13. Lab Class13 (DESeq lab)

June 8, 2024

1 Transcriptomics and the analysis of RNA-Seq data

1.1 Outline

In this class session we will:

- Open a new RStudio Project and Quarto document for today's class;
- Review how to install both Bioconductor and CRAN packages;
- Explore the Himes et al. gene expression data using base R, dplyr and ggplot2 package functions;
- Perform a detailed differential gene expression analysis with the DESeq2 package.
- Render a reproducible PDF report of your work with answers to all questions below.

For full details of the original analysis see the PubMed entry 24926665 and for associated data see the GEO entry GSE52778.

1.2 2. Bioconductor setup

[1]:
```
install.packages("BiocManager")
BiocManager::install()
# For this class we will need DESeq2:
BiocManager::install("DESeq2")

library(BiocManager)
library(DESeq2)
```

```
The downloaded binary packages are in
/var/folders/vw/6c5wjngs433234dthdjypz800000gn/T//Rtmpa66FsR/downloaded_packages
```

```
'getOption("repos")' replaces Bioconductor standard repositories, see
'help("repositories", package = "BiocManager")' for details.
Replacement repositories:
    CRAN: https://cran.r-project.org

Bioconductor version 3.17 (BiocManager 1.30.23), R 4.3.2 (2023-10-31)

Old packages: 'backports', 'BH', 'boot', 'broom', 'bslib', 'cachem',
  'checkmate', 'cli', 'cluster', 'codetools', 'commonmark', 'cowplot', 'cpp11',
  'curl', 'data.table', 'DBI', 'deldir', 'digest', 'dotCall64', 'dqrng',
```

```
  'emmeans', 'estimability', 'fansi', 'farver', 'fastcluster', 'fastmap',
  'FNN', 'foreign', 'fs', 'future', 'future.apply', 'ggplot2', 'ggrepel',
  'ggridges', 'ggsci', 'globals', 'glue', 'gplots', 'gtable', 'hardhat',
  'hdf5r', 'highr', 'Hmisc', 'htmltools', 'htmlwidgets', 'httpuv', 'igraph',
  'ISOcodes', 'jsonlite', 'KernSmooth', 'knitr', 'later', 'lattice', 'lda',
  'listenv', 'locfit', 'markdown', 'matrixStats', 'mgcv', 'minqa', 'munsell',
  'mvtnorm', 'nlme', 'openssl', 'parallelly', 'patchwork', 'pbdZMQ', 'plotly',
  'progress', 'promises', 'quanteda', 'quantreg', 'R.oo', 'Rcpp', 'RcppAnnoy',
  'RcppArmadillo', 'RcppEigen', 'RcppHNSW', 'RCurl', 'readr', 'repr',
  'reticulate', 'rlang', 'rmarkdown', 'rpart', 'RSQLite', 'rstudioapi',
  'Rtsne', 'sass', 'Seurat', 'SeuratObject', 'shape', 'shiny', 'sp', 'SparseM',
  'spatstat.data', 'spatstat.explore', 'spatstat.geom', 'spatstat.random',
  'stm', 'stringi', 'survival', 'tidyr', 'tidyselect', 'tinytex', 'uuid',
  'uwot', 'vctrs', 'viridis', 'vroom', 'WGCNA', 'withr', 'xfun', 'xml2', 'yaml'

'getOption("repos")' replaces Bioconductor standard repositories, see
'help("repositories", package = "BiocManager")' for details.
Replacement repositories:
    CRAN: https://cran.r-project.org

Bioconductor version 3.17 (BiocManager 1.30.23), R 4.3.2 (2023-10-31)

Warning message:
"package(s) not installed when version(s) same as or greater than current; use
  `force = TRUE` to re-install: 'DESeq2'"
Old packages: 'backports', 'BH', 'boot', 'broom', 'bslib', 'cachem',
  'checkmate', 'cli', 'cluster', 'codetools', 'commonmark', 'cowplot', 'cpp11',
  'curl', 'data.table', 'DBI', 'deldir', 'digest', 'dotCall64', 'dqrng',
  'emmeans', 'estimability', 'fansi', 'farver', 'fastcluster', 'fastmap',
  'FNN', 'foreign', 'fs', 'future', 'future.apply', 'ggplot2', 'ggrepel',
  'ggridges', 'ggsci', 'globals', 'glue', 'gplots', 'gtable', 'hardhat',
  'hdf5r', 'highr', 'Hmisc', 'htmltools', 'htmlwidgets', 'httpuv', 'igraph',
  'ISOcodes', 'jsonlite', 'KernSmooth', 'knitr', 'later', 'lattice', 'lda',
  'listenv', 'locfit', 'markdown', 'matrixStats', 'mgcv', 'minqa', 'munsell',
  'mvtnorm', 'nlme', 'openssl', 'parallelly', 'patchwork', 'pbdZMQ', 'plotly',
  'progress', 'promises', 'quanteda', 'quantreg', 'R.oo', 'Rcpp', 'RcppAnnoy',
  'RcppArmadillo', 'RcppEigen', 'RcppHNSW', 'RCurl', 'readr', 'repr',
  'reticulate', 'rlang', 'rmarkdown', 'rpart', 'RSQLite', 'rstudioapi',
  'Rtsne', 'sass', 'Seurat', 'SeuratObject', 'shape', 'shiny', 'sp', 'SparseM',
  'spatstat.data', 'spatstat.explore', 'spatstat.geom', 'spatstat.random',
  'stm', 'stringi', 'survival', 'tidyr', 'tidyselect', 'tinytex', 'uuid',
  'uwot', 'vctrs', 'viridis', 'vroom', 'WGCNA', 'withr', 'xfun', 'xml2', 'yaml'

Bioconductor version '3.17' is out-of-date; the current release version '3.19'
  is available with R version '4.4'; see https://bioconductor.org/install

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

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
```

1.3 3. Import countData and colData

airway_scaledcounts.csv

airway_metadata.csv

[2]:
```
# counts <- read.csv("airway_scaledcounts.csv", row.names=1)
# metadata <- read.csv("airway_metadata.csv")

#or

counts <- read.csv("https://bioboot.github.io/bimm143_W18/class-material/
 ↪airway_scaledcounts.csv", row.names=1)
metadata <- read.csv("https://bioboot.github.io/bimm143_W18/class-material/
 ↪airway_metadata.csv")
```

[3]:
```
head(counts)
```

A data.frame: 6 × 8

	SRR1039508 <dbl>	SRR1039509 <dbl>	SRR1039512 <dbl>	SRR1039513 <dbl>	SRR1039 <dbl>
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

[4]:
```
head(metadata)
```

A data.frame: 6 × 4

	id <chr>	dex <chr>	celltype <chr>	geo_id <chr>
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

