dif_analysis

Kate Weaver

12/20/2021

```
library(limma)
library(SummarizedExperiment)
library(rtracklayer)
library(tidyverse)
library(data.table)
library(readxl)
library(janitor)
library(ggrepel)
library(ggthemes)
library(edgeR)
library(plotly)
library(survcomp)
library(cowplot)
library(AnnotationDbi)
library(EnsDb.Hsapiens.v86)
library(ggplot2)
```

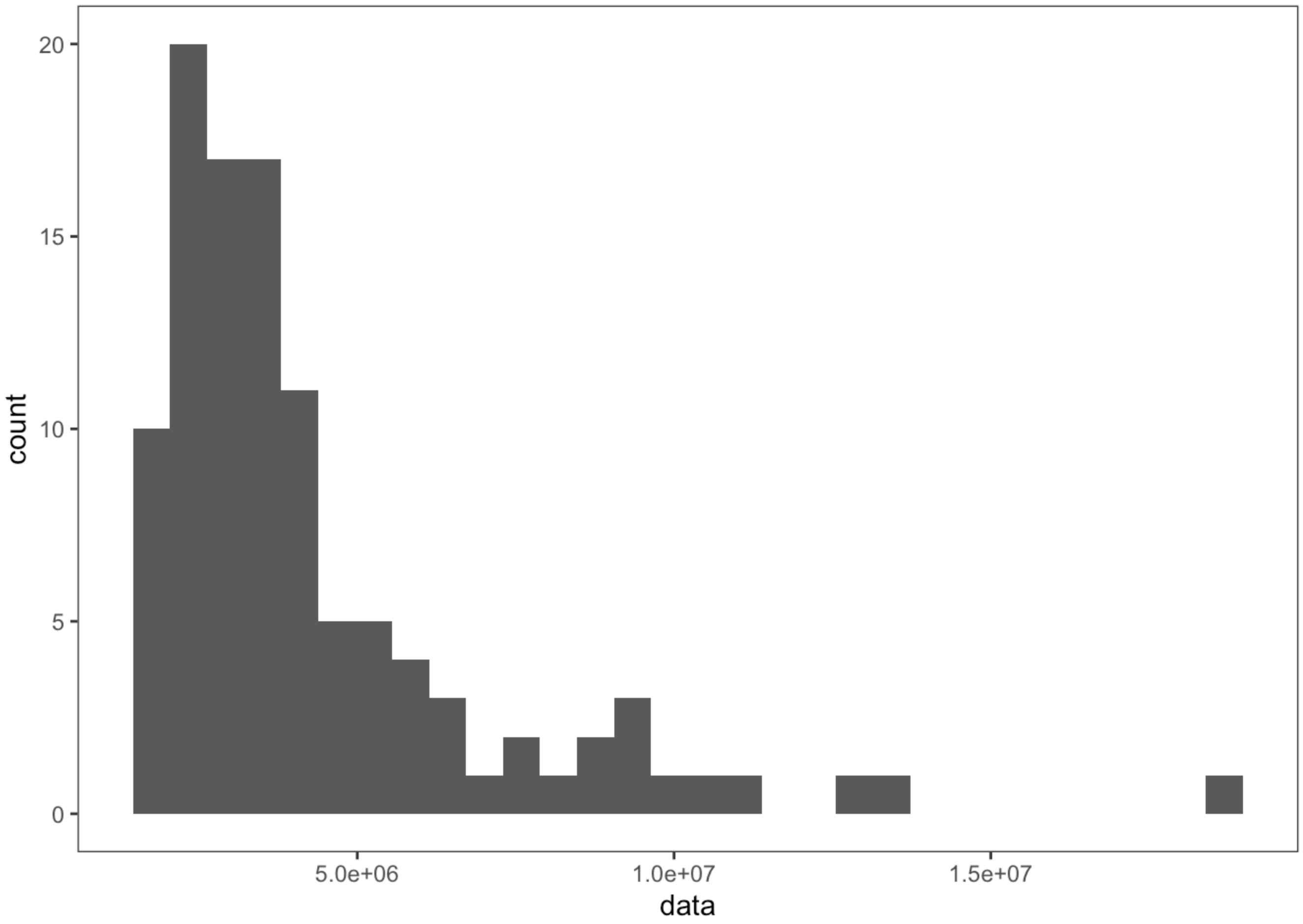
```
qval <- 0.1
```

importing SummarizedExperiment Data and transforming to raw counts data

```
load("~/mccoyLab/collabs/doubleseq_2021/summarized_experiment/create_summarized_experiment_allNov2021.Rdata")
counts <- as.matrix(assays(seAll)$counts)
for (col in 1:ncol(counts)){
  colnames(counts)[col]<-strsplit(sub("Aligned.sortedByCoord.out.bam", "", colnames(counts)[col]), "_")[[1]][1]
}
```

```
#subset to just training data/samples
embryoID_by_set <- read.csv("~/mccoyLab/collabs/doubleseq_2021/data_split/embryo_bySet_full_kw_20211217.csv", col.names=c("embryoID", "set"))
train_embryoIDs = embryoID_by_set$embryoID[embryoID_by_set$set == "train"]
train_cols <- which(colnames(counts) %in% train_embryoIDs)

counts <- counts[,train_cols]

ggplot(data.frame(data=colSums(counts)), aes(x=data)) + geom_histogram() + theme_bw() + theme(panel.background = element_blank(), panel.grid = element_blank()) #library size is right-skewed, normal distribution, but fairly variable
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
message(paste0("nearly ", max(colSums(counts))/min(colSums(counts)), " fold differenc
e between largest and smallest library size for training samples before filtering"))
```

```
## nearly 12.4328171619122 fold difference between largest and smallest library size
for training samples before filtering
```

