lab7_2

2024-05-16

```
library("DESeq2")
```

```
## Warning: package 'DESeq2' was built under R version 4.3.3
```

```
## Loading required package: S4Vectors
```

```
## Warning: package 'S4Vectors' was built under R version 4.3.2
```

```
## Loading required package: stats4
```

```
## Loading required package: BiocGenerics
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min
```

```
##
## Attaching package: 'S4Vectors'
```

```
## The following object is masked from 'package:utils':
##
##     findMatches
```

```
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
```

```
## Loading required package: IRanges
```

```
## Loading required package: GenomicRanges
```

```
## Loading required package: GenomeInfoDb
```

```
## Warning: package 'GenomeInfoDb' was built under R version 4.3.3
```

```
## Loading required package: SummarizedExperiment
```

```
## Loading required package: MatrixGenerics
```

```
## Loading required package: matrixStats
```

```
## Warning: package 'matrixStats' was built under R version 4.3.2
```

```
##
## Attaching package: 'MatrixGenerics'
```

```
## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars
```

```
## Loading required package: Biobase
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
##
## Attaching package: 'Biobase'
```

```
## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians
```

```
## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```
library("ggplot2")
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
# 4
counts <- read.csv("WholeBlood_counts.csv", row.names = 1)
# 5
col.mat <- data.frame( donor=factor(c(1,3,3,2,2,1)),
                       treatment=factor(c("Buffer", "MRSA",
                                          "Buffer", "MRSA",
                                          "Buffer", "MRSA")) )
col.mat
```

```
##   donor treatment
## 1     1    Buffer
## 2     3      MRSA
## 3     3    Buffer
## 4     2      MRSA
## 5     2    Buffer
## 6     1      MRSA
```

```
# 6, 7
pre.dds <- DESeqDataSetFromMatrix( countData = counts, colData=col.mat, design = ~ donor
+treatment )
# 8
dds <- DESeq( pre.dds )
```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

```
# 9
resultsNames(dds)
```

```
## [1] "Intercept"               "donor_2_vs_1"
## [3] "donor_3_vs_1"            "treatment_MRSA_vs_Buffer"
```

