

BB512/BB612 - Week VI

```
suppressPackageStartupMessages(library(Biobase))
suppressPackageStartupMessages(library(bladderbatch))
```

Data

We are going to use data from the paper Evaluating gene expression in C57BL/6J and DBA/2J mouse striatum using RNA-Seq and microarrays. that is a comparative RNA-seq analysis of different mouse strains.

```
con <- url("http://bowtie-bio.sourceforge.net/recount/ExpressionSets/bottomly_eset.RData")
load(file = con)
close(con)

pdata <- pData(bottomly.eset)
edata <- as.matrix(exprs(bottomly.eset))
fdata <- fData(bottomly.eset)
```

Transform the expression data and remove lowly expressed genes:

```
edata <- log2(as.matrix(edata) + 1)
edata <- edata[rowMeans(edata) > 10, ]
```

```
head(pdata)
```

```
##      sample.id num.tech.reps  strain experiment.number lane.number
## SRX033480 SRX033480          1 C57BL/6J                6           1
## SRX033488 SRX033488          1 C57BL/6J                7           1
## SRX033481 SRX033481          1 C57BL/6J                6           2
## SRX033489 SRX033489          1 C57BL/6J                7           2
## SRX033482 SRX033482          1 C57BL/6J                6           3
## SRX033490 SRX033490          1 C57BL/6J                7           3
```

```
table(pdata$strain)
```

```
##
## C57BL/6J  DBA/2J
##      10      11
```

Calculate p-values parametrically

t-test

```
idxA <- which(colnames(edata) %in% pdata$sample.id[pdata$strain == "C57BL/6J"])
idxB <- which(colnames(edata) %in% pdata$sample.id[pdata$strain == "DBA/2J"])
raw_p_vals <- apply(edata, 1, function(x) t.test(x[idxA], x[idxB])$p.value)
p_vals <- sort(raw_p_vals)
head(p_vals, 10)
```

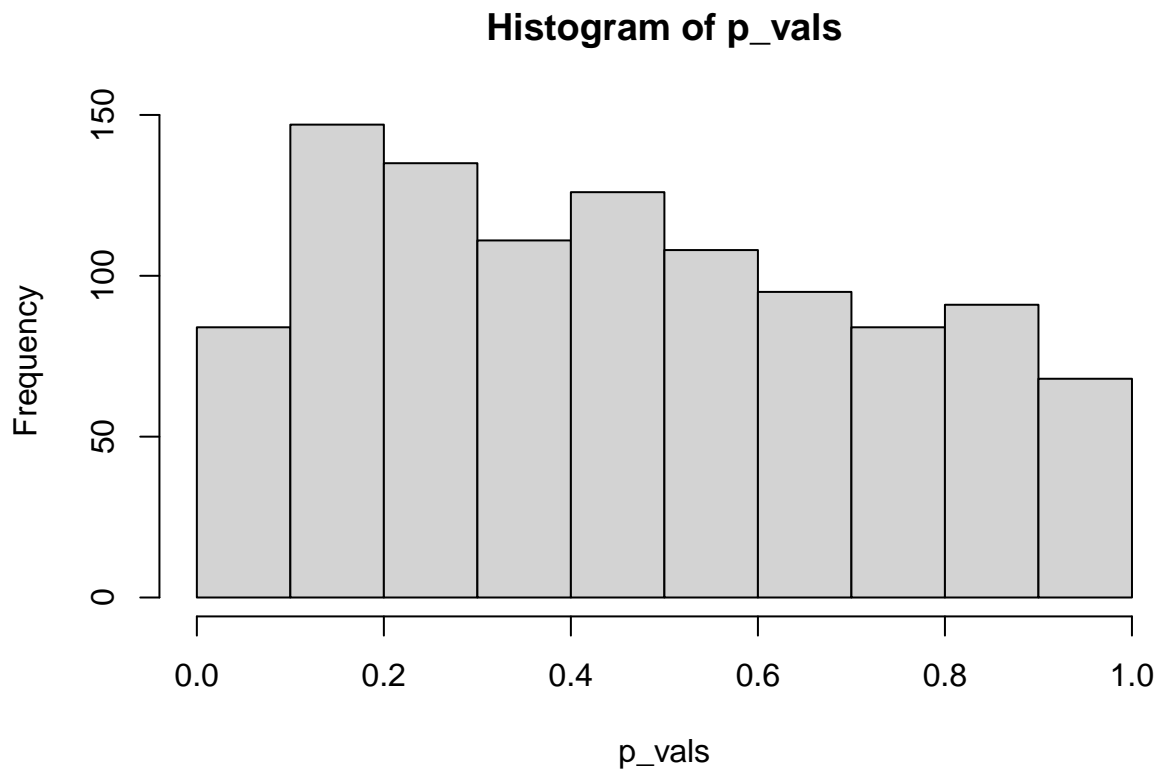
```
## ENSMUSG00000033585 ENSMUSG00000070003 ENSMUSG00000022212 ENSMUSG00000044708
```

```
##          0.00019311          0.00348277          0.00573420          0.00738418
## ENSMUSG00000024883 ENSMUSG00000036473 ENSMUSG00000038020 ENSMUSG00000074045
##          0.00927920          0.01023011          0.01119852          0.01435736
## ENSMUSG00000032340 ENSMUSG00000041297
##          0.01560463          0.01826414
```

```
p_adj_vals <- p.adjust(p_vals, method = "fdr")
head(p_adj_vals, 10)
```

```
## ENSMUSG00000033585 ENSMUSG00000070003 ENSMUSG00000022212 ENSMUSG00000044708
##          0.20257          0.84267          0.84267          0.84267
## ENSMUSG00000024883 ENSMUSG00000036473 ENSMUSG00000038020 ENSMUSG00000074045
##          0.84267          0.84267          0.84267          0.84267
## ENSMUSG00000032340 ENSMUSG00000041297
##          0.84267          0.84267
```

```
hist(p_vals)
```

