# Differential Expression Analysis

First we will load the necessary libraries.

```
library(tidyverse)
library(DESeq2)
library(magrittr)
library(edgeR)
```

The data has been extracted from https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE226134 from the GSE226134_CK_10___norm.xlsx file. Lauren Mock selected for pre-treatment samples and performed data quality control.

```
data <- read.csv("data/input/JoinedWide.csv")

property <- as.data.frame(cbind(data$SegmentDisplayName, data$METASTATIC))
names(property)=c("SegmentDisplayName","METASTATIC")
normCountData <- data[,59:ncol(data)]
row.names(normCountData) <- data$SegmentDisplayName
```

The data given is normalized, but the properties include a normalization factor that we can divide the data by to get to integer counts, which are required for DESeq.

```
intCountData <- normCountData / data$NormalizationFactor
intCountData <- data.frame(lapply(intCountData, as.integer))
```

## DESeq

```
dds <- DESeqDataSetFromMatrix(countData = t(intCountData),
                              colData = property,
                              design = ~METASTATIC)
```

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

DESeq recommends that row counts are filtered to remove rows with very few reads, especially rows with less than 10 reads. Here it appears that we end up keeping all rows.

```
keep <- rowSums(counts(dds)) >= 10
dds <- dds[keep,]
```

Next we will run DESeq to get the differentially expressed genes.

```
dds$METASTATIC <- relevel(dds$METASTATIC, ref = "False")
dds <- DESeq(dds)
```

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates
```

```
mean-dispersion relationship
```

```
-- note: fitType='parametric', but the dispersion trend was not well captured by the
   function: y = a/x + b, and a local regression fit was automatically substituted.
   specify fitType='local' or 'mean' to avoid this message next time.
```

```
final dispersion estimates
```

```
fitting model and testing
```

```
-- replacing outliers and refitting for 116 genes
-- DESeq argument 'minReplicatesForReplace' = 7
-- original counts are preserved in counts(dds)
```

```
estimating dispersions
```

```
fitting model and testing
```

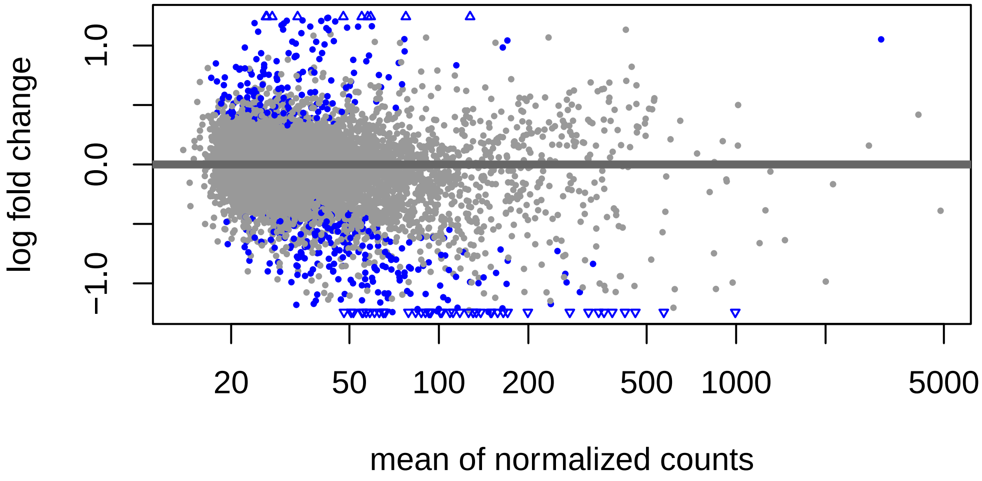
```r
adj_pval_threshold <- 0.05
res <- results(dds, alpha = adj_pval_threshold)
summary(res)
```

```
out of 9223 with nonzero total read count
adjusted p-value < 0.05
LFC > 0 (up)       : 195, 2.1%
LFC < 0 (down)     : 281, 3%
outliers [1]       : 0, 0%
low counts [2]     : 0, 0%
(mean count < 14)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

```r
resultsNames(dds)
```

```
[1] "Intercept"                "METASTATIC_True_vs_False"
```

```r
DESeq2::plotMA(res)
```



```r
resOrdered <- res[order(res$padj),]
resSig <- subset(resOrdered, padj < 0.05)
resSig <- subset(resSig, abs(log2FoldChange) > 1)
```

```r
nrow(resSig) #113 genes
```

```
[1] 113
```

```r
gene_names <- rownames(resSig)
#uncomment the following lines to get a printed list to input for ShinyGO
# for (gene in gene_names) {
#   cat(gene, "\n")
# } #running these genes through ShinyGO shows cancer and metabolism pathways http://bioin
```

