Class 13

Sabrina Koldinger (A16368238)

Class 13

In today's class we will explore and analyze data from a published RNA-seq experiment on smooth muscle cells of the airway treated with dexamethasone.

Data import

We have two import files

```
counts <- read.csv("airway_scaledcounts.csv", row.names=1)
metadata <-  read.csv("airway_metadata.csv")
```

```
head(counts)
```

```
                SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516
ENSG00000000003        723        486        904        445       1170
ENSG00000000005          0          0          0          0          0
ENSG00000000419        467        523        616        371        582
ENSG00000000457        347        258        364        237        318
ENSG00000000460         96         81         73         66        118
ENSG00000000938          0          0          1          0          2
                SRR1039517 SRR1039520 SRR1039521
ENSG00000000003       1097        806        604
ENSG00000000005          0          0          0
ENSG00000000419        781        417        509
ENSG00000000457        447        330        324
ENSG00000000460         94        102         74
ENSG00000000938          0          0          0
```

```r
metadata
```

```
          id     dex celltype     geo_id
1 SRR1039508 control   N61311 GSM1275862
2 SRR1039509 treated   N61311 GSM1275863
3 SRR1039512 control  N052611 GSM1275866
4 SRR1039513 treated  N052611 GSM1275867
5 SRR1039516 control  N080611 GSM1275870
6 SRR1039517 treated  N080611 GSM1275871
7 SRR1039520 control  N061011 GSM1275874
8 SRR1039521 treated  N061011 GSM1275875
```

Data explore

Q1. How many genes are in this dataset?

```r
nrow(counts)
```

```
[1] 38694
```

There are 38694 genes.

Q2. How many 'control' cell lines do we have?

```r
table(metadata$dex)
```

```
control treated
      4       4
```

There are 4 control cell lines.

Toy differential gene expression

Time to analyze.

We have 4 control and 4 treated samples/experiments/cloumns. Make sure the metadata id column matches the columns in our count data.

```r
colnames(counts)
```

```
[1] "SRR1039508" "SRR1039509" "SRR1039512" "SRR1039513" "SRR1039516"
[6] "SRR1039517" "SRR1039520" "SRR1039521"
```

```r
metadata$id==colnames(counts)
```

```
[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

To check that all elements of a vector are TRUE we can use the `all()` fxn.

```r
all(metadata$id==colnames(counts))
```

```
[1] TRUE
```

To start I will calculate the means for the control and treated genes and compare them,

__ Identify and extract the `control` columns - Determine the mean value for each gene.

```r
controls= metadata$dex=="control"
controlonly=counts[, controls]
control.mean=apply(controlonly, 1, mean)
head(control.mean)
```

```
ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
         900.75            0.00          520.50          339.75           97.25
ENSG00000000938
           0.75
```

```r
treateds= metadata$dex=="treated"
treatedonly=counts[,treateds]
treated.mean=apply(treatedonly, 1, mean)
head(treated.mean)
```

```
ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
         658.00            0.00          546.00          316.50           78.75
ENSG00000000938
           0.00
```

```r
mean.counts= data.frame(control.mean, treated.mean)
```

```r
plot(mean.counts)
```

Data needs to be transformed to see the data better.

```r
plot(mean.counts, log="xy")
```

```
Warning in xy.coords(x, y, xlabel, ylabel, log): 15032 x values <= 0 omitted
from logarithmic plot

Warning in xy.coords(x, y, xlabel, ylabel, log): 15281 y values <= 0 omitted
from logarithmic plot
```

I want to compare the treated and control values here and will use fold change in log2 units to do this. log2(treated/control)

```
log2fc=log2(mean.counts$treated.mean/mean.counts$control.mean)
mean.counts$log2fc=log2fc
```

No difference in expression levels results in log2 function equaling zero. If the treated is upregulated and is expressed more: twice as much will equal 1 from a log 2 function. It will be a negative result if the treated is down regulated

A common rule of thumb cut-off for calling a gene "differentially expressed" is a log2 fold change value of either >+2 or <-2 for "upregulated" and "downregulated" respectively.