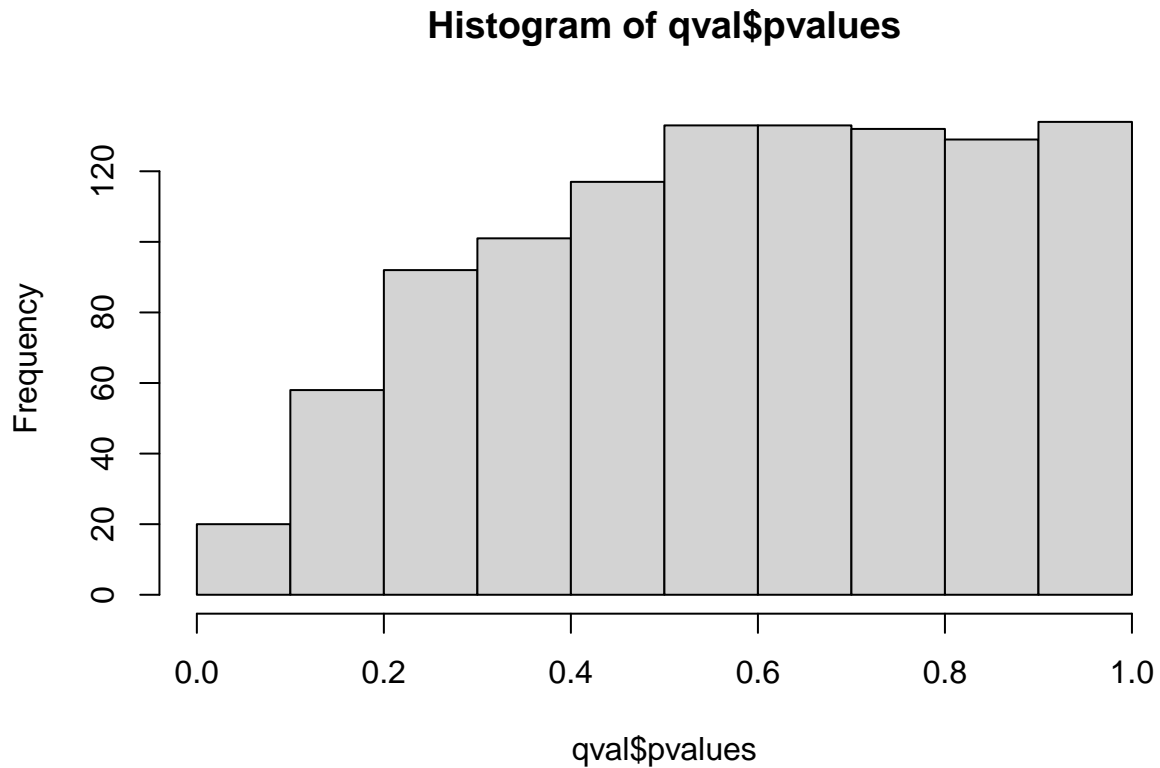


Adjusting for variables with edge

```
# devtools::install_github("jdstorey/edge")
library(edge)

edge_study <- build_study(edata, grp = pdata$strain,
                          adj.var = as.factor(pdata$lane.number))
de_obj <- lrt(edge_study)
qval <- qvalueObj(de_obj)
```

```
hist(qval$pvalues)
```



Using moderated statistics with limma

```
library(limma)
```

