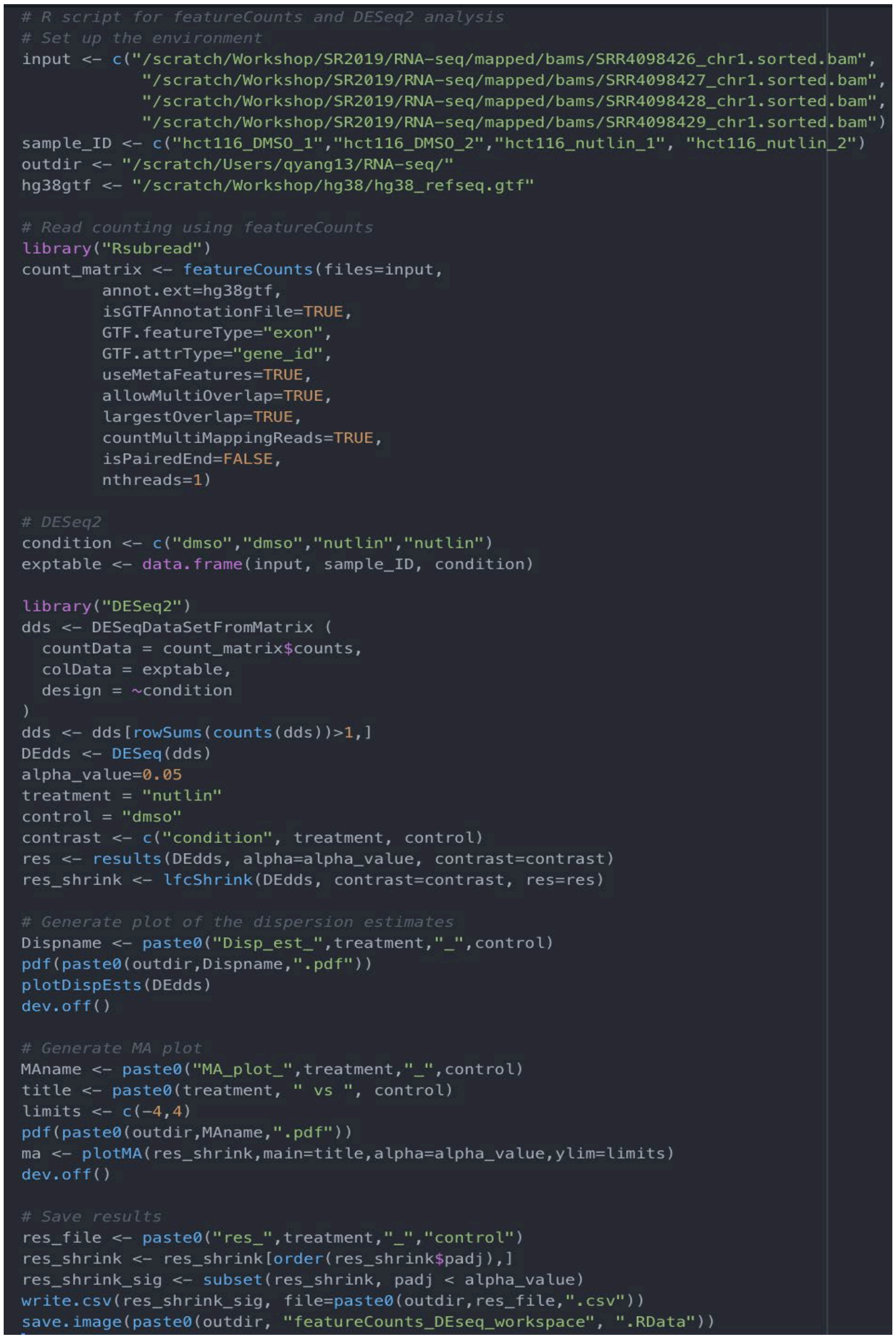
Day 6: RNA-seq Homework

During today's class, we have gone over featureCounts and DEseq2 step by step. In an actual scenario, the dataset will be much larger, so you might want to submit all of the R steps as a sbatch job script. This homework will walk you through the steps to do that.