1.4 Q1. How many genes are in this dataset?

- There are 38694 genes in this data set

1.5 Q2. How many 'control' cell lines do we have?

- There are 4 control cell lines in the data set

1.6 4. Toy differential gene expression

[5]:
```
control <- metadata[metadata[,"dex"]=="control",]
control.counts <- counts[ ,control$id]
control.mean <- rowSums( control.counts )/4
head(control.mean)
```

ENSG00000000003 900.75 **ENSG00000000005** 0 **ENSG00000000419** 520.5
ENSG00000000457 339.75 **ENSG00000000460** 97.25 **ENSG00000000938** 0.75

[6]:
```
library(dplyr)
control <- metadata %>% filter(dex=="control")
control.counts <- counts %>% select(control$id)
control.mean <- rowSums(control.counts)/4
head(control.mean)
```

```
Attaching package: ‘dplyr’


The following object is masked from ‘package:Biobase’:

    combine


The following object is masked from ‘package:matrixStats’:

    count


The following objects are masked from ‘package:GenomicRanges’:

    intersect, setdiff, union


The following object is masked from ‘package:GenomeInfoDb’:

    intersect


The following objects are masked from ‘package:IRanges’:

    collapse, desc, intersect, setdiff, slice, union


The following objects are masked from ‘package:S4Vectors’:

    first, intersect, rename, setdiff, setequal, union
```

```
The following objects are masked from 'package:BiocGenerics':

    combine, intersect, setdiff, union


The following objects are masked from 'package:stats':

    filter, lag


The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

ENSG00000000003	900.75	**ENSG00000000005**	0	**ENSG00000000419**	520.5
ENSG00000000457	339.75	**ENSG00000000460**	97.25	**ENSG00000000938**	0.75

1.7 Q3. How would you make the above code in either approach more robust? Is there a function that could help here?

[7]:
```
control <- metadata[metadata$dex == "control", ]
control.counts <- counts[ , control$id]
control.mean <- rowMeans(control.counts)
head(control.mean)
```

ENSG00000000003	900.75	**ENSG00000000005**	0	**ENSG00000000419**	520.5
ENSG00000000457	339.75	**ENSG00000000460**	97.25	**ENSG00000000938**	0.75

1.8 Follow the same procedure for the treated samples (i.e. calculate the mean per gene across drug treated samples and assign to a labeled vector called treated.mean)

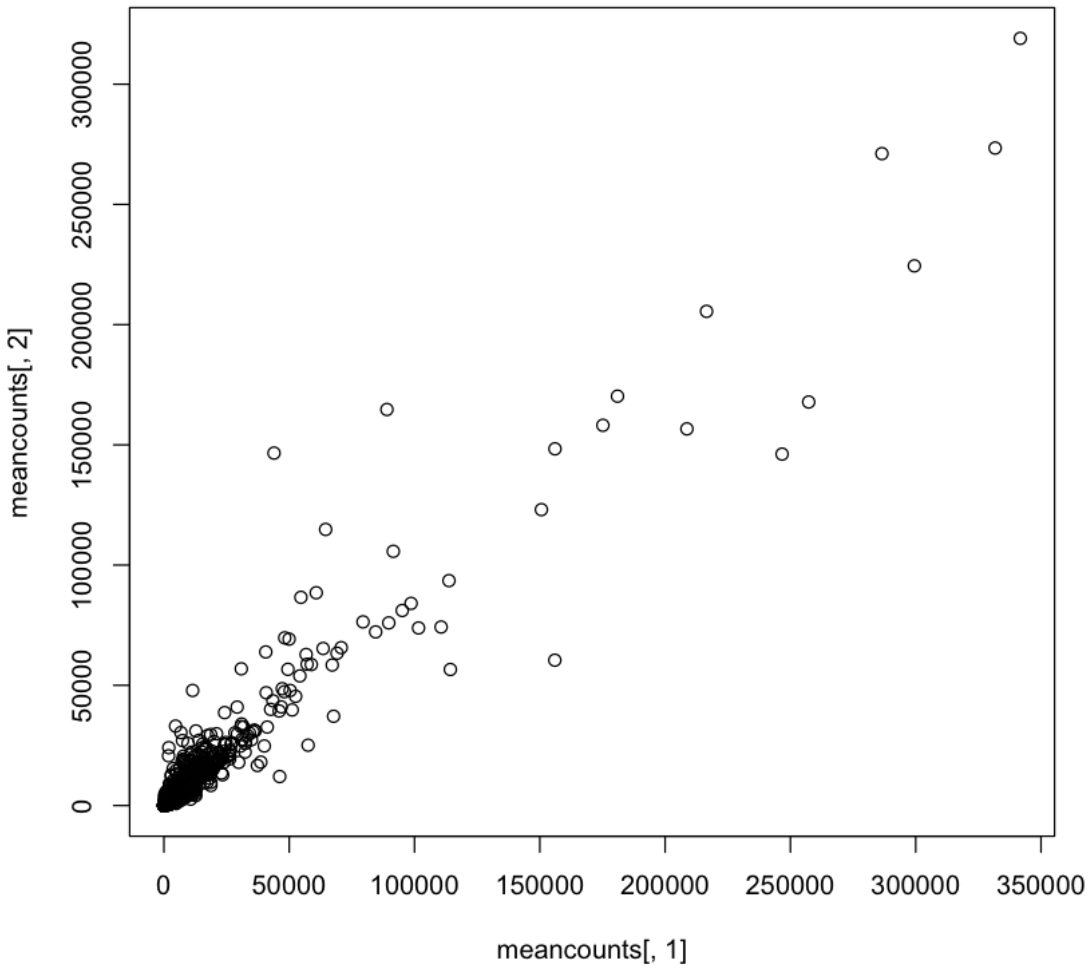
[8]:
```
treated <- metadata[metadata$dex == "treated", ]
treated.counts <- counts[, treated$id]
treated.mean <- rowMeans(treated.counts)
head(treated.mean)
```

ENSG00000000003	658	**ENSG00000000005**	0	**ENSG00000000419**	546
ENSG00000000457	316.5	**ENSG00000000460**	78.75	**ENSG00000000938**	0

[9]:
```
meancounts <- data.frame(control.mean, treated.mean)
```

1.9 Q5 (a). Create a scatter plot showing the mean of the treated samples against the mean of the control samples. Your plot should look something like the following.

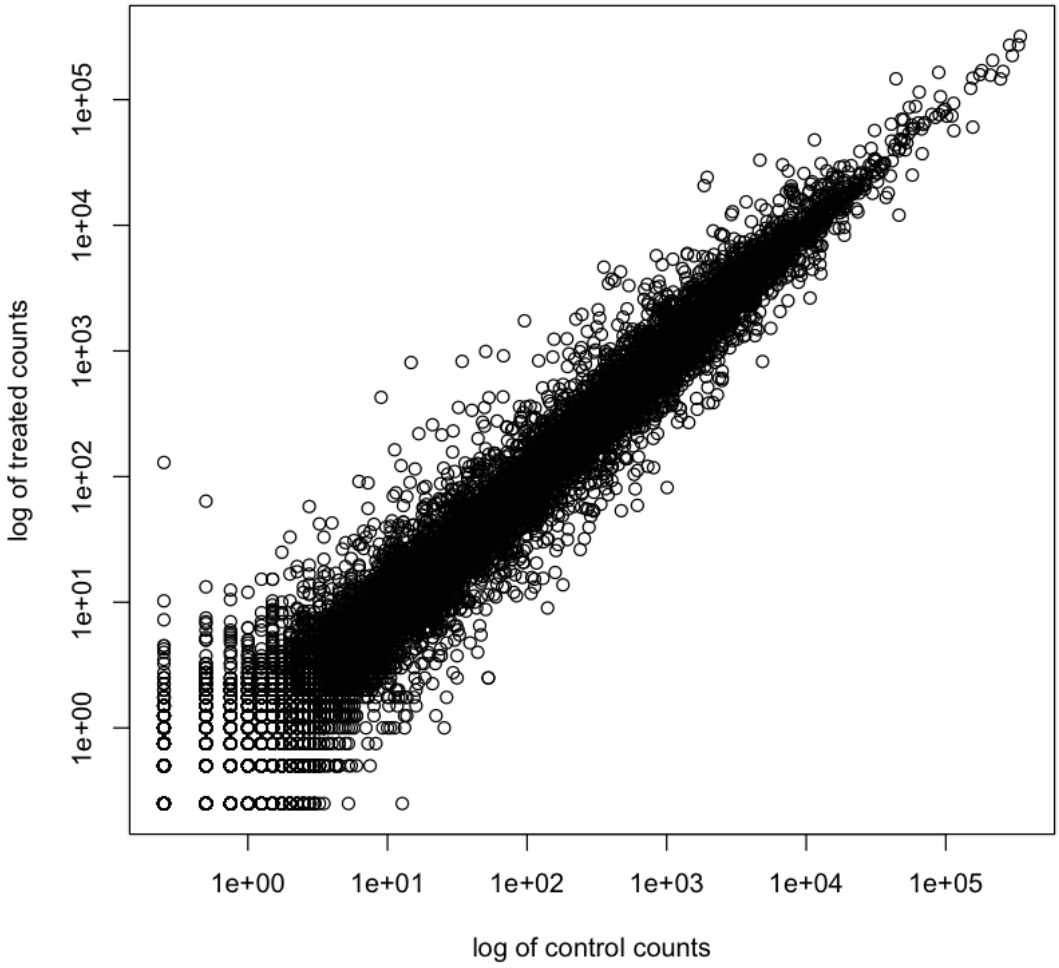
[10]:
```
# plot(meancounts[,1],meancounts[,2], xlab="Control", ylab="Treated")