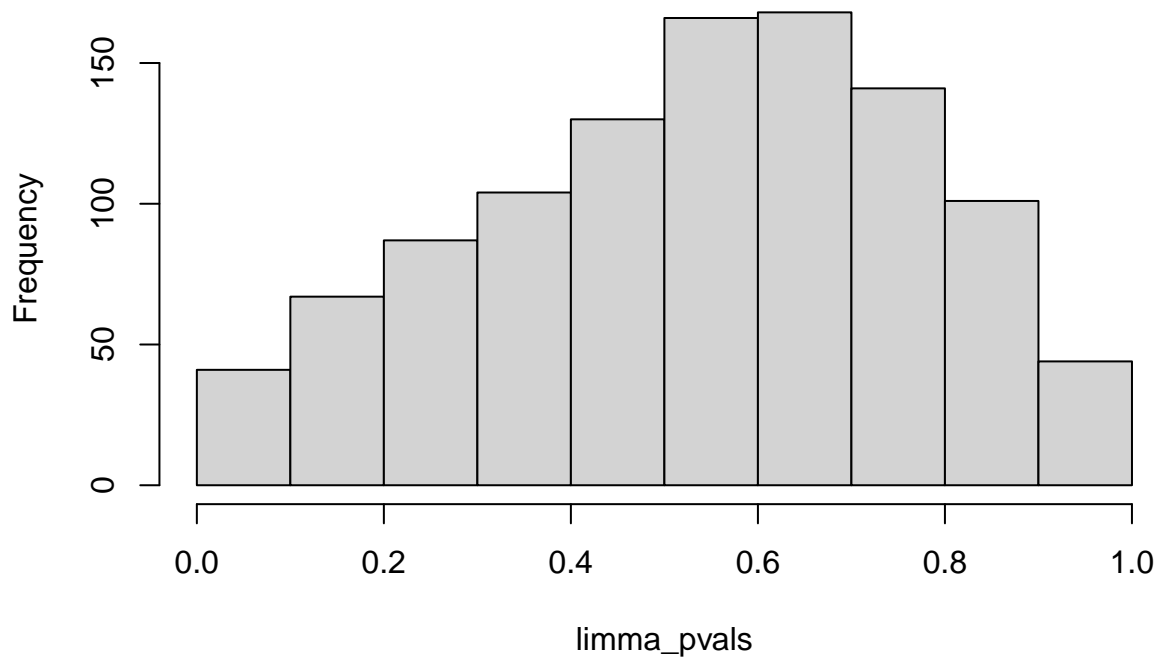
```
##  
## Attaching package: 'limma'  
## The following object is masked from 'package:BiocGenerics':  
##  
## plotMA
```

```
mod <- model.matrix(~ pdata$strain + pdata$lane.number)  
fit_limma <- lmFit(edata, mod)  
ebayes_limma <- eBayes(fit_limma)  
limma_pvals <- topTable(ebayes_limma, number = Inf)$P.Value
```

```
## Removing intercept from test coefficients
```

```
hist(limma_pvals)
```