1. Create a R script containing all the related R commands from **worksheet section 6.1B and section 6.2** using vim:

```
USERNAME> vim featureCounts_DESeq2.r
```

The completed script should look like this (zoom in to see details):

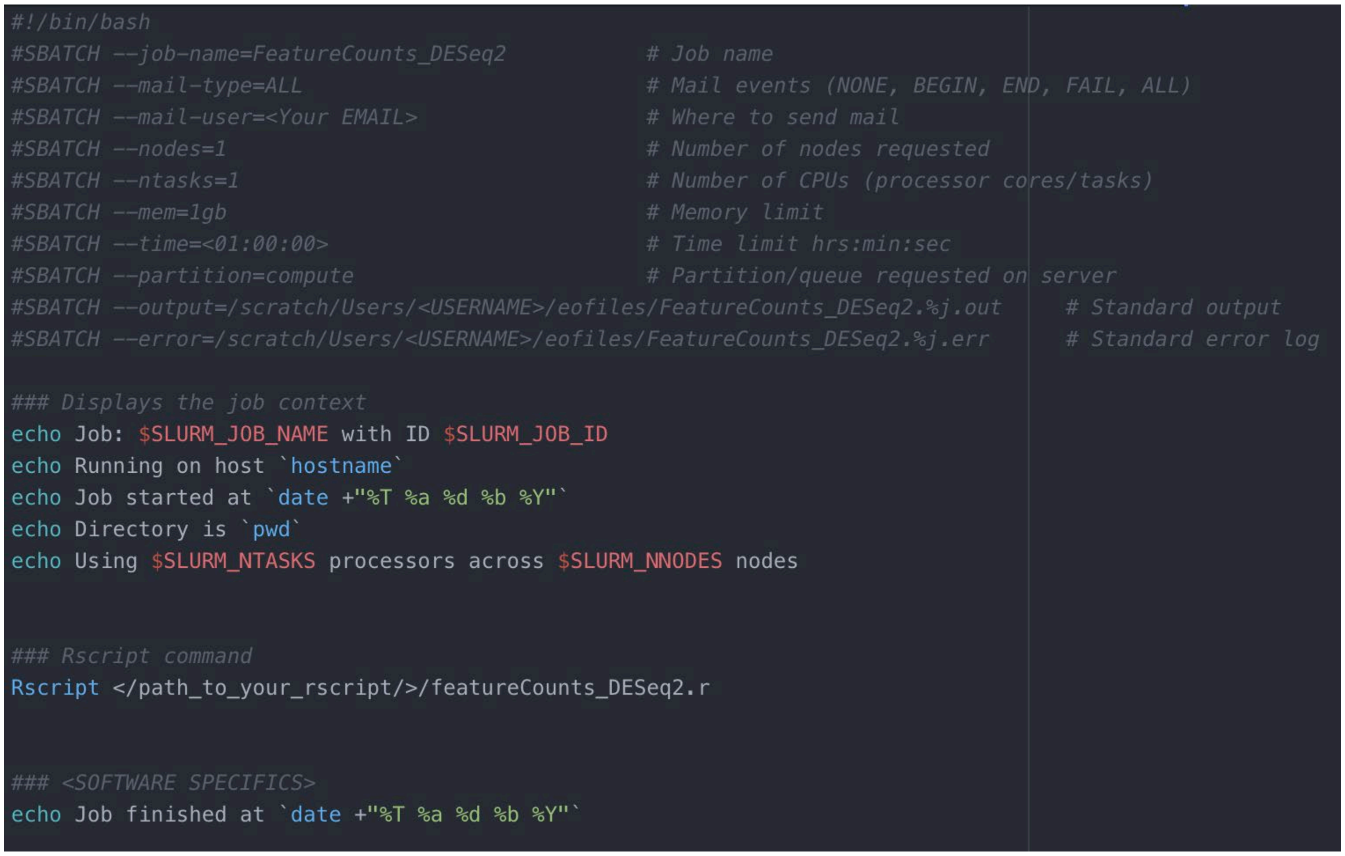


```
# R script for featureCounts and DESeq2 analysis
# Set up the environment
input <- c("/scratch/Workshop/SR2019/RNA-seq/mapped/bams/SRR4098426_chr1.sorted.bam",
           "/scratch/Workshop/SR2019/RNA-seq/mapped/bams/SRR4098427_chr1.sorted.bam",
           "/scratch/Workshop/SR2019/RNA-seq/mapped/bams/SRR4098428_chr1.sorted.bam",
           "/scratch/Workshop/SR2019/RNA-seq/mapped/bams/SRR4098429_chr1.sorted.bam")
sample_ID <- c("hct116_DMSO_1","hct116_DMSO_2","hct116_nutlin_1", "hct116_nutlin_2")
outdir <- "/scratch/Users/qyang13/RNA-seq/"
hg38gtf <- "/scratch/Workshop/hg38/hg38_refseq.gtf"

# Read counting using featureCounts
library("Rsubread")
count_matrix <- featureCounts(files=input,
        annot.ext=hg38gtf,
        isGTFAnnotationFile=TRUE,
        GTF.featureType="exon",
        GTF.attrType="gene_id",
        useMetaFeatures=TRUE,
        allowMultiOverlap=TRUE,
        largestOverlap=TRUE,
        countMultiMappingReads=TRUE,
        isPairedEnd=FALSE,