Therefore, we should probably use voom before limma-trend

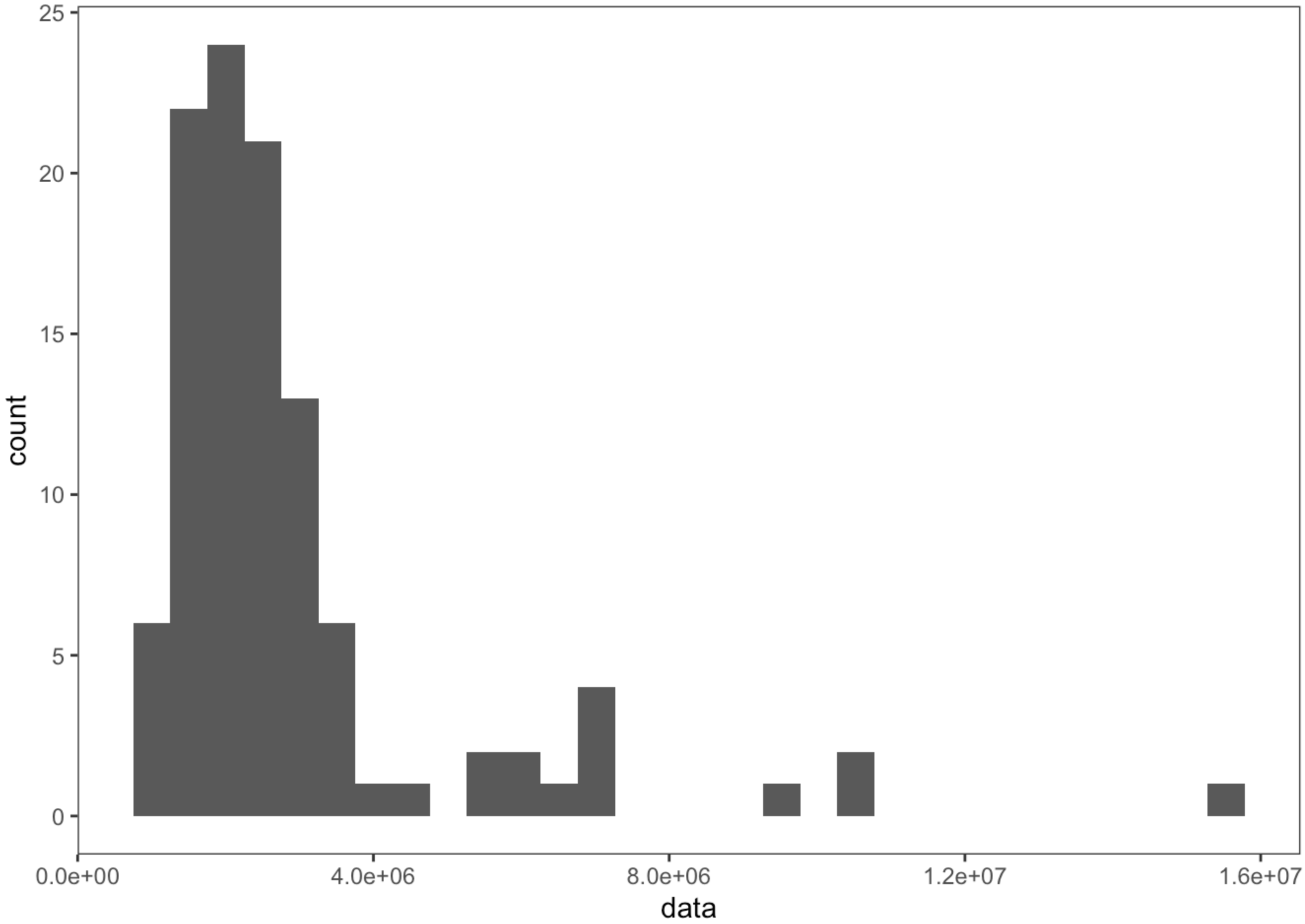
```
#subset to genes with matchind gencode ensembl gene IDs on chr1-22
file_gencode <- "~/genomes/hg38_genome/gencode.v34.annotation.gtf"

gtf <- rtracklayer::import(file_gencode) %>%
  as.data.frame() %>%
  dplyr::filter(type == "gene") %>%
  dplyr::select(gene_id, seqnames, width) %>%
  dplyr::rename(ensembl_gene_id = gene_id) %>%
  dplyr::rename(chromosome_name = seqnames) %>%
  dplyr::rename(length = width)

gene_table <- gtf[match(rownames(counts), gtf$ensembl_gene_id),]
gene_table <- gene_table[gene_table$chromosome_name %in% paste0("chr", 1:22),]
counts <- counts[gene_table$ensembl_gene_id,]
```

```
#Importing metadata to differentiate between pregnant and not pregnant and other main covariates
meta <- read.csv("~/mccoyLab/collabs/doubleseq_2021/tidied_meta/tidied_meta_CREATE_kw_20211217.csv", row.names = 1) %>%
  as.data.frame() %>%
  mutate(across(c("AOD", "GC", "Infertility_type", "Previous_pregnancy", "Past_surgical_hist", "Pregnant", "Ongoing_pregnancy", "Final_outcome", "Embryo_grade_at_freezing", "Interpretation", "cDNA_RT_Date", "Library_Prep_Date", "Sequencing_Date", "Study_Participant_ID"), as.factor)) %>%
  mutate(across(c("InfD_SSM_GC", "InfD_Egg_factor", "InfD_MF", "InfD_Uterine_factor", "InfD_TF", "InfD_RPL", "InfD_RIF", "InfD_Unexplained", "PMdH_none", "PMdH_vasculitis", "PMdH_immune", "PMdH_stress_hormones"), as.factor))

#subsetting meta data to just the training rows
remove <- setdiff(rownames(meta), colnames(counts))
if (length(which(rownames(meta) %in%  remove)) > 0){
  meta <- meta[-which(rownames(meta) %in%  remove),]
}

#need to sort so name order is the same for setting up design for DESeq experiment
counts_order <- order(colnames(counts))
meta_order <- order(rownames(meta))

countdata <- counts[, counts_order]
coldata <- meta[meta_order, ]
```

```r
#OLDFiltering out counts 0 and 1
#keep <- rowSums(countdata) > 1
#countdata <- countdata[keep, ]

#Use more stringent filtering
dgeFullData <- DGEList(countdata, group=as.factor(coldata$Study_Participant_ID))
#normalize counts by TMM
TMMFullData <- calcNormFactors(dgeFullData, method="TMM")
TMMCounts <- as.matrix(TMMFullData$counts)
countsCleaned <- TMMCounts[rowSums(TMMCounts >= 6) > (ncol(TMMCounts)* .2),]

#TPM Calculation
calc_tpm <- function(x, gene.length) {
  x <- as.matrix(x)
  len.norm.lib.size <- colSums(x / gene.length)
  return((t(t(x) / len.norm.lib.size) * 1e06)/ gene.length)
}

#sum(rownames(dgeFullData$counts) == gene_table$ensembl_gene_id) == length(rownames(d
geFullData$counts)) --> TRUE

#creates a matrix with calculated TPM values for each sample from TMM normalized coun
ts and the gene lengths
rawTPMvals <- calc_tpm(TMMFullData, gene.length = gene_table$length)
cleanedTPMVals <- rawTPMvals[rowSums(rawTPMvals > 0.1) > (ncol(rawTPMvals)*.2),]
cleanCountsDf <- as.data.frame(countsCleaned)
cleanTPMdf <- as.data.frame(cleanedTPMVals)

countdata <- cbind(countdata[intersect(rownames(cleanCountsDf), rownames(cleanTPMd
f)),])
```

```r
ggplot(data.frame(data=colSums(countdata)), aes(x=data)) + geom_histogram() + theme_b
w() + theme(panel.grid = element_blank(), panel.background = element_blank()) #librar
y size is right-skewed, normal distribution, but fairly variable
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
message(paste0("nearly ", max(colSums(countdata))/min(colSums(countdata)), " fold difference between largest and smallest library size for training samples before filtering"))
```

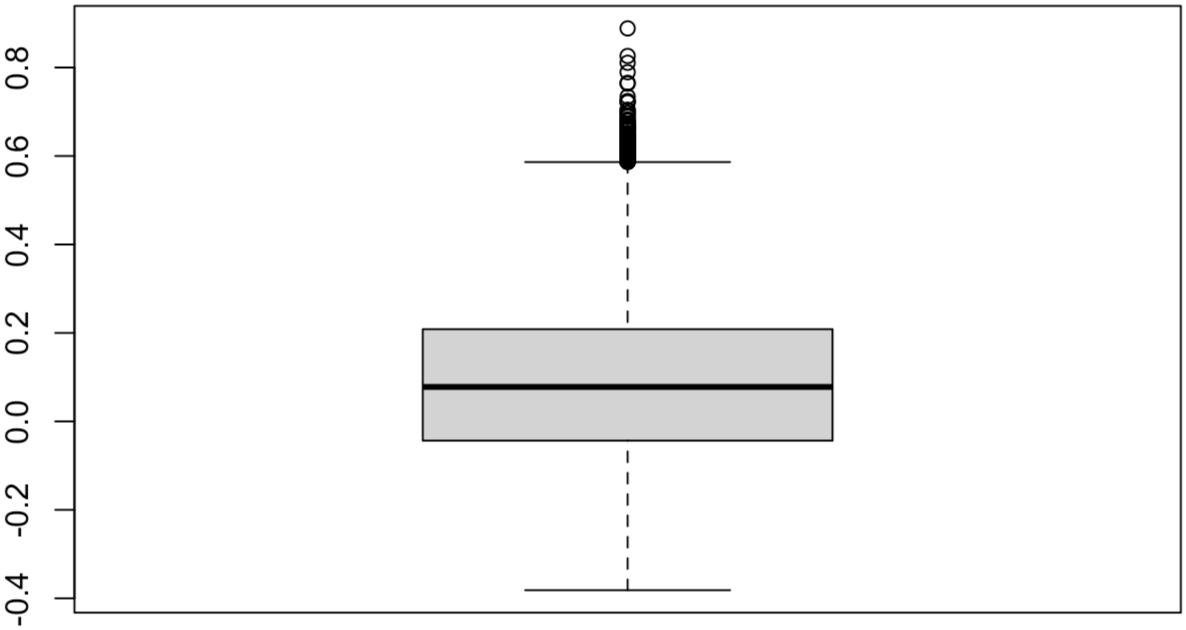
```
## nearly 16.8899467648348 fold difference between largest and smallest library size for training samples before filtering
```

