summary of the results table also verifies this:

```
> summary(high_lfc)

out of 1 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)       : 1, 100%
LFC < 0 (down)     : 0, 0%
outliers [1]       : 0, 0%
low counts [2]     : 0, 0%
(mean count < 5)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results

> summary(low_p)

out of 1 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)       : 0, 0%
LFC < 0 (down)     : 1, 100%
outliers [1]       : 0, 0%
low counts [2]     : 0, 0%
(mean count < 5)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

Figure 11. Summary tables for gene ENST00000452392.2 (high_lfc) and ENST00000424832.6 (low_p).

The summary table for gene ENST00000452392.2 shows an LFC > 0:1, signifying that the pediatric TBM shows higher expression; contrarily, the summary table for gene ENST00000424832.6 shows an LFC < 0:1, signifying that the healthy control shows higher expression. Shrinkage of effect size (LFC estimates) was conducted to visualize and rank genes. The plots for the correlation of unshrunk and shrunk LFC and shrunk LFC against expression are shown below:

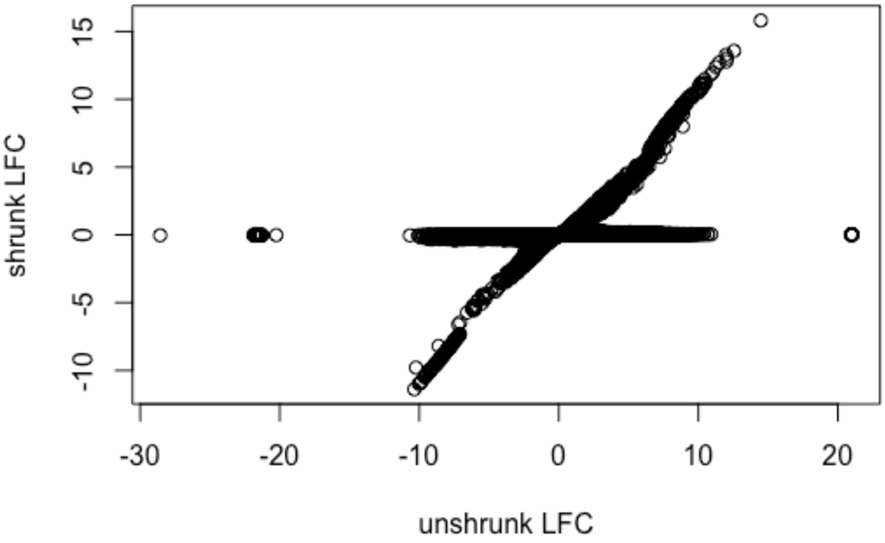


Figure 12. Plot of correlation for unshrunk vs shrunk LFC.

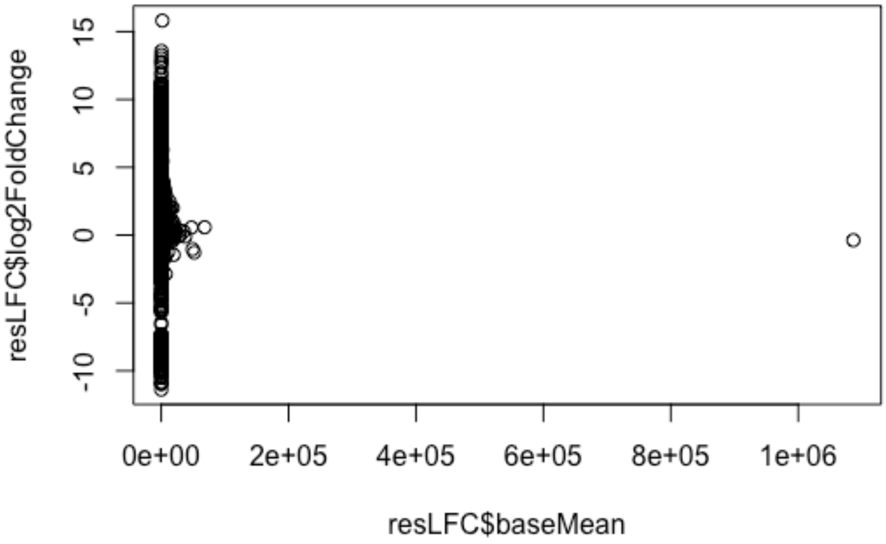


Figure 13. Plot of shrunk LFC against expression.