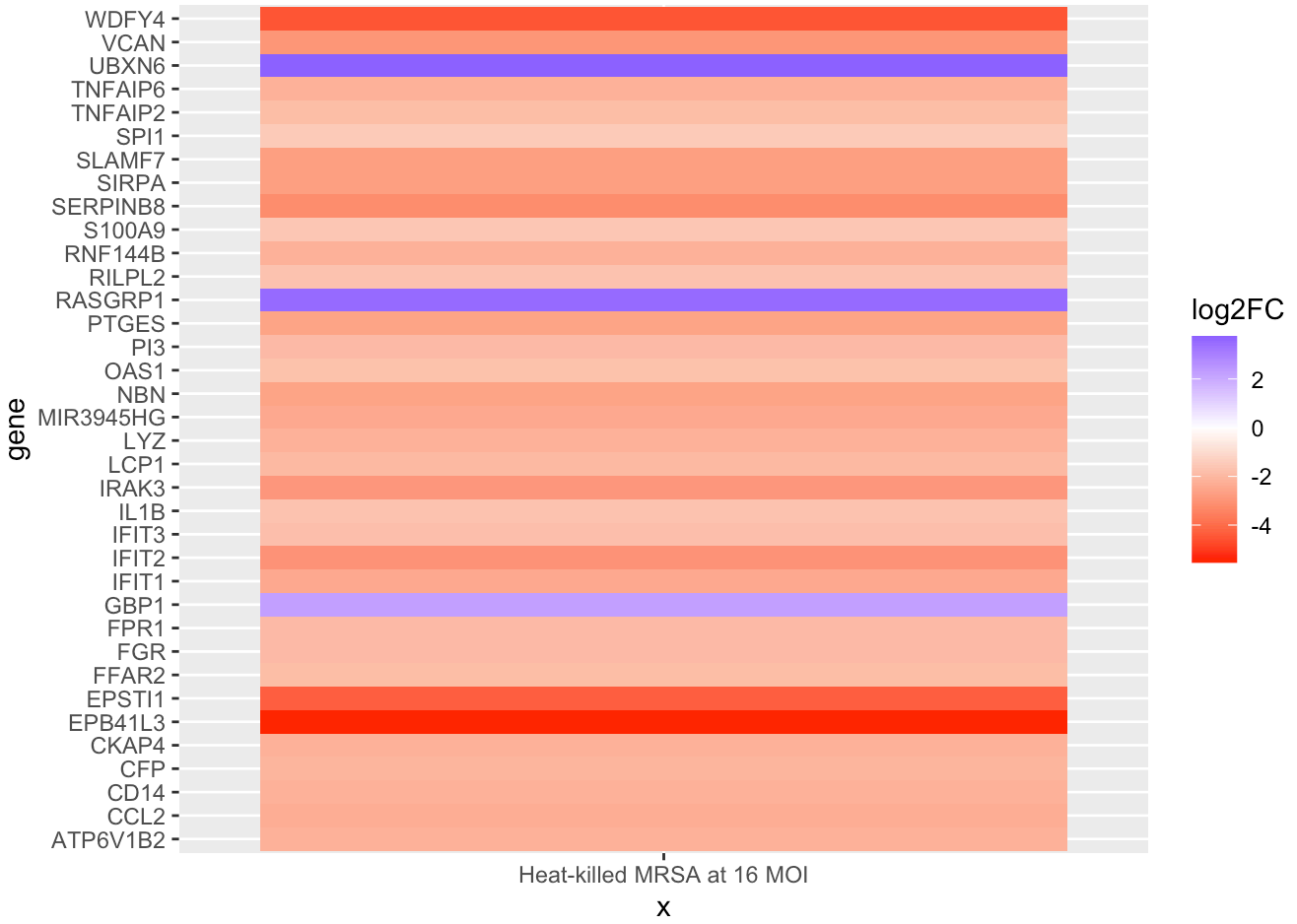
```
# 10
res <- results( dds, name = "treatment_MRSA_vs_Buffer", alpha = 0.05)
summary(res)
```

```
##
## out of 16290 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 3, 0.018%
## LFC < 0 (down)     : 33, 0.2%
## outliers [1]       : 0, 0%
## low counts [2]     : 12436, 76%
## (mean count < 5)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
# 11
# res@rownames
# res$log2FoldChange
# res$padj

# 12
p.adj <- res$padj
log2FC <- res$log2FoldChange

# 13
names(p.adj) <- names(log2FC) <- res@rownames

# 14
which.DEGs <- p.adj<0.05 & abs(log2FC)>1 & !is.na(p.adj)
# 15
DEGs <- data.frame(names(p.adj)[which.DEGs], p.adj[which.DEGs], log2FC[which.DEGs])
colnames(DEGs) <- c("gene", "p.adj", "log2FC")
DEGs
```

```
##                gene        p.adj    log2FC
## ATP6V1B2   ATP6V1B2 0.0333895656 -2.163801
## CCL2           CCL2 0.0089631898 -2.264737
## CD14           CD14 0.0278975780 -2.199438
## CFP             CFP 0.0452106713 -2.101039
## CKAP4         CKAP4 0.0089631898 -2.201746
## EPB41L3     EPB41L3 0.0137537695 -5.538506
## EPSTI1       EPSTI1 0.0089631898 -4.307507
## FFAR2         FFAR2 0.0278975780 -1.863574
## FGR             FGR 0.0101470074 -1.919043
## FPR1           FPR1 0.0491303130 -1.923186
## GBP1           GBP1 0.0091314632  2.253520
## IFIT1         IFIT1 0.0001564909 -2.457976
## IFIT2         IFIT2 0.0001247703 -2.971993
## IFIT3         IFIT3 0.0233698972 -1.827538
## IL1B           IL1B 0.0233698972 -1.732563
## IRAK3         IRAK3 0.0384596189 -2.873872
## LCP1           LCP1 0.0032302783 -1.949097
## LYZ             LYZ 0.0491303130 -2.153569
## MIR3945HG MIR3945HG 0.0436030790 -2.369481
## NBN             NBN 0.0137537695 -2.590427
## OAS1           OAS1 0.0443310671 -1.753340
## PI3             PI3 0.0061139221 -1.908441
## PTGES         PTGES 0.0004925398 -2.597111
## RASGRP1     RASGRP1 0.0080548552  3.584057
## RILPL2       RILPL2 0.0214329602 -1.697297
## RNF144B     RNF144B 0.0116756550 -2.229662
## S100A9       S100A9 0.0491303130 -1.584933
## SERPINB8   SERPINB8 0.0466945759 -3.235457
## SIRPA         SIRPA 0.0061963743 -2.722516
## SLAMF7       SLAMF7 0.0126434617 -2.725524
## SPI1           SPI1 0.0466945759 -1.539702
## TNFAIP2     TNFAIP2 0.0089631898 -1.845831
## TNFAIP6     TNFAIP6 0.0002435253 -2.237214
## UBXN6         UBXN6 0.0080548552  3.753348
## VCAN           VCAN 0.0491303130 -2.962815
## WDFY4         WDFY4 0.0214329602 -4.468822
```