#or

plot(meancounts[,1], meancounts[,2])
```



1.10 Q6. Try plotting both axes on a log scale. What is the argument to plot() that allows you to do this?

[11]:
```
plot(meancounts[,1], meancounts[,2], log="xy",
xlab= "log of control counts",
ylab= "log of treated counts")
```

```
Warning message in xy.coords(x, y, xlabel, ylabel, log):
"15032 x values <= 0 omitted from logarithmic plot"
Warning message in xy.coords(x, y, xlabel, ylabel, log):
"15281 y values <= 0 omitted from logarithmic plot"
```



[12]:
```
meancounts$log2fc <- log2(meancounts[,"treated.mean"]/meancounts[,"control.
  ↪mean"])
head(meancounts)
```

A data.frame: 6 × 3

	control.mean <dbl>	treated.mean <dbl>	log2fc <dbl>
ENSG00000000003	900.75	658.00	-0.45303916
ENSG00000000005	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

[13]:
```
zero.vals <- which(meancounts[,1:2]==0, arr.ind=TRUE)

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

A data.frame: 6 × 3

	control.mean <dbl>	treated.mean <dbl>	log2fc <dbl>
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

1.11 Q7. What is the purpose of the arr.ind argument in the which() function call above? Why would we then take the first column of the output and need to call the unique() function?

[14]:
```
zero.values <- (which(meancounts[,1:2]==0, arr.ind=TRUE))
to.rm <- unique(zero.values[,1])
mycounts <- meancounts[-to.rm,]
head(mycounts)
```

A data.frame: 6 × 3

	control.mean <dbl>	treated.mean <dbl>	log2fc <dbl>
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

[15]:
```
nrow(mycounts)
```

21817

[16]:
```
up.ind <- mycounts$log2fc > 2
down.ind <- mycounts$log2fc < (-2)
```

1.12 Q8. Using the up.ind vector above can you determine how many up regulated genes we have at the greater than 2 fc level?

[17]:
```
sum(up.ind)
```

250

1.13 Q9. Using the down.ind vector above can you determine how many down regulated genes we have at the greater than 2 fc level?