```
head(mean.counts)
```

```
                control.mean treated.mean      log2fc
ENSG00000000003       900.75       658.00 -0.45303916
ENSG00000000005         0.00         0.00         NaN
ENSG00000000419       520.50       546.00  0.06900279
ENSG00000000457       339.75       316.50 -0.10226805
ENSG00000000460        97.25        78.75 -0.30441833
ENSG00000000938         0.75         0.00        -Inf
```

First we need to remove zero count genes as we can't say anything about these genes anyway.

```
to.rm.ind=rowSums(mean.counts[,1:2]==0)> 0
mycounts= mean.counts[!to.rm.ind,]
```

How many genes do we have that we can use for data analysis?

```
nrow(mycounts)
```

```
[1] 21817
```

21817 genes

Q8. Using the up.ind vector above can you determine how many up regulated genes we have at the greater than 2 fc level?

```
up.ind <- mycounts$log2fc > 2
sum(up.ind)
```

```
[1] 250
```

250 genes were up regulated at a greater than 2 fc level.

Q9. Using the down.ind vector above can you determine how many down regulated genes we have at the greater than 2 fc level?

```
down.ind <- mycounts$log2fc < (-2)
sum(down.ind)
```

```
[1] 367
```

367 gene were down regulated.

Q10. Do you trust these results? Why or why not? No, we don't know if these results are statistically significant.

DESeq analysis

Do it properly with the help of DESeq2 package.

```r
library(DESeq2)
```

We have to use a specific data object for working with DESeq.

```r
dds= DESeqDataSetFromMatrix(countData = counts, colData = metadata, design= ~dex)
```

```
converting counts to integer mode

Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
design formula are characters, converting to factors
```

Run our main analysis with `DESeq()` function.

```r
dds= DESeq(dds)
```

```
estimating size factors

estimating dispersions

gene-wise dispersion estimates

mean-dispersion relationship

final dispersion estimates

fitting model and testing
```

TO get the results out of our 'dds" object we can use the `results()` fxn from DESeq.

```r
res= results(dds)
head(res)
```

```
log2 fold change (MLE): dex treated vs control
Wald test p-value: dex treated vs control
DataFrame with 6 rows and 6 columns
                  baseMean log2FoldChange     lfcSE      stat    pvalue
                 <numeric>      <numeric> <numeric> <numeric> <numeric>
```

```
ENSG00000000003 747.194195     -0.3507030  0.168246 -2.084470 0.0371175
ENSG00000000005   0.000000             NA        NA        NA        NA
ENSG00000000419 520.134160      0.2061078  0.101059  2.039475 0.0414026
ENSG00000000457 322.664844      0.0245269  0.145145  0.168982 0.8658106
ENSG00000000460  87.682625     -0.1471420  0.257007 -0.572521 0.5669691
ENSG00000000938   0.319167     -1.7322890  3.493601 -0.495846 0.6200029
                     padj
                <numeric>
ENSG00000000003  0.163035
ENSG00000000005        NA
ENSG00000000419  0.176032
ENSG00000000457  0.961694
ENSG00000000460  0.815849
ENSG00000000938        NA
```

Data visualization/ Volcano Plot

A very common and useful summary results figure from this type of analysis is called a volcano plot. Plot log2fc vs. P-value. We usd the `padj` the adjusted P-valuse for multiple testing.

```r
plot(res$log2FoldChange, -log(res$padj))
```

```
plot( res$log2FoldChange,  -log(res$padj),
      xlab="Log2 FoldChange ",
      ylab="-Log of adjusted P-value")
```

Add color

```
mycols= rep("gray", nrow(res))
mycols[abs(res$log2FoldChange) > 2]="blue"
mycols[abs(res$log2FoldChange) < -2]="blue"

plot( res$log2FoldChange,  -log(res$padj),
      xlab="Log2 FoldChange ",
      ylab="-Log of P-value", col=mycols, abline(v=c(-2,2)))
```

Adding annotation data

Here we load the AnnotationDbi package and the annotation data package for humans org.Hs.eg.db.

