

Class 14: RNASeq Mini Project

Mudit

Import Data

we need two things “Counts” and “MetaData” (what DESeq calls colData - as it describes the columns in Counts).

```
counts <- read.csv("GSE37704_featurecounts.csv", row.names = 1)
metadata <- read.csv("GSE37704_metadata.csv")
```

Data CleanUp

Start with a wee peak:

```
head(counts)
```

| | length | SRR493366 | SRR493367 | SRR493368 | SRR493369 | SRR493370 |
|-----------------|-----------|-----------|-----------|-----------|-----------|-----------|
| ENSG00000186092 | 918 | 0 | 0 | 0 | 0 | 0 |
| ENSG00000279928 | 718 | 0 | 0 | 0 | 0 | 0 |
| ENSG00000279457 | 1982 | 23 | 28 | 29 | 29 | 28 |
| ENSG00000278566 | 939 | 0 | 0 | 0 | 0 | 0 |
| ENSG00000273547 | 939 | 0 | 0 | 0 | 0 | 0 |
| ENSG00000187634 | 3214 | 124 | 123 | 205 | 207 | 212 |
| | SRR493371 | | | | | |
| ENSG00000186092 | 0 | | | | | |
| ENSG00000279928 | 0 | | | | | |
| ENSG00000279457 | 46 | | | | | |
| ENSG00000278566 | 0 | | | | | |
| ENSG00000273547 | 0 | | | | | |
| ENSG00000187634 | 258 | | | | | |

```
head(metadata)
```

```
      id      condition
1 SRR493366 control_sirna
2 SRR493367 control_sirna
3 SRR493368 control_sirna
4 SRR493369      hoxa1_kd
5 SRR493370      hoxa1_kd
6 SRR493371      hoxa1_kd
```

We want the columns in `counts` to match the rows in the `metadata`.

```
colnames(counts)
```

```
[1] "length"      "SRR493366" "SRR493367" "SRR493368" "SRR493369" "SRR493370"
[7] "SRR493371"
```

```
metadata$id
```

```
[1] "SRR493366" "SRR493367" "SRR493368" "SRR493369" "SRR493370" "SRR493371"
```

we can get rid of the first column in `counts` to make these match

```
countData <- counts[,-1]
head(countData)
```

| | SRR493366 | SRR493367 | SRR493368 | SRR493369 | SRR493370 | SRR493371 |
|-----------------|-----------|-----------|-----------|-----------|-----------|-----------|
| ENSG00000186092 | 0 | 0 | 0 | 0 | 0 | 0 |
| ENSG00000279928 | 0 | 0 | 0 | 0 | 0 | 0 |
| ENSG00000279457 | 23 | 28 | 29 | 29 | 28 | 46 |
| ENSG00000278566 | 0 | 0 | 0 | 0 | 0 | 0 |
| ENSG00000273547 | 0 | 0 | 0 | 0 | 0 | 0 |
| ENSG00000187634 | 124 | 123 | 205 | 207 | 212 | 258 |

```
#compare all the entries
all(colnames(countData) == metadata$id)
```

```
[1] TRUE
```

```
all(c(T,T,T,T))
```

```
[1] TRUE
```

```
all(c(T,T,T,F))
```

```
[1] FALSE
```

```
x <- c(T,F,T,T)
if(all(x)){
  cat("Me Happy")
}else{
  cat("me sad")
}
```

```
me sad
```

If the standard practice to remove any genes/transcripts that we have no data for - i.e. zero counts in all columns

```
tp.keep.inds <- rowSums(countData) > 0
cleanCounts <- countData[tp.keep.inds,]
```

Setup for DESeq

```
#!/ message: false
library(DESeq2)
```

```
Loading required package: S4Vectors
```

```
Loading required package: stats4
```

```
Loading required package: BiocGenerics
```

```
Attaching package: 'BiocGenerics'
```

The following objects are masked from 'package:stats':

IQR, mad, sd, var, xtabs

The following objects are masked from 'package:base':

anyDuplicated, aperm, append, as.data.frame, basename, cbind,
colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
Position, rank, rbind, Reduce, rownames, sapply, saveRDS, setdiff,
table, tapply, union, unique, unsplit, which.max, which.min

Attaching package: 'S4Vectors'

The following object is masked from 'package:utils':

findMatches

The following objects are masked from 'package:base':

expand.grid, I, unname

Loading required package: IRanges

Loading required package: GenomicRanges

Loading required package: GenomeInfoDb

Loading required package: SummarizedExperiment

Loading required package: MatrixGenerics

Loading required package: matrixStats

Attaching package: 'MatrixGenerics'

The following objects are masked from 'package:matrixStats':

```
colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
colWeightedMeans, colWeightedMedians, colWeightedSds,
colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
rowWeightedSds, rowWeightedVars
```

