

# 12biowriteup

2023-05-22

```
library(BiocManager)
library(DESeq2)

## Loading required package: S4Vectors

## 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

##
## Attaching package: 'IRanges'
```

```

## The following object is masked from 'package:grDevices':
##
##      windows

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## 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

## Warning: replacing previous import 'S4Arrays::read_block' by
## 'DelayedArray::read_block' when loading 'SummarizedExperiment'

counts <- read.csv("airway_scaledcounts.csv", row.names=1)
metadata <- read.csv("airway_metadata.csv")

head(counts)

##          SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516
## ENSG00000000003    723        486       904       445      1170
## ENSG00000000005      0         0         0         0         0
## ENSG00000000419    467       523       616       371      582
## ENSG00000000457    347       258       364       237      318
## ENSG00000000460     96        81        73        66      118
## ENSG00000000938      0         0         1         0         2
##          SRR1039517 SRR1039520 SRR1039521
## ENSG00000000003   1097       806       604
## ENSG00000000005      0         0         0
## ENSG00000000419    781       417       509
## ENSG00000000457    447       330       324
## ENSG00000000460     94        102       74
## ENSG00000000938      0         0         0

head(metadata)

##      id   dex celltype geo_id
## 1 SRR1039508 control N61311 GSM1275862
## 2 SRR1039509 treated N61311 GSM1275863
## 3 SRR1039512 control N052611 GSM1275866
## 4 SRR1039513 treated N052611 GSM1275867
## 5 SRR1039516 control N080611 GSM1275870
## 6 SRR1039517 treated N080611 GSM1275871

```

```
View(metadata)
```

- Q1. How many genes in the set? 38694  
 Q2. How many ‘control’ cell lines are there? 4

```

control <- metadata[metadata[, "dex"]=="control",]
control.counts <- counts[,control$id]
control.mean <- rowSums(control.counts)/4
head(control.mean)

## ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
##           900.75          0.00        520.50       339.75        97.25
## ENSG00000000938
##           0.75

```

Q4.

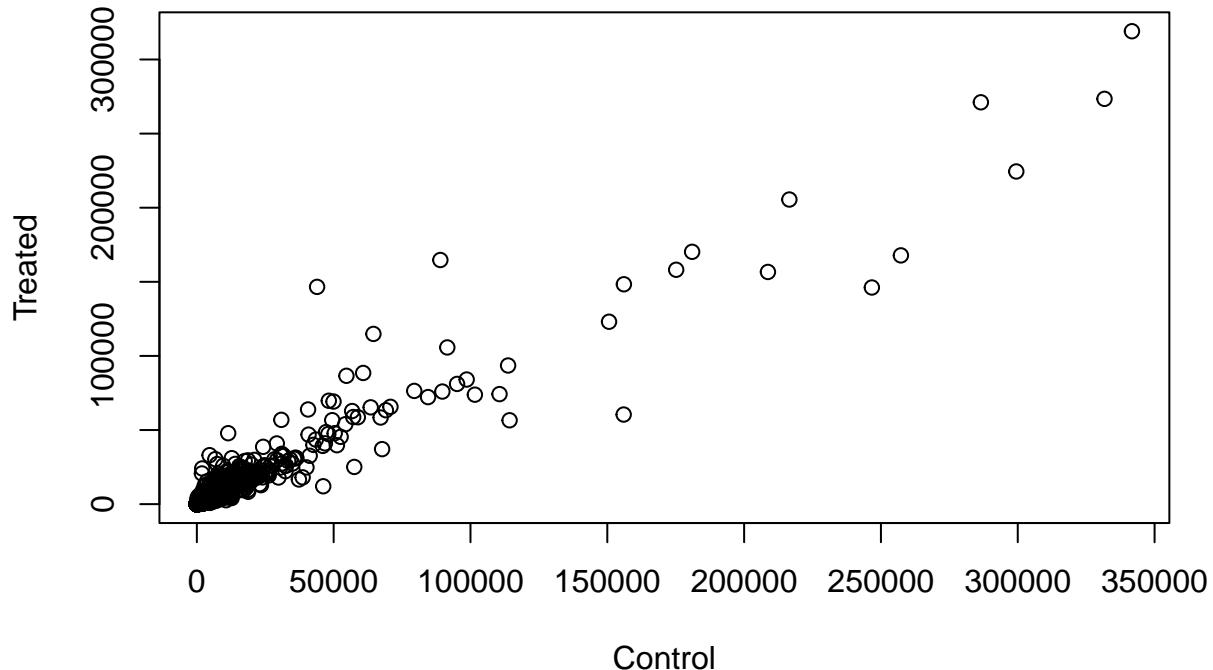
```
## mean per gene across drug treated samples
treated <- metadata[metadata[, "dex"]=="treated",]
treated.mean <- rowSums( counts[ ,treated$id] )/4
names(treated.mean) <- counts$ensgene

##combine the mean datas together
meancounts <- data.frame(control.mean, treated.mean)

## sum of mean counts across all genes for each group
colSums(meancounts)

## control.mean treated.mean
##      23005324      22196524

plot(meancounts[,1],meancounts[,2], xlab="Control", ylab="Treated")
```



