Differential Gene Expression Analysis

Biol 350L: Microbiology Lab

Spring 2021

TA: Dimitri Krutkin

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

```
#Clears the global environment (for those who might use R)
rm(list = ls())
```

Reading raw expression data into R:

Reads raw count data (in the form of comma-seperated values - CSV) and assigns the data in the file to the varriable "raw_rna_seq_data"

```
#read.csv() is a function
raw_rna_seq_data = read.csv("dataset1.csv")
```

"Calling" the variable dumps its data to the screen

Large datasets would take up all of your screenspace: un-commenting the variable below and running the code chunk would print the entire data matrix to the screen:

```
#printing the entire matrix would take up far too much space
#raw_rna_seq_data
```

Some shorter ways to look at your data:

```
#functions head() and tail() can show the top and bottom of the data:
head(raw_rna_seq_data)
```

```
##      Gene_name Control.BR1 Control.BR2 Treatment.BR1 Treatment.BR2
## 1 plasmid_0001         206         313           837           938
## 2 plasmid_0002         130         140           361           463
## 3 plasmid_0003         101          85           232           247
## 4 plasmid_0004         181         260           516           749
## 5 plasmid_0005         252         298           523           681
## 6 plasmid_0006          80         107           186           205
```

```
tail(raw_rna_seq_data)
```

```
##            Gene_name Control.BR1 Control.BR2 Treatment.BR1 Treatment.BR2
## 4196 chromosome_4085           0           0             0             0
## 4197 chromosome_4086          84         186           263           213
## 4198 chromosome_4087         106         158           246           231
## 4199 chromosome_4088         272         482           850           858
## 4200 chromosome_4089          20          15            16            15
## 4201 chromosome_4090      300201      204214        380880        693195
```

The structure of "raw_rna_seq_data" above is referred to as "data frame" in R - it can be thought of as an excel spreadsheet

In an excel spreadsheet, gene counts for each experimental group (control vs. treated) are associated with their respective gene, which usually have their unique identifiers in a column on the left-most side

In R, the row names are stored in a seperate category from data columns - the column "gene_name" above should be removed and each entry from the column should be assigned to the row names seperate category

```
#Extracts all of the gene names and assigns them to a variable called "row_names"
row_names = raw_rna_seq_data$Gene_name

head(row_names)
```

```
## [1] "plasmid_0001" "plasmid_0002" "plasmid_0003" "plasmid_0004" "plasmid_0005"
## [6] "plasmid_0006"
```

```
#Creates a new data frame, only including experimental groups and their respective counts
final_count_data = raw_rna_seq_data[2:5]

#Assigns the values from the first columnn of the raw count matrix to the row names of the final count matrix
row.names(final_count_data) = row_names
```

```
colnames(raw_rna_seq_data[2:5])
```

```
## [1] "Control.BR1"   "Control.BR2"   "Treatment.BR1" "Treatment.BR2"
```

DESeq2 needs a minimum amount of additional information regarding columns/biological replicates:

```
#There are two 2 conditions (Control and Treatment) with 4 samples, at 2 biological replicates each
#Biological replicates belong to the same condition
condition_info = data.frame(Condition = c("Control", "Control", "Treatment", "Treatment"))
condition_info
```

```
##   Condition
## 1   Control
## 2   Control
## 3 Treatment
## 4 Treatment
```

```
#Assigns the column names of the count matrix as the row names for each sample
row.names(condition_info) = colnames(raw_rna_seq_data[2:5])
condition_info
```

```
##               Condition
## Control.BR1     Control
## Control.BR2     Control
## Treatment.BR1 Treatment
## Treatment.BR2 Treatment
```

```
#Checks to see that "Condition" in "condition_info" only has two "levels" (Control and Treated)
levels(condition_info$Condition)
```

```
## NULL
```

```
#Be default, R will not recognize that you are trying to cluster the biological replicates together.
#The entries in "Condition" within condition_info are "characters" - they are just a string of charcters with no symolic meaning
```

```
#as.factor() function groups the entries together into "levels" (Control and treatment)
condition_info$Condition = as.factor(condition_info$Condition)
condition_info
```

```
##               Condition
## Control.BR1     Control
## Control.BR2     Control
## Treatment.BR1 Treatment
## Treatment.BR2 Treatment
```

```
#Re-checks to make sure there are only two unique levels within "Condition"
levels(condition_info$Condition)
```

```
## [1] "Control"   "Treatment"
```