## EdgeR

Reference: https://web.stanford.edu/class/bios221/labs/rnaseq/lab_4_rnaseq.html

First we will prepare the data and calculate the dispersion so we will next be able to find the differential expression

```r
d <- DGEList(counts=t(intCountData),group=property$METASTATIC)
dim(d)
```

```
[1] 9223   59
```

```r
#head(d$counts)
apply(d$counts, 2, sum)
```

```
 Sample1  Sample2  Sample3  Sample4  Sample5  Sample6  Sample7  Sample8 
  910619    49480   131443   348687   915774  1045122   649427   378320 
 Sample9 Sample10 Sample11 Sample12 Sample13 Sample14 Sample15 Sample16 
  380713  2014498   823838   802316   238557   883929   514514   496125 
Sample17 Sample18 Sample19 Sample20 Sample21 Sample22 Sample23 Sample24 
  361452   314568   442702  1115658   369882   156975   488535  1658825 
Sample25 Sample26 Sample27 Sample28 Sample29 Sample30 Sample31 Sample32 
 2145876   485456   291752  2437146   381860   524198    39357    42942 
Sample33 Sample34 Sample35 Sample36 Sample37 Sample38 Sample39 Sample40 
  970369    86043   883825   254777  1124356   122495   309227   567186 
Sample41 Sample42 Sample43 Sample44 Sample45 Sample46 Sample47 Sample48 
 2001258   517700   323444    57887   528169  1187802  1153804   211023 
Sample49 Sample50 Sample51 Sample52 Sample53 Sample54 Sample55 Sample56 
```

```
   64074    108979    528054    500160    802961   2543986    474835    303885
Sample57 Sample58 Sample59
  152894   1453165   1118159
```

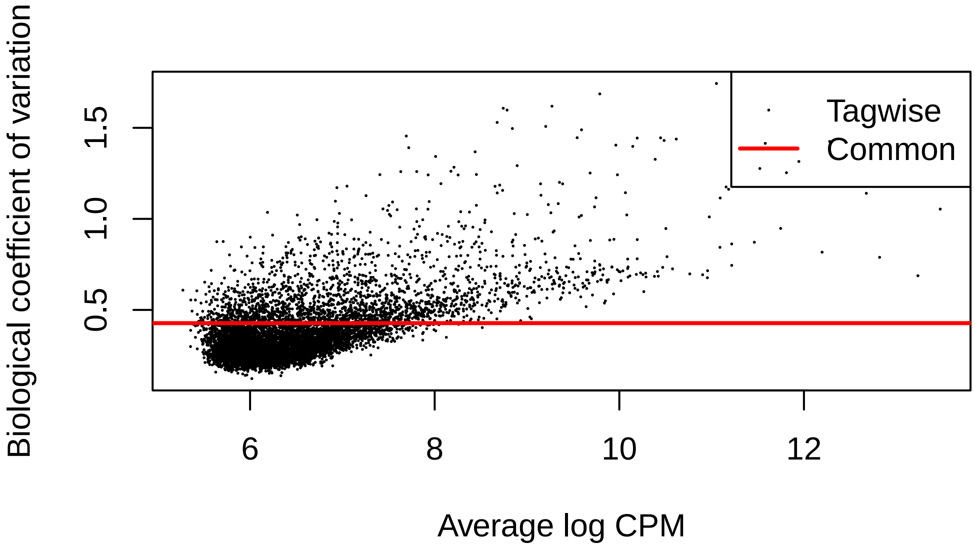
```
#filtering steps for DESeq
keep <- rowSums(cpm(d)>100) >= 2
d <- d[keep,]
dim(d) #cuts down about 600 genes
```

```
[1] 8680   59
```

```
d$samples$lib.size <- colSums(d$counts)
d <- calcNormFactors(d)
d1 <- estimateCommonDisp(d, verbose=T)
```

```
Disp = 0.18261 , BCV = 0.4273
```

```
d1 <- estimateTagwiseDisp(d1)
plotBCV(d1)
```



We will now use our information from the dispersion calculation to check for differential expression and then compare to the results from DESeq.

