References:

DE lap from class and this really detailed vignette...<https://bioconductor.org/packages/release/bioc/vignettes/tximport/inst/doc/tximport.html>.

### This code is what I described on Sat.

### defining the locations of the kallisto est. files

dirs = list.files("C:/Users/moham/Downloads/kallisto\_out/")

estimate\_files = list.files(

"C:/Users/moham/Downloads/kallisto\_out/",

pattern = "abundance.tsv",

recursive = TRUE, full.names = TRUE)

names(estimate\_files) = dirs

estimate\_files

#looking at the first estimate file

quants = read\_tsv(estimate\_files[1])

head(quants)

#load in the tx2g dataset

t2g\_file = read.table(

file = 'transcripts\_to\_genes.txt',

header = FALSE, sep = '\t')

# bellow code sets the data frame headers

t2g\_file = t2g\_file[, c(1,3)]

newheaders= c("TXNAME", "GENEID")

colnames(t2g\_file)= newheaders

head(t2g\_file)

#### tximport time

tx2gene= t2g\_file

write.csv(

tx2gene,file="tx2gene.csv",

row.names = FALSE,quote=FALSE)

txi= tximport(

estimate\_files,

type="kallisto",

tx2gene = tx2gene)

names(txi)

head(txi$counts)

x= as.data.frame(txi$counts)

# Read the sample information into R

sampleinfo= read.delim("SRP295171\_sampleinfo.txt")

#View(sampleinfo)

rownames(sampleinfo)= sampleinfo$SampleName

#need to rearrange the metadata rownames to match

#######the order of the column names of the txi$count data set

Matched\_indeces= match(colnames(txi$counts), rownames(sampleinfo))

Matched\_indeces

sampleinfo= sampleinfo[Matched\_indeces,]

head(sampleinfo)

#now check to see if they match

all(rownames(sampleinfo) == colnames(txi$counts))

sampleinfo$Treatment=c("LPS\_OVA", "LPS\_OVA", "LPS\_OVA",

"LPS\_Ctrl", "LPS\_Ctrl", "LPS\_Ctrl")

############### DGE analysis step

# setup the deseq2 object

dds= DESeqDataSetFromTximport(txi,

colData = sampleinfo,

design = ~Treatment)

### remove genes with on average < 6 reads across the samples

### we consider these genes with less than 6 reads as not expressed

### this logic holds that a gene should have altleast one read per sample

### and we have six samples in total for the study

#### also remove genes expressed in less than 3 samples given our cutoff

#### went from 35,999 genes to 15,050 genes after that

dds2 =dds

keep = rowSums(assay(dds) >= 6) >= 3

table(keep)

dds = dds[keep,]

### size factor estimation, dispersion estimation

dds = DESeq(dds)

dds2 = DESeq(dds2)

# this hist shows how many genes are expressed in only sample...before filtering

# note: we still remove genes with low sample counts here

is\_expressed2 = assay(dds2) >=6

head(is\_expressed2)

png('bioinf545\_hist\_before\_filtering.png',

width = 7.925,

height = 5,

units = 'in',

res = 600)

hist(rowSums(is\_expressed2),

main="Number of samples expressing a given gene: before filtering step",

xlab="Sample Count")

dev.off()

# this hist shows how the data looks when filtering out genes not expressed

# in at least three samples

is\_expressed = assay(dds) >=6

head(is\_expressed)

png('bioinf545\_hist\_after\_filtering.png',

width = 7.925,

height = 5,

units = 'in',

res = 600)

hist(rowSums(is\_expressed),

main="Number of samples expressing a given gene: after filtering step",

xlab="Sample Count")

dev.off()

### differential expression: comparing LPS\_OVA vs LPS\_Ctrl

res = results(dds,contrast=c("Treatment", "LPS\_OVA", "LPS\_Ctrl"),pAdjustMethod="fdr")

sort(res$log2FoldChange)

summary(res)

# write results into file and then load results into a df

write.table(res, file = "Deseq2\_results.txt", sep = "\t",

row.names = TRUE)