Should definitely use voom before limma-trend

```r
#set up design matrix for edgeR, limma, & voom
#Response variable / outcome of interest later
Pregnant <- factor(coldata$Pregnant)
# main covariates to consider
Batch <- factor(coldata$Sequencing_Date)
# participant age
Age_O <- as.numeric(coldata$Oocyte_Age)
#lining thickness
lthick <- as.numeric(coldata$lining_thickness_mm)
#embryo grade at freezing
egaf <- factor(coldata$Embryo_grade_at_freezing)
#design0 <- model.matrix( ~ Pregnant + Batch)
#design1 <- model.matrix( ~ Pregnant + Batch + Age_O)
#design2 <- model.matrix( ~ Pregnant + Batch + Age_O + lthick)
design3 <- model.matrix( ~ Pregnant + Batch + Age_O + lthick + egaf)

dge <- edgeR::DGEList(counts = countdata, samples = coldata)

#log counts per million and use of prior.count to damp down the variance of logs of low counts
logCPM <- edgeR::cpm(dge, log=TRUE, prior.count=3)

#requires statmod package

corfit <- duplicateCorrelation(logCPM, design=design3, ndups=1, block=coldata$Study_Participant_ID) # A slow computation
message(corfit$consensus.correlation)
```

```
## 0.0825105560731058
```

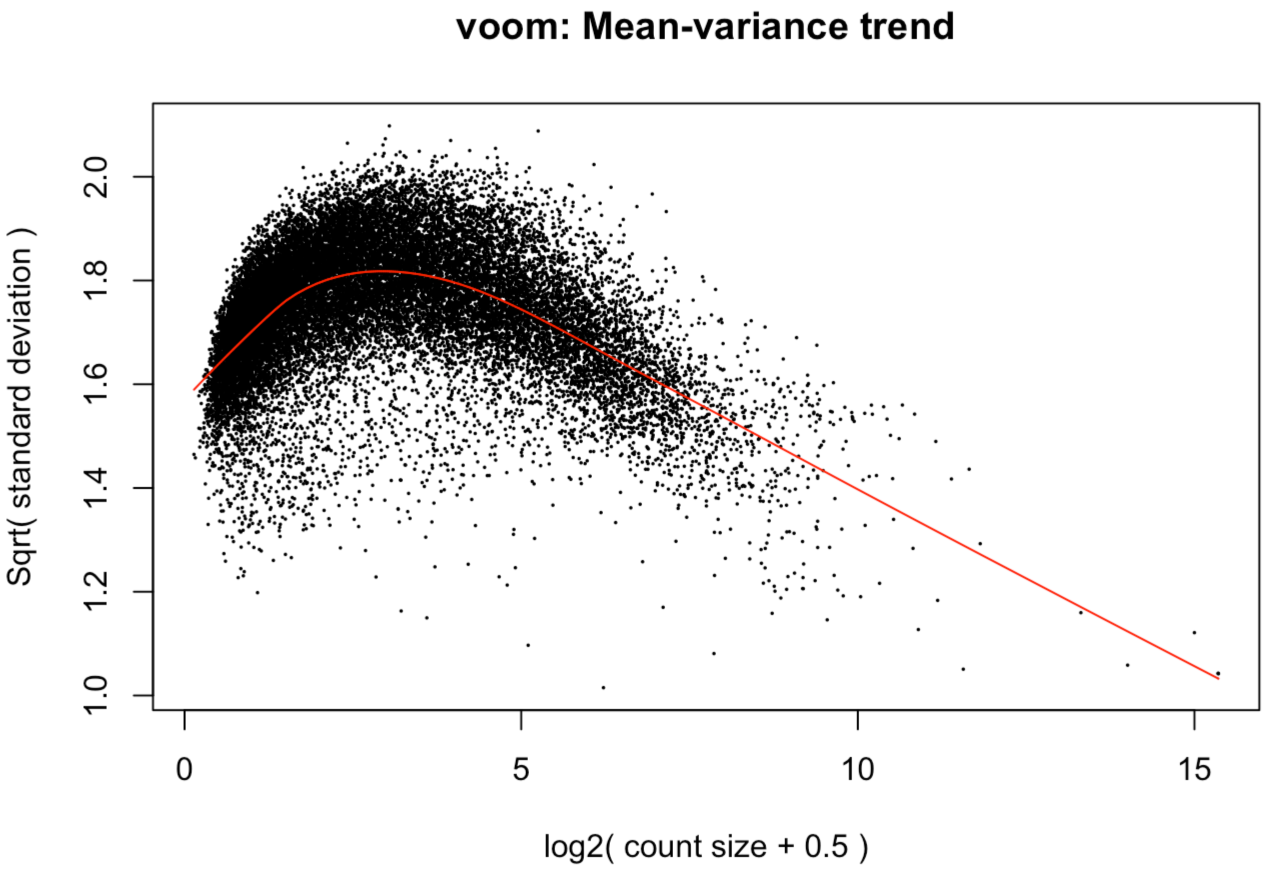
```r
g4 <- boxplot(tanh(corfit$atanh.correlations))
```



```
g4
```

```
## $stats
##             [,1]
## [1,] -0.38159651
## [2,] -0.04354100
## [3,]  0.07787973
## [4,]  0.20847693
## [5,]  0.58628051
## 
## $n
## [1] 22601
## 
## $conf
##            [,1]
## [1,] 0.07523108
## [2,] 0.08052838
## 
## $out
##   [1] 0.6112359 0.6446197 0.5867888 0.6161717 0.6601489 0.5948071 0.6072769
##   [8] 0.6384512 0.6592120 0.6399844 0.5876341 0.8884607 0.6056894 0.5925431
##  [15] 0.6519914 0.6598277 0.6038710 0.6319103 0.6234811 0.6051692 0.6969287
##  [22] 0.6062153 0.5899089 0.5975795 0.6049420 0.6153552 0.6320571 0.5990499
##  [29] 0.5920309 0.6060627 0.6041030 0.6206476 0.6725248 0.7241776 0.5988283
##  [36] 0.5888514 0.5875110 0.7033344 0.6195734 0.6235942 0.6277653 0.6029558
##  [43] 0.6313106 0.6356459 0.6722642 0.7340924 0.6793613 0.6244466 0.6288418
##  [50] 0.6175924 0.6616796 0.6129824 0.5965946 0.6208838 0.6981250 0.6414248
##  [57] 0.6402290 0.6421618 0.6099647 0.6045395 0.6256691 0.5879901 0.6734688
##  [64] 0.6000244 0.6226704 0.6051352 0.6885483 0.6501034 0.5908466 0.6369326
##  [71] 0.6444619 0.5929097 0.7894076 0.6397800 0.6118927 0.6396437 0.6500173
##  [78] 0.5971489 0.6602478 0.6910881 0.5945160 0.5966649 0.6328221 0.6658316
##  [85] 0.5873314 0.5998353 0.5875379 0.6221664 0.6228958 0.6020593 0.6129997
##  [92] 0.6095366 0.6272825 0.5934938 0.8107765 0.5950699 0.6321683 0.6796551
##  [99] 0.6116479 0.6182190 0.7654541 0.5869769 0.6827029 0.6101261 0.6229803
## [106] 0.7637836 0.6514624 0.7044587 0.6082900 0.5875352 0.8261413 0.5923178
## [113] 0.5895519 0.6632305 0.5929197 0.6644561 0.6670204 0.7226311 0.6148179
## [120] 0.7213392 0.6075156 0.6377610 0.6159585 0.6238620
## 
## $group
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1
## 
## $names
## [1] "1"
```

