RNAseq: Visualizations

C. Ryan Campbell

Duke University

c.ryan.campbell@duke.edu

9 Nov 2017

Overview

1. DESeq2
   - cluster
   - RStudio
2. Running DESeq2
3. Plotting DESeq2
   - Check Data Generation
   - Check for Outliers
4. DESeq2 & ggplot
   - Managing the Data
   - Plotting
   - Investigating Further
   - DESeq & gene families

Today's Goals

- Basic Plots
- ggplot and DESeq
- Advanced Plots

Deleting Old Files

- If you're finished with your downloaded fastq files PLEASE delete them!

```
rm -r
rm *fastq
```

- The -r command removes directories
- Or you can use the "*" to select all reads ending in .fastq

Combining htseq Output

- And `cut` to eliminate redundant columns:

```
> paste s01.norm.counts s02.norm.counts s03.hypo.counts
s04.hypo.counts | cut -f1,2,4,6,8
3.8-1.4 0 0 0 0
3.8-1.5 0 0 0 0
5-HT3C2 0 0 0 0
A1BG 252 192 175 153
A1BG-AS1 47 28 31 35
```

Finally write to a file (notice the grep to remove non-gene rows):

```
> paste s01.norm.counts s02.norm.counts s03.hypo.counts
s04.hypo.counts | cut -f1,2,4,6,8 | grep -v __ >
hypoxia_hsap.counts
```

DESeq2

- See the website for installation instructions
- Needs two things:
  1. A matrix of counts = “cts”
  2. A matrix of sample conditions = “coldata”

DESeq2

In R:

```
> head(cts, n = 4)
s01norm s02norm s03hypox s04hypox
3.8-1.4 0 0 0 0
3.8-1.5 0 0 0 0
5-HT3C2 0 0 0 0
A1BG 252 192 175 153

> coldata
condition type
s01norm "untreated" "paired-end"
s02norm "untreated" "paired-end"
s03hypox "treated" "paired-end"
s04hypox "treated" "paired-end"
```

DESeq2

- Order must match (01, 02, 03...)
- Name the columns and rows appropriately

```
> rownames(coldata)
[1] "s01norm" "s02norm" "s03hypox" "s04hypox"


> colnames(coldata)
[1] "condition" "type"
```

You use the same function to call (see above) define (see below) column and row names:

```
> colnames(coldata) <- c("condition","type")
```

DESeq2

- Get your data in this format
- Keep track of your work in a .Rmd file!!
- Then (and only then) proceed to running DESeq (see further slides)

DESeq2 Formatting Tips: Reading Data

- In R:

```
> countfile <- read.table("hypoxia_hsap.counts")
> head(countfile, n = 5)
V1 V2 V3 V4 V5
1 3.8-1.4 0 0 0 0
2 3.8-1.5 0 0 0 0
3 5-HT3C2 0 0 0 0
4 A1BG 252 192 175 153
```

Now we need to get the data into matrices with the correct row and column names

DESeq2 Formatting Tips: Transforming Data

- `countfile` contains the count data
- We need to format it appropriately

```
> cts <- as.matrix(countfile[,2:5])
> colnames(cts) <- c("s01norm","s02norm","s03hypox","s04hypox")
> rownames(cts) <- countfile[,1]
> head(cts, n = 4)
s01norm s02norm s03hypox s04hypox
3.8-1.4 0 0 0 0
3.8-1.5 0 0 0 0
5-HT3C2 0 0 0 0
A1BG 252 192 175 153
```

First take in only the data (columns 2-5) as a matrix
Then name the columns the sample IDs

DESeq2 Formatting: coldata matrix

- `coldata` needs to contain the appropriate sample info
- Each row is a sample, each column is information about the sample
- Experimental Treatment, Read Format, Cell Type can all be pertinent info
- First we will make a vector with the information
- Then we'll take the data and make it into a 2x4 matrix

DESeq2 Formatting: coldata matrix

```
> sampleinfo <- c("untreated","untreated","treated","treated",
"paired-end","paired-end","paired-end","paired-end")
> sampleinfo
[1] "untreated" "untreated" "treated" "treated" "paired-end"
"paired-end"
[7] "paired-end" "paired-end"
> coldata <- matrix(sampleinfo, nrow = 4, ncol = 2, byrow = F)
> coldata
[,1] [,2]
[1,] "untreated" "paired-end"
[2,] "untreated" "paired-end"
[3,] "treated" "paired-end"
[4,] "treated" "paired-end"
```

DESeq2 Formatting: coldata matrix

- `coldata` now needs correct row and column names
- What should they be? (hint: it is in this slide deck)
- Once you decide, use `rownames()` and `colnames()` to add them

Today's (Remaining) Goals

1. Basic Plots
2. ggplot and DESeq
3. Advanced Plots

DESeq2 guides

- Here are the DESeq guides that I have summarized in this walkthrough:
- Walkthrough Link
- Focus on “Quick Start” and more specifically:
- Setting the R objects `cts` and `coldata` correctly
- Using `paste` (a unix command) to format your data into `cts`

http://bioconductor.org/packages/devel/bioc/vignettes/DESeq2/inst/doc/DESeq2.html

DESeq2

- To run DESeq, first create a “DESeq dataset”

```
> dds <- DESeqDataSetFromMatrix(countData = cts,
colData = coldata,
design = ~ condition)

> dds
```

Where cts and coldata are the files described earlier

Output:

```
class:  DESeqDataSet
dim:  45381 4
metadata(1):  version
```

DESeq2

- Let's throw out genes with low expression levels (here, below 10)

```
> keep <- rowSums(counts(dds)) >= 10
> dds <- dds[keep,]
> dds$condition <- relevel(dds$condition, ref = "untreated")
> dds
```

Output:

```
class:  DESeqDataSet
dim:  17380 4
metadata(1):  version
```

We've shrunk our gene list by roughly 60 percent

DESeq2

- Now... Run DESeq!!!