        nthreads=1)

# DESeq2
condition <- c("dmso","dmso","nutlin","nutlin")
exptable <- data.frame(input, sample_ID, condition)

library("DESeq2")
dds <- DESeqDataSetFromMatrix (
  countData = count_matrix$counts,
  colData = exptable,
  design = ~condition
)
dds <- dds[rowSums(counts(dds))>1,]
DEdds <- DESeq(dds)
alpha_value=0.05
treatment = "nutlin"
control = "dmso"
contrast <- c("condition", treatment, control)
res <- results(DEdds, alpha=alpha_value, contrast=contrast)
res_shrink <- lfcShrink(DEdds, contrast=contrast, res=res)

# Generate plot of the dispersion estimates
Dispname <- paste0("Disp_est_",treatment,"_",control)
pdf(paste0(outdir,Dispname,".pdf"))
plotDispEsts(DEdds)
dev.off()

# Generate MA plot
MAname <- paste0("MA_plot_",treatment,"_",control)
title <- paste0(treatment, " vs ", control)
limits <- c(-4,4)
pdf(paste0(outdir,MAname,".pdf"))
ma <- plotMA(res_shrink,main=title,alpha=alpha_value,ylim=limits)
dev.off()

# Save results
res_file <- paste0("res_",treatment,"_","control")
res_shrink <- res_shrink[order(res_shrink$padj),]
res_shrink_sig <- subset(res_shrink, padj < alpha_value)
write.csv(res_shrink_sig, file=paste0(outdir,res_file,".csv"))
save.image(paste0(outdir, "featureCounts_DESeq_workspace", ".RData"))
```