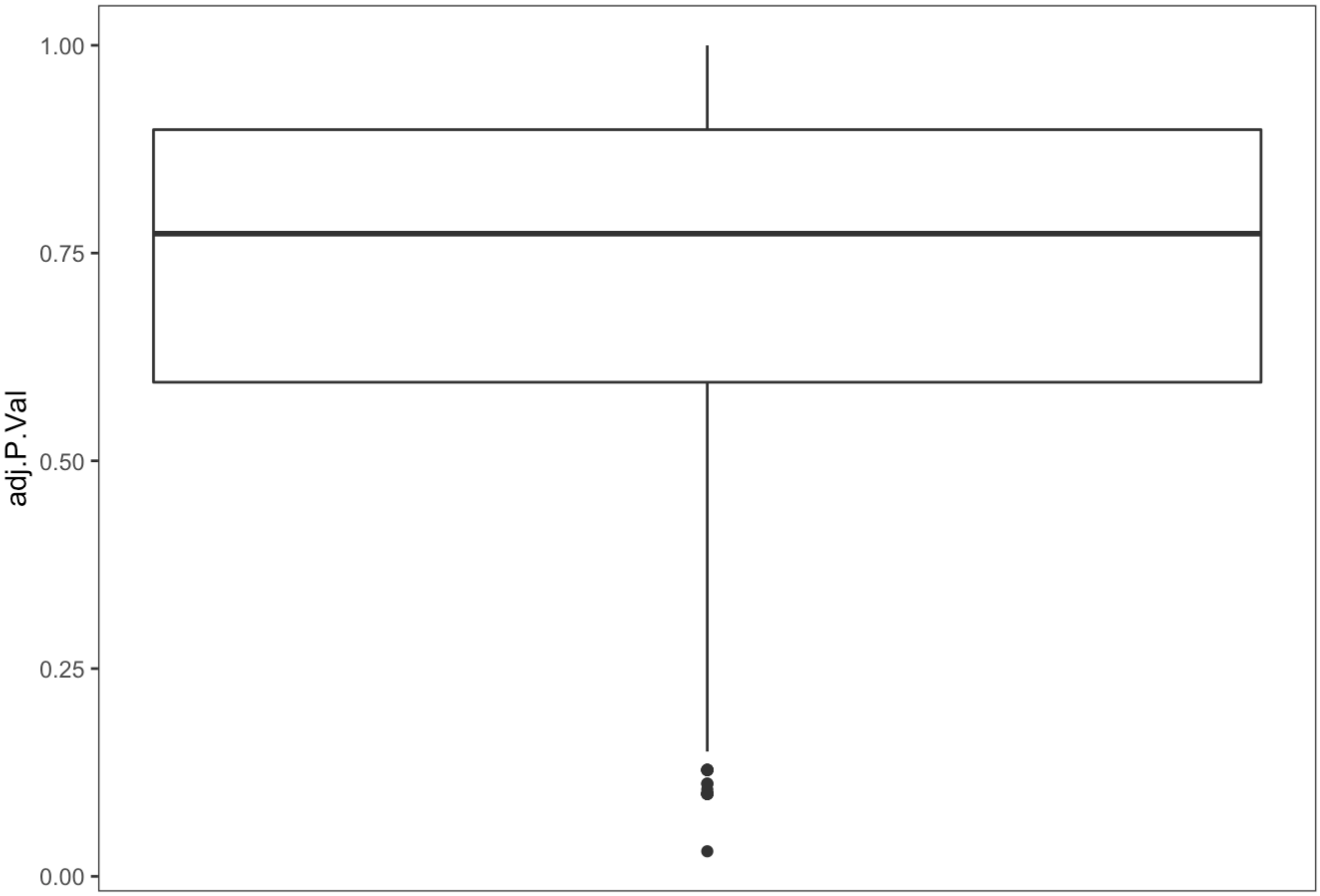
```
v <- limma::voom(dge, design3, plot=TRUE, save.plot = TRUE)
```