Q5B. What geom?() function would you use to create a ggplot out of this data?

geom\_point()

Q6. What argument to plot() allows you to plot both axes on a log scale?

log=xy

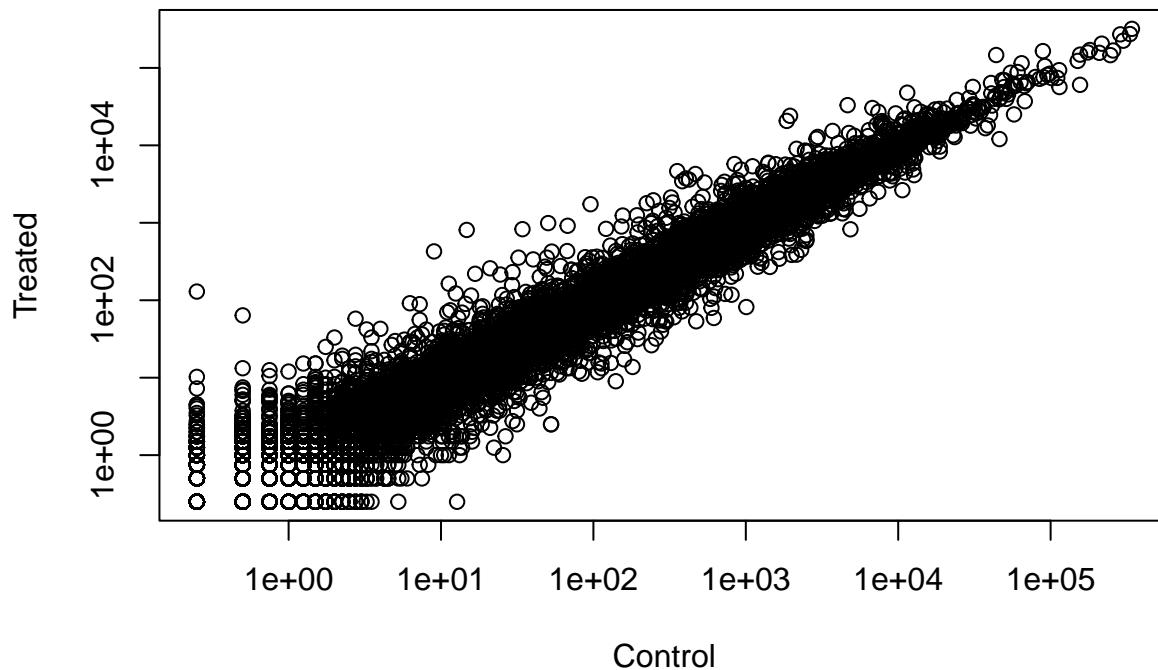
```

plot(meancounts[,1],meancounts[,2], log = "xy", xlab="Control", ylab="Treated")

## Warning in xy.coords(x, y, xlabel, ylabel, log): 15032 x values <= 0 omitted
## from logarithmic plot

## Warning in xy.coords(x, y, xlabel, ylabel, log): 15281 y values <= 0 omitted
## from logarithmic plot

```



```

#calculate log2foldchange and add it to meancounts dataframe
meancounts$log2fc <- log2(meancounts[,"treated.mean"]/meancounts[,"control.mean"])
head(meancounts)

```

|                     | control.mean | treated.mean | log2fc      |
|---------------------|--------------|--------------|-------------|
| ## ENSG000000000003 | 900.75       | 658.00       | -0.45303916 |
| ## ENSG000000000005 | 0.00         | 0.00         | NaN         |
| ## ENSG00000000419  | 520.50       | 546.00       | 0.06900279  |
| ## ENSG00000000457  | 339.75       | 316.50       | -0.10226805 |
| ## ENSG00000000460  | 97.25        | 78.75        | -0.30441833 |
| ## ENSG00000000938  | 0.75         | 0.00         | -Inf        |

```

##filtering out NaN/zero and negative infinity results
zero.vals <- which(meancounts[,1:2]==0, arr.ind=TRUE)

```

```

to.rm <- unique(zero.vals[,1])
mycounts <- meancounts[-to.rm,]
head(mycounts)

