FDR

```
setwd("/Users/hjshim/Documents/Work/github/mycomments4projects/HMT/script")
pval = read.table("../data/pvalues.multiseq.DESeq2.txt", sep=" ", header=TRUE)
dim(pval)
```

```
## [1] 242714      2
```

```
names(pval) # pvalues from two different methods
```

```
## [1] "pval.multiseq" "pval.deseq2"
```

```
length(pval$pval.multiseq)
```

```
## [1] 242714
```

```
length(pval$pval.deseq2)
```

```
## [1] 242714
```

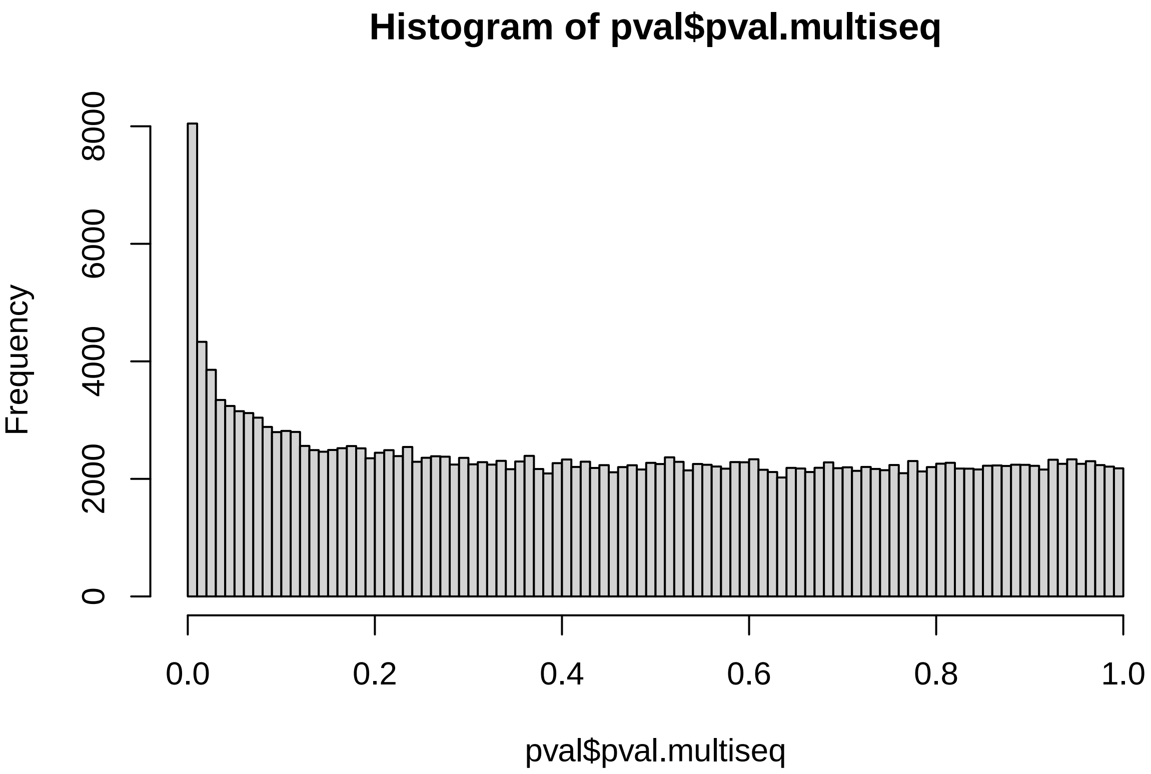
```
# check if we have all p-values; if we don't have a complete set of p-values, we should remove those ca
sum(is.na(pval$pval.multiseq)==TRUE)
```

```
## [1] 0
```

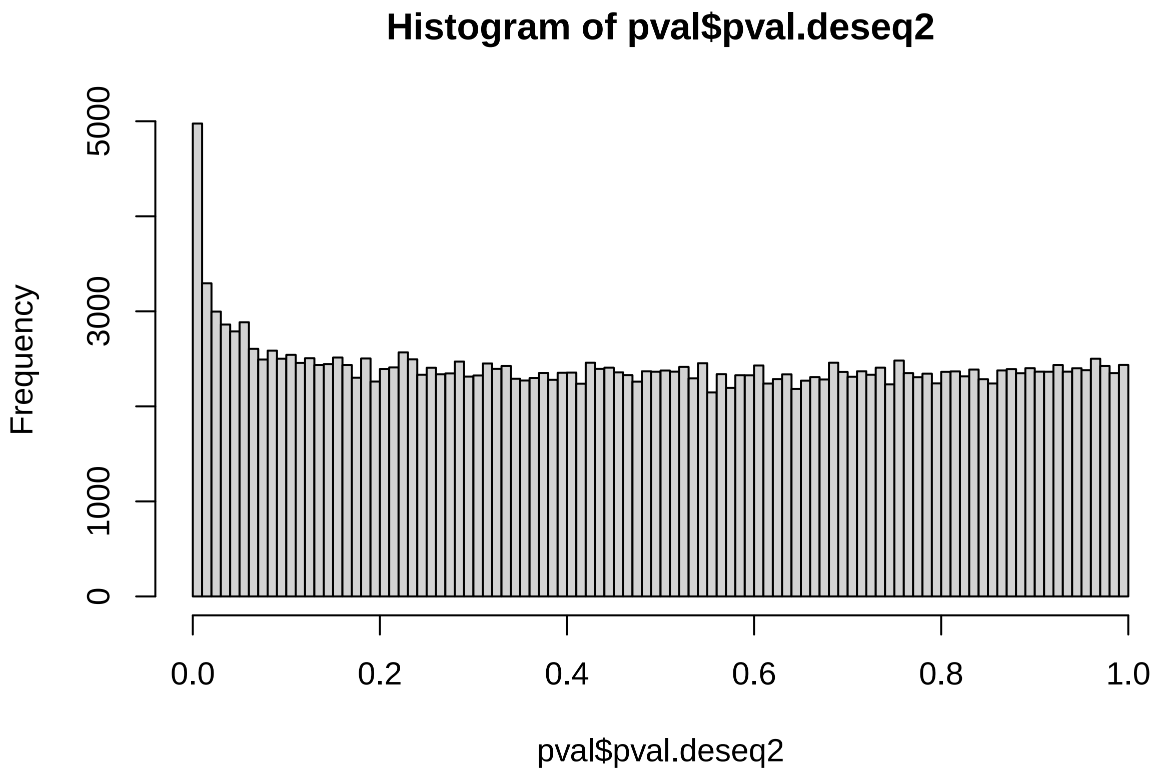
```
sum(is.na(pval$pval.deseq2)==TRUE)
```

```
## [1] 0
```

```