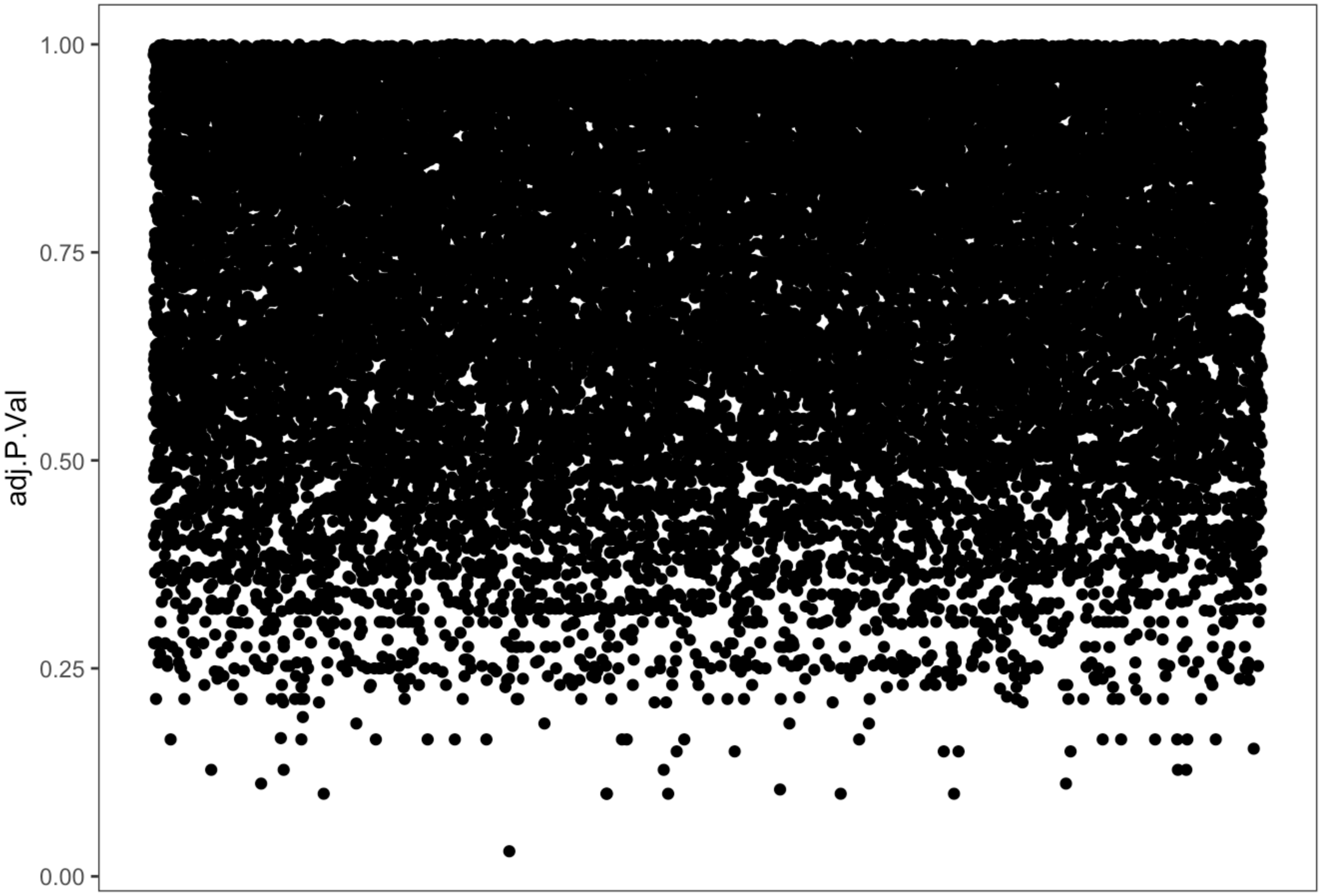
```
save(v, file="~/mccoyLab/collabs/doubleseq_2021/dif_expression_results/voomOut_trainSPA_withBatch.Rdata")

fit <- lmFit(v, design3, correlation = corfit$consensus)
fit
```

```
## An object of class "MArrayLM"
## $coefficients
##                    (Intercept)    Pregnant1      Batch1     Batch2       Age_O
## ENSG00000000419.12   8.6311051  0.008254271 -1.39483510 -1.2284238 -0.13029281
## ENSG00000000457.14   3.8947377 -1.676818764 -1.61528319 -0.9311951 -0.04289475
## ENSG00000000460.17   4.3780885 -0.811935783 -0.68355242  0.7421455 -0.03387664
## ENSG00000000938.13   0.1631185  0.434141172 -0.30779717  0.5203889  0.07291627
## ENSG00000000971.16   5.8760514 -0.151324315  0.09613538 -0.1898835 -0.10597728
##                          lthick       egaf2      egaf3      egaf4      egaf5
## ENSG00000000419.12 -0.154256303 0.002148364  0.5018600  0.8020383  4.3015116
## ENSG00000000457.14 -0.001652822 1.719739776  0.5299222  0.3873885  0.0757614
## ENSG00000000460.17  0.031441902 1.542830411 -0.3975013 -0.4580717  0.2292071
## ENSG00000000938.13 -0.163809553 0.076591805 -1.8837309 -1.1029650 -3.5551158
## ENSG00000000971.16 -0.123587523 3.266275902 -0.5904614  0.5147126  0.7389159
##                         egaf6       egaf7       egaf8      egaf9     egaf10
## ENSG00000000419.12  0.9897865  0.04016172 -0.01086929 -0.6984611 -0.4709502
## ENSG00000000457.14  0.5205278 -2.04048280  0.40620437  0.6203677 -3.4281351
## ENSG00000000460.17  0.4563336 -2.77235332  3.67750247 -0.7427119 -1.6051898
## ENSG00000000938.13 -2.6480132 -1.62125057  3.94539489 -0.6408281  3.7104288
## ENSG00000000971.16 -2.8617688 -2.07854813  0.53641280  1.6791194  2.6585347
##                        egaf11      egaf12
## ENSG00000000419.12 -0.7871771  0.94500565
## ENSG00000000457.14  3.6481867  0.52109505
## ENSG00000000460.17 -6.1410616 -0.07898724
## ENSG00000000938.13 -2.8739169 -3.07519924
## ENSG00000000971.16 -3.0605034  3.86783259
## 22596 more rows ...
##
## $stdev.unscaled
##                    (Intercept) Pregnant1    Batch1    Batch2      Age_O
## ENSG00000000419.12    3.550931 0.6886322 0.8266728 0.7729604 0.08921811
## ENSG00000000457.14    3.448963 0.6724390 0.7814184 0.7482368 0.08699972
## ENSG00000000460.17    3.550856 0.6768345 0.8238515 0.7535519 0.08955509
## ENSG00000000938.13    3.348842 0.6673634 0.7985575 0.7372030 0.08420000
## ENSG00000000971.16    3.432902 0.6744289 0.8221301 0.7341357 0.08534739
##                       lthick    egaf2    egaf3     egaf4    egaf5    egaf6
## ENSG00000000419.12 0.1779428 1.536258 1.309991 0.9911118 1.883718 1.607650
## ENSG00000000457.14 0.1747133 1.500071 1.266831 0.9605443 1.969307 1.585864
## ENSG00000000460.17 0.1785683 1.378922 1.287004 0.9816908 2.125904 1.550283
## ENSG00000000938.13 0.1673022 1.539980 1.257397 0.9953175 1.801031 1.469052
## ENSG00000000971.16 0.1738277 1.430044 1.245167 0.9868146 2.142343 1.424458
##                       egaf7    egaf8    egaf9   egaf10   egaf11   egaf12
## ENSG00000000419.12 1.274674 2.555970 1.121305 3.304065 3.528097 2.114631
## ENSG00000000457.14 1.152537 2.554701 1.103764 2.697298 2.732678 2.131287
## ENSG00000000460.17 1.241499 1.966441 1.151566 3.435910 2.809692 2.053548
## ENSG00000000938.13 1.268344 2.081146 1.152139 2.714899 2.981825 1.761367
## ENSG00000000971.16 1.175438 2.529336 1.151741 3.050489 2.797278 2.001074
## 22596 more rows ...
##
## $sigma
## [1] 0.9487968 1.0615387 0.9795905 1.1025547 1.1160472
## 22596 more elements ...
##
## $df.residual
## [1] 90 90 90 90 90
## 22596 more elements ...
##
## $cov.coefficients
##              (Intercept)    Pregnant1        Batch1        Batch2         Age_O
## (Intercept)  1.316450744 -0.037883784 -0.0397780419 -0.0065617172 -0.0279186291
## Pregnant1   -0.037883784  0.048528744  0.0061837230  0.0089474091  0.0005205790
## Batch1      -0.039778042  0.006183723  0.0682048597  0.0255006851  0.0002482662
## Batch2      -0.006561717  0.008947409  0.0255006851  0.0596221271 -0.0008260296
## Age_O       -0.027918629  0.000520579  0.0002482662 -0.0008260296  0.0008176764
##                    lthick         egaf2        egaf3         egaf4        egaf5
## (Intercept) -2.736124e-02 -1.105305e-01 -0.020541746 -0.0002357444  0.079781787
## Pregnant1   -2.323062e-03  7.055642e-03  0.021725582  0.0153912441  0.018798900
## Batch1      -1.134501e-04  8.815889e-03  0.001865631  0.0045789616 -0.038870427
## Batch2       3.344692e-04  8.538123e-05  0.005181515  0.0002305227  0.006398339
## Age_O       -4.811847e-05  1.696204e-03 -0.001293556 -0.0012467435 -0.003366241
##                    egaf6         egaf7        egaf8        egaf9        egaf10
## (Intercept) -0.014386465 -0.0298661515  0.107086764  0.009455159 -0.1657234882
## Pregnant1   -0.002006960  0.0005258287  0.014344757  0.013047829 -0.0115626366
## Batch1       0.008074117  0.0152109393  0.027585044 -0.007407154  0.0261956862
## Batch2       0.009436482  0.0109598398  0.032777769  0.002670168  0.0225320008
## Age_O       -0.001712637 -0.0008412626 -0.005409268 -0.001453557  0.0007515577
##                   egaf11        egaf12
## (Intercept)  0.129744124 -0.0426993823
## Pregnant1    0.039250081  0.0365821422
## Batch1       0.030485364  0.0150572597
## Batch2       0.037910269 -0.0070693368
## Age_O       -0.005942596 -0.0002132919
## 12 more rows ...
##
## $pivot
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17
##
## $rank
## [1] 17
##
## $Amean
## ENSG00000000419.12 ENSG00000000457.14 ENSG00000000460.17 ENSG00000000938.13
##          2.0561139          0.9538320          2.6394283          0.4770247
## ENSG00000000971.16
##          1.0267008
## 22596 more elements ...
##
## $method
## [1] "ls"
##
## $design
##   (Intercept) Pregnant1 Batch1 Batch2 Age_O lthick egaf2 egaf3 egaf4 egaf5
## 1           1         1      1      0    34     10      0     0     1     0
## 2           1         0      0      1    36      8      0     0     1     0
## 3           1         1      0      0    31     10      0     0     1     0
## 4           1         0      0      0    31      8      0     1     0     0
## 5           1         1      0      0    34     13      0     0     0     0
##   egaf6 egaf7 egaf8 egaf9 egaf10 egaf11 egaf12
## 1     0     0     0     0      0      0      0
## 2     0     0     0     0      0      0      0
## 3     0     0     0     0      0      0      0
## 4     0     0     0     0      0      0      0
## 5     0     1     0     0      0      0      0
## 102 more rows ...
```

```
fit <- eBayes(fit)
res = topTable(fit, n=Inf, sort="p", coef=2)

ggplot(res, aes(x=1, y=adj.P.Val)) + geom_boxplot() + theme_bw() + theme(panel.grid =
element_blank(), panel.background = element_blank()) + xlab("") + theme(axis.text.x=e
lement_blank(), axis.ticks.x=element_blank())
```



```
ggplot(res, aes(x=1, y=adj.P.Val)) + geom_jitter(width=0.3, height=0) + theme_bw() +
theme(panel.grid = element_blank(), panel.background = element_blank()) + xlab("") +
theme(axis.text.x=element_blank(), axis.ticks.x=element_blank())
```