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

Output:

```
log2 fold change (MAP): condition treated vs untreated
Wald test p-value:  condition treated vs untreated
DataFrame with 17380 rows and 6 columns
baseMean log2FoldChange lfcSE stat pvalue

A1BG 189.927175 -0.20541335 0.16773404 -1.22463727 0.2207119
A1BG-AS1 34.902293 0.02668120 0.26939459 0.09904134 0.9211054
```

DESeq2

- A few final things to do and explore:
- Sort the results by p-value

```
> resOrdered <- res[order(res$pvalue),]
```

- Summarize the results

```
> summary(res)
```

- How many genes are significant (at 0.10 and 0.05)?

```
> sum(res$padj < 0.1, na.rm=TRUE)
> sum(res$padj < 0.05, na.rm=TRUE)
```

Plotting in DESeq2

- There are several ways to plot this data
- You can used the built-in DESeq functions:

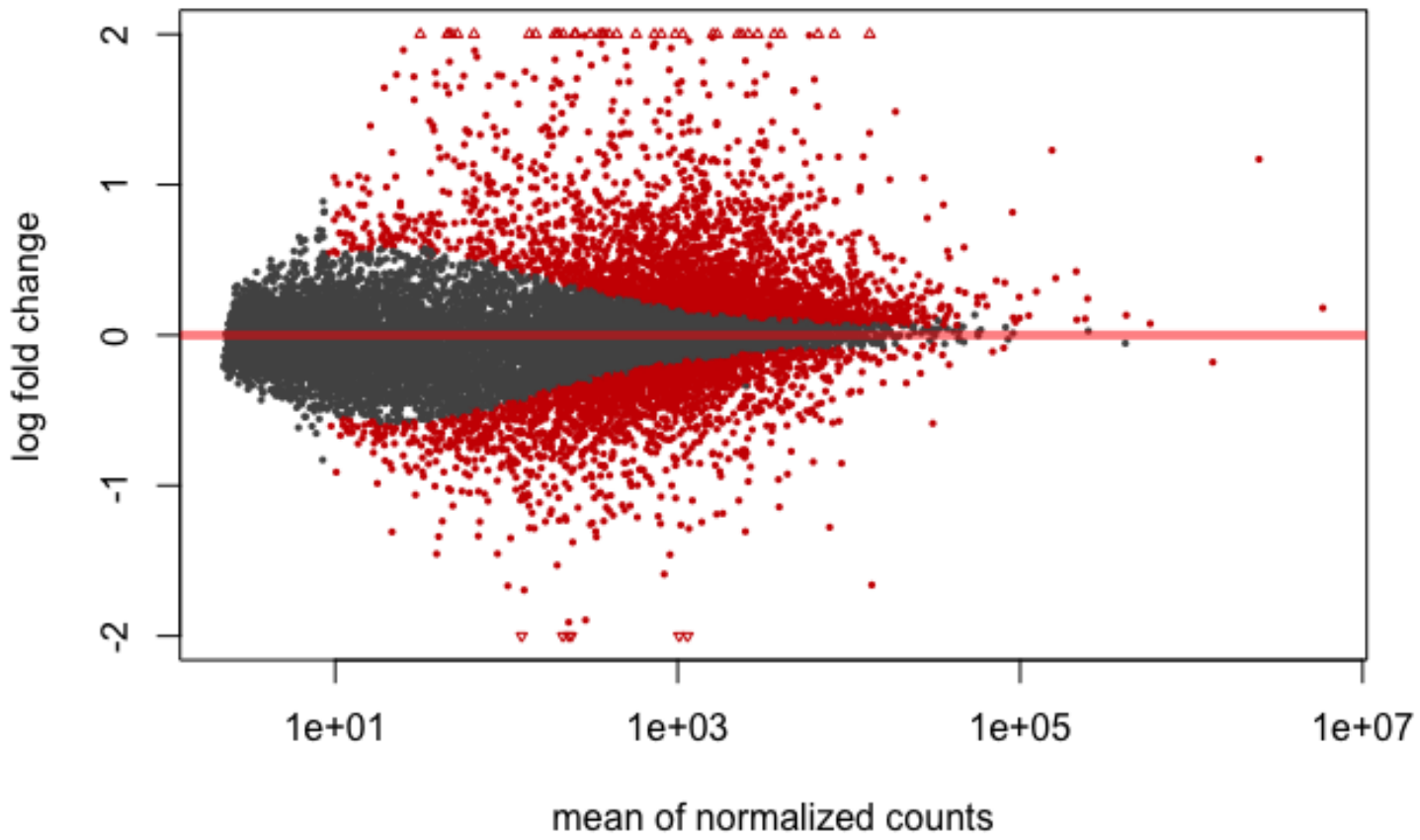
```
> plotMA(res, ylim=c(-2,2))


> plotCounts(dds, gene=which.min(res$padj),
intgroup="condition")
```

- Or the R plotting function:

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

Plotting in DESeq2

```
> plotMA(res, ylim=c(-2,2))
```

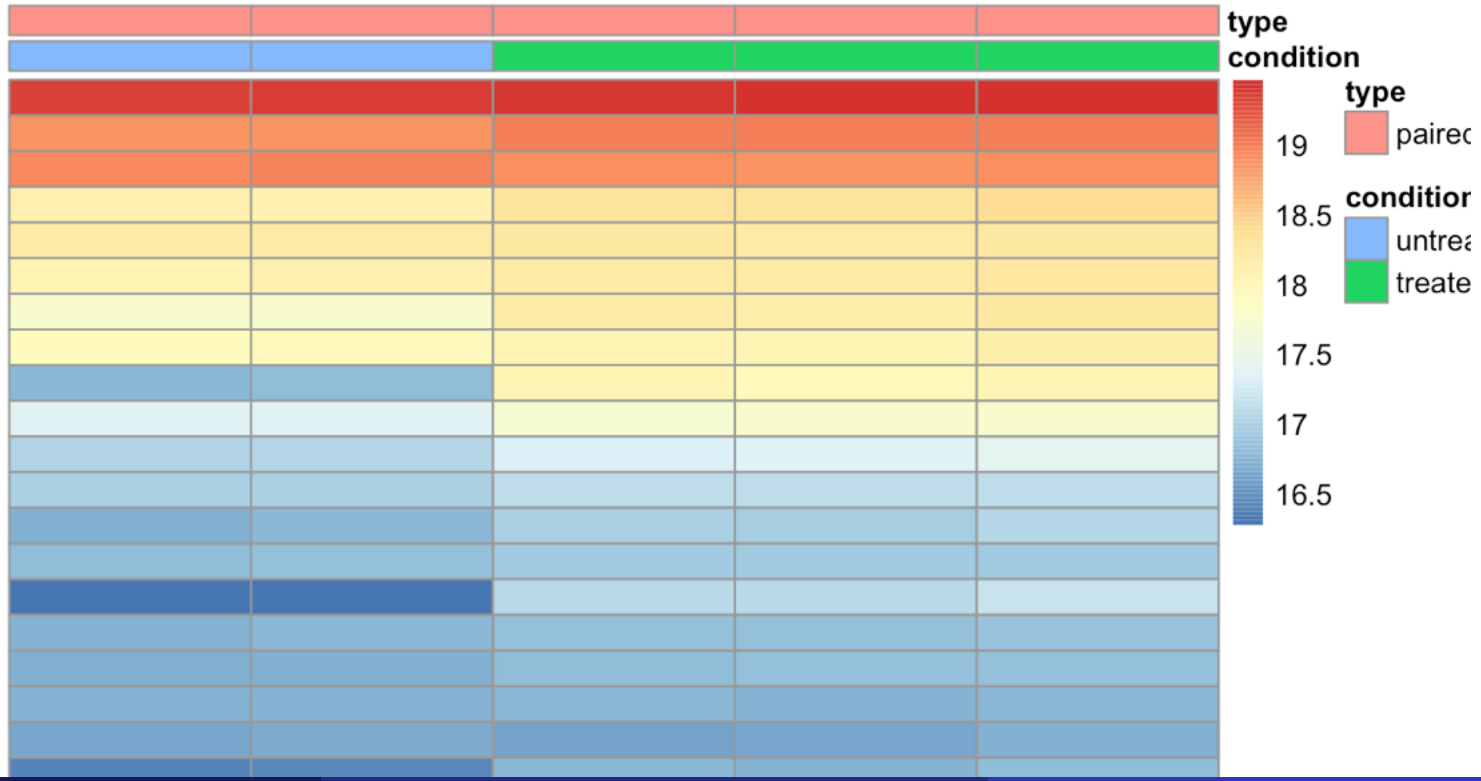


Check for Outliers

- Once you have the data in, use these methods to investigate sample quality
- Heatmaps - sorts normalized data and visualizes sample by sample
- Clustered Heatmaps - clusters samples by similarity
- PCA - plots similarity graphically
- If a sample looks out of the norm in these plots, remove it and rerun DESeq

Heatmap in DESeq2

```
> library("pheatmap")
> ntd <- normTransform(dds)
> df <- as.data.frame(colData(dds)[,c("condition","type")])
pheatmap(assay(ntd)[select,], cluster_rows=FALSE,
show_rownames=FALSE, cluster_cols=FALSE, annotation_col=df)
```