Figure 13 shows that expression and LFC seem to cluster around the origin (0,0), with a few outliers visible. An MA-plot was used for both the unshrunk LFC and shrunken LFC in order to see differences in reduction of noise associated with $\log_2$ fold changes.

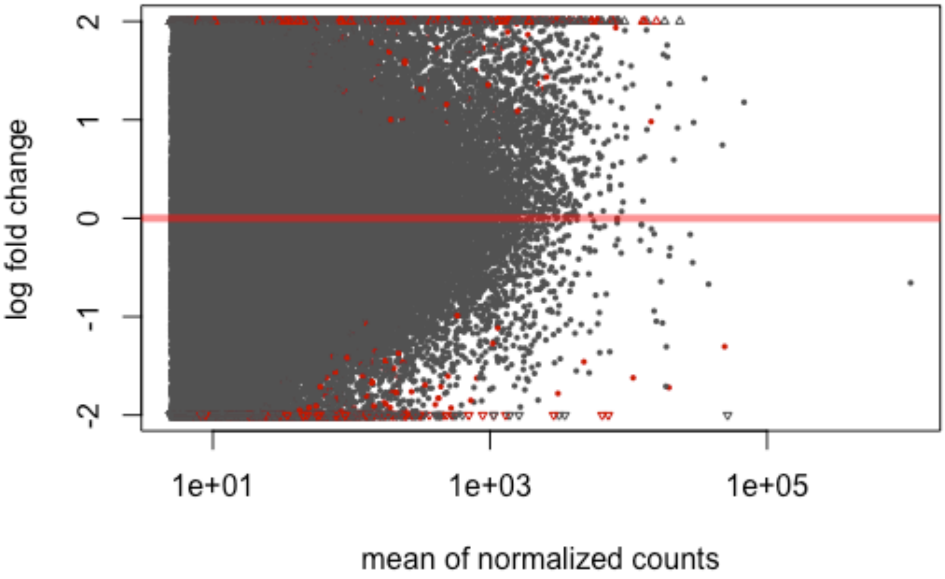


Figure 14. MA-plot for unshrunk LFC.

The above MA-plot shows a substantial amount of noise. Majority of the genes are not statistical significant (grey colored). Genes that are statistical significant (red colored) appear more densely packed towards the top and bottom edge of the plot, signifying that majority of those genes will have a $\log_2$ fold change greater than 2 or less than -2.

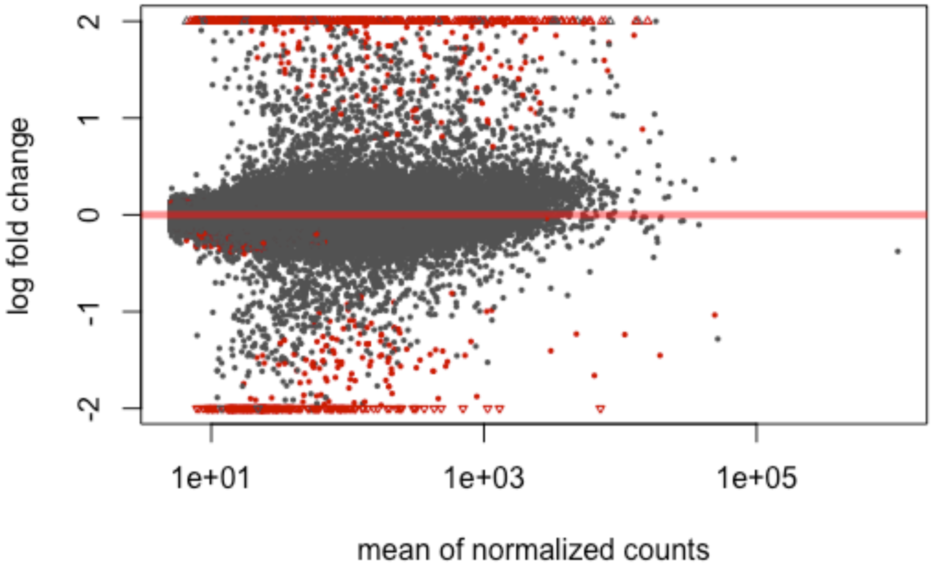


Figure 15. MA-plot of shrunk LFC.

Figure 15 shows the MA-plot with less noise. There are visibly more genes that are statistical significant above 1 or less than -1. Still, majority of the genes seem to be above a LFC of 2 and below -2. The count data was then transformed to in order to be used for visualizations and clustering. The data was transformed using a variance stabilizing transformation in order to remove the dependence of the variance on the mean. The following heatmap shows the expression of each gene in each sample using the transformed data:

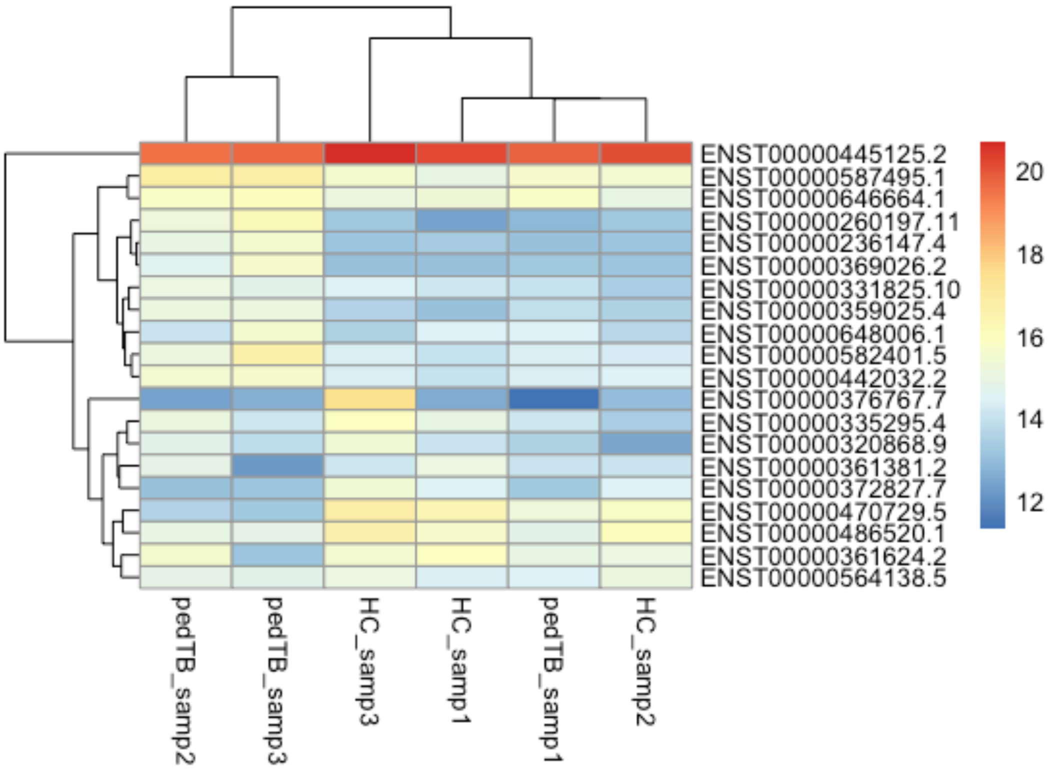


Figure 16. Heatmap of transformed data showing the top 20 genes with highest counts.