[18]:
```
sum(down.ind)
```

367

1.14 Q10. Do you trust these results? Why or why not?

- No, the next section will better encapsulate the results using statistics

2 5. Setting up for DESeq

[19]:
```
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},
  }
```

2.1 Importing data

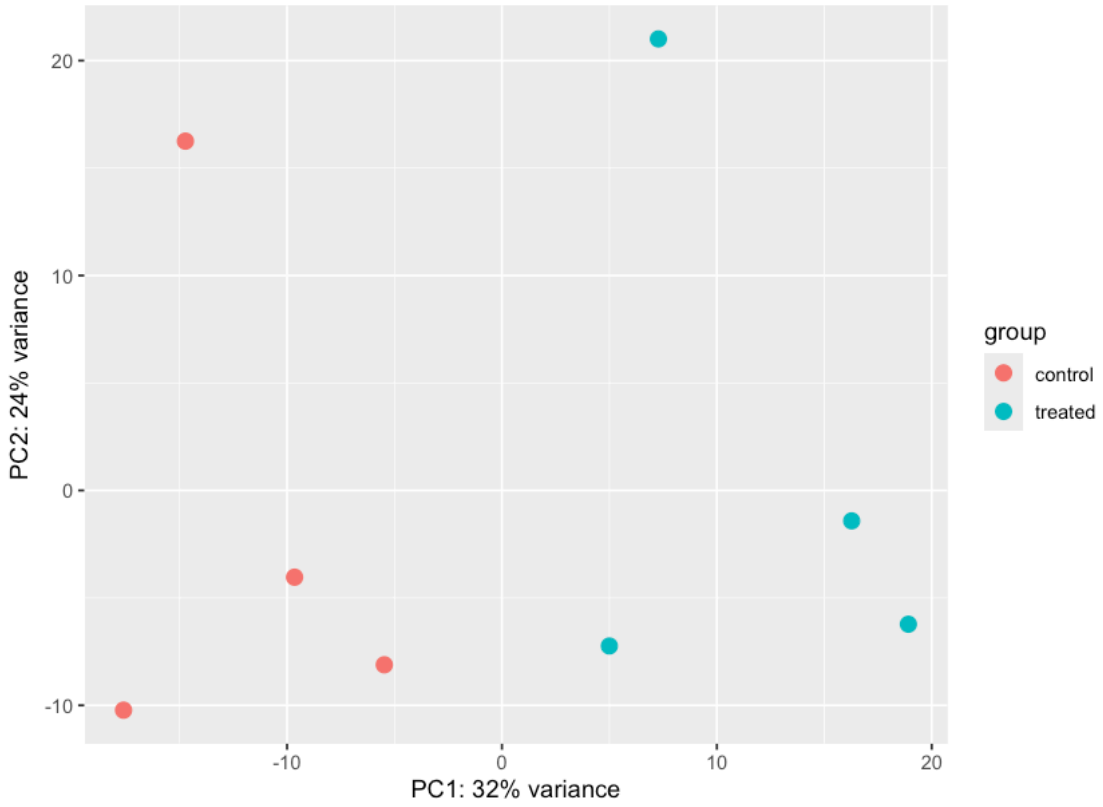
[20]:
```
dds <- DESeqDataSetFromMatrix(countData=counts,
                              colData=metadata,
                              design=~dex)
dds
```

```
converting counts to integer mode

Warning message in DESeqDataSet(se, design = design, ignoreRank):
"some variables in design formula are characters, converting to factors"
```

```
class: DESeqDataSet
dim: 38694 8
metadata(1): version
assays(1): counts
rownames(38694): ENSG00000000003 ENSG00000000005 … ENSG00000283120
  ENSG00000283123
rowData names(0):
colnames(8): SRR1039508 SRR1039509 … SRR1039520 SRR1039521
colData names(4): id dex celltype geo_id
```

2.2 6. Principal Component Analysis (PCA)

[21]:
```
vsd <- vst(dds, blind = FALSE)
plotPCA(vsd, intgroup = c("dex"))
```



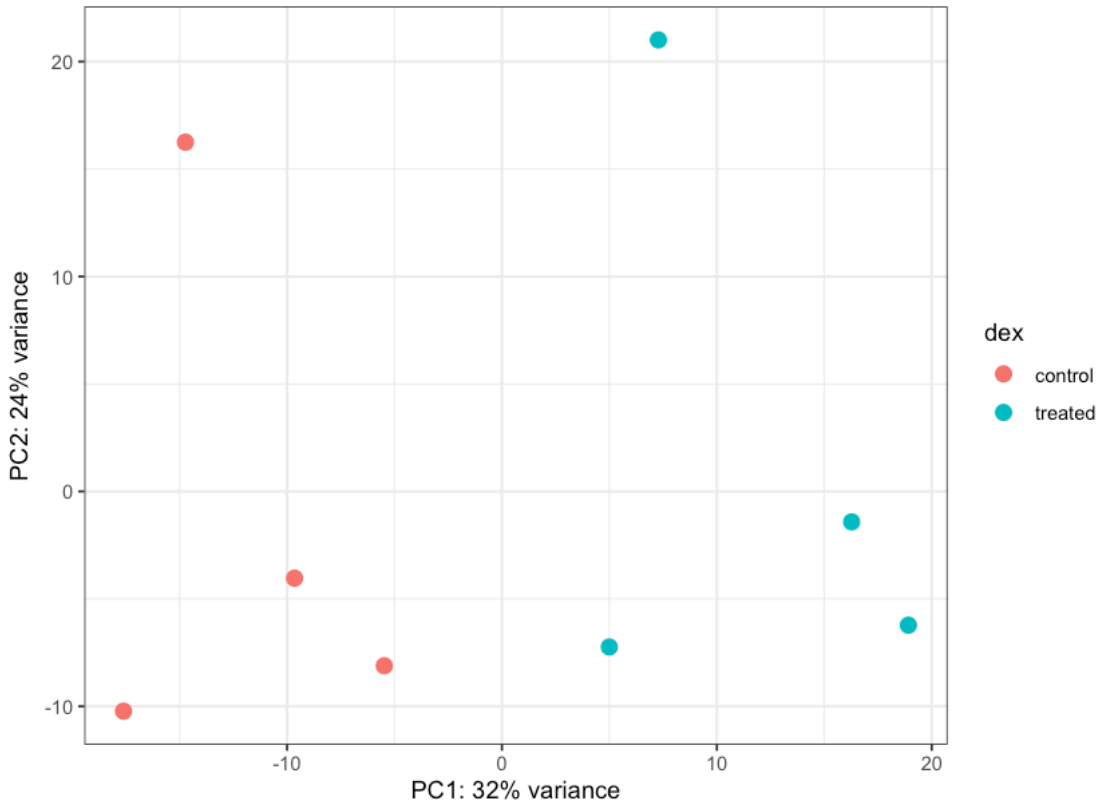
[22]:
```
pcaData <- plotPCA(vsd, intgroup=c("dex"), returnData=TRUE)
head(pcaData)
```

A data.frame: 6 × 5

	PC1 <dbl>	PC2 <dbl>	group <fct>	dex <fct>	name <chr>
SRR1039508	-17.607922	-10.225252	control	control	SRR1039508
SRR1039509	4.996738	-7.238117	treated	treated	SRR1039509
SRR1039512	-5.474456	-8.113993	control	control	SRR1039512
SRR1039513	18.912974	-6.226041	treated	treated	SRR1039513
SRR1039516	-14.729173	16.252000	control	control	SRR1039516
SRR1039517	7.279863	21.008034	treated	treated	SRR1039517

[23]:
```
# Calculate percent variance per PC for the plot axis labels
percentVar <- round(100 * attr(pcaData, "percentVar"))
```

[24]:
```
library(ggplot2)

ggplot(pcaData) +
  aes(x = PC1, y = PC2, color = dex) +
  geom_point(size =3) +
  xlab(paste0("PC1: ", percentVar[1], "% variance")) +
  ylab(paste0("PC2: ", percentVar[2], "% variance")) +
  coord_fixed() +
  theme_bw()
```



2.3 7. DESeq analysis

[25]:
```
dds <- DESeq(dds)
```

```
estimating size factors

estimating dispersions

gene-wise dispersion estimates

mean-dispersion relationship

final dispersion estimates

fitting model and testing
```