Loading required package: Biobase

Welcome to Bioconductor

```
Vignettes contain introductory material; view with
'browseVignettes()'. To cite Bioconductor, see
'citation("Biobase")', and for packages 'citation("pkgname")'.
```

Attaching package: 'Biobase'

The following object is masked from 'package:MatrixGenerics':

```
rowMedians
```

The following objects are masked from 'package:matrixStats':

```
anyMissing, rowMedians
```

```
#!/ message: false
dds <- DESeqDataSetFromMatrix(countData = cleanCounts,
                              colData = metadata,
                              design = ~condition)
```

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

DESeq

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

estimating size factors

estimating dispersions

gene-wise dispersion estimates

mean-dispersion relationship

final dispersion estimates

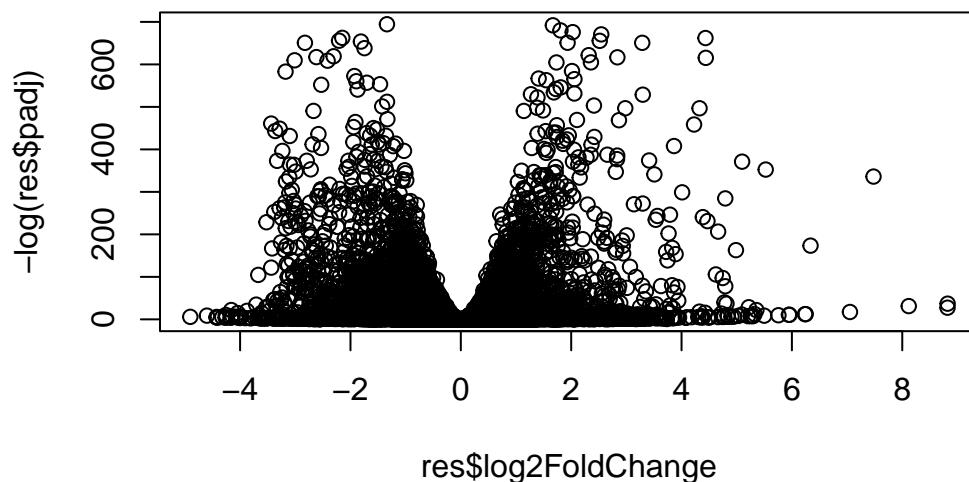
fitting model and testing

```
res = results(dds)
```

Inspect Results

Data Viz

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



```
head(res)
```

log2 fold change (MLE): condition hoxa1 kd vs control sirna

Wald test p-value: condition hoxa1 kd vs control sirna

DataFrame with 6 rows and 6 columns

| | baseMean | log2FoldChange | lfcSE | stat | pvalue |
|-----------------|-------------|----------------|-----------|------------|-------------|
| | <numeric> | <numeric> | <numeric> | <numeric> | <numeric> |
| ENSG00000279457 | 29.9136 | 0.1792571 | 0.3248216 | 0.551863 | 5.81042e-01 |
| ENSG00000187634 | 183.2296 | 0.4264571 | 0.1402658 | 3.040350 | 2.36304e-03 |
| ENSG00000188976 | 1651.1881 | -0.6927205 | 0.0548465 | -12.630158 | 1.43990e-36 |
| ENSG00000187961 | 209.6379 | 0.7297556 | 0.1318599 | 5.534326 | 3.12428e-08 |
| ENSG00000187583 | 47.2551 | 0.0405765 | 0.2718928 | 0.149237 | 8.81366e-01 |
| ENSG00000187642 | 11.9798 | 0.5428105 | 0.5215598 | 1.040744 | 2.97994e-01 |
| | padj | | | | |
| | <numeric> | | | | |
| ENSG00000279457 | 6.86555e-01 | | | | |
| ENSG00000187634 | 5.15718e-03 | | | | |
| ENSG00000188976 | 1.76549e-35 | | | | |
| ENSG00000187961 | 1.13413e-07 | | | | |
| ENSG00000187583 | 9.19031e-01 | | | | |
| ENSG00000187642 | 4.03379e-01 | | | | |

Pathway Analysis

Annotation of Genes

First I need to translate my Ensembl IDs in my `res` object to Entrez and gene symbol formats.