The heatmap in figure 16 shows that in terms of samples, there are two large clusters, one consisting of samples pedTB_samp2 and pedTB_samp3 for the first half of the half of the genes, and another cluster consisting of samples HC_samp3, HC_samp1, HC_samp2, and pedTB_samp1 for the second half of the genes. The samples from the pedTB cluster showed higher gene expressions for genes ENST00000260197.11, ENST00000236147.4, ENST00000369026.2, ENST00000331825.10, ENST00000359025.4, ENST00000648006.1, ENST00000582401.5, and ENST00000442032.2. The samples from the majority HC cluster showed higher gene expression for genes ENST00000470729.5, ENST00000486520.1, ENST00000361624.2, and ENST00000564138.5. In terms of sample distances, there is a clear separation as well between clusters.

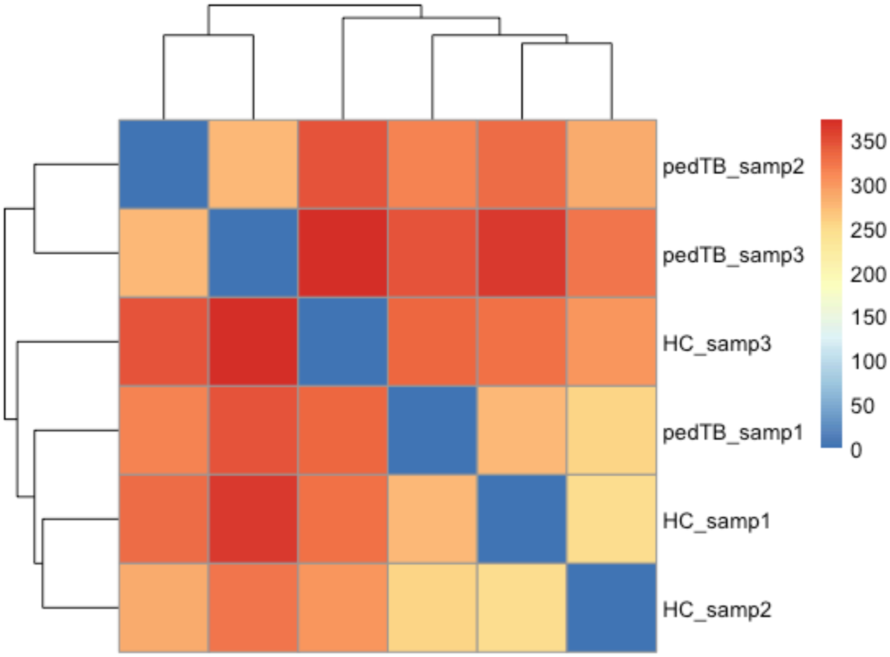


Figure 17. Heatmap of distance matrix between samples.

The above figure shows two groups: one group consisting of samples pedTB_samp2 and pedTB_samp3, and another group consisting of samples HC_samp3, pedTB_samp1, HC_samp1, and HC_samp2. When comparing the two heatmaps, it can be seen that pedTB_samp1 shows higher gene expression for genes also highly expressed in the healthy control samples compared to the other TBM samples which may explain why pedTB_samp1 clustered with the healthy control samples. Apart from this, pedTB_samp2 shows the highest similarity with HC_samp3, pedTB_samp3 shows the highest similarity with HC_samp1 and HC_samp3, HC_samp3 also showed the highest similarity with pedTB_samp3, pedTB_samp1 showed the highest similarity