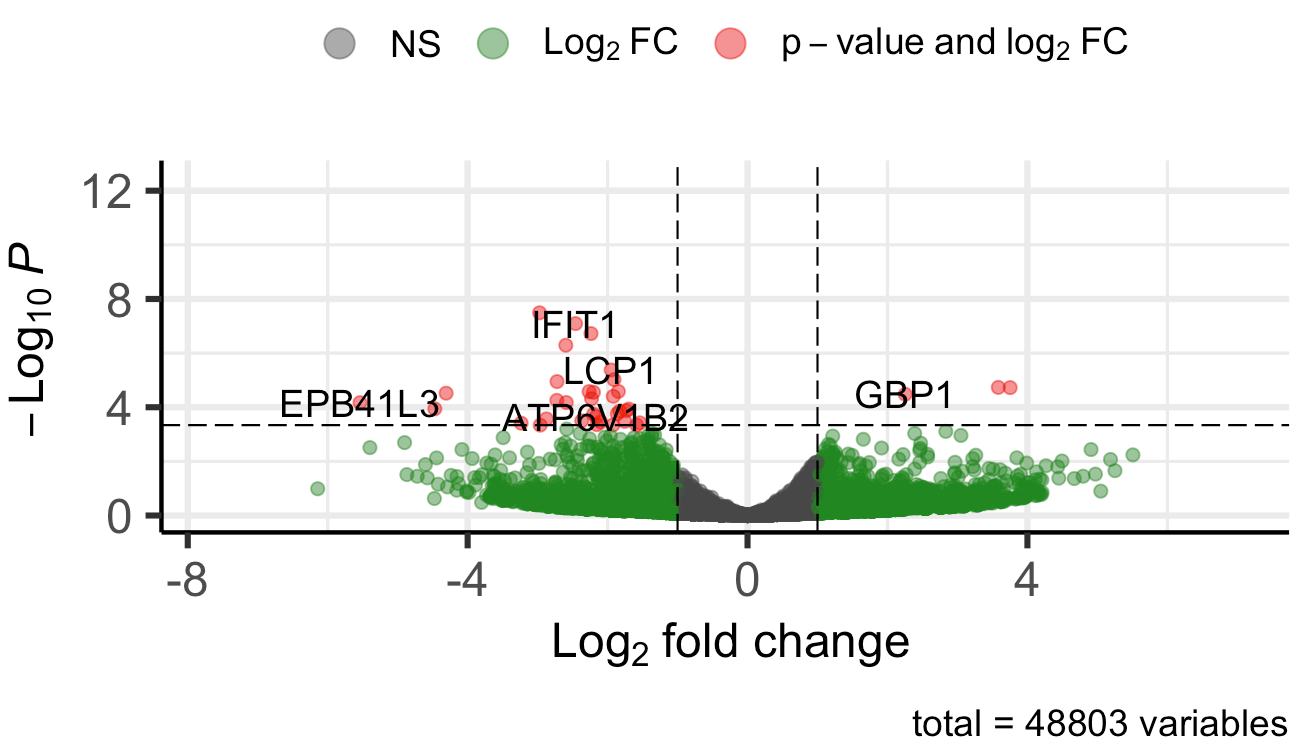
```
# 16
ggplot( data=DEGs, aes(x="Heat-killed MRSA at 16 MOI", y=gene, fill = log2FC))+
  geom_tile()+
  scale_fill_gradient2(low="red", mid="white", high="blue")
```




```
# 17
library("EnhancedVolcano")
```

```
## Loading required package: ggrepel
```

```
EnhancedVolcano(res, lab=rownames(res),
                x="log2FoldChange", y="pvalue",
                pCutoffCol = "padj",
                pCutoff=0.05, FCcutoff = 1)
```




```
# 18, 19
DEG.check <- function(gene, alpha, lfc){
  # 20, 21
  if(p.adj[gene]<alpha & abs(log2FC[gene])>lfc & !is.na(p.adj[gene]) ){
    # 22
    return("Significant")
  }
  else{
    # 23
    return("Not significant")
  }
}
# 24
DEG.check("WDFY4", 0.05, 1)
```

```
## [1] "Significant"
```

```
DEG.check("A2M", 0.05, 1)
```

```
## [1] "Not significant"
```

Section 4

Q1

Number of upregulated: 3 Number of downregulated: 33

Q2

This gene is not a DEGs as it has a very small p value and log2 is very close to 0.

Q3

UBXN6 is most up regulated and EPB41L3 is most down regulated according the value of log2FC where UBXN6 has the highest positive value and EPB41L3 has the lowest negative value in DEGs.

Q4

The heatmap focus on how gene expression changed relatived to the control condition. The log2(fold-change) values already use the control condition as the reference point, so we don't need additional control buffer column.

Q5

The two vertical lines in the volcano plot indicated the boundaries for log2 fold-change, helping to visually identify genes that are significantly up-regulated or down-regulated. One the left side, the gene are down regulated and on the right side, the gene are up regulated. It also distinguishing significant DEGs and non DEGs.

Q6

They are more significantly differential expressed. As the higher point in the graph indicated the lower p value which means it is more significantly.

Q7

```
DEG.check("IFIT1", 0.1, 0)
```

```
## [1] "Significant"
```

```
DEG.check("IFIT1B", 0.1, 0)
```

```
## [1] "Not significant"
```

```
DEG.check("IFIT2", 0.1, 0)
```

```
## [1] "Significant"
```

```
DEG.check("IFIT3", 0.1, 0)
```

```
## [1] "Significant"
```

```
DEG.check("IFIT5", 0.1, 0)
```

```
## [1] "Not significant"
```

Therefore, IFIT1, IFIT2, IFIT3 are DEGs.