```

```

##                  control.mean treated.mean      log2fc
## ENSG00000000003      900.75     658.00 -0.45303916
## ENSG00000000419      520.50     546.00  0.06900279
## ENSG00000000457      339.75     316.50 -0.10226805
## ENSG00000000460      97.25      78.75 -0.30441833
## ENSG00000000971     5219.00    6687.50  0.35769358
## ENSG00000001036     2327.00    1785.75 -0.38194109

```

Q7. arr.ind will have which() return values that match the given value, in this case: 0. Which will be looking at both rows and columns. Since we don't want any values with 0 in the gene row, the unique function will prevent a certain entry from being counted twice if it also has a 0 value in it's column.

```

up.ind <- mycounts$log2fc > 2
down.ind <- mycounts$log2fc < (-2)

```

```
which(up.ind, arr.ind=TRUE)
```

```

## [1]   59   149   203   290   338   344   576   1048  1055   1196   1618   1680
## [13] 1781  1939  1981  2038  2050  2118  2355  2358  2443  2514  2528  2599
## [25] 2686  3186  3265  3459  3548  3788  3836  4052  4128  4129  4244  4453
## [37] 4567  4659  4899  5124  5180  5181  5290  5386  5423  5484  5496  5498
## [49] 5584  5725  5964  6055  6167  6246  6277  6288  6364  6453  6518  6553
## [61] 6636  6869  6959  7034  7107  7108  7115  7183  7306  7348  7369  7422
## [73] 7617  7790  7913  7970  7989  8330  8365  8396  8530  8706  8714  8723
## [85] 8729  8741  8934  9175  9176  9213  9215  9270  9297  9392  9608  9753
## [97] 9828  9924  9940  9968  9971  10079 10146 10174 10229 10234 10515 10577
## [109] 10631 10790 10955 10981 11019 11143 11177 11215 11341 11497 11520 11650
## [121] 11798 11899 12087 12153 12237 12254 12269 12365 12637 12742 12828 12894
## [133] 12922 12952 13071 13363 13426 13604 13813 13822 13832 13833 13851 13938
## [145] 14099 14140 14151 14419 14472 14493 14519 14700 14820 14908 15111 15511
## [157] 15526 15564 15567 15675 15766 15807 16310 16341 16361 16365 16415 16416
## [169] 16421 16494 16630 16631 16638 16725 16929 16996 17115 17132 17206 17211
## [181] 17285 17318 17332 17338 17357 17385 17485 17530 17550 17779 17918 17996
## [193] 18061 18120 18163 18186 18246 18456 18493 18576 18650 18744 18839 18914
## [205] 18960 19045 19052 19078 19129 19139 19150 19296 19297 19462 19530 19550
## [217] 19666 19803 19915 20073 20183 20214 20314 20374 20384 20432 20445 20465
## [229] 20553 20668 20766 20771 20774 20984 21093 21113 21119 21189 21190 21216
## [241] 21266 21283 21317 21321 21391 21403 21409 21454 21544 21603

```

```
which(down.ind, arr.ind=TRUE)
```

```

## [1]   341   365   418   443   451   623   624   662   684   825   925   1124
## [13] 1142  1234  1238  1315  1351  1518  1551  1576  1623  1887  1911  1928
## [25] 1965  1974  2131  2245  2342  2375  2670  3015  3020  3091  3279  3320
## [37] 3332  3334  3355  3419  3794  3899  4001  4108  4286  4288  4331  4348
## [49] 4504  4605  4906  4933  4984  5009  5015  5053  5067  5154  5282  5320
## [61] 5397  5501  5518  5525  5535  5550  5555  5608  5684  5689  5724  5749