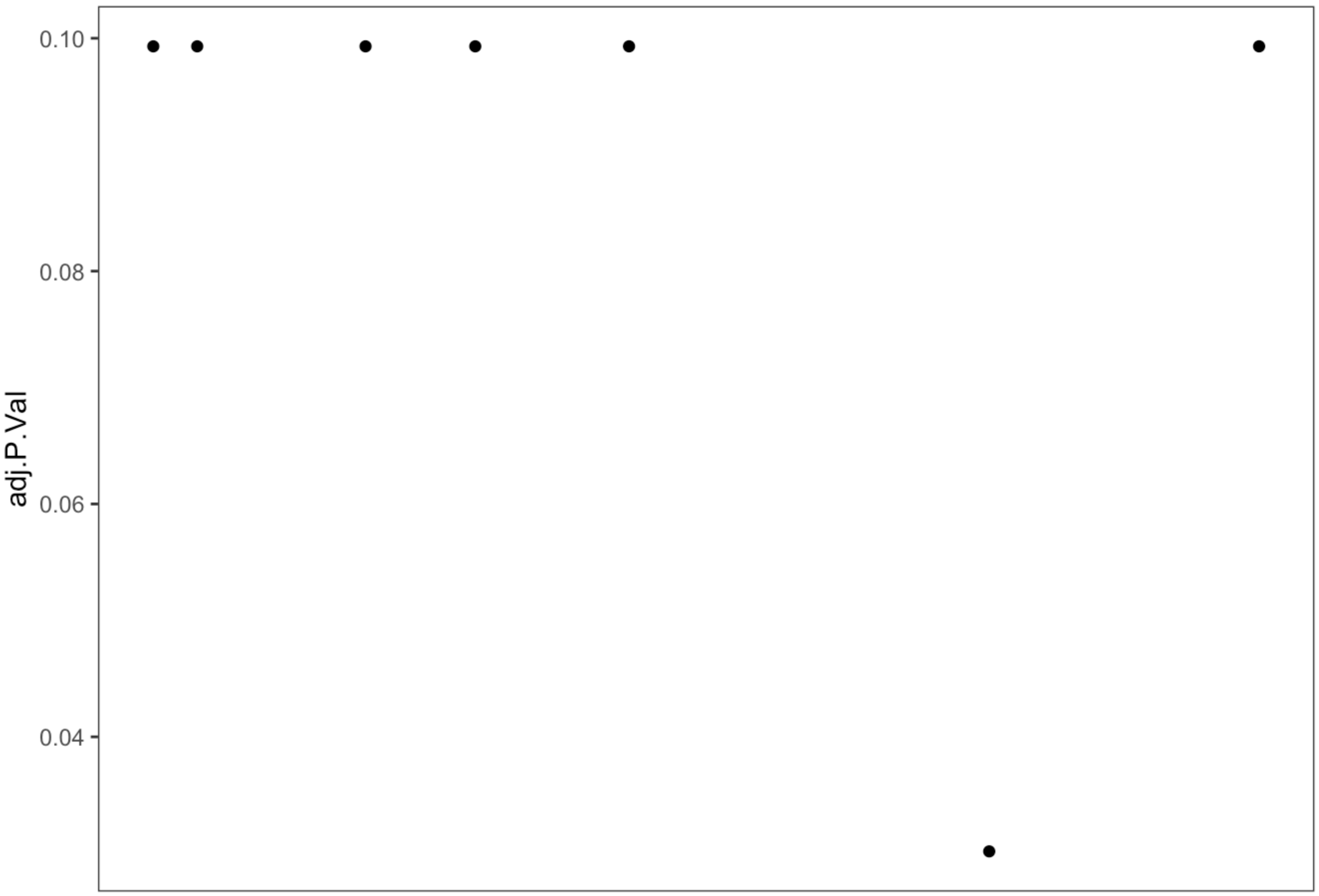
```
#mapping from ENSEMBL Gene ID to SYMBOL and ENTREZ
ens.str <- substr(rownames(res), 1, 15)
edb <- EnsDb.Hsapiens.v86
res$symbol <- mapIds(edb, keys=ens.str, column="SYMBOL", keytype="GENEID", multiVals
="first")
```

```
## Warning: Unable to map 897 of 22601 requested IDs.
```

```
res$entrez <- mapIds(edb, keys=ens.str, column="ENTREZID", keytype="GENEID", multiVal
s="first")
```

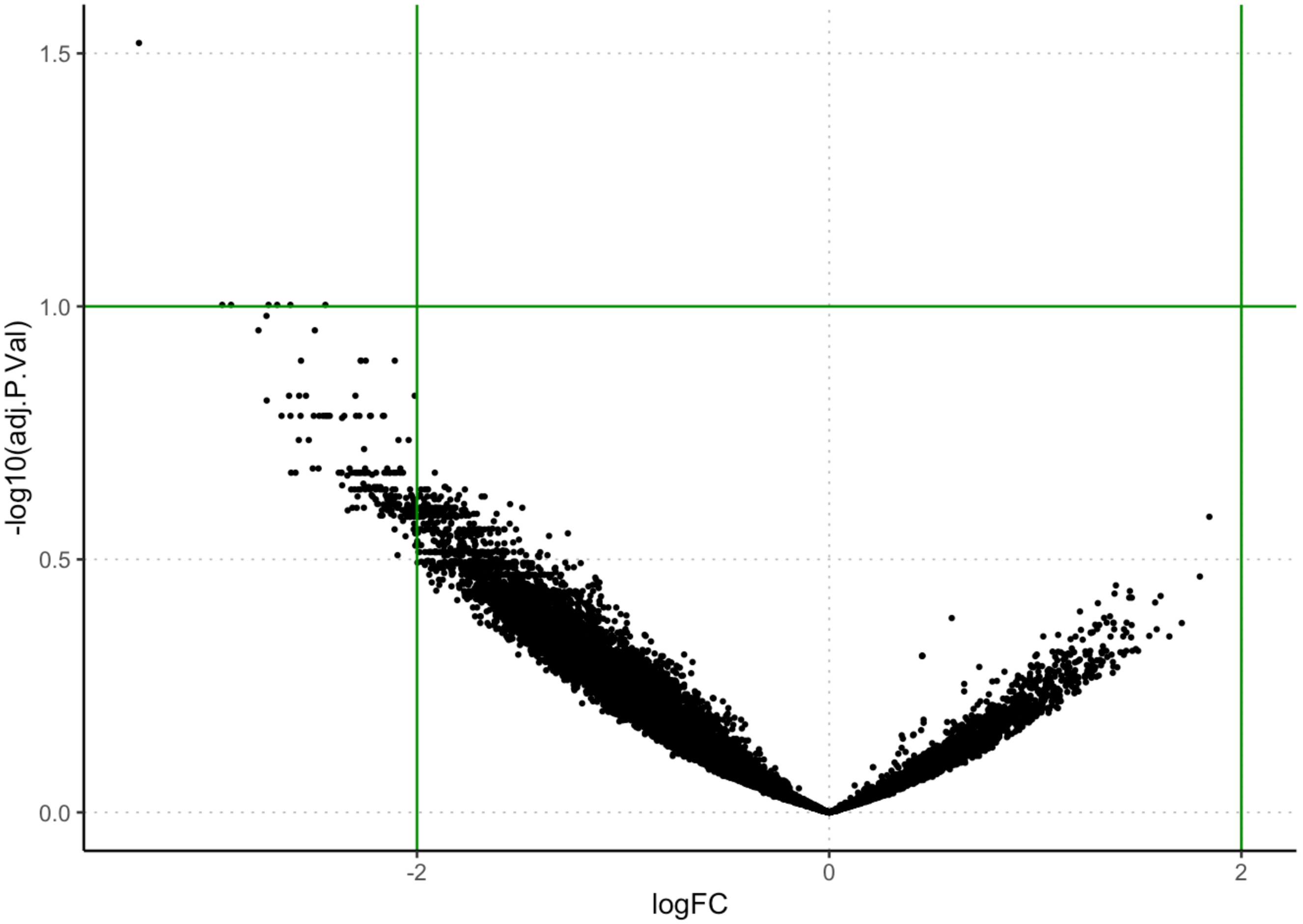
```
## Warning: Unable to map 6666 of 22601 requested IDs.
```

```
res.qval <- subset(res, adj.P.Val<qval) %>%
  as.data.frame()
write.csv(res.qval, file = "~/mccoyLab/collabs/doubleseq_2021/dif_expression_results/
results_trainSPA_dupCor_withBatch.qval.csv")

ggplot(res.qval, aes(x=1, y=adj.P.Val)) + geom_jitter(width=0.1, height=0) + theme_bw
() + theme(panel.grid = element_blank(), panel.background = element_blank()) + xlab
("") + theme(axis.text.x=element_blank(), axis.ticks.x=element_blank())
```