2.4 Getting results

[26]:
```
res <- results(dds)
res
```

```
log2 fold change (MLE): dex treated vs control
Wald test p-value: dex treated vs control
DataFrame with 38694 rows and 6 columns
                 baseMean log2FoldChange     lfcSE      stat    pvalue
                <numeric>      <numeric> <numeric> <numeric> <numeric>
ENSG00000000003  747.1942     -0.3507030  0.168246 -2.084470 0.0371175
ENSG00000000005    0.0000             NA        NA        NA        NA
ENSG00000000419  520.1342      0.2061078  0.101059  2.039475 0.0414026
ENSG00000000457  322.6648      0.0245269  0.145145  0.168982 0.8658106
ENSG00000000460   87.6826     -0.1471420  0.257007 -0.572521 0.5669691
…                   …             …       …        …        …
ENSG00000283115  0.000000             NA        NA        NA        NA
ENSG00000283116  0.000000             NA        NA        NA        NA
ENSG00000283119  0.000000             NA        NA        NA        NA
ENSG00000283120  0.974916      -0.668258   1.69456 -0.394354  0.693319
ENSG00000283123  0.000000             NA        NA        NA        NA
                     padj
                <numeric>
ENSG00000000003  0.163035
ENSG00000000005        NA
ENSG00000000419  0.176032
ENSG00000000457  0.961694
ENSG00000000460  0.815849
…                   …
ENSG00000283115        NA
ENSG00000283116        NA
```

```
ENSG00000283119        NA
ENSG00000283120        NA
ENSG00000283123        NA
```

[27]:
```
summary(res, alpha=0.05)

#or

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

```
out of 25258 with nonzero total read count
adjusted p-value < 0.05
LFC > 0 (up)       : 1242, 4.9%
LFC < 0 (down)     : 939, 3.7%
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
```

2.5 8. Adding annotation data

[28]:
```
library("AnnotationDbi")
library("org.Hs.eg.db")

columns(org.Hs.eg.db)
```

```
Attaching package: ‘AnnotationDbi’


The following object is masked from ‘package:dplyr’:

    select
```

1. ’ACCNUM’ 2. ’ALIAS’ 3. ’ENSEMBL’ 4. ’ENSEMBLPROT’ 5. ’ENSEMBLTRANS’ 6. ’ENTREZID’ 7. ’ENZYME’ 8. ’EVIDENCE’ 9. ’EVIDENCEALL’ 10. ’GENENAME’ 11. ’GENETYPE’ 12. ’GO’ 13. ’GOALL’ 14. ’IPI’ 15. ’MAP’ 16. ’OMIM’ 17. ’ONTOLOGY’ 18. ’ONTOLOGYALL’ 19. ’PATH’ 20. ’PFAM’ 21. ’PMID’ 22. ’PROSITE’ 23. ’REFSEQ’ 24. ’SYMBOL’ 25. ’UCSCKG’ 26. ’UNIPROT’

[29]:
```
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
```

[30]:
```
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>
ENSG00000000003 747.194195     -0.3507030  0.168246 -2.084470 0.0371175
ENSG00000000005   0.000000             NA        NA        NA        NA
ENSG00000000419 520.134160      0.2061078  0.101059  2.039475 0.0414026
ENSG00000000457 322.664844      0.0245269  0.145145  0.168982 0.8658106
ENSG00000000460  87.682625     -0.1471420  0.257007 -0.572521 0.5669691
ENSG00000000938   0.319167     -1.7322890  3.493601 -0.495846 0.6200029
                     padj      symbol
                <numeric> <character>
ENSG00000000003  0.163035      TSPAN6
ENSG00000000005        NA        TNMD
ENSG00000000419  0.176032        DPM1
ENSG00000000457  0.961694       SCYL3
ENSG00000000460  0.815849       FIRRM
ENSG00000000938        NA         FGR
```

2.6 Q11. Run the mapIds() function two more times to add the Entrez ID and UniProt accession and GENENAME as new columns called res$entrez, res$uniprot and res$genename.

[31]:
```
res$entrez <- mapIds(org.Hs.eg.db,
                     keys=row.names(res),
                     column="ENTREZID",
                     keytype="ENSEMBL",
                     multiVals="first")

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

res$genename <- mapIds(org.Hs.eg.db,
```

```
                    keys=row.names(res),
                    column="GENENAME",
                    keytype="ENSEMBL",
                    multiVals="first")

head(res)
```

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

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

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


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>
ENSG00000000003 747.194195     -0.3507030  0.168246 -2.084470 0.0371175
ENSG00000000005   0.000000             NA        NA        NA        NA
ENSG00000000419 520.134160      0.2061078  0.101059  2.039475 0.0414026
ENSG00000000457 322.664844      0.0245269  0.145145  0.168982 0.8658106
ENSG00000000460  87.682625     -0.1471420  0.257007 -0.572521 0.5669691
ENSG00000000938   0.319167     -1.7322890  3.493601 -0.495846 0.6200029
                     padj      symbol      entrez     uniprot
                <numeric> <character> <character> <character>
ENSG00000000003  0.163035      TSPAN6        7105  A0A024RCI0
ENSG00000000005        NA        TNMD       64102      Q9H2S6
ENSG00000000419  0.176032        DPM1        8813      O60762
ENSG00000000457  0.961694       SCYL3       57147      Q8IZE3
ENSG00000000460  0.815849       FIRRM       55732  A0A024R922
ENSG00000000938        NA         FGR        2268      P09769
                          genename
                       <character>
ENSG00000000003          tetraspanin 6
ENSG00000000005            tenomodulin
ENSG00000000419 dolichyl-phosphate m..
ENSG00000000457 SCY1 like pseudokina..
ENSG00000000460 FIGNL1 interacting r..
ENSG00000000938 FGR proto-oncogene, ..
```