```

```

## [73] 5756 5779 5811 5873 5890 5904 5950 6073 6116 6293 6310 6370
## [85] 6504 6533 6684 6766 7000 7150 7189 7332 7396 7438 7521 7609
## [97] 7627 7858 7869 7936 8003 8013 8049 8052 8081 8114 8210 8212
## [109] 8237 8407 8492 8501 8653 8667 8762 8875 9043 9189 9199 9274
## [121] 9457 9600 9628 9807 9843 9853 9859 9983 9998 10020 10040 10176
## [133] 10272 10361 10371 10487 10495 10526 10569 10586 10779 11138 11147 11159
## [145] 11204 11237 11243 11332 11345 11417 11424 11436 11642 11710 11769 11848
## [157] 12032 12240 12290 12395 12456 12466 12520 12549 12566 12618 12640 12671
## [169] 12678 12687 12741 12826 12915 12964 13027 13193 13197 13211 13230 13324
## [181] 13340 13348 13408 13410 13451 13464 13473 13484 13627 13677 13682 13812
## [193] 13991 14033 14207 14226 14239 14285 14336 14350 14380 14406 14544 14628
## [205] 14666 14720 14898 15147 15275 15282 15325 15388 15406 15420 15492 15655
## [217] 15677 15731 15759 15787 15789 15840 15981 16063 16144 16234 16266 16283
## [229] 16318 16343 16371 16395 16500 16502 16662 16709 16860 16876 16932 16949
## [241] 16976 17024 17043 17069 17121 17162 17168 17196 17259 17326 17327 17344
## [253] 17488 17543 17546 17587 17652 17667 17682 17709 17745 17808 17931 17932
## [265] 17944 17957 18018 18051 18059 18197 18210 18299 18482 18490 18519 18521
## [277] 18546 18578 18582 18618 18652 18763 18785 18812 18834 18841 18922 18968
## [289] 19064 19094 19114 19187 19228 19243 19246 19270 19302 19311 19338 19347
## [301] 19363 19443 19499 19551 19554 19613 19616 19654 19669 19801 19813 19814
## [313] 19830 19890 19959 19976 20018 20074 20078 20096 20114 20128 20138 20233
## [325] 20242 20254 20258 20270 20273 20277 20390 20404 20413 20434 20480 20483
## [337] 20543 20559 20561 20599 20624 20631 20698 20709 20809 20813 20821 20894
## [349] 20969 21009 21084 21118 21148 21154 21188 21249 21280 21335 21371 21416
## [361] 21621 21627 21669 21673 21713 21744 21764

```

Q8. How many up regulated genes? 250

Q9. How many downregulated genes? 367

Q10. Can we trust these results? No. We have not been able to test the results for statistic significance.

```

library(DESeq2)
citation("DESeq2")

```

```

## To cite package 'DESeq2' in publications use:
##
## Love, M.I., Huber, W., Anders, S. Moderated estimation of fold change
## and dispersion for RNA-seq data with DESeq2 Genome Biology 15(12):550
## (2014)
##
## A BibTeX entry for LaTeX users is
##
## @Article{,
##   title = {Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2},
##   author = {Michael I. Love and Wolfgang Huber and Simon Anders},
##   year = {2014},
##   journal = {Genome Biology},
##   doi = {10.1186/s13059-014-0550-8},
##   volume = {15},
##   issue = {12},
##   pages = {550},
## }

```



```

res2 <- as.data.frame(res)

View(res2)

summary(res)

## 
## out of 25258 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)      : 1563, 6.2%
## LFC < 0 (down)    : 1188, 4.7%
## outliers [1]       : 142, 0.56%
## low counts [2]     : 9971, 39%
## (mean count < 10)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

res05 <- results(dds, alpha=0.05)
summary(res05)

## 
## out of 25258 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)      : 1236, 4.9%
## LFC < 0 (down)    : 933, 3.7%
## outliers [1]       : 142, 0.56%
## low counts [2]     : 9033, 36%
## (mean count < 6)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

library("AnnotationDbi")
library("org.Hs.eg.db")

## 

columns(org.Hs.eg.db)

##   [1] "ACNUM"        "ALIAS"         "ENSEMBL"        "ENSEMLPROT"    "ENSEMLTRANS"
##   [6] "ENTREZID"     "ENZYME"        "EVIDENCE"       "EVIDENCEALL"   "GENENAME"
##  [11] "GENETYPE"     "GO"            "GOALL"          "IPI"           "MAP"
##  [16] "OMIM"          "ONTOLOGY"      "ONTOLOGYALL"   "PATH"          "PFAM"
##  [21] "PMID"          "PROSITE"        "REFSEQ"         "SYMBOL"        "UCSCKG"
##  [26] "UNIPROT"

# adding columns to results table with mapIds
res$symbol <- mapIds(org.Hs.eg.db,
                      keys=row.names(res), # Our genenames
                      keytype="ENSEMBL",   # The format of our genenames
                      column="SYMBOL",    # The new format we want to add
                      multiVals="first")