# check histograms
hist(pval$pval.multiseq, breaks=100)
```



```r
hist(pval$pval.deseq2, breaks=100)
# look good!

# check
# apply qvalue package
library("qvalue")
```



```
qval.multiseq = qvalue(pval$pval.multiseq)
qval.deseq2 = qvalue(pval$pval.deseq2)

# check the proportion of null cases
qval.multiseq$pi0
```

```
## [1] 0.9223634
```

```
qval.deseq2$pi0
```

```
## [1] 0.9897486
```

```
# possible values of FDR
alpha.list = seq(0, 0.1, by=0.001)
length(alpha.list)
```

```
## [1] 101
```

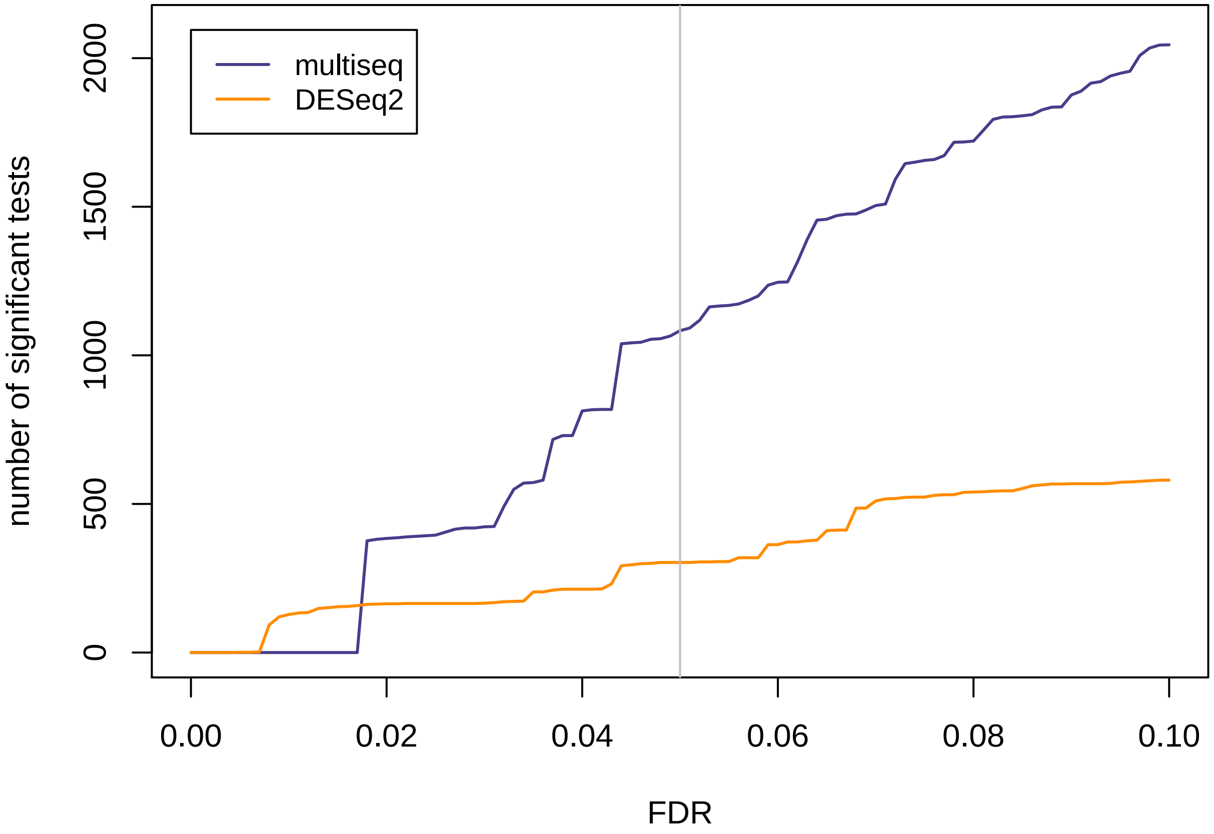
```
## 101

# count the number of significant tests at a given FDR
num.multiseq = num.deseq2 = rep(NA, length(alpha.list))
for(i in 1:length(alpha.list)){
    num.multiseq[i] = sum(qval.multiseq$qvalues < alpha.list[i])