[32]:
```
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
```

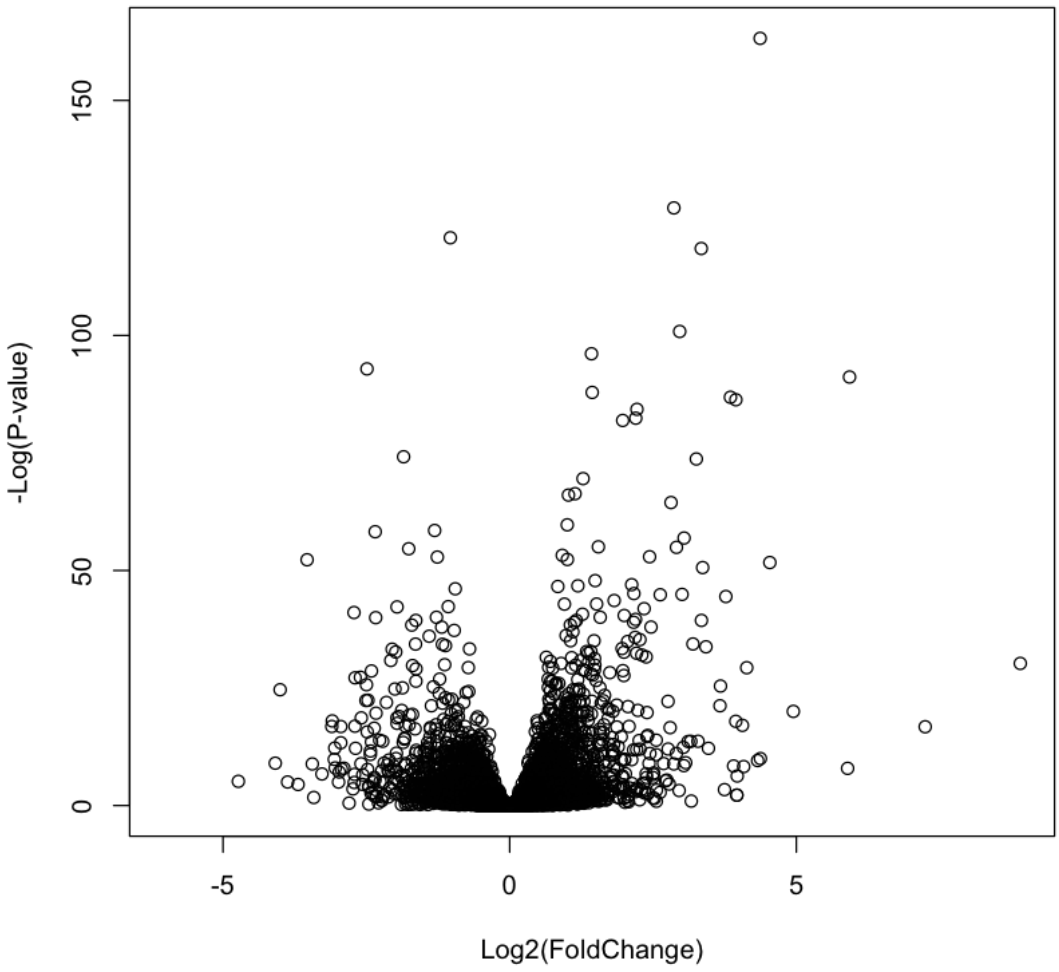
```
                 baseMean log2FoldChange     lfcSE      stat      pvalue
                <numeric>      <numeric> <numeric> <numeric>   <numeric>
ENSG00000152583   954.771        4.36836 0.2371268   18.4220 8.74490e-76
ENSG00000179094   743.253        2.86389 0.1755693   16.3120 8.10784e-60
ENSG00000116584  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  A0A024RDE1
ENSG00000179094 6.13966e-56        PER1        5187      O15534
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
```