Histogram of limma_pvals



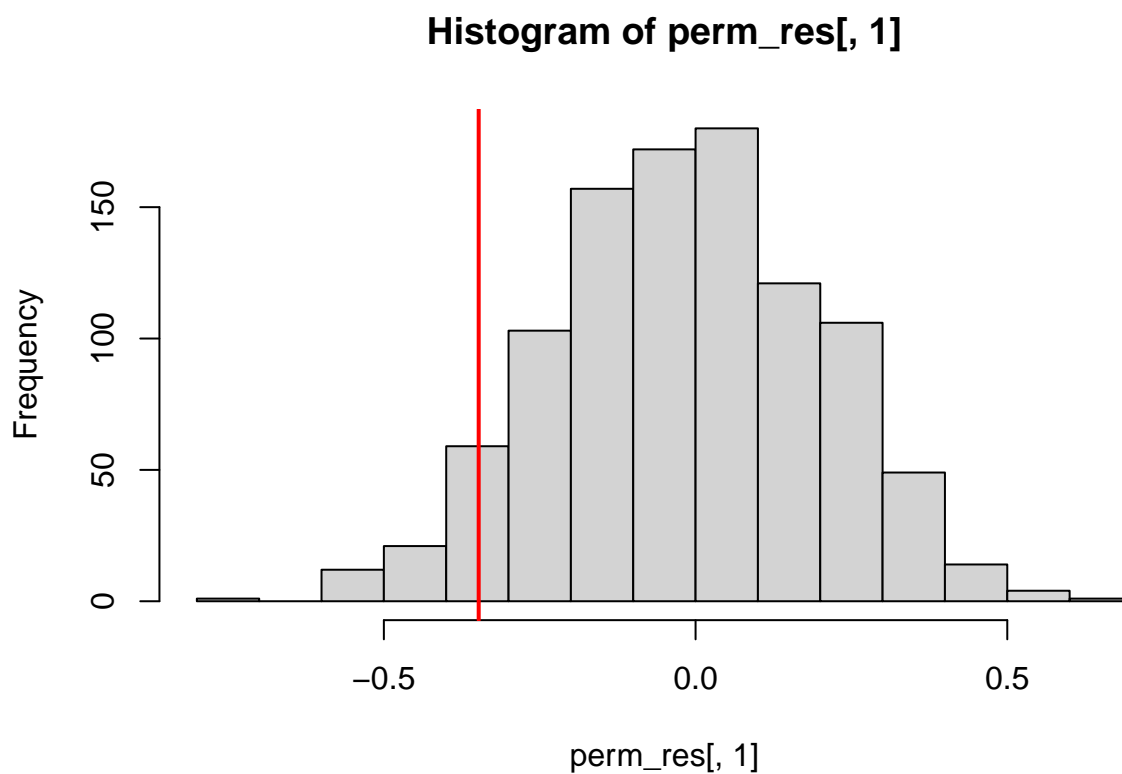
Calculating empirical permutation p-values

```
idxA <- which(colnames(edata) %in% pdata$sample.id[pdata$strain == "C57BL/6J"])
idxB <- which(colnames(edata) %in% pdata$sample.id[pdata$strain == "DBA/2J"])

actual_diffs <- apply(edata, 1, function(x) mean(x[idxA]) - mean(x[idxB]))

B <- 1000
perm_res <- c()
set.seed(123)
for (i in seq_len(B)) {
  cur_idxA <- sample(seq_len(ncol(edata)), 10)
  cur_idxB <- setdiff(seq_len(ncol(edata)), cur_idxA)
  cur_diffs <- apply(edata, 1, function(x) mean(x[cur_idxA]) - mean(x[cur_idxB]))
  perm_res <- rbind(perm_res, cur_diffs)
}

hist(perm_res[, 1])
abline(v = actual_diffs[1], col = "red", lwd = 2)
```