HC_samp3 and pedTB_samp3, HC_samp1 showed the highest similarity with pedTB_samp3, and HC_samp2 showed the highest similarity wit pedTB_samp3.

Finally, principal component analysis was conducted. A PCA plot was plotted in order to see the first two principal components. The first two principal components explain the most variance across the samples.

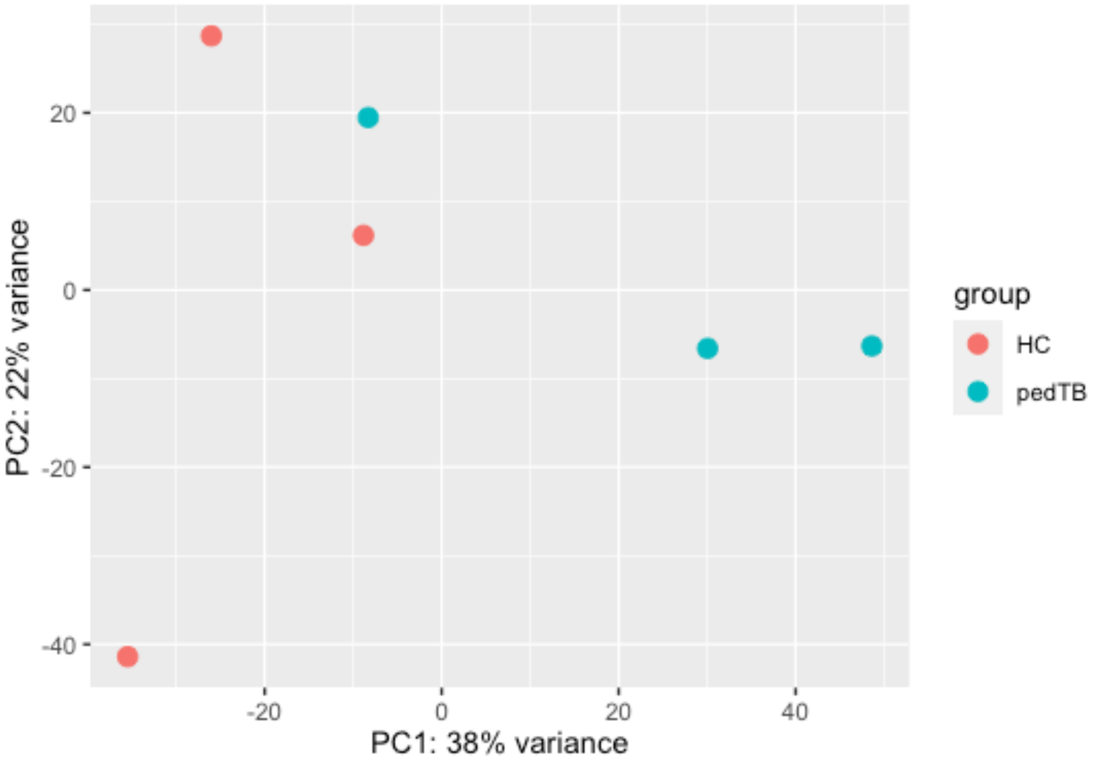


Figure 18. PCA plot of the first to principal component vectors.

The first two principal components explain 60% of the total variance. PC1 separates the groups, showing that the results from the heatmaps previously discussed are valid. PC2 seems to separate the groups as well.

Conclusion

Differential expression between pediatric tuberculosis meningitis cases and non-infected healthy controls were analyzed using kallisto and DESeq2. Pediatric TBM showed higher expression for the gene ENST00000452392.2, the gene with the highest LFC value. Healthy control showed higher expression for the gene ENST00000424832.6, the gene with the lowest p-value. 8 genes showed higher expression for the pedTB samples and clustered together, while 4 genes showed higher expression for the HC samples and clustered together. Between samples, there were two clusters where one group consisted of samples pedTB_samp2 and pedTB_samp3, and another group consisted of samples HC_samp3, pedTB_samp1, HC_samp1, and HC_samp2. PCA analysis showed clear separation between the samples.

