Sample GEO RNA-seq Dataset Processing Code using DESeq2 (GSE124422)

Shayna Kay

07/04/2022

## Standard set-up of R Markdown environment

knitr::opts\_chunk$set(echo = TRUE)  
rm(list=ls()) # Clear R environment's memory prior to beginning analysis

## Install packages and load libraries

Required packages include: “htmltools”, “DESeq2” from “Bioconductor”, “ggplot2”, and “dplyr”.

## Read count data into RStudio and process data for analysis

Prepare the count data table as required for input into DESeq2 functions.

# file.choose()  
  
# Read in .csv file containing RNA-seq count data  
countData <- read.csv('C:\\Users\\shayn\\OneDrive\\Documents\\Thesis Documents\\DATA\\Bioinformatics\\GEO Files\\Select Datasets for Analysis 1\\3-GSE124422 - Copy\_done\\GSE124422\_htseq\_counts\_atra-ctrl.csv', header = TRUE, sep = ",")   
  
head(countData) # Print the first 6 lines of the count table to confirm data import was successful

## ï..ID X1\_ATRA\_counts X2\_ATRA\_counts X3\_ATRA\_counts X1\_untreated\_counts  
## 1 1/2-SBSRNA4 3 7 17 4  
## 2 A1BG 1 0 0 0  
## 3 A1BG-AS1 0 0 1 0  
## 4 A1CF 0 0 0 0  
## 5 A2LD1 4 8 2 7  
## 6 A2M 2 2 4 3  
## X2\_untreated\_counts X3\_untreated\_counts  
## 1 6 7  
## 2 1 0  
## 3 0 0  
## 4 0 0  
## 5 10 4  
## 6 5 0

countData <- countData %>% rename(ID = ï..ID) # Rename column containing gene symbols to "ID"  
head(countData) # Print the first 6 lines of the count table to confirm processing was successful

## ID X1\_ATRA\_counts X2\_ATRA\_counts X3\_ATRA\_counts X1\_untreated\_counts  
## 1 1/2-SBSRNA4 3 7 17 4  
## 2 A1BG 1 0 0 0  
## 3 A1BG-AS1 0 0 1 0  
## 4 A1CF 0 0 0 0  
## 5 A2LD1 4 8 2 7  
## 6 A2M 2 2 4 3  
## X2\_untreated\_counts X3\_untreated\_counts  
## 1 6 7  
## 2 1 0  
## 3 0 0  
## 4 0 0  
## 5 10 4  
## 6 5 0

## Generate a metadata table

DESeq2 requires a metadata table containing information about the RNA-seq samples.

condition <- factor(c("X1\_ATRA\_counts","X2\_ATRA\_counts","X3\_ATRA\_counts","X1\_untreated\_counts","X2\_untreated\_counts","X3\_untreated\_counts")) # Condition column links metadata table to count table  
treatment <- factor(c("ATRA", "ATRA", "ATRA", "control", "control", "control"))  
GEO\_ID <- factor(c("GSM3532958","GSM3532965", "GSM3532972", "GSM3532961", "GSM3532968", "GSM3532975"))  
metadata <- DataFrame(condition, treatment, GEO\_ID) # Create the dataframe containing above sample information  
head(metadata) # Print the first 4 lines of the metadata table to confirm it was created successfully

## DataFrame with 6 rows and 3 columns  
## condition treatment GEO\_ID  
## <factor> <factor> <factor>  
## 1 X1\_ATRA\_counts ATRA GSM3532958  
## 2 X2\_ATRA\_counts ATRA GSM3532965  
## 3 X3\_ATRA\_counts ATRA GSM3532972  
## 4 X1\_untreated\_counts control GSM3532961  
## 5 X2\_untreated\_counts control GSM3532968  
## 6 X3\_untreated\_counts control GSM3532975

## Construct DESEQDataSet Object to analyze differential gene expression

Export a CSV file containing results for further analysis of genes of interest.

# The "DESeqDataSetfromMatrix" function stores intermediate calculations and results of the analysis of   
# differential expression based on a formula which specifies the experimental design  
dds <- DESeqDataSetFromMatrix(countData=countData,   
 colData=metadata,   
 design=~treatment, tidy = TRUE)

## it appears that the last variable in the design formula, 'treatment',  
## has a factor level, 'control', which is not the reference level. we recommend  
## to use factor(...,levels=...) or relevel() to set this as the reference level  
## before proceeding. for more information, please see the 'Note on factor levels'  
## in vignette('DESeq2').

# Re-level experimental treatments, making control the reference treatment  
dds$treatment <- relevel(dds$treatment, ref = "control")   
dds # View intermediate results of differential expression analysis

## class: DESeqDataSet   
## dim: 23368 6   
## metadata(1): version  
## assays(1): counts  
## rownames(23368): 1/2-SBSRNA4 A1BG ... ZZZ3 tAKR  
## rowData names(0):  
## colnames(6): X1\_ATRA\_counts X2\_ATRA\_counts ... X2\_untreated\_counts  
## X3\_untreated\_counts  
## colData names(3): condition treatment GEO\_ID

dds <- DESeq(dds) # Complete differential expression analysis and store output in "dds" variable

## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

res <- results(dds) # Store the results of differential expression analysis in "res" variable  
head(results(dds, tidy = TRUE)) # View the first 6 lines of the results table to confirm expected output