```
library("AnnotationDbi")
library("org.Hs.eg.db")
```

```
columns(org.Hs.eg.db)
```

```
 [1] "ACCNUM"       "ALIAS"        "ENSEMBL"      "ENSEMBLPROT"  "ENSEMBLTRANS"
 [6] "ENTREZID"     "ENZYME"       "EVIDENCE"     "EVIDENCEALL"  "GENENAME"
[11] "GENETYPE"     "GO"           "GOALL"        "IPI"          "MAP"
[16] "OMIM"         "ONTOLOGY"     "ONTOLOGYALL"  "PATH"         "PFAM"
[21] "PMID"         "PROSITE"      "REFSEQ"       "SYMBOL"       "UCSCKG"
[26] "UNIPROT"
```

```r
res$symbol=mapIds(org.Hs.eg.db,
                     keys=row.names(res),
                     keytype="ENSEMBL",
                     column="SYMBOL",
                     multiVals="first")
```

```
'select()' returned 1:many mapping between keys and columns
```

```r
head(res)
```

```
log2 fold change (MLE): dex treated vs control
Wald test p-value: dex treated vs control
DataFrame with 6 rows and 7 columns
                  baseMean log2FoldChange     lfcSE      stat    pvalue
                 <numeric>      <numeric> <numeric> <numeric> <numeric>
ENSG00000000003 747.194195     -0.3507030  0.168246 -2.084470 0.0371175
ENSG00000000005   0.000000             NA        NA        NA        NA
ENSG00000000419 520.134160      0.2061078  0.101059  2.039475 0.0414026
ENSG00000000457 322.664844      0.0245269  0.145145  0.168982 0.8658106
ENSG00000000460  87.682625     -0.1471420  0.257007 -0.572521 0.5669691
ENSG00000000938   0.319167     -1.7322890  3.493601 -0.495846 0.6200029
                     padj      symbol
                 <numeric> <character>
ENSG00000000003  0.163035      TSPAN6
ENSG00000000005        NA        TNMD
ENSG00000000419  0.176032        DPM1
ENSG00000000457  0.961694       SCYL3
ENSG00000000460  0.815849       FIRRM
ENSG00000000938        NA         FGR
```

Also want entrez IDs.

```r
res$entrez=mapIds(org.Hs.eg.db,
                     keys=row.names(res),
                     keytype="ENSEMBL",
                     column="ENTREZID",
                     multiVals="first")
```

```
'select()' returned 1:many mapping between keys and columns
```

```r
head(res)
```

```
log2 fold change (MLE): dex treated vs control
Wald test p-value: dex treated vs control
DataFrame with 6 rows and 8 columns
                  baseMean log2FoldChange     lfcSE      stat    pvalue
                 <numeric>      <numeric> <numeric> <numeric> <numeric>
ENSG00000000003 747.194195     -0.3507030  0.168246 -2.084470 0.0371175
ENSG00000000005   0.000000             NA        NA        NA        NA
ENSG00000000419 520.134160      0.2061078  0.101059  2.039475 0.0414026
ENSG00000000457 322.664844      0.0245269  0.145145  0.168982 0.8658106
ENSG00000000460  87.682625     -0.1471420  0.257007 -0.572521 0.5669691
ENSG00000000938   0.319167     -1.7322890  3.493601 -0.495846 0.6200029
                     padj      symbol      entrez
                 <numeric> <character> <character>
ENSG00000000003  0.163035      TSPAN6        7105
ENSG00000000005        NA        TNMD       64102
ENSG00000000419  0.176032        DPM1        8813
ENSG00000000457  0.961694       SCYL3       57147
ENSG00000000460  0.815849       FIRRM       55732
ENSG00000000938        NA         FGR        2268
```

Add genename column

```r
res$name=mapIds(org.Hs.eg.db,
                keys=row.names(res),
                keytype="ENSEMBL",
                column="GENENAME",
                multiVals="first")
```

```
'select()' returned 1:many mapping between keys and columns
```

```r
head(res)
```

```
log2 fold change (MLE): dex treated vs control
Wald test p-value: dex treated vs control
DataFrame with 6 rows and 9 columns
                  baseMean log2FoldChange     lfcSE      stat    pvalue
                 <numeric>      <numeric> <numeric> <numeric> <numeric>
```