For this I will use the Annotationa DBI Package and its `mapIDs()` function

```
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" | | | | |

Let's map to "SYMBOLS", "ENTREZID", "GENENAME" from our "ENSEMBL" ids

```
res$genename <- mapIds(org.Hs.eg.db,
  keys = rownames(res),
  keytype = "ENSEMBL",
  column = "GENENAME")
```

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

```
res$symbol <- mapIds(org.Hs.eg.db,
  keys = rownames(res),
  keytype = "ENSEMBL",
  column = "SYMBOL")
```

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


```
res$entrez <- mapIds(org.Hs.eg.db,
                     keys = rownames(res),
                     keytype = "ENSEMBL",
                     column = "ENTREZID")
```

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

```
head(res)
```

log2 fold change (MLE): condition hoxa1 kd vs control sirna

Wald test p-value: condition hoxa1 kd vs control sirna

DataFrame with 6 rows and 9 columns

| | baseMean | log2FoldChange | lfcSE | stat | pvalue |
|-----------------|-----------|----------------|-----------|------------|-------------|
| | <numeric> | <numeric> | <numeric> | <numeric> | <numeric> |
| ENSG00000279457 | 29.9136 | 0.1792571 | 0.3248216 | 0.551863 | 5.81042e-01 |
| ENSG00000187634 | 183.2296 | 0.4264571 | 0.1402658 | 3.040350 | 2.36304e-03 |
| ENSG00000188976 | 1651.1881 | -0.6927205 | 0.0548465 | -12.630158 | 1.43990e-36 |
| ENSG00000187961 | 209.6379 | 0.7297556 | 0.1318599 | 5.534326 | 3.12428e-08 |
| ENSG00000187583 | 47.2551 | 0.0405765 | 0.2718928 | 0.149237 | 8.81366e-01 |
| ENSG00000187642 | 11.9798 | 0.5428105 | 0.5215598 | 1.040744 | 2.97994e-01 |

| | padj | genename | symbol | entrez |
|-----------------|-------------|------------------------|-------------|-------------|
| | <numeric> | <character> | <character> | <character> |
| ENSG00000279457 | 6.86555e-01 | NA | NA | NA |
| ENSG00000187634 | 5.15718e-03 | sterile alpha motif .. | SAMD11 | 148398 |
| ENSG00000188976 | 1.76549e-35 | NOC2 like nucleolar .. | NOC2L | 26155 |
| ENSG00000187961 | 1.13413e-07 | kelch like family me.. | KLHL17 | 339451 |
| ENSG00000187583 | 9.19031e-01 | pleckstrin homology .. | PLEKHN1 | 84069 |
| ENSG00000187642 | 4.03379e-01 | PPARGC1 and ESRR ind.. | PERM1 | 84808 |

```
#rownames(res)
```

Before going any further lets focus in on a subset of “top” hits

We can use as a starting point log2FC of +2/-2 and a adjusted P-value of 0.05

```
top.inds <- (abs(res$log2FoldChange) > 2) & (res$padj < 0.05)
top.inds[is.na(top.inds)] = FALSE
```

```
c(T,T,T,F) & c(T,T,F,F)
```

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

Let's save our "top genes" to a CSV file...

```
top.genes <- res[top.inds,]  
write.csv(top.genes, file = "top_geneset.csv")
```

```
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>).

```
#####
```

```
library(gage)
```

```
library(gageData)  
  
data(kegg.sets.hs)  
data(sigmet.idx.hs)  
  
# Focus on signaling and metabolic pathways only  
kegg.sets.hs = kegg.sets.hs[sigmet.idx.hs]  
  
# Examine the first 3 pathways  
head(kegg.sets.hs, 3)
```

```
$`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"

$`hsa00230 Purine metabolism`
[1] "100"      "10201"   "10606"   "10621"   "10622"   "10623"   "107"     "10714"
[9] "108"      "10846"   "109"     "111"     "11128"   "11164"   "112"     "113"
[17] "114"      "115"     "122481"  "122622"  "124583"  "132"     "158"     "159"
[25] "1633"     "171568"  "1716"    "196883"  "203"     "204"     "205"     "221823"
[33] "2272"     "22978"   "23649"   "246721"  "25885"   "2618"    "26289"   "270"
[41] "271"     "27115"   "272"     "2766"    "2977"    "2982"    "2983"    "2984"
[49] "2986"     "2987"    "29922"   "3000"    "30833"   "30834"   "318"     "3251"
[57] "353"      "3614"    "3615"    "3704"    "377841"  "471"     "4830"    "4831"
[65] "4832"     "4833"    "4860"    "4881"    "4882"    "4907"    "50484"   "50940"
[73] "51082"    "51251"   "51292"   "5136"    "5137"    "5138"    "5139"    "5140"
[81] "5141"     "5142"    "5143"    "5144"    "5145"    "5146"    "5147"    "5148"
[89] "5149"     "5150"    "5151"    "5152"    "5153"    "5158"    "5167"    "5169"
[97] "51728"    "5198"    "5236"    "5313"    "5315"    "53343"   "54107"   "5422"
[105] "5424"     "5425"    "5426"    "5427"    "5430"    "5431"    "5432"    "5433"
[113] "5434"     "5435"    "5436"    "5437"    "5438"    "5439"    "5440"    "5441"
[121] "5471"     "548644"  "55276"   "5557"    "5558"    "55703"   "55811"   "55821"
[129] "5631"     "5634"    "56655"   "56953"   "56985"   "57804"   "58497"   "6240"
[137] "6241"     "64425"   "646625"  "654364"  "661"     "7498"    "8382"    "84172"
[145] "84265"    "84284"   "84618"   "8622"    "8654"    "87178"   "8833"    "9060"
[153] "9061"     "93034"   "953"     "9533"    "954"     "955"     "956"     "957"
[161] "9583"     "9615"

