- Download.sh was downloaded
    - After “cd” in scratch/zmk256/FinalProjectA, I copied download.sh to the directory, and ran “bash download.sh” to download the contents.
- Download the recommended reference transcriptome and annotation file from provided links.

```
latest_release=$(curl -s 'http://rest.ensembl.org/info/software?content-type=
application/json' | grep -o '"release":[0-9]*' | cut -d: -f2)

wget -L ftp://ftp.ensembl.org/pub/release-${latest_release}/fasta/homo_sapie
ns/dna/Homo_sapiens.GRCh38.dna_sm.primary_assembly.fa.gz

wget -L ftp://ftp.ensembl.org/pub/release-${latest_release}/gtf/homo_sapiens
/Homo_sapiens.GRCh38.${latest_release}.gtf.gz
```

- Create a slurm script to execute nextflow
    - Load the most recent Nextflow module:

```
module load nextflow/23.04.1
```

    - Enter the execution slurm script by using the nano command.

```
#!/bin/bash
#SBATCH --job-name=RNA_seq
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=24:00:00
#SBATCH --mem=4G
#SBATCH --mail-type=END
#SBATCH --mail-user=zmk256@nyu.edu


module load nextflow/23.04.1

#Execute nf-core/rnaseq pipeline
nextflow run nf-core/rnaseq -r 3.14.0 \
--input /scratch/zmk256/FinalProjectA/samplesheet.csv \
--outdir res \
--fasta "/scratch/zmk256/FinalProjectA/Homo_sapiens.GRCh
38.dna_sm.primary_assembly.fa.gz" \
--gtf "/scratch/zmk256/FinalProjectA/Homo_sapiens.GRCh38.111.gtf" \
--extra_salmon_quant_args "--gcBias " \
-profile nyu_hpc \
-params-file /scratch/zmk256/FinalProjectA/rna.json
```

- Create a configuration file (rna.json)

```
{
        "max_memory": "22.GB",
        "max_cpus": 4,
        "max_time": "4.h",
        "skip_trimming": false,
        "skip_alignment": true,
        "pseudo_aligner": "salmon",
        "save_reference": true
}
```

- MultiQC and execution reports were generated after the successful execution of Nextflow workflow, along with the trimmed reads using TrimGalore!/Cutadept (versions provided in Appendix D).
-

- Four tools were used to process the generated files from the nextflow execution.
    - Generated files:
        - Quant.sf files for each sample (six in total)
        - Tx2gene.tsv in scratch/zmk256/FinalPrjectA/res/salmon
    - Four necessary tools
        - RStudio
        - DESeq1
        - Tximport

```
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("tximport")
# ------------------
# source credit: Soneson C, Love MI, Robinson MD (2015).
 "Differential analyses for RNA-seq:
# transcript-level estimates improve gene-level inferences."
 F1000Research, 4. doi:
# 10.12688/f1000research.7563.1.
```