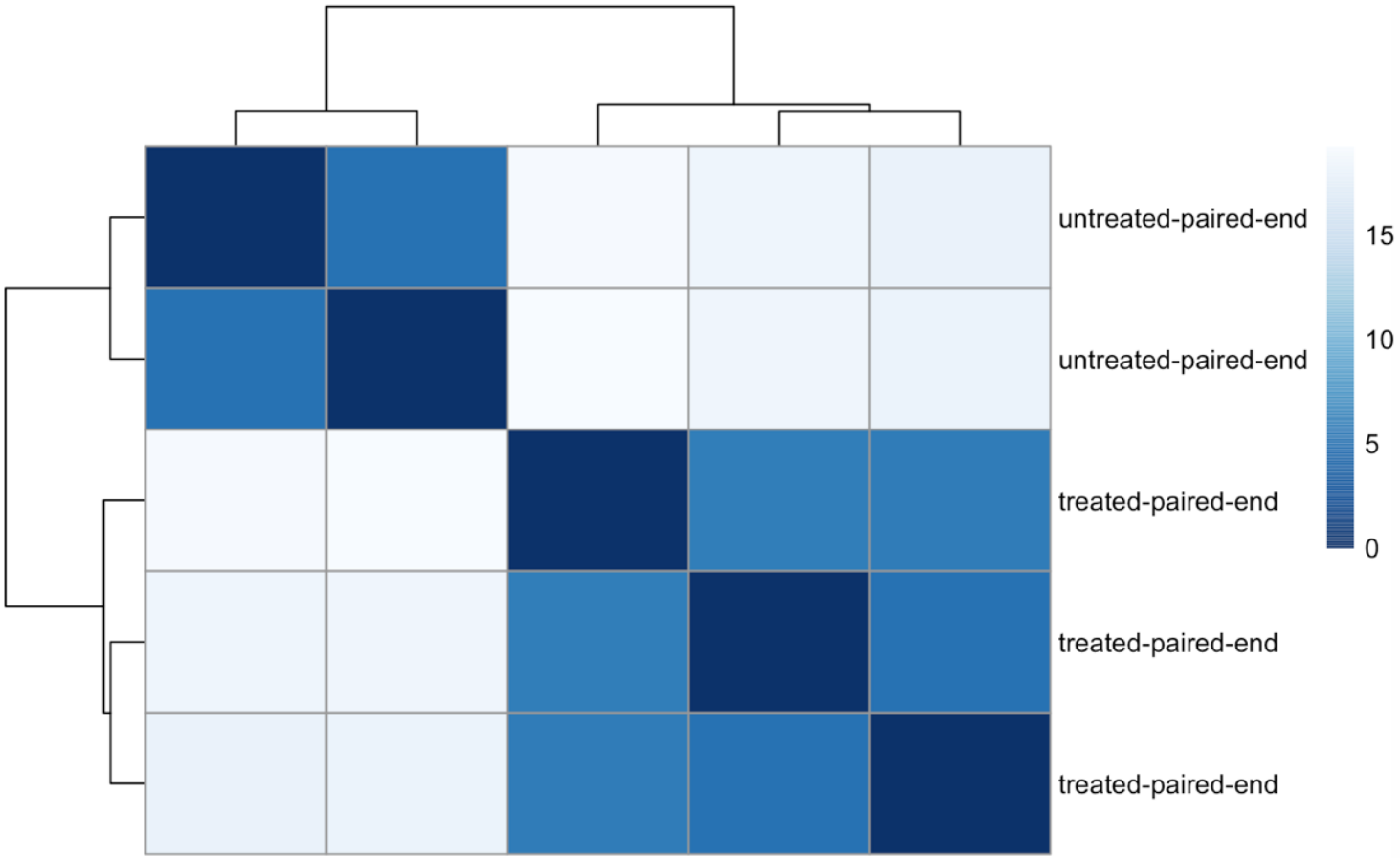


Heatmap in DESeq2

```
> sampleDists <- dist(t(assay(vsd)))
> library("RColorBrewer")
> sampleDistMatrix <- as.matrix(sampleDists)
> rownames(sampleDistMatrix) <- paste(vsd$condition, vsd$type,
sep="-")
> colnames(sampleDistMatrix) <- NULL
> colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
> pheatmap(sampleDistMatrix,
clustering_distance_rows=sampleDists,
clustering_distance_cols=sampleDists, col=colors)
```

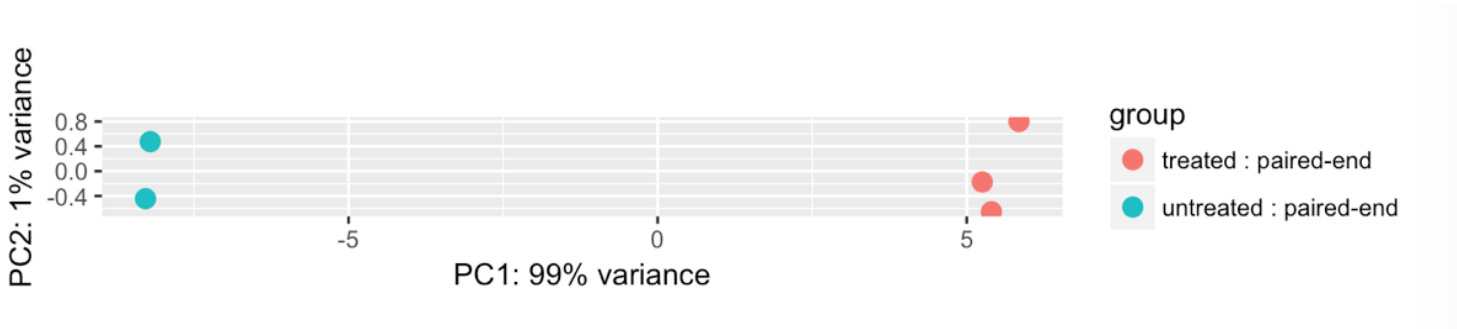
Heatmap in DESeq2

```
> pheatmap(sampleDistMatrix,
clustering_distance_rows=sampleDists,
clustering_distance_cols=sampleDists, col=colors)
```



PCA in DESeq2

```
> plotPCA(vsd, intgroup=c("condition", "type"))
```



ggplot and DESeq2

- First, take the DESeq results `res`, and convert them to a data frame

```
> library("ggplot2")


> resDF <- as.data.frame(res)
```

ggplot and DESeq2

- Now how can we plot the data frame `resDF` with `ggplot` to match `plotMA`?
- What are the x and y axes?
- What column do we need to add?

ggplot and DESeq2

- Now how can we plot the data frame `resDF` with `ggplot` to match `plotMA`?

```
ggplot(dataframe, aes()) +
geom_point() +
scale_x_log10()
```