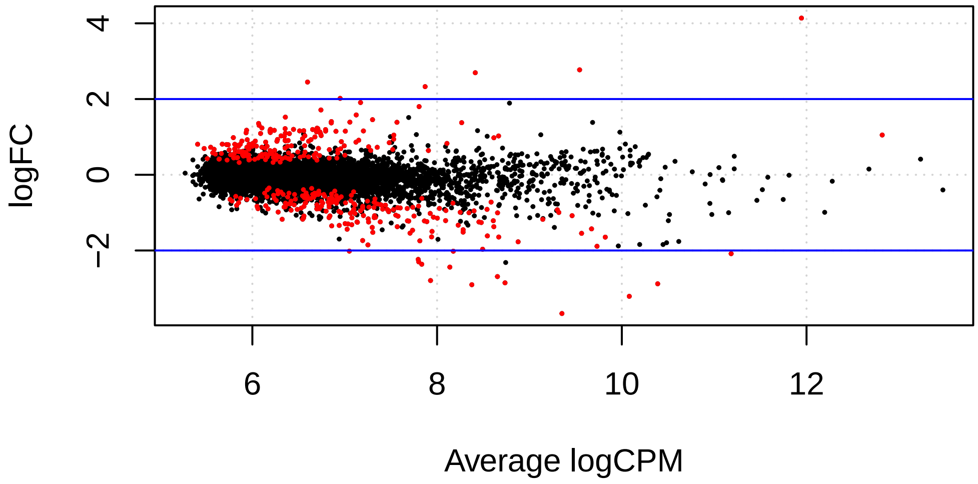
```r
et12 <- exactTest(d1, pair=c(1,2))
topTags(et12, n=10)
```

```
Comparison of groups:  True-False
           logFC    logCPM       PValue          FDR
PITX1  4.1377711 11.945164 1.817350e-15 1.577460e-11
MT1G   1.9075098  7.171275 5.273978e-13 1.806617e-09
ANK1   2.6948864  8.413772 6.244069e-13 1.806617e-09
GSTA1  2.4484216  6.597395 4.285723e-12 9.300019e-09
DCXR   1.1614314  6.191021 4.497128e-11 7.807015e-08
ASH2L  1.3051355  6.072951 1.168861e-09 1.690953e-06
TNC   -3.2101590 10.080863 1.991779e-09 2.438533e-06
BAG4   1.0346380  6.745474 2.247496e-09 2.438533e-06
LUM    2.3276325  7.870496 2.841152e-09 2.740133e-06
FGFR1  0.9251594  5.944759 4.159412e-09 3.610369e-06
```

```r
de1 <- decideTestsDGE(et12, adjust.method="BH", p.value=0.05)
summary(de1)
```

```
       True-False
Down          211
NotSig       8296
Up            173
```

```r
de1tags12 <- rownames(d1)[as.logical(de1)]
plotSmear(et12, de.tags=de1tags12)
abline(h = c(-2, 2), col = "blue")
```



```
tags <- topTags(et12, n=Inf)
top_genes <- rownames(tags$table)[tags$table$FDR < 0.05 & abs(tags$table$logFC) > 1]
```

### Compare DESeq and EdgeR

```
sum(top_genes %in% rownames(resSig)) #106 of 113 genes match
```

```
[1] 106
```

```
sum(rownames(resSig) %in% top_genes) #106; serves as a check
```

```
[1] 106
```

```
gene <- top_genes[top_genes %in% rownames(resSig)]
#the output of the following lines is very long so it will be omitted from our rendered do
#to get the list of genes for ShinyGO please uncomment the following lines:
# for (gene_name in gene) {
#   cat(gene_name, "\n")
# }
```

We can see that there are 106 genes in common between the results from DESeq and from EdgeR. When we plug in the overlapping genes into ShinyGO, we see pathways enriched for receptor interactions, cancer, adhesion, and signaling pathways, which make sense given the biological basis of metastasis.

We will use these 106 genes as the differential expression genes for downstream steps in the process. We will use the p-value information from DESeq.

```
res_match <- subset(resSig, rownames(resSig) %in% top_genes)

res_match_df <- cbind(gene,as.data.frame(res_match@listData))
write.csv(res_match_df,file="data/output/metastasis_results.csv")
```