```
ENSG00000000003 747.194195     -0.3507030  0.168246 -2.084470 0.0371175
ENSG00000000005   0.000000             NA        NA        NA        NA
ENSG00000000419 520.134160      0.2061078  0.101059  2.039475 0.0414026
ENSG00000000457 322.664844      0.0245269  0.145145  0.168982 0.8658106
ENSG00000000460  87.682625     -0.1471420  0.257007 -0.572521 0.5669691
ENSG00000000938   0.319167     -1.7322890  3.493601 -0.495846 0.6200029
                    padj      symbol      entrez                   name
                <numeric> <character> <character>            <character>
ENSG00000000003  0.163035      TSPAN6        7105          tetraspanin 6
ENSG00000000005        NA        TNMD       64102            tenomodulin
ENSG00000000419  0.176032        DPM1        8813 dolichyl-phosphate m..
ENSG00000000457  0.961694       SCYL3       57147 SCY1 like pseudokina..
ENSG00000000460  0.815849       FIRRM       55732 FIGNL1 interacting r..
ENSG00000000938        NA         FGR        2268 FGR proto-oncogene, ..
```

Pathway analysis

Now with the necessary annotation data, we can use different databases that use the IDs.

The `gage` packacge is used to do geneset analysis(or pathway analysis, geneset enrichment, overlap analysis).

```r
library(pathview)
```

```
##############################################################################
Pathview is an open source software package distributed under GNU General
Public License version 3 (GPLv3). Details of GPLv3 is available at
http://www.gnu.org/licenses/gpl-3.0.html. Particullary, users are required to
formally cite the original Pathview paper (not just mention it) in publications
or products. For details, do citation("pathview") within R.

The pathview downloads and uses KEGG data. Non-academic uses may require a KEGG
license agreement (details at http://www.kegg.jp/kegg/legal.html).
##############################################################################
```

```r
library(gage)
```

```r
library(gageData)

data(kegg.sets.hs)
```

We will use KEGG first

```r
head(kegg.sets.hs, 2)
```

```
$`hsa00232 Caffeine metabolism`
[1] "10"   "1544" "1548" "1549" "1553" "7498" "9"

$`hsa00983 Drug metabolism - other enzymes`
 [1] "10"     "1066"   "10720"  "10941"  "151531" "1548"   "1549"   "1551"
 [9] "1553"   "1576"   "1577"   "1806"   "1807"   "1890"   "221223" "2990"
[17] "3251"   "3614"   "3615"   "3704"   "51733"  "54490"  "54575"  "54576"
[25] "54577"  "54578"  "54579"  "54600"  "54657"  "54658"  "54659"  "54963"
[33] "574537" "64816"  "7083"   "7084"   "7172"   "7363"   "7364"   "7365"
[41] "7366"   "7367"   "7371"   "7372"   "7378"   "7498"   "79799"  "83549"
[49] "8824"   "8833"   "9"      "978"
```

The main `gage()` function requires a named vector of fold changes, where the names of the values are the Entrez gene IDs.

```r
foldchange= res$log2FoldChange
names(foldchange)= res$symbol
head(foldchange)
```

```
     TSPAN6        TNMD        DPM1       SCYL3       FIRRM         FGR
-0.35070302          NA  0.20610777  0.02452695 -0.14714205 -1.73228897
```

Run the analysis

```r
keggres = gage(foldchange, gsets=kegg.sets.hs)
```

Lets lookat what is in our results here.

```r
attributes(keggres)
```

```
$names
[1] "greater" "less"    "stats"
```

```
head(keggres$less, 3)
```

```
                                        p.geomean stat.mean p.val q.val
hsa00232 Caffeine metabolism                   NA       NaN    NA    NA
hsa00983 Drug metabolism - other enzymes       NA       NaN    NA    NA
hsa01100 Metabolic pathways                    NA       NaN    NA    NA
                                        set.size exp1
hsa00232 Caffeine metabolism                   0   NA
hsa00983 Drug metabolism - other enzymes       0   NA
hsa01100 Metabolic pathways                    0   NA
```

I can now use the returned pathway IDs from KEGG as input to the `pathview` package.

```
pathview(gene.data=foldchange, pathway.id="hsa05310")
```

```
Warning: None of the genes or compounds mapped to the pathway!
Argument gene.idtype or cpd.idtype may be wrong.

'select()' returned 1:1 mapping between keys and columns

Info: Working in directory /Users/sabrinakoldinger/Desktop/BIMM 143 Bioinformatics/Class13

Info: Writing image file hsa05310.pathview.png
```