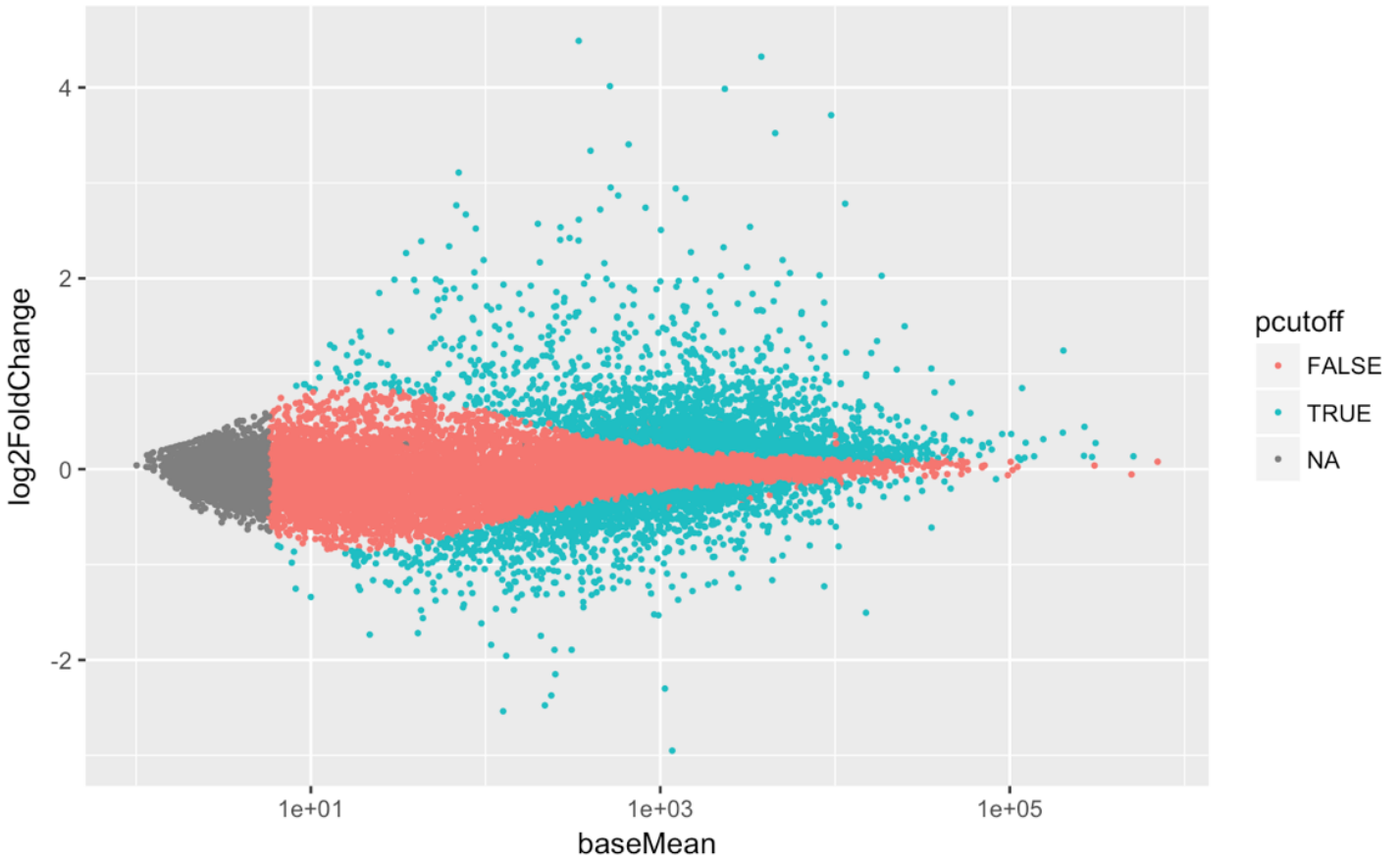
ggplot and DESeq2

- Now how can we plot the data frame `resDF` with `ggplot` to match `plotMA`?

```
> resDF$pcutoff <- resDF$padj < .01
> ggplot(resDF, aes(y = log2FoldChange, x =
baseMean, col = pcutoff)) + geom_point(size =
.5) + scale_x_log10()
```