References

Blog, RNA-Seq. "Introduction to RNA Sequencing and Analysis." *RNA*, 16 July 2015, www.rna-seqblog.com/introduction-to-rna-sequencing-and-analysis/.

Israni, Anil V., et al. "Tubercular Meningitis in Children: Clinical, Pathological, and Radiological Profile and Factors Associated with Mortality." *Journal of Neurosciences in Rural Practice*, vol. 07, no. 03, 2016, pp. 400–404., doi:10.4103/0976-3147.181475.

"Meningitis, Tuberculous." *NORD (National Organization for Rare Disorders)*, rarediseases.org/rare-diseases/meningitis-tuberculous/.

"Tuberculous Meningitis in Pediatric Patients (ID 437114)." *National Center for Biotechnology Information*, U.S. National Library of Medicine, www.ncbi.nlm.nih.gov/bioproject/PRJNA437114.

"Tuberculous Meningitis." *Background, Pathophysiology, Etiology*, 9 Nov. 2019, emedicine.medscape.com/article/1166190-overview.

Well, Gijs T. J. van, et al. "Twenty Years of Pediatric Tuberculous Meningitis: A Retrospective Cohort Study in the Western Cape of South Africa." *American Academy of Pediatrics*, American Academy of Pediatrics, 1 Jan. 2009, pediatrics.aappublications.org/content/123/1/e1.

Supplementary Information - Code