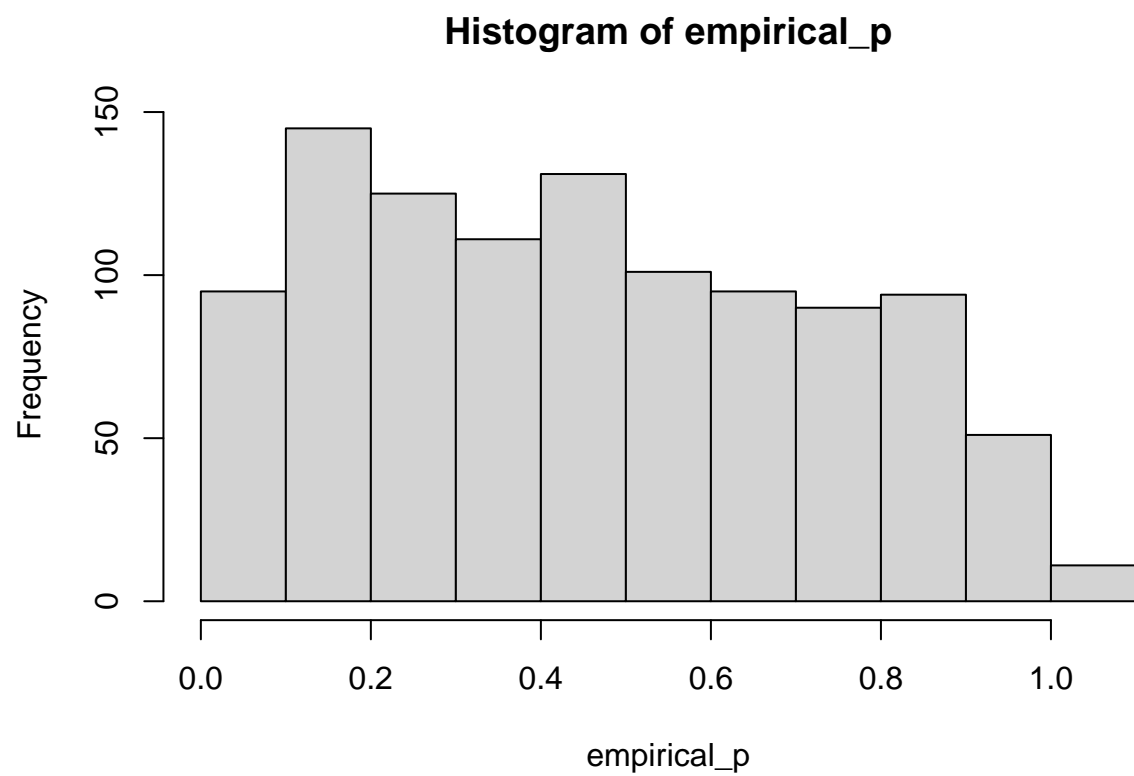


```
# empirical p value
2 * sum(perm_res[, 1] <= actual_diffs[1]) / B

## [1] 0.114

empirical_p <- c()
for (j in 1:ncol(perm_res)) {
  if (actual_diffs[j] < 0) {
    empirical_p <- c(empirical_p, 2 * sum(perm_res[, j] <= actual_diffs[j]) / B)
  } else {
    empirical_p <- c(empirical_p, 2 * sum(perm_res[, j] >= actual_diffs[j]) / B)
  }
}

hist(empirical_p)
```



```
plot(raw_p_vals, empirical_p)
abline(a = 0, b = 1, col = "red", lwd = 3)
```