2. Once this script is complete, save and quit the script using :**wq**. In a general situation, you can run the R script using the command **Rscript:**

```
USERNAME>  Rscript /path/to/your/rscript/featureCounts_DESeq2.r
```

But, in this case, we are using the login node of the server to handle a large amount data taking up a large amount of memory and CPU, and IT might not like it. A better practice is submitting the single line of command above as a job script through sbatch. Now create a sbatch script similar to the one below (**Modify the content in <>**):



```
#!/bin/bash
#SBATCH --job-name=FeatureCounts_DESeq2          # Job name
#SBATCH --mail-type=ALL                          # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=<Your EMAIL>                 # Where to send mail
#SBATCH --nodes=1                                # Number of nodes requested
#SBATCH --ntasks=1                               # Number of CPUs (processor cores/tasks)
#SBATCH --mem=1gb                                # Memory limit
#SBATCH --time=<01:00:00>                        # Time limit hrs:min:sec
#SBATCH --partition=compute                      # Partition/queue requested on server
#SBATCH --output=/scratch/Users/<USERNAME>/eofiles/FeatureCounts_DESeq2.%j.out     # Standard output
#SBATCH --error=/scratch/Users/<USERNAME>/eofiles/FeatureCounts_DESeq2.%j.err      # Standard error log

### Displays the job context
echo Job: $SLURM_JOB_NAME with ID $SLURM_JOB_ID
echo Running on host `hostname`
echo Job started at `date +"%T %a %d %b %Y"`
echo Directory is `pwd`
echo Using $SLURM_NTASKS processors across $SLURM_NNODES nodes


### Rscript command
Rscript </path_to_your_rscript/>/featureCounts_DESeq2.r


### <SOFTWARE SPECIFICS>
echo Job finished at `date +"%T %a %d %b %Y"`
```

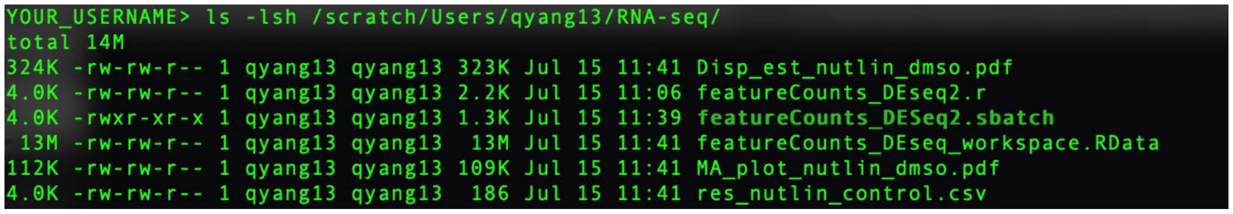
IMPORTANT: Do **NOT** module load R, as the R in that instance would not have the necessary libraries for completing the analysis. Simply use **Rscript** command to run the script you just wrote in step 1.

3. The job should take about 3 mins to complete. Once the job is completed successfully, let's first examine the output from the job:

```
USERNAME > cat /scratch/Users/USERNAME/eofiles/FeatureCounts_DESeq2.<jobID>.out
USERNAME > cat /scratch/Users/USERNAME/eofiles/FeatureCounts_DESeq2.<jobID>.err
```