Creating the DESeq data object

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

```
## Bioconductor version '3.14' is out-of-date; the current release version '3.15'
##   is available with R version '4.2'; see https://bioconductor.org/install
```

```
BiocManager::install("DESeq2")
```

```
## Bioconductor version 3.14 (BiocManager 1.30.18), R 4.1.3 (2022-03-10)
```

```
## Warning: package(s) not installed when version(s) same as current; use `force = TRUE` to
##   re-install: 'DESeq2'
```

```
#Every package needs to be loaded manually into R with the library() function
library(DESeq2)
```

```
deseq_dataset = DESeqDataSetFromMatrix(countData = final_count_data,
                                       colData = condition_info,
                                       design = ~Condition)
deseq_dataset
```

```
## class: DESeqDataSet
## dim: 4201 4
## metadata(1): version
## assays(1): counts
## rownames(4201): plasmid_0001 plasmid_0002 ... chromosome_4089
##   chromosome_4090
## rowData names(0):
## colnames(4): Control.BR1 Control.BR2 Treatment.BR1 Treatment.BR2
## colData names(1): Condition
```

Conducting differential expression analysis, quality control, and retrieving results

The standard steps for a differential gene expression analysis protocol have been wrapped into a single function, "DESeq"

```
diff_exp_analysis = DESeq(deseq_dataset)
```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

Principal component plots are a way of visualizing mass data between different groups; it is a common way of visualizing data and seeing how close biological replicates are to each other

```
#To be able to effectively compare between-samples, raw count matrix must first be transformed
#vst() function imposes a variance-stablizing transformation on the DESeq object (diff_exp_analysis)
diff_exp_analysis_variance_stabalized = vst(diff_exp_analysis)

#plotPCA() function takes transformed DESeq object as input
plotPCA(diff_exp_analysis_variance_stabalized, intgroup = "Condition")
```


