Class 13 RNA Seq

Sindy Chavez

Background

We will do all the following:

- Read countData and colData
- Check and fix countData if required
- DESeq Analysis
- Visualization
- Gene Annotation
- Pathway Analysis

About the data

The data for for hands-on session comes from GEO entry: GSE37704, which is associated with the following publication:

Trapnell C, Hendrickson DG, Sauvageau M, Goff L et al. "Differential analysis of gene regulation at transcript resolution with RNA-seq". Nat Biotechnol 2013 Jan;31(1):46-53. PMID: 23222703

1. Read countData and colData

We need at least two things for this tupe of analysis:

- countData
- colData (a.k.a. Metadata)

```
colData <- read.csv("GSE37704_metadata.csv", row.names = 1)
colData
```

```
             condition
SRR493366 control_sirna
SRR493367 control_sirna
SRR493368 control_sirna
SRR493369      hoxa1_kd
SRR493370      hoxa1_kd
SRR493371      hoxa1_kd
```

```
countData <- read.csv("GSE37704_featurecounts.csv", row.names = 1)
head(countData)
```

```
                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
```

2. Fix countData

> Q. Complete the code below to remove the troublesome first column from countData

length is in the first column position, and we need to get rid of it. You can do [,-1], but don't run many times or you risk removing the first column over and over again

```
countData <- as.matrix(countData[,-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

Make sure that the data in CountData match the metadata

```
all(rownames(colData)==colnames(countData))
```

```
[1] TRUE
```

All looks good apart from all those zero count genes!! We should remove these before doing any testing.

> Q. Complete the code below to filter countData to exclude genes (i.e. rows) where we have 0 read count across all samples (i.e. columns).

We can sum across the rows and if we get a zero then we have no counts in any exp for a given gene.

```
keep.inds <- rowSums(countData)!=0
counts <- countData[keep.inds, ]
```

```
nrow(counts)
```

```
[1] 15975
```

QC with PCA

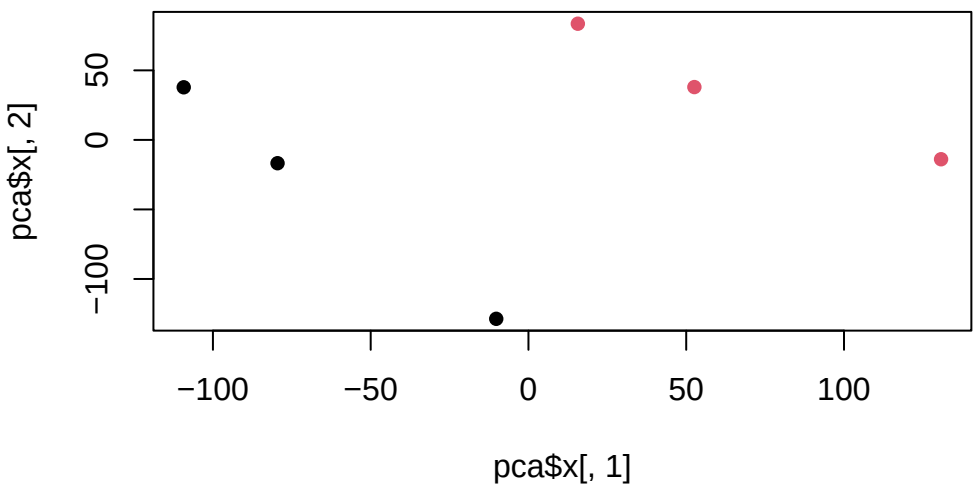
The 'prcomp()' function in base R is often used to check the

```
pca <- prcomp(t(counts), scale. = TRUE)
summary(pca)
```

```
Importance of components:
                          PC1     PC2      PC3      PC4      PC5       PC6
Standard deviation     87.7211 73.3196 32.89604 31.15094 29.18417 6.648e-13
Proportion of Variance  0.4817  0.3365  0.06774  0.06074  0.05332 0.000e+00
Cumulative Proportion   0.4817  0.8182  0.88594  0.94668  1.00000 1.000e+00
```

Our PCA score plot (a.k.a. PC1 vs PC2)

```
plot(pca$x[,1], pca$x[,2], col=as.factor(colData$condition), pch=16)
```



3. DESeq

```
library(DESeq2)
```

First I need to setup the input object required by DESeq

```
dds <- DESeqDataSetFromMatrix(countData = counts,
                              colData=colData,
                              design=~condition)
```

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

```
dds <- DESeq(dds)
res <- results(dds)
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
```

Q. Call the summary() function on your results to get a sense of how many genes are up or down-regulated at the default 0.1 p-value cutoff.

