**RNA Seq differential gene expression analysis**

Terminal commands preceded by $

R Studio commands preceded by >

-Download hg38 and annotations from UCSC or Ensembl

Genome:

$ rsync -avzP rsync://hgdownload.cse.ucsc.edu/goldenPath/hg38/bigZips/ .

Annotations:

$ rsync -avzP rsync://hgdownload.cse.ucsc.edu/goldenPath/hg38/database/ .

-Install STAR (helpful video: https://www.youtube.com/watch?v=Ju6PtQD-H34)

<https://github.com/alexdobin/STAR/releases>

Uncompress STAR:

$ tar –zxvf STAR-2.5.2a.tar.gz

Locate the correct executable for your OS:

$ cd STAR-2.5.2a/bin/MacOSX\_x86\_64

Change file permission of the executable:

$ sudo chmod 755 STAR

Add the executable to path:

$ sudo cp STAR /usr/local/bin

Drag the entire STAR-2.5.2a folder to your PATH in terminal:

$ nano .bash\_profile

export PATH=”$PATH: /Users/…”

-Create a directory (I called it "genomepath") for the indexed genome files

$ mkdir genome path

Make sure you are in the directory *containing* this folder. Check using this command.

$ pwd

To list all file permissions in current directory:

$ ls –l

Change permissions to genome path folder to allow STAR to write in this file. Note: rwx (4,2,1) for user, rwx (4,2,1) for group, rwx (4,2,1) for world. Add up permissions. Ex: 777 indicates you allow user, group, world to read, write, and execute (r,w,x) this file.

$ sudo chmod 777 genomepath

-In terminal, index the genome into directory "genomepath" so that STAR can use the indexed genome to align your reads. Make sure you are in the genomepath directory.

lski1944:genomepath tangs$

/Users/tangs/STAR-2.5.2a/bin/MacOSX\_x86\_64/STAR --runThreadN 8 --runMode genomeGenerate --genomeDir /Users/tangs/genomepath --genomeFastaFiles /Users/tangs/hg38/hg38.fa --sjdbGTFfile /Users/tangs/hg38anno --sjdbOverhang 100

Without annotations:

$ /Users/tangs/STAR-2.5.2a/bin/MacOSX\_x86\_64/STAR --runThreadN 8 --runMode genomeGenerate --genomeDir /Users/tangs/genomepath --genomeFastaFiles /Users/tangs/hg38/hg38.fa

-Align reads to genome using STAR, should get bam files as output

$ /Users/tangs/STAR-2.5.2a/bin/MacOSX\_x86\_64/STAR --runThreadN 8 --runMode alignReads --genomeDir /Users/tangs/genomepath --readFilesIn /Users/tangs/RNAseqdata/A375-2\_AGTTCC\_L003\_R1\_001.fastq –outSAMtype BAM SortedByCoordinate

Repeat for each fastq file.

Make sure to save the bam file in another folder (outside of genomepath) so that STAR does not overwrite the bam file just generated. Also save the log.out, log.final.out, log.progress.out, and SJ.out.tab files associated with the bam file so you have the summary results of the run.

How to “parallelize” STAR (meaning use multiple cores to run program instead of just one core; may make alignment process faster) –there is a command you can find through google…

-Use R Studio and Bioconductor packages to generate count table and count table matrix from bam files

Adapted from: <http://homer.salk.edu/homer/basicTutorial/rnaseqR.html>

In R Studio:

**> source("http://bioconductor.org/biocLite.R")**

**> biocLite("GenomicRanges")  
> biocLite("GenomicFeatures")**

**> biocLite(“Rsamtools”)  
> biocLite("DESeq2")  
> biocLite("edgeR")**  
**> biocLite("org.Hs.eg.db") Note: use Hs for Homo sapiens**

Before checking differential expression, we need to quantify reads in transcripts.  
  
