20.109 RNASeq Ex3: a549 cell line analysis

Amanda Kedaigle, Ernest Fraenkel

11/4/2018

Published RNA-seq Data

In addition to the experiment you and your instructors have conducted yourself, previous authors who have published on this topic have made their data available to the scientific community. Thanks to them, we can analyze that data ourselves and see if their results agree with ours. This time, you'll be writing much of the R code on your own. But to start off, we'll help you prepare the data.

The authors of this paper (link) from 2017 were studying what happens to cancer cells after they enter senescence - a state of cell cycle arrest. As part of their study, they treated a common cell line model of lung cancer, a549 cells, with etoposide and showed that it induced senescence. They offer the data from that paper from here (link). That website is part of the Gene Expression Omnibus, or GEO, a great database for published high-throughput data. The file at the bottom of that page, "GSE102639_A549_Etoposide___Aurora_treated.txt.gz" contains the read counts per gene in their experiment, and has been downloaded to the RStudio.cloud server in Exercise #3. Let's turn it into a DESeqDataSet object together:

```
library("DESeq2")
#Here we read in the file
countsf = read.table("GSE102639_A549_Etoposide__Aurora_treated.txt",sep="\t",header=T,quote="")

#We are only interested in some of the columns in this file - namely, the a549 cells treated with etopo
#Here, we cut our data frame "counts" down to those columns.
counts = countsf[,c('X4422_1_A549_untreated_R1_GCCAATA','X4422_2_A549_untreated_R2_CAGATCA','X4422_9_A5

#We make the rownames of our dataframe the gene names.
#Note that they are Ensembl IDs, not gene symbols. That will be important later.
rownames(counts) = countsf$ensembl_gene_id
head(counts)
```

```
##                 X4422_1_A549_untreated_R1_GCCAATA
## ENSG00000000457                               430
## ENSG00000000460                               160
## ENSG00000000938                                 0
## ENSG00000000971                              6745
## ENSG00000001460                               150
## ENSG00000001461                               482
##                 X4422_2_A549_untreated_R2_CAGATCA
## ENSG00000000457                               592
## ENSG00000000460                               224
## ENSG00000000938                                 0
## ENSG00000000971                             10268
## ENSG00000001460                               213
## ENSG00000001461                               742
##                 X4422_9_A549_Etoposide_R1_GTGAAAC
## ENSG00000000457                               268
## ENSG00000000460                                52
## ENSG00000000938                                 0
## ENSG00000000971                              5490
```

```
## ENSG00000001460                                198
## ENSG00000001461                               2186
##                  X4422_10_A549_Etoposide_R2_ATCACGA
## ENSG00000000457                                405
## ENSG00000000460                                 36
## ENSG00000000938                                  0
## ENSG00000000971                              18010
## ENSG00000001460                                 91
## ENSG00000001461                               1044
```

```
#We'll add information about the experimental design to a new data frame called coldata
#In this case, the first two columns are untreated, and the second two are treated with etoposide.
coldata = data.frame(row.names = colnames(counts), group = c("untreated","untreated","etoposide","etopos

#Here, we make the DESeqDataSet object out of our two dataframes!
#We called the column in coldata containing the treatment "group", so we will set the design to "~group
ddsa549 <- DESeqDataSetFromMatrix(countData = counts,
                                  colData = coldata,
                                  design = ~ group)
```

Now that we've made the "ddsa549" object, you can write your own code to calculate the differentially expressed genes after etoposide treatment in this experiment. Use your previous lab for clues about the code to write, and follow the steps below.

First, explore the data. How many genes do we have data for? How many reads per sample are there in this dataset? How is the metadata (or "coldata") for this dataset structured?

Second, make sure the data clusters in a sensible way. Create a new object with rlog transformed data. Use that object to make a heatmap and a PCA plot and save the images. What samples cluster together in these analyses?

Now, run DESeq2 on this dataset, and get the results of a contrast between "etoposide" and "untreated" in the "group" column. How many genes are up- and down-regulated by etoposide in this experiment?

Check out the differentially expressed genes with the head() function. Here's where we notice that the gene names are with Ensembl IDs, rather than the more easily understandable Gene Symbols. We'll use the "AnnotationDbi" library to map Ensembl IDs to Gene Symbols. Here, we're assuming that the dataframe where you stored the results of running DESeq2 on the data was called "resa549". You should change the two spots where we've used that term to whatever you called your own data frame.

```
library("AnnotationDbi")
library("org.Hs.eg.db")
resa549$geneSymbol = mapIds(org.Hs.eg.db,
                        keys=row.names(resa549),
                        column="SYMBOL",
                        keytype="ENSEMBL",
                        multiVals="first")
```

Finally, use the "AnnotationDbi" package to add the Gene Ontology terms to the resulting data frame, and then use the "topGO" library to create a topGOdata object, and run Fisher and K-S tests to find the most significant gene ontology terms in the downregulated and upregulated genes of this dataset. How do they compare to the terms we found in the previous dataset?