[33]:
```
write.csv(res[ord,], "deseq_results.csv")
```

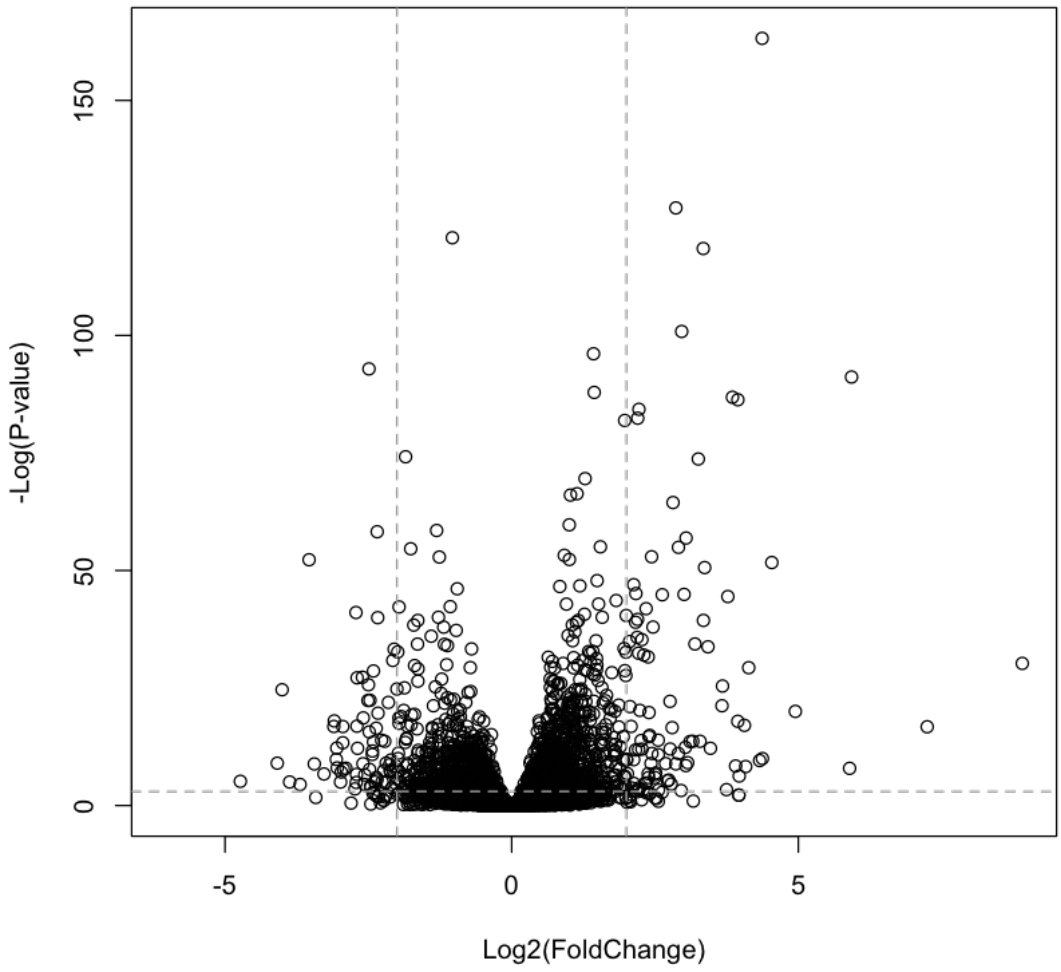
2.7 9. Data Visualization

2.8 Volcano plots

[34]:
```
#plot(res$log2FoldChange,  -log(res$padj)

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

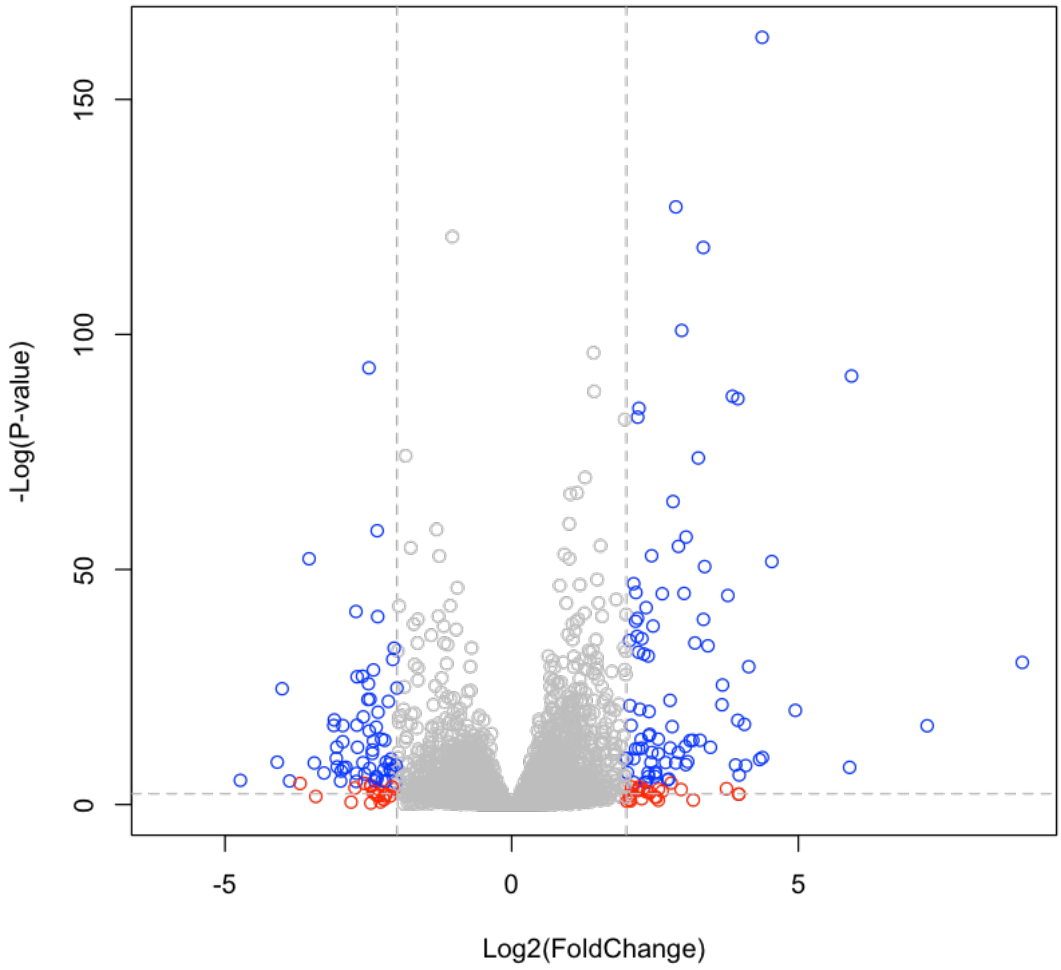


[35]:
```
plot( res$log2FoldChange,  -log(res$padj),
 ylab="-Log(P-value)", xlab="Log2(FoldChange)")

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



[36]:
```
# 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)
```

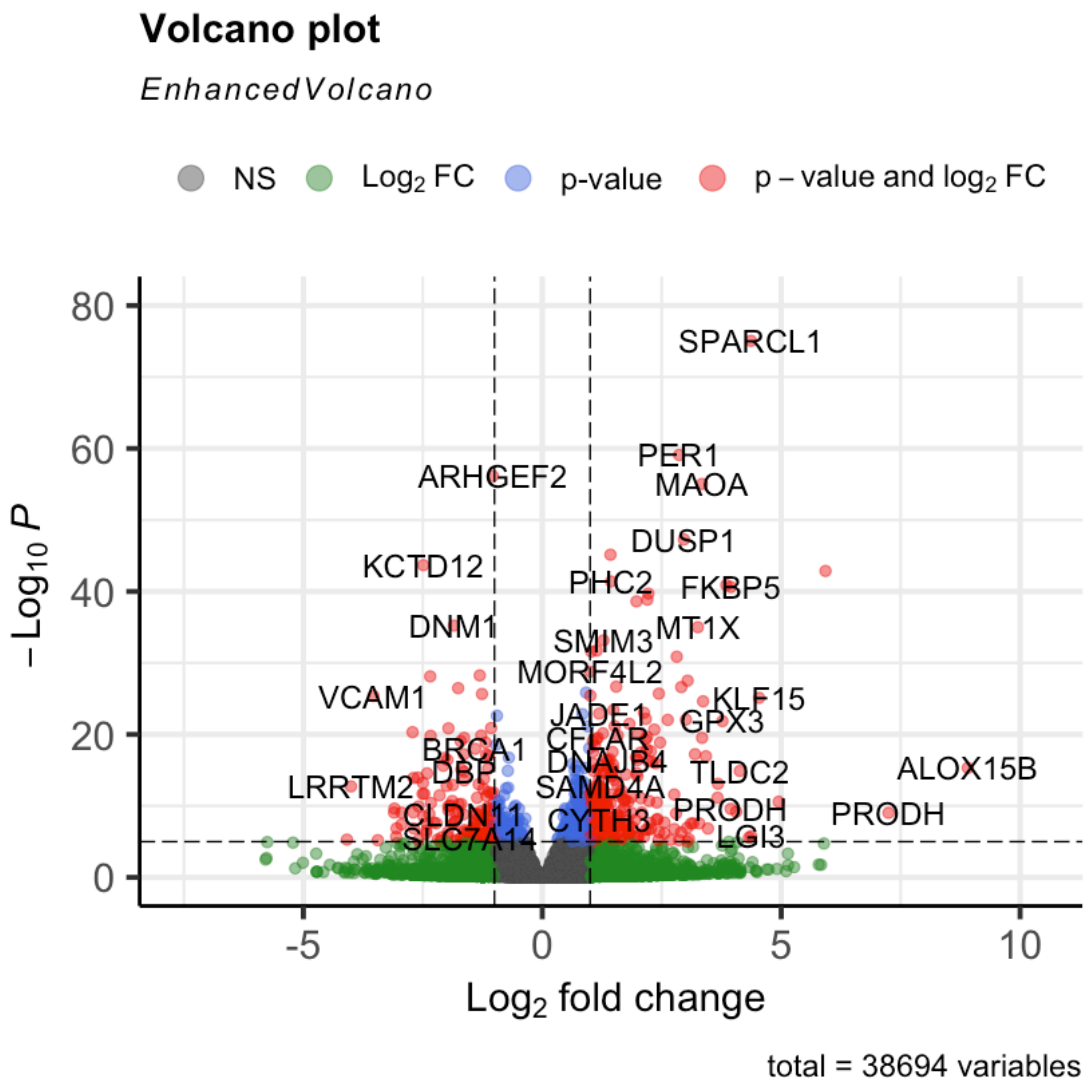
```
abline(h=-log(0.1), col="gray", lty=2)
```