# load library for genomic annotations   
**> library(GenomicFeatures)**  
**> library(GenomicRanges)**  
  
# load the transcript annotation file from UCSC.  Make sure to enter the correct genome version  
**> txdb=makeTxDbFromUCSC(genome='hg38',tablename='refGene')**  
  
# Use the function transcriptsBy(txdb,'gene') for the whole genic region instead of just exons  **> ex\_by\_gene=exonsBy(txdb,'gene')**  
  
# load the samtools library for R  
**> library(Rsamtools)**

**Make sure you are in the directory containing the bam files. Use following command to get working or current directory.**

**> getwd**

**> biocLite("GenomicAlignments")**

**> library(GenomicAlignments)**  
  
# read the sequencing read alignment into R (combine with next step to save memory)  
**> reads1r1=readGAlignments("A375-2\_Aligned.sortedByCoord.out.bam")  
> reads2r1=readGAlignments("CC-A375-2\_Aligned.sortedByCoord.out.bam")**  
#repeat as necessary for more samples  
  
# count reads overlapping the exons  
**> counts1r1 = countOverlaps(ex\_by\_gene,reads1r1)  
> counts2r1 = countOverlaps(ex\_by\_gene,reads2r1)**  
  
# create count table  
**> countTable = data.frame(condition1r1=counts1r1,condition2r1=counts2r1, stringsAsFactors=FALSE)**  
# set the gene IDs to the table row names  
**> rownames(countTable)=names(ex\_by\_gene)**

**Can view countTable by typing:**

**> countTable**  
  
#output tag counts to a file  
**> write.table(countTable,file="countTable.txt",sep="\t")**

**txt file will appear in working directory.**  
  
#removing rows that are zero for all genes (edgeR and DESeq have trouble with these)  
**> x <- rowSums(countTable==0)!=ncol(countTable)  
> newCountTable <- countTable[x,]**

**> write.table(newCountTable,file="newCountTable.txt",sep="\t")**

**txt file will appear in working directory.**

**To generate count table matrix:**

# install edgeR  
**> biocLite("edgeR")**  
  
# load the edgeR library  
**> library(edgeR)**  
  
# either use the input data from above, or load your own table of read counts.  
**> data <- as.matrix(read.table("newCountTable.txt"))**  
# - or -  
**> data = newCountTable**  
  
# assign groups to the samples in the counts table. Note: 0=condition 1, 1=condition 2  
**> g <- c(0,1)**  
# get the library sizes (total counts in annotated genes)  
**> libSizes <- as.vector(colSums(data))**  
  
# edgeR command pipeline (basically the same for each sample)  
**> d <- DGEList(counts=data,group=g,lib.size=libSizes)  
> d <- calcNormFactors(d)  
> d <- estimateCommonDisp(d)  
> d <- estimateTagwiseDisp(d)  
> de.com <- exactTest(d)  
> results <- topTags(de.com,n = length(data[,1]))**  
  
# write the output to a text file  
**> write.table(as.matrix(results$table),file="outputFile.txt",sep="\t")**

**Output file should appear in working directory.**  
  
# No replicates  
# If you don't have replicates, edgeR will throw a fit.  In place of:  
**> d <- estimateCommonDisp(d)  
> d <- estimateTagwiseDisp(d)**you'll want to use**:  
> d$common.dispersion <- 0.05**

Need to add columns to the count table to specify what the conditions are.

Adapted from: https://dwheelerau.com/2014/02/17/how-to-use-deseq2-to-analyse-rnaseq-data/

> library('DESeq2')

> sampleFiles <- c("counts1r1", "counts2r1")

> sampleCondition <- c("untreated", "treated")

> CountTable <- read.table("newCountTable.txt", header=TRUE, row.names=1)

> samples <- data.frame(sampleName=sampleFiles, fileName=sampleFiles, condition=sampleCondition)

> dds <- DESeqDataSetFromMatrix(countData = CountTable, colData=samples, design=~condition)

> dds <-DESeq(dds)

> res<-results(dds)

To order the padj values:

> res<-res[order(res$padj),]

> head(res)

To look at what the columns mean:

> mcols(res,use.names=TRUE)

To save the table as csv file:

> write.csv(as.data.frame(res),file='sim\_condition\_treated\_results\_deseq2.csv')

-Differential gene expression analysis using DESeq2

See manual for generation of plots and explanations of data: <https://bioc.ism.ac.jp/packages/2.14/bioc/vignettes/DESeq2/inst/doc/beginner.pdf>

<https://bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.pdf>

To make plots:

> biocLite(ggplot2)

> library(ggplot2)

Need to unmask plotMA (plotMA was masked because we loaded DESeq before edgeR or vice versa…not sure)

> DESeq2::plotMA(dds,ylim=c(-2,2),main='DESeq2')

To get histogram of p-values:

> hist( res$pvalue, breaks=20, col="grey" )

To get dispersion plot:

> plotDispEsts( dds, ylim = c(1e-6, 1e1) )

See manual for further analysis.