```

```

## 'select()' returned 1:many mapping between keys and columns

head(res)

## log2 fold change (MLE): dex treated vs control
## Wald test p-value: dex treated vs control
## DataFrame with 6 rows and 7 columns
##           baseMean log2FoldChange      lfcSE      stat     pvalue
##           <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG000000000003 747.194195     -0.3507030  0.168246 -2.084470 0.0371175
## ENSG000000000005  0.000000          NA        NA        NA        NA
## ENSG000000000419 520.134160      0.2061078  0.101059  2.039475 0.0414026
## ENSG000000000457 322.664844      0.0245269  0.145145  0.168982 0.8658106
## ENSG000000000460 87.682625     -0.1471420  0.257007 -0.572521 0.5669691
## ENSG000000000938 0.319167     -1.7322890  3.493601 -0.495846 0.6200029
##           padj      symbol
##           <numeric> <character>
## ENSG000000000003 0.163035    TSPAN6
## ENSG000000000005   NA        TNMD
## ENSG000000000419 0.176032    DPM1
## ENSG000000000457 0.961694    SCYL3
## ENSG000000000460 0.815849    FIRRM
## ENSG000000000938   NA        FGR

res$entrez <- mapIds(org.Hs.eg.db,
                      keys=row.names(res),
                      column="ENTREZID",
                      keytype="ENSEMBL",
                      multiVals="first")

## 'select()' returned 1:many mapping between keys and columns

res$uniprot <- mapIds(org.Hs.eg.db,
                      keys=row.names(res),
                      column="UNIPROT",
                      keytype="ENSEMBL",
                      multiVals="first")

## 'select()' returned 1:many mapping between keys and columns

res$genename <- mapIds(org.Hs.eg.db,
                      keys=row.names(res),
                      column="GENENAME",
                      keytype="ENSEMBL",
                      multiVals="first")

## 'select()' returned 1:many mapping between keys and columns

head(res)

```

```

## log2 fold change (MLE): dex treated vs control
## Wald test p-value: dex treated vs control
## DataFrame with 6 rows and 10 columns
##           baseMean log2FoldChange      lfcSE      stat     pvalue
##           <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG000000000003  747.194195    -0.3507030  0.168246 -2.084470 0.0371175
## ENSG000000000005   0.000000      NA        NA        NA        NA
## ENSG000000000419  520.134160    0.2061078  0.101059  2.039475 0.0414026
## ENSG000000000457  322.664844    0.0245269  0.145145  0.168982 0.8658106
## ENSG000000000460   87.682625    -0.1471420  0.257007 -0.572521 0.5669691
## ENSG000000000938   0.319167    -1.7322890  3.493601 -0.495846 0.6200029
##           padj      symbol      entrez      uniprot
##           <numeric> <character> <character> <character>
## ENSG000000000003   0.163035    TSPAN6       7105 AOA024RCIO
## ENSG000000000005      NA        TNMD       64102 Q9H2S6
## ENSG000000000419   0.176032    DPM1        8813 060762
## ENSG000000000457   0.961694    SCYL3       57147 Q8IZE3
## ENSG000000000460   0.815849    FIRRM       55732 AOA024R922
## ENSG000000000938      NA        FGR        2268 P09769
##           genename
##           <character>
## ENSG000000000003      tetraspanin 6
## ENSG000000000005      tenomodulin
## ENSG000000000419      dolichyl-phosphate m..
## ENSG000000000457      SCY1 like pseudokina..
## ENSG000000000460      FIGNL1 interacting r..
## ENSG000000000938      FGR proto-oncogene, ..

```

#arrange and view results by p-value  
ord <- order( res\$padj )  
#View(res[ord,])  
head(res[ord,])

```

## log2 fold change (MLE): dex treated vs control
## Wald test p-value: dex treated vs control
## DataFrame with 6 rows and 10 columns
##           baseMean log2FoldChange      lfcSE      stat     pvalue
##           <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000152583   954.771     4.36836  0.2371268  18.4220 8.74490e-76
## ENSG00000179094   743.253     2.86389  0.1755693  16.3120 8.10784e-60
## ENSG00000116584  2277.913    -1.03470  0.0650984 -15.8944 6.92855e-57
## ENSG00000189221   2383.754     3.34154  0.2124058  15.7319 9.14433e-56
## ENSG00000120129   3440.704     2.96521  0.2036951  14.5571 5.26424e-48
## ENSG00000148175  13493.920     1.42717  0.1003890  14.2164 7.25128e-46
##           padj      symbol      entrez      uniprot
##           <numeric> <character> <character> <character>
## ENSG00000152583  1.32441e-71 SPARCL1       8404 AOA024RDE1
## ENSG00000179094  6.13966e-56 PER1        5187 015534
## ENSG00000116584  3.49776e-53 ARHGEF2       9181 Q92974
## ENSG00000189221  3.46227e-52 MAOA        4128 P21397
## ENSG00000120129  1.59454e-44 DUSP1        1843 B4DU40
## ENSG00000148175  1.83034e-42 STOM        2040 F8VSL7
##           genename
##           <character>