    num.deseq2[i] = sum(qval.deseq2$qvalues < alpha.list[i])
}

# number of significant tests at FDR = 0.05
wh = which(alpha.list == 0.05)
num.multiseq[wh]
```

```
## [1] 1083
```

```
num.deseq2[wh]
```

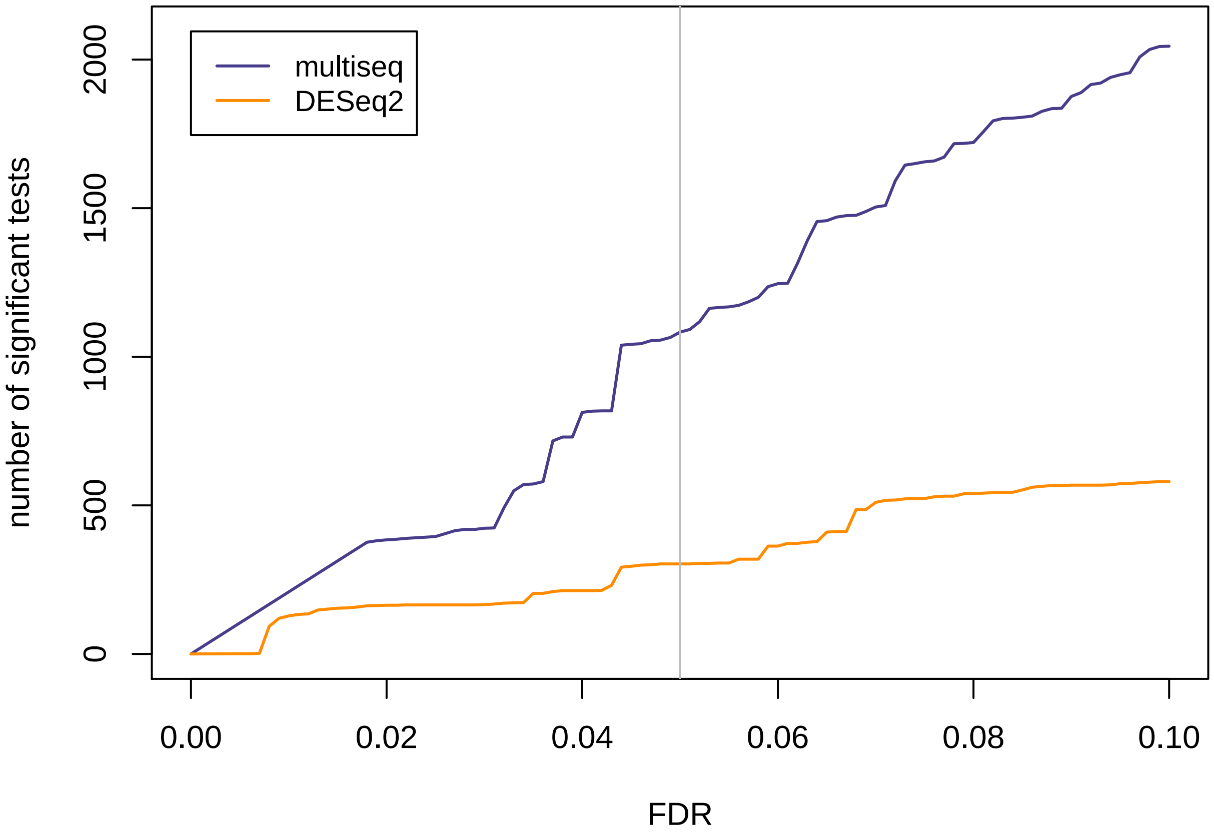
```
## [1] 303
```

```
# Make FDR curves
multiseq.col = "#483D8B"
deseq2.col = "#FF8C00"
par(mar = c(4, 4, 1, 1))
ymax = max(num.multiseq, num.deseq2) + 50
ymin = 0
plot(alpha.list, num.multiseq, ylim=c(ymin,ymax), col=multiseq.col, type = "l", lty = 1, lwd = 1.5, xlab
points(alpha.list, num.deseq2, ylim=c(ymin,ymax), col=deseq2.col, type="l", lty = 1, lwd = 1.5)
abline(v=0.05, col="grey")
legend(0,ymax, c("multiseq", "DESeq2"), col = c(multiseq.col, deseq2.col), lty = c(1,1), cex = 0.9, lwd
```



```
# if you want to make smooth lines...
wh = max(which(num.multiseq == 0))
num.multiseq[1:wh] = num.multiseq[wh+1]*seq(0,wh-1)/wh
wh = max(which(num.deseq2 == 0))
num.deseq2[1:wh] = num.deseq2[wh+1]*seq(0,wh-1)/wh

multiseq.col = "#483D8B"
deseq2.col = "#FF8C00"
par(mar = c(4, 4, 1, 1))
ymax = max(num.multiseq, num.deseq2) + 50
ymin = 0
plot(alpha.list, num.multiseq, ylim=c(ymin,ymax), col=multiseq.col, type = "l", lty = 1, lwd = 1.5, xlab
points(alpha.list, num.deseq2, ylim=c(ymin,ymax), col=deseq2.col, type="l", lty = 1, lwd = 1.5)
```

```
abline(v=0.05, col="grey")
legend(0,ymax, c("multiseq", "DESeq2"), col = c(multiseq.col, deseq2.col), lty = c(1,1), cex = 0.9, lwd
```


multiseq
DESeq2
number of significant tests
FDR