        - Tidyverse

```
> library(tximport, lib.loc = "/ext3/apps/r/4.3.3/lib/R/library")
> library(DESeq2, lib.loc = "/ext3/apps/r/4.3.3/lib/R/library")
> netid <- 'fb2148'
> sampleID <- c('SRR7819990', 'SRR7819991', 'SRR7819992', 'SRR7819993', 'SRR7819994',
 'SRR7819995')
> sample_condition <- c(rep('control', 3),rep('silenced', 3))
> files <- file.path('/scratch',zmk256,'FinalProjectA','res','salmon',sampleID,'
quant.sf')
> names(files) <- sampleID
> tx2gene <- read.table(file.path("/scratch",zmk256,"FinalProjectA","res","salmon",
"tx2gene.tsv"),header=F,sep="\t")
> txi <- tximport(files, type='salmon', tx2gene=tx2gene)

> metadata.df <- data.frame(sample = factor(sampleID),
+                           cell_line = factor( c(sampleID)),
+                           condition = factor(sample_condition,levels = c('control',
'silenced')) )

> row.names(metadata.df) <- sampleID

> metadata.df
              sample  cell_line condition
```

```
SRR7819990 SRR7819990 SRR7819990   control
SRR7819991 SRR7819991 SRR7819991   control
SRR7819992 SRR7819992 SRR7819992   control
SRR7819993 SRR7819993 SRR7819993  silenced
SRR7819994 SRR7819994 SRR7819994  silenced
SRR7819995 SRR7819995 SRR7819995  silenced

> dds <- DESeqDataSetFromTximport(txi,
+                                 colData = metadata.df,
+                                 design = ~ condition)

> counts(dds) %>%
+     dim()

> counts(dds) %>%
+     head()
> keep <- rowSums(counts(dds)) >= 10 # pre-filtering genes with less than 10 reads
> dds <- dds[keep,]
> counts(dds) %>%
+     dim()

> dds <- DESeq(dds)

counts(dds,normalized=T) %>%
  head(n=10) # 10 rows

normalizedcounts.tbl.df <- counts(dds,normalized=T) %>%
  as.data.frame() %>%
  rownames_to_column(var = 'feature_id') %>%
  as_tibble() # convert data.frame to tibble

normalizedcounts.tbl.df

normalizedcounts.long.tbl.df <- normalizedcounts.tbl.df %>%
                                 pivot_longer( cols = -feature_id,
                                               names_to = 'sample',
                                               values_to = 'normalized_count')
Normalizedcounts.long.tbl.df # to visualize

normalizedcounts.long.tbl.df %>%
  ggplot(aes(x = normalized_count)) +
  geom_histogram(binwidth = 20) +
  xlim(0,1000) +
  ylim(0,7500) +
  facet_wrap( ~ sample, ncol = 4)

normalizedcounts.long.tbl.df %>%
  filter(is.finite(normalized_count)) %>%
  group_by(feature_id) %>%
  summarise(mean = mean(normalized_count),
            variance = var(normalized_count)) %>%
  ggplot(aes(x = mean,y = variance)) +
    geom_point(size = .6) +
    scale_y_log10(limits = c(1,1e9)) +
    scale_x_log10(limits = c(1,1e9)) +
    geom_abline(intercept = 0, slope = 1, color="dark blue")

#logarithmic transformation
rld <- rlog(dds)

#perform PCA plot
plot(PCA)

> res <- results(dds, contrast = c('condition','silenced','control') )
> resultsNames(dds)
[1] "Intercept"                      "condition_silenced_vs_control"
> res.lfcShrink <- lfcShrink(dds, coef = 'condition_silenced_vs_control', type='apeglm')

> plotMA(res)
```

```
> plotMA(res.lfcShrink)
> plotDispEsts(dds)
> res <- results(dds, alpha=0.05)
> res.lfcShrink <- lfcShrink(dds, res=res, coef='condition_silenced_vs_control', type='apeglm')

> res.lfcShrink %>%
+     as_tibble() %>% # coerce DESeqResults object to tibble (a tidyverse data.frame with benefits)
+     summarise(padj_NA = sum(is.na(padj)), # summarise collapses output to a single row with new
 columns with
 summaries of the data
+               padj_notNA = sum(!is.na(padj)))


plot(metadata(res.lfcShrink)$filterNumRej,
     type="b", ylab="number of rejections",
     xlab="quantiles of filter")
lines(metadata(res)$lo.fit, col="red")
abline(v=metadata(res)$filterTheta)

metadata(res.lfcShrink)$filterThreshold

# p value and multiple correction test portion (source: Week 10 assignment)
res.lfcShrink %>%
  as_tibble() %>% # coerce to tibble
  ggplot(aes(pvalue)) +
  geom_histogram(fill="light blue",color='black',bins = 40)

res.lfcShrink.tbl_df <- res.lfcShrink %>%
  as.data.frame() %>%
  rownames_to_column(var = "feature_id") %>%
  as_tibble()

res.lfcShrink.tbl_df %>%
  arrange(padj)

res.lfcShrink.tbl_df %>%
  filter(padj < 0.05) %>%
  arrange(padj)

res.lfcShrink.tbl_df %>%
  summarise(`FDR < 0.05` = sum(padj < 0.05,na.rm = T))

res.lfcShrink.tbl_df %>%
  mutate(`LFC < 0` = case_when(log2FoldChange < 0 & padj < 0.05 ~ 1, # add a column "LFC < 0" and set
 to 1 if gene has LFC < 0 and FDR < 0.05
                               TRUE ~ 0)) %>%                       # and set to zero otherwise
  mutate(`LFC > 0` = case_when(log2FoldChange > 0 & padj < 0.05 ~1,  # add a column "LFC < 0" and set
 to 1 if gene has LFC > 0 and FDR < 0.05
                               TRUE ~ 0)) %>%                       # and set to zero otherwise
  summarise(`LFC < 0 count`= sum(`LFC < 0`),
            `LFC > 0 count` = sum( `LFC > 0` ))

summary(res.lfcShrink)
```