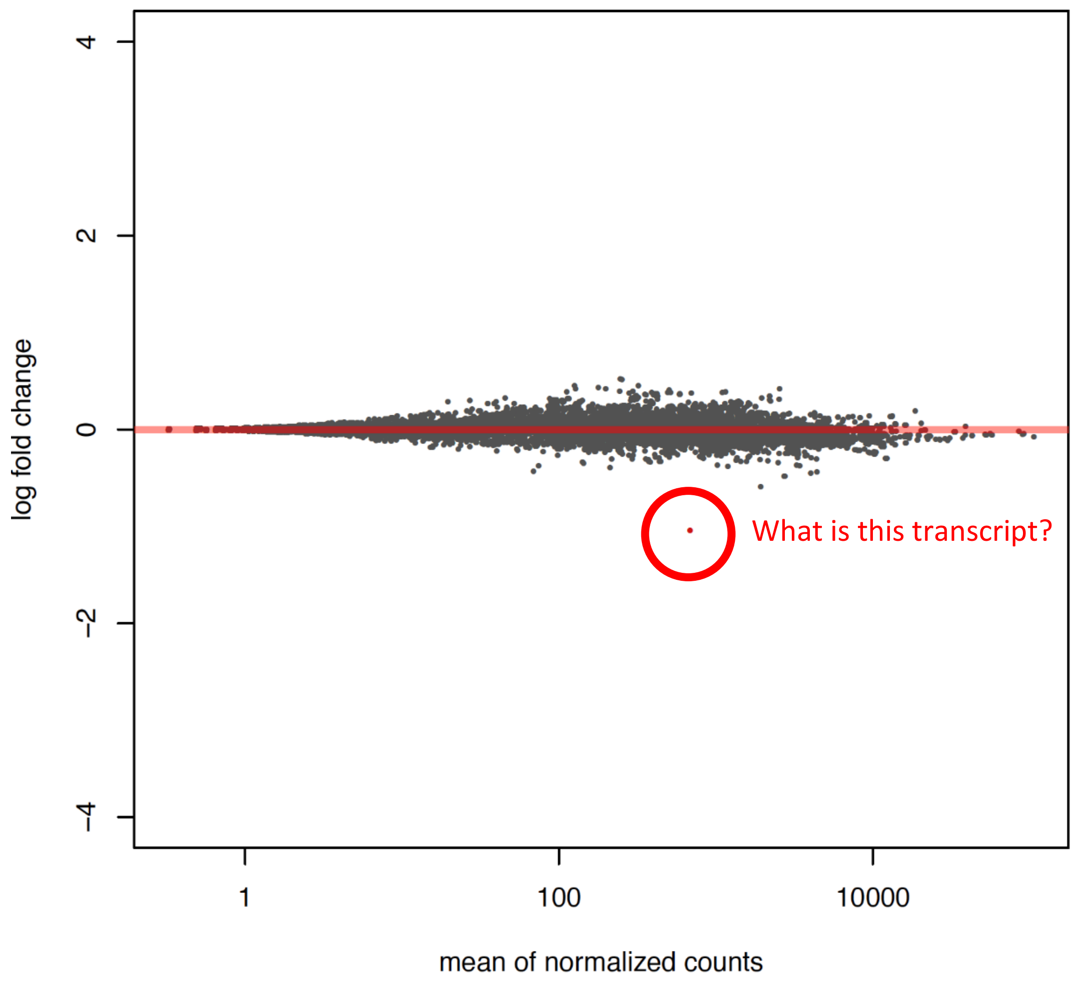
In these error and output files, you should get the same output like the ones we explained during the class.

4. More importantly, we should also get the results and plots from the analysis in our directory:



```
YOUR_USERNAME> ls -lsh /scratch/Users/qyang13/RNA-seq/
total 14M
324K -rw-rw-r-- 1 qyang13 qyang13 323K Jul 15 11:41 Disp_est_nutlin_dmso.pdf
4.0K -rw-rw-r-- 1 qyang13 qyang13 2.2K Jul 15 11:06 featureCounts_DESeq2.r
4.0K -rwxr-xr-x 1 qyang13 qyang13 1.3K Jul 15 11:39 featureCounts_DESeq2.sbatch
 13M -rw-rw-r-- 1 qyang13 qyang13  13M Jul 15 11:41 featureCounts_DEseq_workspace.RData
112K -rw-rw-r-- 1 qyang13 qyang13 109K Jul 15 11:41 MA_plot_nutlin_dmso.pdf
4.0K -rw-rw-r-- 1 qyang13 qyang13  186 Jul 15 11:41 res_nutlin_control.csv
```

5. Next, download the .pdf and .csv files from the output directory to your local machine using **rsync**. From the **MA_plot_nutlin_dmso.pdf** file, you should notice there's only one transcript that's significantly downregulated. What is that gene?
(hint: check **res_nutlin_control.csv**, which contains significantly up/dow-regulated genes)


log fold change
mean of normalized counts
What is this transcript?