```
SRR6809855 <- read.delim("SRR6809855_GRCh38.tsv")
SRR6809856 <- read.delim("SRR6809856_GRCh38.tsv")
SRR6809862 <- read.delim("SRR6809862_GRCh38.tsv")
SRR6809883 <- read.delim("SRR6809883_GRCh38.tsv")
SRR6809886 <- read.delim("SRR6809886_GRCh38.tsv")
SRR6809890 <- read.delim("SRR6809890_GRCh38.tsv")

plot(SRR6809855$tpm, SRR6809883$tpm)

head(SRR6809855)
head(subset(SRR6809855, est_counts > 0))

ped <- data.frame(
  pedTB_samp1 = round(SRR6809855$est_counts),
  pedTB_samp2 = round(SRR6809856$est_counts),
  pedTB_samp3 = round(SRR6809862$est_counts),
  HC_samp1 = round(SRR6809883$est_count),
  HC_samp2 = round(SRR6809886$est_count),
  HC_samp3 = round(SRR6809890$est_count)
)

rownames(ped) <- SRR6809855$target_id

head(ped)
head(subset(ped, pedTB_samp1>0))

coldata <- data.frame(
  treatment = c(rep("pedTB",3), rep("HC",3))
)

rownames(coldata) <- colnames(ped)
coldata

library("DESeq2")
dds <- DESeqDataSetFromMatrix(countData = ped,
                              colData = coldata,
                              design = ~treatment)
dds <- DESeq(dds)
res <- results(dds)
res_full <- results(dds)
summary(res_full)

#perform minimum pre-filtering
dds_filt <- dds[rowSums(counts(dds)) >= 10,]
```

```
dds_filtered <- DESeq(dds_filt)
res_filt <- results(dds_filtered)
summary(res_filt)
summary(res)
subset(res, baseMean>0)
res <- na.omit(res)
summary(res)
res
res_filt2 <- na.omit(res_filt)
summary(res_filt2)
#remove NAs

#look at gene with highest LFC
res_ordered <- res_filt2[order(-res_filt2$log2FoldChange),]
high_lfc <- res_ordered[1,]

#look at gene with lowest p-value
low_p <- res_filt2[which.min(res_filt2$padj),]


#look at summary for highlfc gene and lowp gene
summary(high_lfc)
summary(low_p)

plotCounts(dds_filtered, gene = "ENST00000424832.6", intgroup = "treatment")

plotCounts(dds_filtered, gene = "ENST00000452392.2", intgroup = "treatment")

library(apeglm)
resultsNames(dds_filtered)
resLFC <- lfcShrink(dds_filtered,
                    coef="treatment_pedTB_vs_HC",
                    type="apeglm")
resLFC <- na.omit(resLFC)
plot(res_filt2$log2FoldChange, resLFC$log2FoldChange,
     xlab="unshrunk LFC", ylab="shrunk LFC")

plot(resLFC$baseMean, resLFC$log2FoldChange)
#res[order(res$padj),]
#res[order(res$log2FoldChange),]
plotMA(res_filt2, ylim=c(-2,2))
plotMA(resLFC, ylim=c(-2,2))

plotCounts(dds, gene=which.min(res$padj), intgroup="treatment")

dev.off()
```

```
d <- plotCounts(dds, gene=which.min(res$padj), intgroup="treatment", returnData=TRUE)
library("ggplot2")
ggplot(d, aes(x=treatment, y=count)) +
  geom_point(position=position_jitter(w=0.1,h=0)) + scale_y_log10(breaks=c(1,5,10))

ntd <- normTransform(dds_filtered)
vsd <- vst(dds_filtered, blind=FALSE)
rld <- rlog(dds_filtered, blind=FALSE)
head(assay(vsd), 3)

colData(dds_filtered)

dds_filtered$treatment
library("pheatmap")
select <- order(rowMeans(counts(dds_filtered,normalized=TRUE)), decreasing=TRUE)[1:20]
annot <- dds_filtered$treatment
pheatmap(assay(vsd)[select,],
        cluster_rows = FALSE,
        show_rownames=FALSE,
        cluster_cols=FALSE,
        annotation_col=df)
annotations <- c(rep("pedTB",3), rep("HC",3))
df <- as.data.frame(colData(dds_filtered)[,c("treatment")])
pheatmap(assay(vsd)[select,],cluster_cols=FALSE,labels_col=annotations)

pheatmap(assay(vsd)[select,])
#get order of genes as seen on heatmap
h_m<- pheatmap(assay(vsd)[select,])
hc <- as.hclust( h_m$tree_row)
cutree( hc, h=11 )[hc$order] #tb
cutree( hc, h=10 )[hc$order] #hc
sampleDists <- dist(t(assay(vsd)))
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- paste(vsd$treatment)
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette(rev(brewer.pal(7,"Blues")))(255)
colors <- colorRampPalette(c("blue","black","red"))(n=600)

pheatmap(sampleDistMatrix,clustering_distance_rows=sampleDists,
clustering_distance_cols=sampleDists)
library("RColorBrewer")
plotPCA(vsd,intgroup=("treatment"))
```