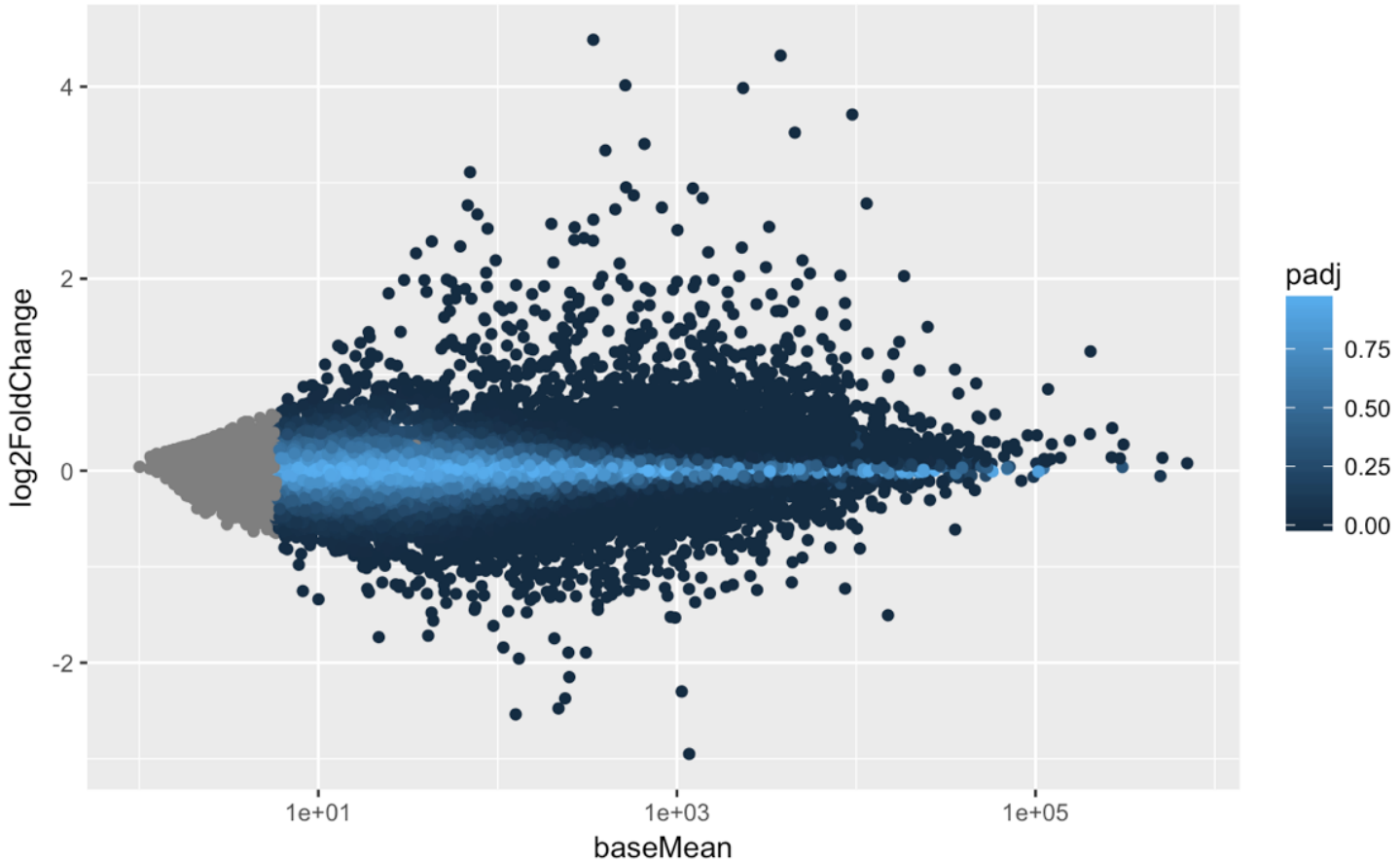
Plotting in DESeq2

```
> ggplot(resDF, aes(y = log2FoldChange, x =
baseMean, col = pcutoff)) + geom_point(size =
.5) + scale_x_log10()
```



Plotting in DESeq2

```
> ggplot(resDF, aes(y = log2FoldChange, x =
baseMean, col = padj)) + geom_point() +
scale_x_log10()
```



ggplot and DESeq2

- Now how can we plot the data frame with text labels?

```
ggplot(<dataframe>, aes( x = , y = ) +
geom_point() +
scale_x_log10() +
geom_text(aes(label = ))
```