Results of the differential expression analysis are retrieved with the "results" function:

```
diff_exp_results = results(diff_exp_analysis)
diff_exp_results
```

```
## log2 fold change (MLE): Condition Treatment vs Control
## Wald test p-value: Condition Treatment vs Control
## DataFrame with 4201 rows and 6 columns
##                    baseMean log2FoldChange     lfcSE      stat      pvalue
##                   <numeric>      <numeric> <numeric> <numeric>   <numeric>
## plasmid_0001        502.904       0.965054  0.189795   5.08472 3.68163e-07
## plasmid_0002        243.963       0.778384  0.220349   3.53250 4.11653e-04
## plasmid_0003        152.642       0.527147  0.260556   2.02316 4.30566e-02
## plasmid_0004        380.635       0.700175  0.212193   3.29971 9.67865e-04
## plasmid_0005        407.370       0.302956  0.192557   1.57333 1.15642e-01
## ...                     ...            ...       ...       ...         ...
## chromosome_4086    176.9425      0.0396238  0.288361  0.137410   0.8907065
## chromosome_4087    176.4232      0.0480236  0.241658  0.198726   0.8424773
## chromosome_4088    566.4100      0.3843858  0.205275  1.872541   0.0611319
## chromosome_4089     17.7041     -1.0148012  0.612887 -1.655771   0.0977682
## chromosome_4090 370886.1718      0.2127649  0.251002  0.847663   0.3966259
##                        padj
##                   <numeric>
## plasmid_0001    3.83183e-06
## plasmid_0002    1.99809e-03
## plasmid_0003    9.83344e-02
## plasmid_0004    4.25373e-03
## plasmid_0005    2.13862e-01
## ...                     ...
## chromosome_4086    0.928892
## chromosome_4087    0.896057
## chromosome_4088    0.129704
## chromosome_4089    0.187385
## chromosome_4090    0.539494
```

```
#Dpylr is a library for common data transformations, it might need to be installed before using
library(dplyr, quietly = TRUE)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:Biobase':
##
##     combine
```

```
## The following object is masked from 'package:matrixStats':
##
##     count
```

```
## The following objects are masked from 'package:GenomicRanges':
##
##     intersect, setdiff, union
```

```
## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect
```

```
## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union
```

```
## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union
```

```
## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, union
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#subset() takes the results of the differential expression analysis and removes all entries with adjusted p value < 0.05
diff_exp_results_subset = subset(diff_exp_results, padj < 0.05)
diff_exp_results_subset
```

```
## log2 fold change (MLE): Condition Treatment vs Control
## Wald test p-value: Condition Treatment vs Control
## DataFrame with 1536 rows and 6 columns
##                  baseMean log2FoldChange     lfcSE      stat      pvalue
##                 <numeric>      <numeric> <numeric> <numeric>   <numeric>
## plasmid_0001      502.904       0.965054  0.189795   5.08472 3.68163e-07
## plasmid_0002      243.963       0.778384  0.220349   3.53250 4.11653e-04
## plasmid_0004      380.635       0.700175  0.212193   3.29971 9.67865e-04
## plasmid_0007      396.986       0.743554  0.203575   3.65248 2.59723e-04
## plasmid_0008      536.550       0.582527  0.176267   3.30480 9.50457e-04
## ...                   ...            ...       ...       ...         ...
## chromosome_4064 1312.0898      -0.944086  0.187988  -5.02207 5.11188e-07
## chromosome_4067  147.2987      -1.216485  0.281228  -4.32561 1.52109e-05
## chromosome_4076  174.3578       0.895872  0.258616   3.46411 5.31993e-04
## chromosome_4077   33.3960      -1.353145  0.454434  -2.97765 2.90468e-03
## chromosome_4084   85.6157       0.855650  0.332654   2.57219 1.01058e-02
##                        padj
##                   <numeric>
## plasmid_0001    3.83183e-06
## plasmid_0002    1.99809e-03
## plasmid_0004    4.25373e-03
## plasmid_0007    1.33335e-03
## plasmid_0008    4.19407e-03
## ...                     ...
## chromosome_4064 5.15376e-06
## chromosome_4067 1.11069e-04
## chromosome_4076 2.54093e-03
## chromosome_4077 1.08705e-02
## chromosome_4084 3.08182e-02
```

```
#Orders the subset results table by descending log2foldchange value
diff_exp_results_subset_ordered = diff_exp_results_subset[order(-diff_exp_results_subset$log2FoldChange), , drop = FALSE]
diff_exp_results_subset_ordered
```

```
## log2 fold change (MLE): Condition Treatment vs Control
## Wald test p-value: Condition Treatment vs Control
## DataFrame with 1536 rows and 6 columns
##                  baseMean log2FoldChange     lfcSE      stat       pvalue
##                 <numeric>      <numeric> <numeric> <numeric>    <numeric>
## chromosome_0464  93186.33        7.81214  0.161549   48.3576  0.00000e+00
## chromosome_0465   6755.29        6.29277  0.173206   36.3311 5.22226e-289
## chromosome_0115  10782.72        5.80861  0.164094   35.3981 1.82584e-274
## chromosome_0466   3363.49        5.68865  0.186060   30.5742 2.69305e-205
## chromosome_2799  53633.93        5.52680  0.148934   37.1092 1.99967e-301
## ...                   ...            ...       ...       ...          ...
## chromosome_3409  18465.89       -7.73483  0.169915  -45.5218            0
## chromosome_3410   7793.96       -7.85337  0.191701  -40.9667            0
## chromosome_3411  16420.87       -8.64655  0.191760  -45.0905            0
## chromosome_3412  37500.50       -8.97268  0.180406  -49.7360            0
## chromosome_3413 141657.13       -9.40242  0.165789  -56.7133            0
##                         padj
##                    <numeric>
## chromosome_0464  0.00000e+00
## chromosome_0465 2.18499e-286
## chromosome_0115 6.94481e-272
## chromosome_0466 7.04233e-203
## chromosome_2799 1.04583e-298
## ...                      ...
## chromosome_3409            0
## chromosome_3410            0
## chromosome_3411            0
## chromosome_3412            0
## chromosome_3413            0
```

```
#Writes the final, statistically significant results to a comma-seperated value spreadsheet
#row.names = TRUE ensures the gene names are kept instead of removed
#The final table will be saved as what you title it; the table will be saved in the same location as the .rmd document
write.csv(diff_exp_results_subset_ordered, "diff_exp_results_subset_ordered.csv", row.names = TRUE)
```