```

The **gage** function wants a vector of importance as input with gene names as labels - KEGG speaks ENTREZ

```

foldchanges = res$log2FoldChange
names(foldchanges) = res$entrez
head(foldchanges)

```

```

      <NA>      148398      26155      339451      84069      84808
0.17925708 0.42645712 -0.69272046 0.72975561 0.04057653 0.54281049

```

Run gage with these values

```
# Get the results
keggres <- gage(foldchanges, gsets=kegg.sets.hs)
```

```
attributes(keggres)
```

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

```
# Look at the first few down (less) pathways
head(keggres$less)
```

| | p.geomean | stat.mean | p.val |
|---------------------------------------|--------------|-----------|--------------|
| hsa04110 Cell cycle | 8.995727e-06 | -4.378644 | 8.995727e-06 |
| hsa03030 DNA replication | 9.424076e-05 | -3.951803 | 9.424076e-05 |
| hsa03013 RNA transport | 1.246882e-03 | -3.059466 | 1.246882e-03 |
| hsa03440 Homologous recombination | 3.066756e-03 | -2.852899 | 3.066756e-03 |
| hsa04114 Oocyte meiosis | 3.784520e-03 | -2.698128 | 3.784520e-03 |
| hsa00010 Glycolysis / Gluconeogenesis | 8.961413e-03 | -2.405398 | 8.961413e-03 |

| | q.val | set.size | exp1 |
|---------------------------------------|-------------|----------|--------------|
| hsa04110 Cell cycle | 0.001448312 | 121 | 8.995727e-06 |
| hsa03030 DNA replication | 0.007586381 | 36 | 9.424076e-05 |
| hsa03013 RNA transport | 0.066915974 | 144 | 1.246882e-03 |
| hsa03440 Homologous recombination | 0.121861535 | 28 | 3.066756e-03 |
| hsa04114 Oocyte meiosis | 0.121861535 | 102 | 3.784520e-03 |
| hsa00010 Glycolysis / Gluconeogenesis | 0.212222694 | 53 | 8.961413e-03 |

```
pathview(gene.data=foldchanges, pathway.id="hsa04110")
```

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

Info: Working in directory /Users/muditg19/Downloads/BGCN213/Class 14

Info: Writing image file hsa04110.pathview.png

```
data(go.sets.hs)
data(go.subs.hs)
```

```
# Focus on Biological Process subset of GO
```

```
gobpsets = go.sets.hs[go.subs.hs$BP]

gores = gage(foldchanges, gsets=gobpsets, same.dir=TRUE)

head(gores$less)
```

| | | p.geomean | stat.mean | p.val |
|------------|-------------------------------|--------------|-----------|--------------|
| G0:0048285 | organelle fission | 1.536227e-15 | -8.063910 | 1.536227e-15 |
| G0:0000280 | nuclear division | 4.286961e-15 | -7.939217 | 4.286961e-15 |
| G0:0007067 | mitosis | 4.286961e-15 | -7.939217 | 4.286961e-15 |
| G0:0000087 | M phase of mitotic cell cycle | 1.169934e-14 | -7.797496 | 1.169934e-14 |
| G0:0007059 | chromosome segregation | 2.028624e-11 | -6.878340 | 2.028624e-11 |
| G0:0000236 | mitotic prometaphase | 1.729553e-10 | -6.695966 | 1.729553e-10 |
| | | q.val | set.size | exp1 |
| G0:0048285 | organelle fission | 5.841698e-12 | 376 | 1.536227e-15 |
| G0:0000280 | nuclear division | 5.841698e-12 | 352 | 4.286961e-15 |
| G0:0007067 | mitosis | 5.841698e-12 | 352 | 4.286961e-15 |
| G0:0000087 | M phase of mitotic cell cycle | 1.195672e-11 | 362 | 1.169934e-14 |
| G0:0007059 | chromosome segregation | 1.658603e-08 | 142 | 2.028624e-11 |
| G0:0000236 | mitotic prometaphase | 1.178402e-07 | 84 | 1.729553e-10 |

TO run reactome online we need to make a little text file with a gene id per line,

```
sig_genes <- res[res$padj <= 0.05 & !is.na(res$padj), "symbol"]
print(paste("Total number of significant genes:", length(sig_genes)))
```

```
[1] "Total number of significant genes: 8147"
```

```
write.table(sig_genes, file="significant_genes.txt", row.names=FALSE, col.names=FALSE, quote=
```