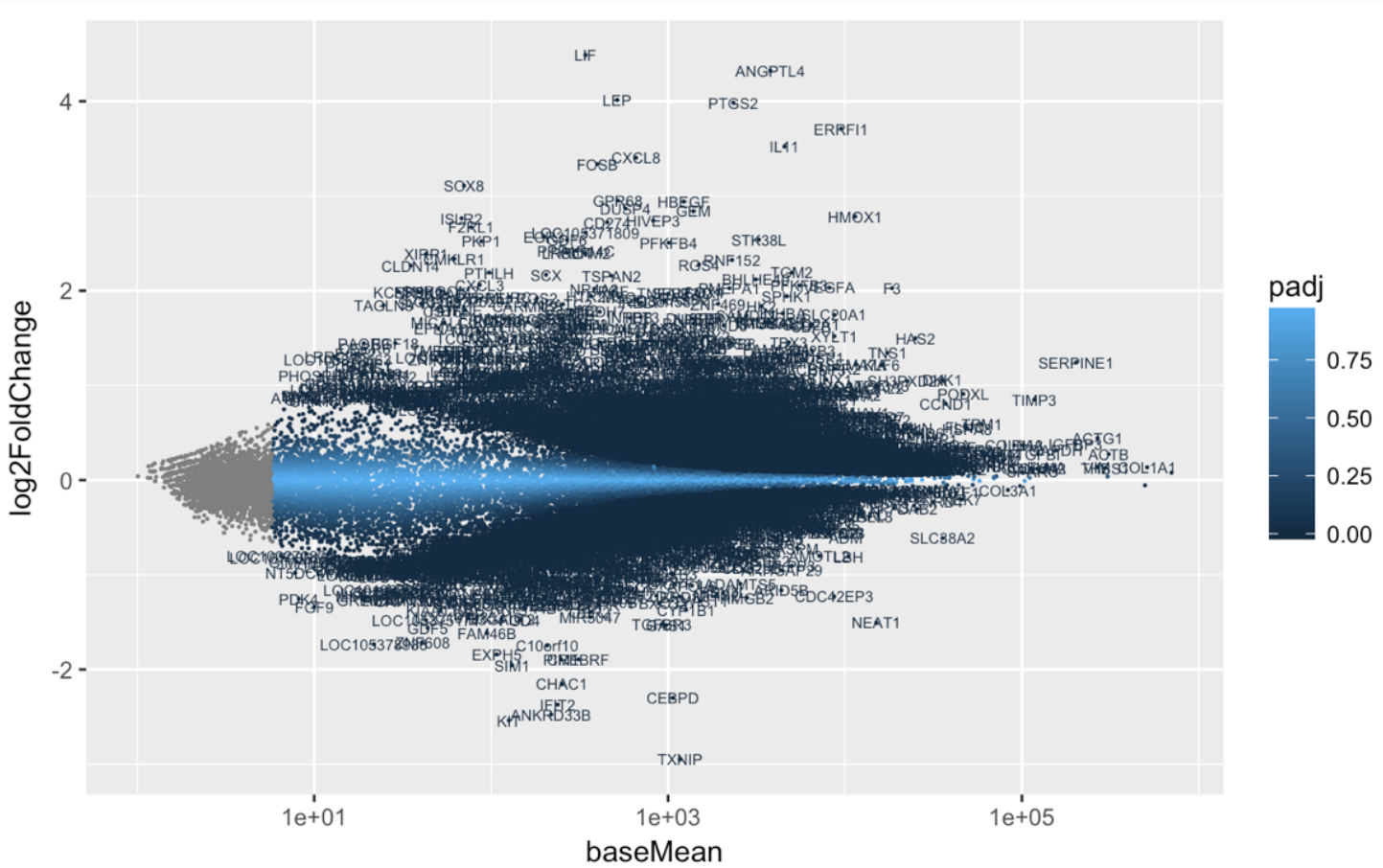
ggplot and DESeq2

- Now how can we plot the data frame with text labels?

```
> ggplot(resDF, aes(y = log2FoldChange, x =
baseMean, col = padj)) + geom_point(size =
.1) + scale_x_log10() + geom_text(aes(label =
ifelse( padj < .01 ,rownames(resDF),'')), size
= 2)
```

Plotting in DESeq2

```
> ggplot(resDF, aes(y = log2FoldChange, x = baseMean, col =
padj)) + geom_point(size = .1) + scale_x_log10() +
geom_text(aes(label = ifelse( padj < .01 ,rownames(resDF),'')),
size = 2)
```



ggplot and DESeq2

- Now how about gene families?

```
genes <- read.table("TIMM.genelist")
```

ggplot and DESeq2

- Now how about gene families?

```
genes <- read.table("TIMM.genelist")
resDF$genefam <- ".NULL"

for (n in 1:dim(genes)[1]) {
match <- which(rownames(resDF) == as.character(genes[n,1]))
resDF$genefam[match] <- "TIMM"
}
```

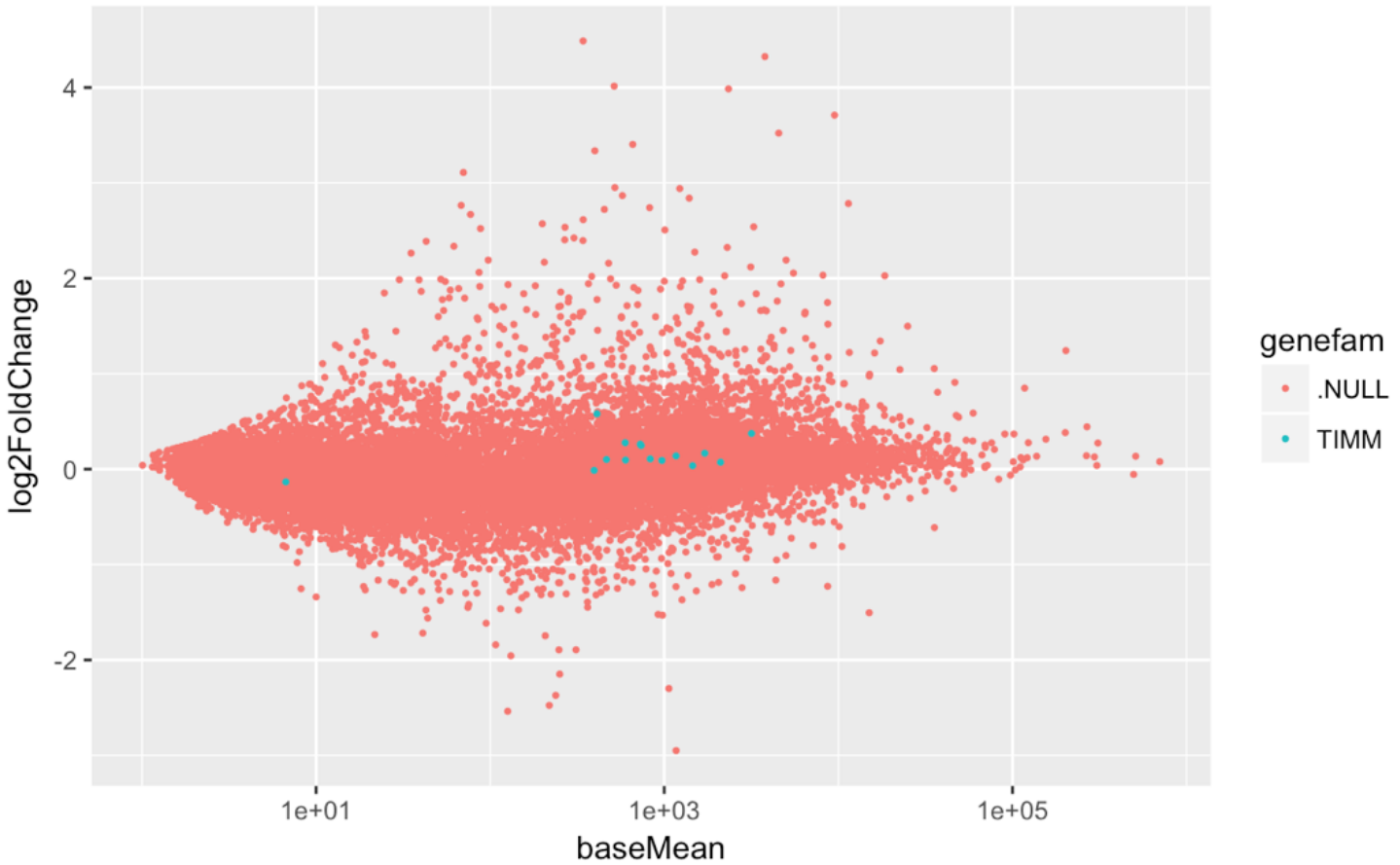
Read in a list of the genes you want to identify