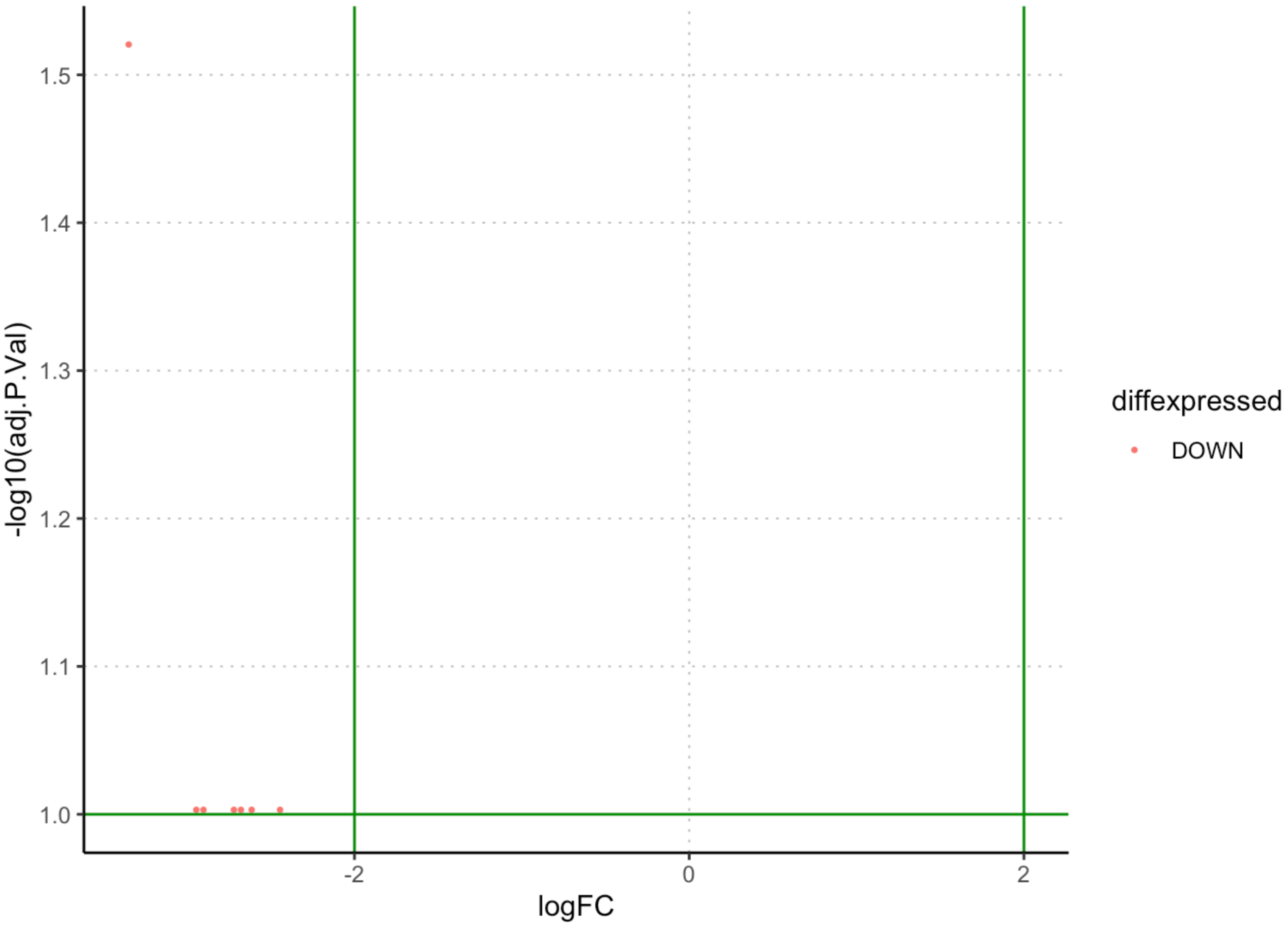


```
#exporting all genes
write.csv(as.data.frame(res), file="~/mccoyLab/collabs/doubleseq_2021/dif_expression_
results/results_trainSPA_dupCor_withBatch.all.csv")
```