```

```

## ENSG00000152583           SPARC like 1
## ENSG00000179094 period circadian reg..
## ENSG00000116584 Rho/Rac guanine nucl..
## ENSG00000189221 monoamine oxidase A
## ENSG00000120129 dual specificity pho..
## ENSG00000148175          stomatin

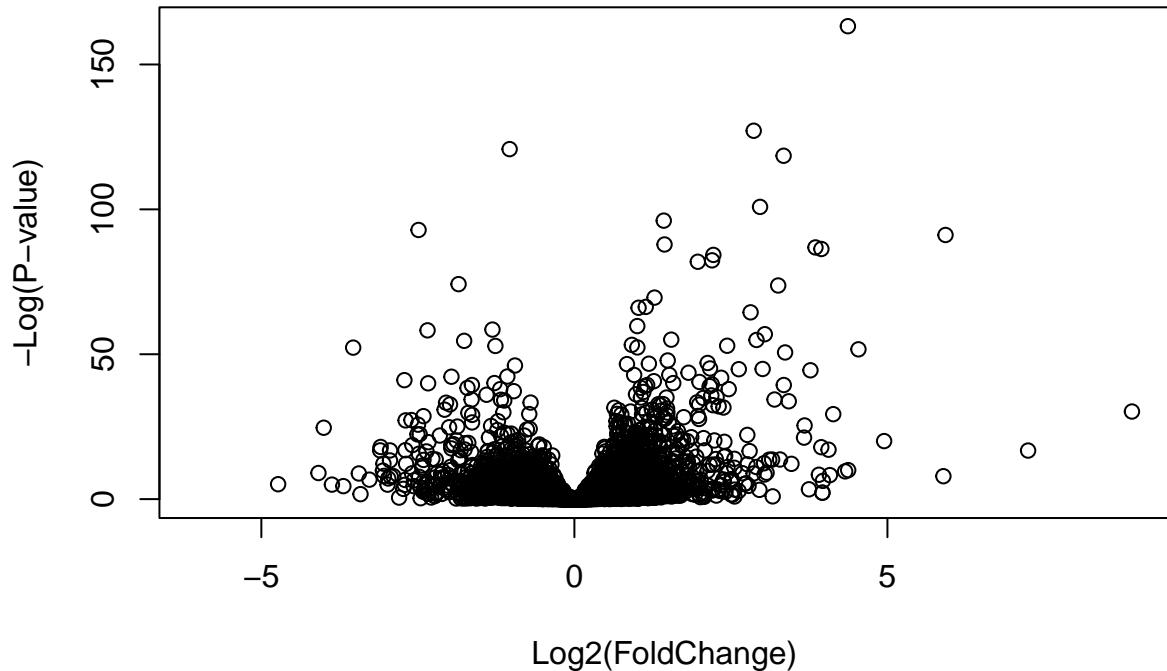
```

```
write.csv(res[ord,], "deseq_results.csv")
```

```

plot(res$log2FoldChange, -log(res$padj),
      xlab="Log2(FoldChange)",
      ylab="-Log(P-value)")

```

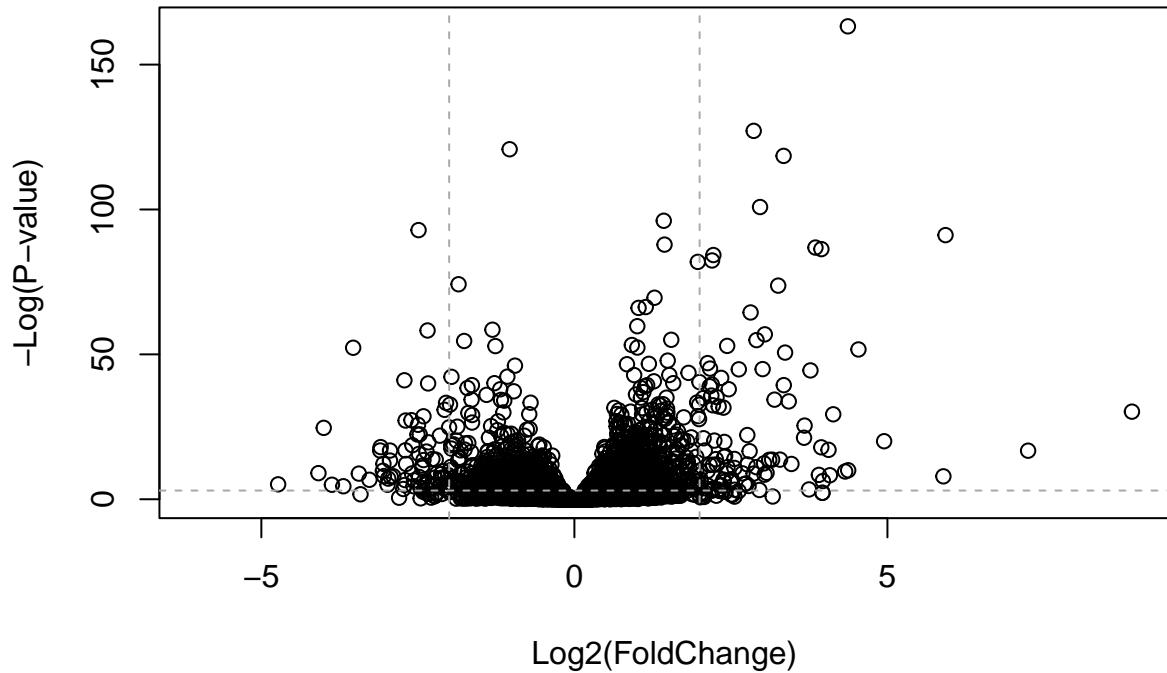


```

plot( res$log2FoldChange, -log(res$padj),
      ylab="-Log(P-value)", xlab="Log2(FoldChange)")

# Add some cut-off lines
abline(v=c(-2,2), col="darkgray", lty=2)
abline(h=-log(0.05), col="darkgray", lty=2)