Add a column for “gene family” to the data frame

Loop through the list of genes, adding a label “TIMM” to the data

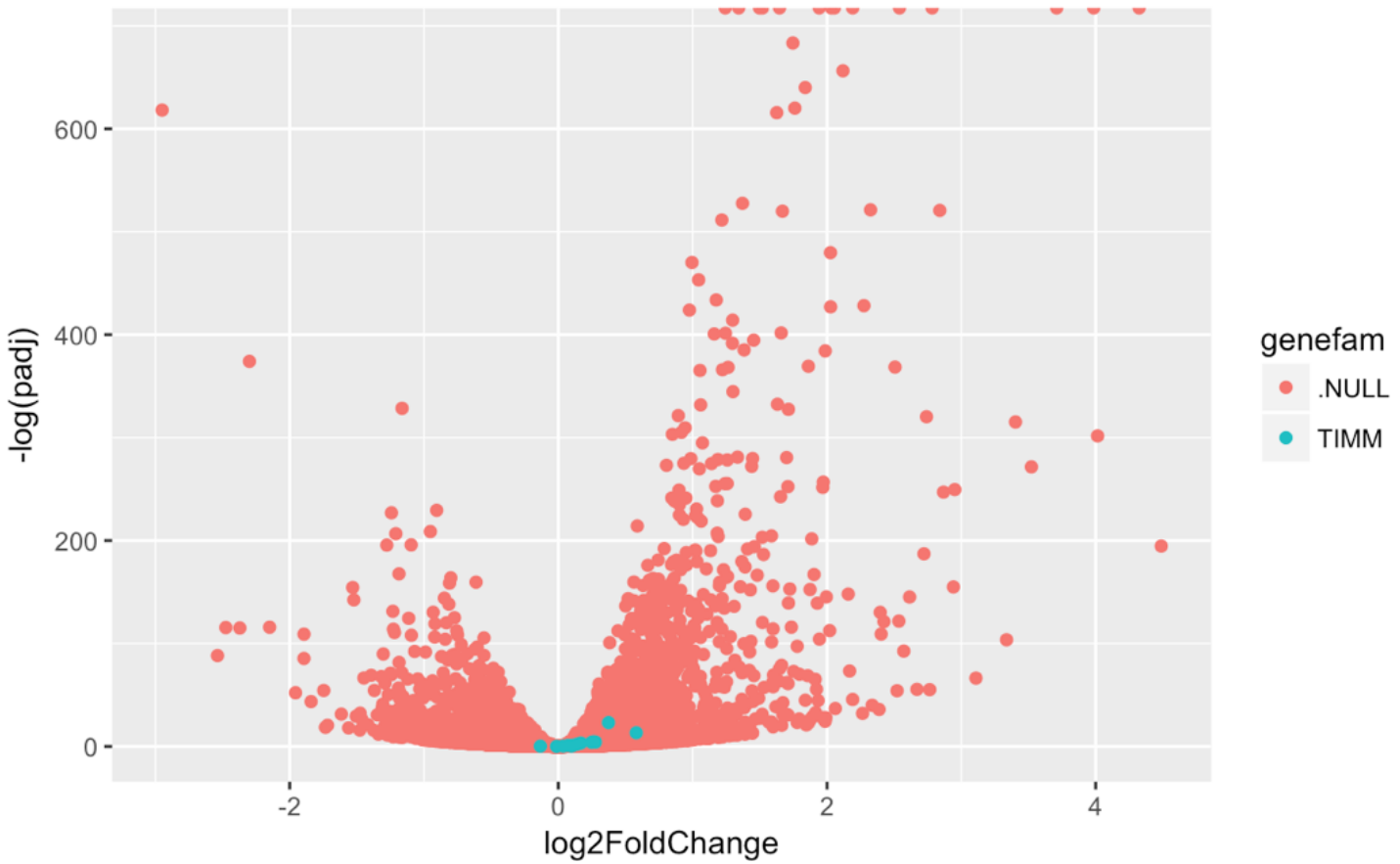
ggplot and DESeq2

```
ggplot(resDF %>% arrange(genefam), aes(y = log2FoldChange, x =
baseMean, col = genefam)) + geom_point( size = .5) +
scale_x_log10()
```



ggplot and DESeq2

```
ggplot(resDF %>% arrange(genefam), aes(y = -log(padj), x =
log2FoldChange, col = genefam)) + geom_point()
```



The End