```
summary(res)
```

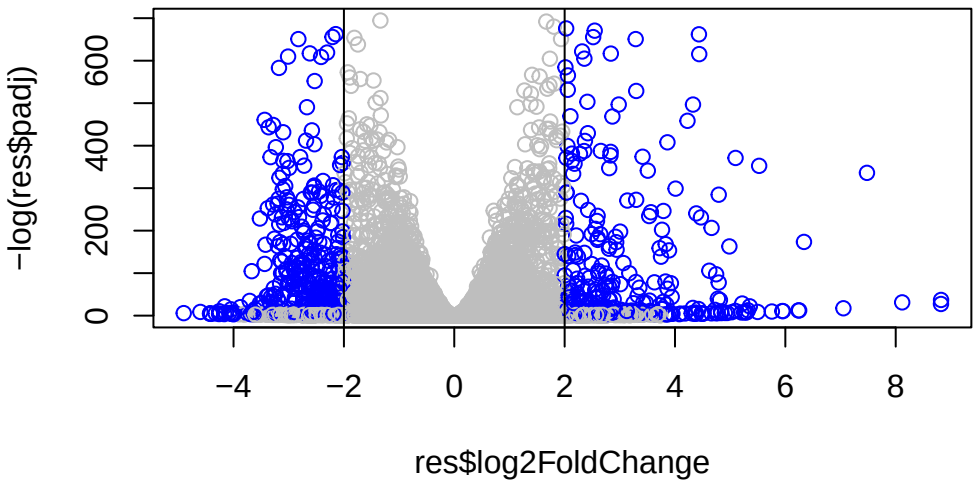
```
out of 15975 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)       : 4349, 27%
LFC < 0 (down)     : 4396, 28%
outliers [1]       : 0, 0%
low counts [2]     : 1237, 7.7%
(mean count < 0)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

Q. Improve this plot by completing the below code, which adds color and axis labels

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

```
mycols[res$padj > 0.05] <- "gray"

plot(res$log2FoldChange, -log(res$padj), col=mycols)
abline(v=c(-2,+2))
```



4. Add Gene Annotation

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

Q. Use the mapIDs() function multiple times to add SYMBOL, ENTREZID and GENENAME annotation to our results by completing the code below.

I will use the 'mapIDs()' to add SYMBOL and ENTREZID annotation to our results.

```
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"
```

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

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

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

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

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

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

Q. Finally for this section let's reorder these results by adjusted p-value and save them to a CSV file in your current project directory.

```
res = res[order(res$pvalue),]
write.csv(res,file = "deseq_results.csv")
```

5. Pathway Analysis or Gene Set Enrichment

We can use 'gage()' with KEGG and GO

```
library(gage)
library(gageData)
library(pathview)
```

What 'gage()' wants as input is that vector of importance - in our case that will be the log2 fold-change values. This vector should have 'names()' that are entrez IDs.

```
foldchange <- res$log2FoldChange
names(foldchange) <- res$entrez
```

And run gage with KEGG Human set

```
#Focus on signaling and metabolic pathways only
data("kegg.sets.hs")
data("sigmet.idx.hs")
kegg.sets.hs = kegg.sets.hs[sigmet.idx.hs]

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

```
head(keggres$less, 5)
```

```
                                      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             1.375901e-03 -3.028500 1.375901e-03
hsa03440 Homologous recombination  3.066756e-03 -2.852899 3.066756e-03
hsa04114 Oocyte meiosis            3.784520e-03 -2.698128 3.784520e-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             0.073840037      144 1.375901e-03
hsa03440 Homologous recombination  0.121861535       28 3.066756e-03
hsa04114 Oocyte meiosis            0.121861535      102 3.784520e-03
```

Let's have a closer look at one of these pathways and our intersecting genes

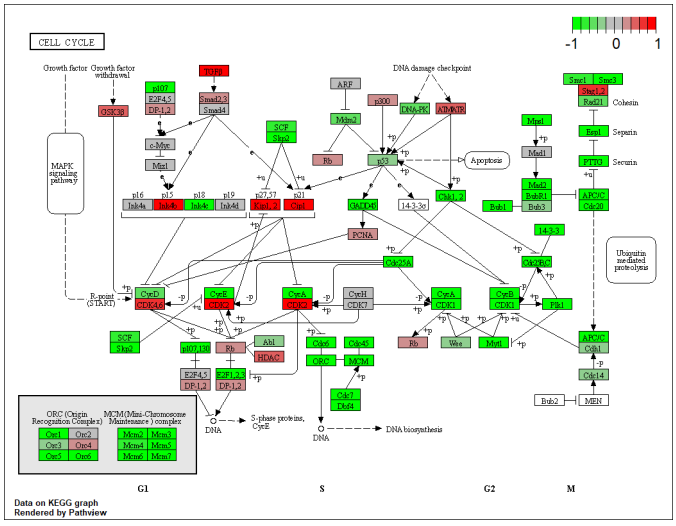
hsa04110 Cell cycle

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

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

```
Info: Working in directory C:/Users/sindy/OneDrive/Documents/BGGN213 Bioinformatics/Class Not
```

```
Info: Writing image file hsa04110.pathview.png
```



6. Gene Ontology (GO)

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

# Focus on Biological Process subset of GO
gobpsets = go.sets.hs[go.subs.hs$BP]

gobpres = gage(foldchange, gsets=gobpsets, same.dir=TRUE)

head(gobpres$less)
```

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

Reactome Analysis

We can use the online version for a fancy display

```
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"
```

```
#sig_genes
```

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

Q: What pathway has the most significant "Entities p-value"? Do the most significant pathways listed match your previous KEGG results? What factors could cause differences between the two methods?

Endosomal/Vacuolar pathway No