```



```

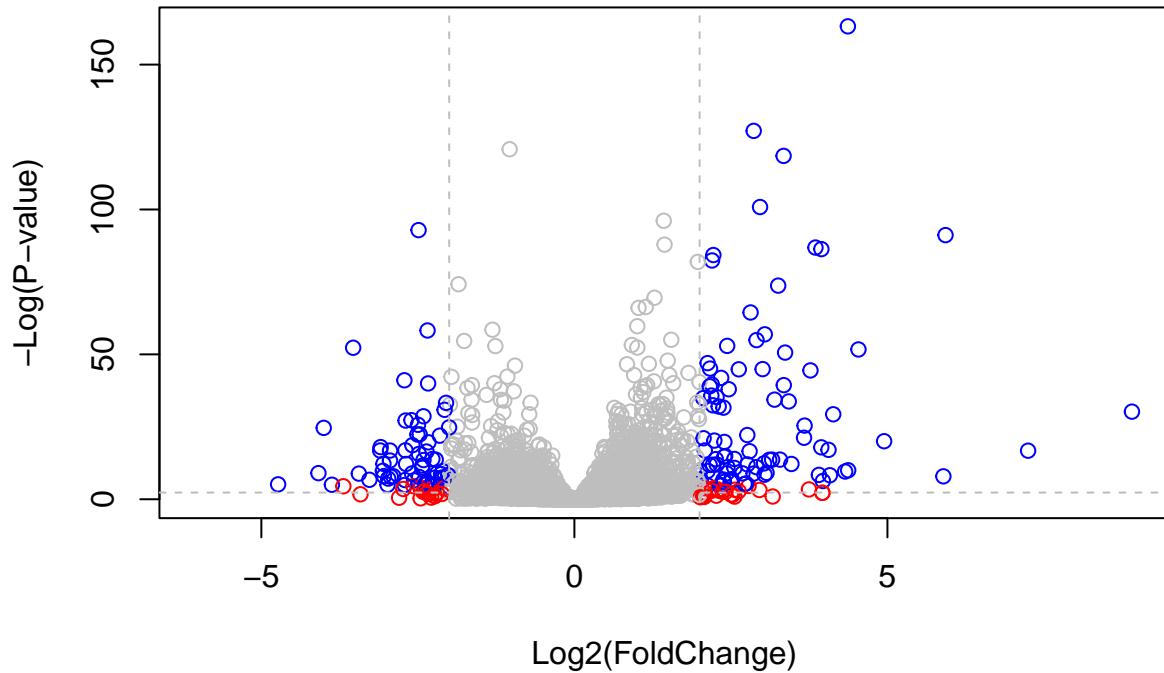
# Setup our custom point color vector
mycols <- rep("gray", nrow(res))
mycols[ abs(res$log2FoldChange) > 2 ] <- "red"

inds <- (res$padj < 0.01) & (abs(res$log2FoldChange) > 2 )
mycols[ inds ] <- "blue"

# Volcano plot with custom colors
plot( res$log2FoldChange, -log(res$padj),
      col=mycols, ylab="-Log(P-value)", xlab="Log2(FoldChange)" )

# Cut-off lines
abline(v=c(-2,2), col="gray", lty=2)
abline(h=-log(0.1), col="gray", lty=2)