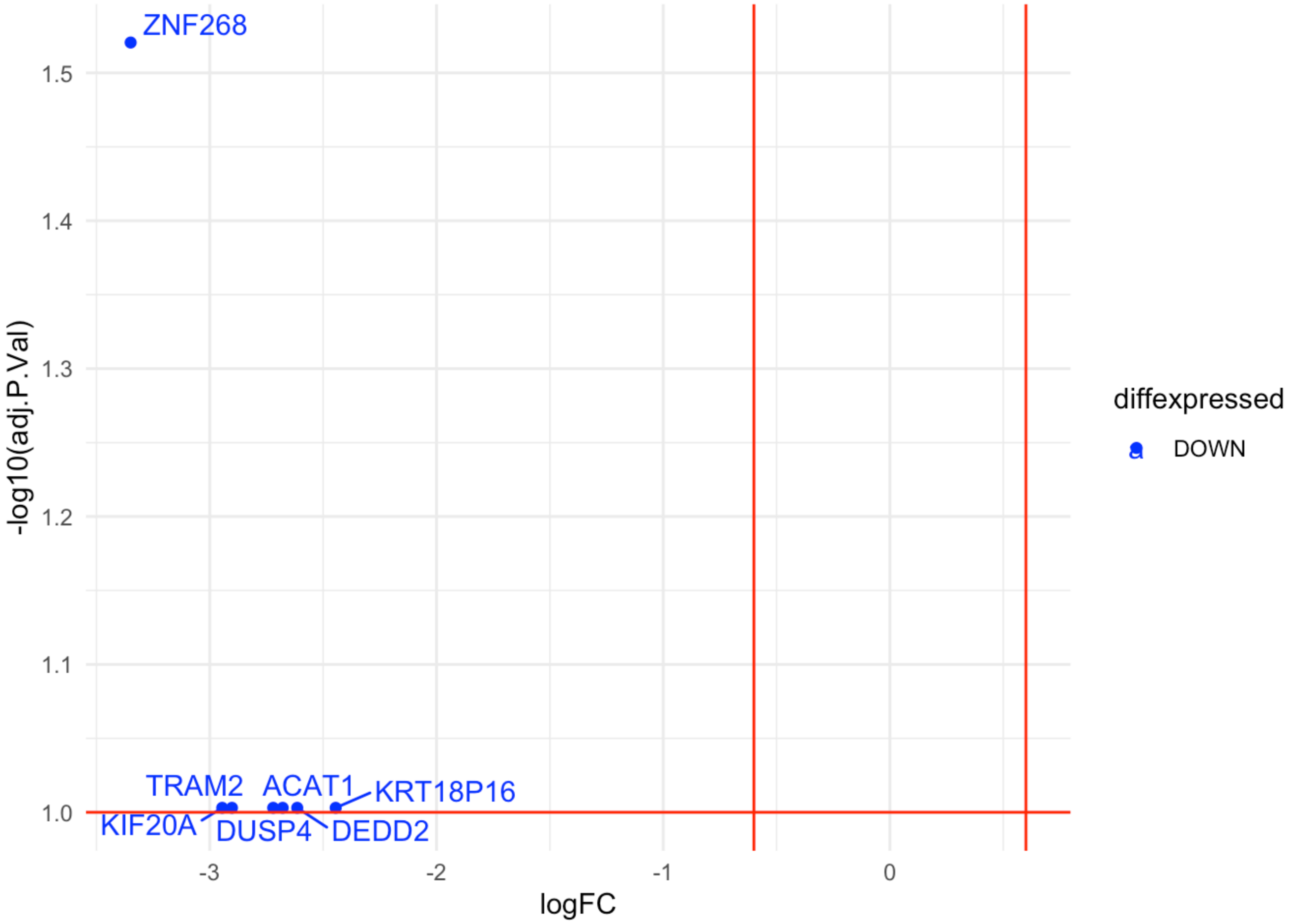
```
#plotting volcano plot
res.qval$diffexpressed <- "NO"
res.qval$diffexpressed[res.qval$logFC > 2 & res.qval$adj.P.Val < qval] <- "UP"
res.qval$diffexpressed[res.qval$logFC < -2 & res.qval$adj.P.Val < qval] <- "DOWN"

g0 <-  ggplot(data = res, aes(x=logFC, y=-log10(adj.P.Val))) + geom_point(size=0.5) +
theme_classic()
g0 <- g0 + theme(panel.grid.major = element_line(color="gray70", size=0.3, linetype=
3)) + geom_vline(xintercept=c(-2,2), col="green4") + geom_hline(yintercept=-log10(qva
l), col="green4")
g0
```



```
g <- ggplot(data=res.qval, aes(x=logFC, y=-log10(adj.P.Val), col=diffexpressed)) + ge
om_point(size=0.5) + theme_classic()
g <- g + theme(panel.grid.major = element_line(color="gray70", size=0.3, linetype=3))
+ geom_vline(xintercept=c(-2,2), col="green4") + geom_hline(yintercept=-log10(qval),
col="green4")
g
```



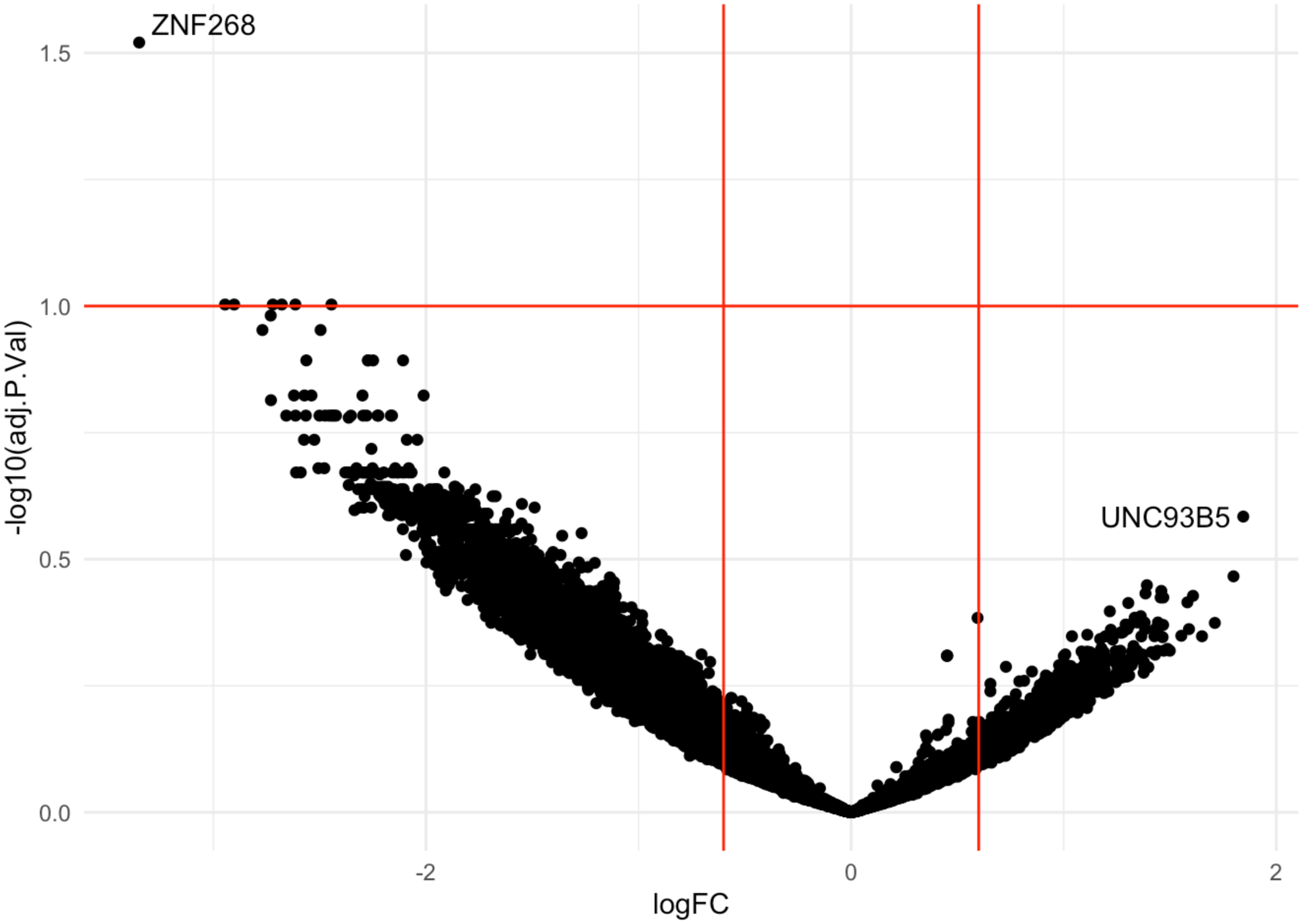
```
g1 <- g + scale_color_manual(values=c("blue", "gray", "red"))

res.qval$delabel <- NA
res.qval$delabel[res.qval$diffexpressed != "NO"] <- res.qval$symbol[res.qval$diffexpr
essed != "NO"]

g1 <- ggplot(data = res.qval, aes(x=logFC, y=-log10(adj.P.Val), col=diffexpressed, la
bel=delabel)) +
  geom_point() +
  theme_minimal() +
  geom_text_repel() +
  scale_color_manual(values=c("blue", "black", "red")) +
  geom_vline(xintercept=c(-0.6, 0.6), col="red") +
  geom_hline(yintercept=-log10(qval), col="red")
g1
```



```
g1 <- ggplot(data = res, aes(x=logFC, y=-log10(adj.P.Val))) +
  geom_point() +
  theme_minimal() +
  geom_text_repel(data = res[(res$adj.P.Val < 0.4),], aes(x=logFC, y=-log10(adj.P.Va
l), label = symbol)) +
  scale_color_manual(values=c("blue", "black", "red")) +
  geom_vline(xintercept=c(-0.6, 0.6), col="red") +
  geom_hline(yintercept=-log10(qval), col="red")
g1
```

```
## Warning: Removed 10 rows containing missing values (geom_text_repel).
```

```
## Warning: ggrepel: 1452 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



Down-expressed genes are those that are down-expressed in outcomes that lead to implantation Up-expressed genes are those that are up-expressed in outcomes that lead to implantation