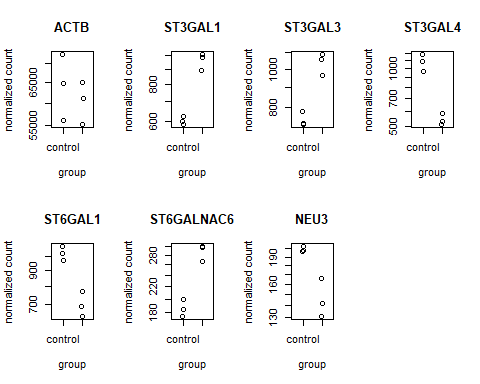
## row baseMean log2FoldChange lfcSE stat pvalue  
## 1 1/2-SBSRNA4 7.4358983 0.7987534 0.7850037 1.01751549 0.3089083  
## 2 A1BG 0.3025880 0.1330469 3.5666530 0.03730302 0.9702434  
## 3 A1BG-AS1 0.1724002 1.0948160 4.0804729 0.26830617 0.7884637  
## 4 A1CF 0.0000000 NA NA NA NA  
## 5 A2LD1 5.7048824 -0.4023918 0.8452236 -0.47607736 0.6340193  
## 6 A2M 2.5645457 0.2089302 1.2593390 0.16590464 0.8682320  
## padj  
## 1 0.5957031  
## 2 NA  
## 3 NA  
## 4 NA  
## 5 NA  
## 6 NA

# Export the results of differential expression analysis and save on computer as a CSV file:  
write.csv(res,"C:\\Users\\shayn\\OneDrive\\Documents\\Thesis Documents\\DATA\\Bioinformatics\\GEO Files\\Select Datasets for Analysis 1\\3-GSE124422 - Copy\_done\\DESeqResults.csv")   
# Adjusted p-values<0.05 were extracted from this file to identify significantly differentially expressed  
# sialyltransferase and neuraminidase genes   
# Log2(FoldChange) values were extracted from this file to create a heat map (Figure 10) of differential   
# sialyltransferase and neuraminidase gene expression  
  
summary(res) # View summary of differential gene expression

##   
## out of 16179 with nonzero total read count  
## adjusted p-value < 0.1  
## LFC > 0 (up) : 1184, 7.3%  
## LFC < 0 (down) : 1283, 7.9%  
## outliers [1] : 0, 0%  
## low counts [2] : 4289, 27%  
## (mean count < 6)  
## [1] see 'cooksCutoff' argument of ?results  
## [2] see 'independentFiltering' argument of ?results

# Create plots of normalized counts for sialyltransferase and neuraminidase genes of interest

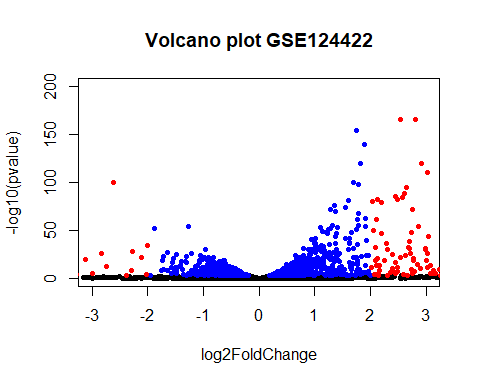
# Use the "plotCounts" function to compare the normalized counts between treated and control groups  
  
# Gemerate plots of normalized counts for ACTB and all sialyltransferases and neuraminidases with significant differences  
par(mfrow=c(2,4)) # Set plot dimensions and layout  
plotCounts(dds, gene="ACTB", intgroup="treatment")  
plotCounts(dds, gene="ST3GAL1", intgroup="treatment")  
plotCounts(dds, gene="ST3GAL3", intgroup="treatment")  
plotCounts(dds, gene="ST3GAL4", intgroup="treatment")  
plotCounts(dds, gene="ST6GAL1", intgroup="treatment")  
plotCounts(dds, gene="ST6GALNAC6", intgroup="treatment")  
plotCounts(dds, gene="NEU3", intgroup="treatment")



## Create a volcano plot

Visualize significantly differentially expressed genes in the entire dataset.

par(mfrow=c(1,1)) # Set plot dimensions and layout  
with(res, plot(log2FoldChange, -log10(pvalue), pch=20, main="Volcano plot GSE124422", xlim=c(-3,3), ylim=c(0,200))) # Create plot with the given specifications  
  
# Add coloured points: blue if padj<0.01, red if log2FC>1 and padj<0.05)  
with(subset(res, padj<0.01), points(log2FoldChange, -log10(pvalue), pch=20, col="blue"))  
with(subset(res, padj<0.01 & abs(log2FoldChange)>2), points(log2FoldChange, -log10(pvalue), pch=20, col="red"))



## Principal Component Analysis

Shows similarity of samples within the same group in transformed RNA-seq count data.

# Transform the raw count data  
# The "vst" function will perform variance stabilizing transformation  
vsdata <- vst(dds, blind=FALSE)  
par(mfrow=c(1,1)) # Set plot dimensions and layout  
  
# Using the "plotPCA" function from the "DESeq2" package to look at how samples group by treatment and day:  
plotPCA(vsdata, intgroup="treatment")