```



```
library(EnhancedVolcano)

## Loading required package: ggplot2

## Loading required package: ggrepel

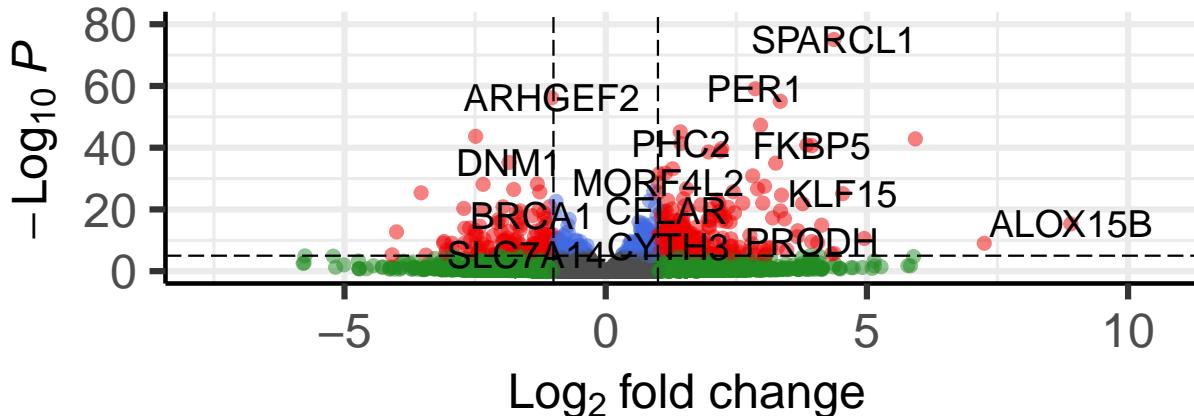
x <- as.data.frame(res)

EnhancedVolcano(x,
  lab = x$symbol,
  x = 'log2FoldChange',
  y = 'pvalue')
```

# Volcano plot

*EnhancedVolcano*

● NS ● Log<sub>2</sub> FC ● p-value ● p-value and log<sub>2</sub> FC



total = 38694 variables

```
library(pathview)
```

```
## #####  
## Pathview is an open source software package distributed under GNU General  
## Public License version 3 (GPLv3). Details of GPLv3 is available at  
## http://www.gnu.org/licenses/gpl-3.0.html. Particullary, users are required to  
## formally cite the original Pathview paper (not just mention it) in publications  
## or products. For details, do citation("pathview") within R.  
##  
## The pathview downloads and uses KEGG data. Non-academic uses may require a KEGG  
## license agreement (details at http://www.kegg.jp/kegg/legal.html).  
## #####
```

```
library(gage)
```

```
##
```

```
library(gageData)
```

```
data(kegg.sets.hs)
```

```
# Examine the first 2 pathways in this kegg set for humans  
head(kegg.sets.hs, 2)
```

```

## $`hsa00232 Caffeine metabolism`
## [1] "10"    "1544"   "1548"   "1549"   "1553"   "7498"   "9"
##
## $`hsa00983 Drug metabolism - other enzymes`
## [1] "10"    "1066"   "10720"  "10941"  "151531"  "1548"   "1549"   "1551"
## [9] "1553"  "1576"   "1577"   "1806"   "1807"   "1890"   "221223"  "2990"
## [17] "3251"  "3614"   "3615"   "3704"   "51733"   "54490"  "54575"   "54576"
## [25] "54577" "54578"  "54579"  "54600"  "54657"   "54658"  "54659"   "54963"
## [33] "574537" "64816"  "7083"   "7084"   "7172"   "7363"   "7364"   "7365"
## [41] "7366"   "7367"   "7371"   "7372"   "7378"   "7498"   "79799"  "83549"
## [49] "8824"   "8833"   "9"      "978"

foldchanges = res$log2FoldChange
names(foldchanges) = res$entrez
head(foldchanges)

##          7105       64102       8813       57147       55732       2268
## -0.35070302           NA  0.20610777  0.02452695 -0.14714205 -1.73228897

# Get the results
keggres = gage(foldchanges, gsets=kegg.sets.hs)

attributes(keggres)

## $names
## [1] "greater" "less"     "stats"

# Look at the first three down (less) pathways
head(keggres$less, 3)

##                                     p.geomean stat.mean      p.val
## hsa05332 Graft-versus-host disease 0.0004250461 -3.473346 0.0004250461
## hsa04940 Type I diabetes mellitus 0.0017820293 -3.002352 0.0017820293
## hsa05310 Asthma                  0.0020045888 -3.009050 0.0020045888
##                                     q.val set.size      exp1
## hsa05332 Graft-versus-host disease 0.09053483      40 0.0004250461
## hsa04940 Type I diabetes mellitus 0.14232581      42 0.0017820293
## hsa05310 Asthma                  0.14232581      29 0.0020045888

pathview(gene.data=foldchanges, pathway.id="hsa05310")

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/sutor/OneDrive/Desktop/bimm143/12bioconDESeq2

## Info: Writing image file hsa05310.pathview.png

```