[37]:
```
BiocManager::install("EnhancedVolcano")
library(EnhancedVolcano)

x <- as.data.frame(res)

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

```
'getOption("repos")' replaces Bioconductor standard repositories, see
```

```
'help("repositories", package = "BiocManager")' for details.
Replacement repositories:
    CRAN: https://cran.r-project.org

Bioconductor version 3.17 (BiocManager 1.30.23), R 4.3.2 (2023-10-31)

Warning message:
"package(s) not installed when version(s) same as or greater than current; use
  `force = TRUE` to re-install: 'EnhancedVolcano'"
Old packages: 'backports', 'BH', 'boot', 'broom', 'bslib', 'cachem',
  'checkmate', 'cli', 'cluster', 'codetools', 'commonmark', 'cowplot', 'cpp11',
  'curl', 'data.table', 'DBI', 'deldir', 'digest', 'dotCall64', 'dqrng',
  'emmeans', 'estimability', 'fansi', 'farver', 'fastcluster', 'fastmap',
  'FNN', 'foreign', 'fs', 'future', 'future.apply', 'ggplot2', 'ggrepel',
  'ggridges', 'ggsci', 'globals', 'glue', 'gplots', 'gtable', 'hardhat',
  'hdf5r', 'highr', 'Hmisc', 'htmltools', 'htmlwidgets', 'httpuv', 'igraph',
  'ISOcodes', 'jsonlite', 'KernSmooth', 'knitr', 'later', 'lattice', 'lda',
  'listenv', 'locfit', 'markdown', 'matrixStats', 'mgcv', 'minqa', 'munsell',
  'mvtnorm', 'nlme', 'openssl', 'parallelly', 'patchwork', 'pbdZMQ', 'plotly',
  'progress', 'promises', 'quanteda', 'quantreg', 'R.oo', 'Rcpp', 'RcppAnnoy',
  'RcppArmadillo', 'RcppEigen', 'RcppHNSW', 'RCurl', 'readr', 'repr',
  'reticulate', 'rlang', 'rmarkdown', 'rpart', 'RSQLite', 'rstudioapi',
  'Rtsne', 'sass', 'Seurat', 'SeuratObject', 'shape', 'shiny', 'sp', 'SparseM',
  'spatstat.data', 'spatstat.explore', 'spatstat.geom', 'spatstat.random',
  'stm', 'stringi', 'survival', 'tidyr', 'tidyselect', 'tinytex', 'uuid',
  'uwot', 'vctrs', 'viridis', 'vroom', 'WGCNA', 'withr', 'xfun', 'xml2', 'yaml'

Loading required package: ggrepel
```



2.9 10. Pathway analysis

2.10 Patway analysis with R and Bioconductor

[38]:
```
# Run in your R console (i.e. not your Rmarkdown doc!)
BiocManager::install( c("pathview", "gage", "gageData") )
```

```
'getOption("repos")' replaces Bioconductor standard repositories, see
'help("repositories", package = "BiocManager")' for details.
Replacement repositories:
    CRAN: https://cran.r-project.org

Bioconductor version 3.17 (BiocManager 1.30.23), R 4.3.2 (2023-10-31)
```

```
Warning message:
"package(s) not installed when version(s) same as or greater than current; use
  `force = TRUE` to re-install: 'pathview' 'gage' 'gageData'"
Old packages: 'backports', 'BH', 'boot', 'broom', 'bslib', 'cachem',
  'checkmate', 'cli', 'cluster', 'codetools', 'commonmark', 'cowplot', 'cpp11',
  'curl', 'data.table', 'DBI', 'deldir', 'digest', 'dotCall64', 'dqrng',
  'emmeans', 'estimability', 'fansi', 'farver', 'fastcluster', 'fastmap',
  'FNN', 'foreign', 'fs', 'future', 'future.apply', 'ggplot2', 'ggrepel',
  'ggridges', 'ggsci', 'globals', 'glue', 'gplots', 'gtable', 'hardhat',
  'hdf5r', 'highr', 'Hmisc', 'htmltools', 'htmlwidgets', 'httpuv', 'igraph',
  'ISOcodes', 'jsonlite', 'KernSmooth', 'knitr', 'later', 'lattice', 'lda',
  'listenv', 'locfit', 'markdown', 'matrixStats', 'mgcv', 'minqa', 'munsell',
  'mvtnorm', 'nlme', 'openssl', 'parallelly', 'patchwork', 'pbdZMQ', 'plotly',
  'progress', 'promises', 'quanteda', 'quantreg', 'R.oo', 'Rcpp', 'RcppAnnoy',
  'RcppArmadillo', 'RcppEigen', 'RcppHNSW', 'RCurl', 'readr', 'repr',
  'reticulate', 'rlang', 'rmarkdown', 'rpart', 'RSQLite', 'rstudioapi',
  'Rtsne', 'sass', 'Seurat', 'SeuratObject', 'shape', 'shiny', 'sp', 'SparseM',
  'spatstat.data', 'spatstat.explore', 'spatstat.geom', 'spatstat.random',
  'stm', 'stringi', 'survival', 'tidyr', 'tidyselect', 'tinytex', 'uuid',
  'uwot', 'vctrs', 'viridis', 'vroom', 'WGCNA', 'withr', 'xfun', 'xml2', 'yaml'
```

[39]:
```
library(pathview)
library(gage)
library(gageData)

data(kegg.sets.hs)

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

```
#############################################################################
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).
#############################################################################
```

$‘hsa00232 Caffeine metabolism‘ 1. ’10’ 2. ’1544’ 3. ’1548’ 4. ’1549’ 5. ’1553’ 6. ’7498’ 7. ’9’

$‘hsa00983 Drug metabolism - other enzymes‘ 1. ’10’ 2. ’1066’ 3. ’10720’ 4. ’10941’ 5. ’151531’ 6. ’1548’ 7. ’1549’ 8. ’1551’ 9. ’1553’ 10. ’1576’ 11. ’1577’ 12. ’1806’ 13. ’1807’

14. '1890' 15. '221223' 16. '2990' 17. '3251' 18. '3614' 19. '3615' 20. '3704' 21. '51733' 22. '54490' 23. '54575' 24. '54576' 25. '54577' 26. '54578' 27. '54579' 28. '54600' 29. '54657' 30. '54658' 31. '54659' 32. '54963' 33. '574537' 34. '64816' 35. '7083' 36. '7084' 37. '7172' 38. '7363' 39. '7364' 40. '7365' 41. '7366' 42. '7367' 43. '7371' 44. '7372' 45. '7378' 46. '7498' 47. '79799' 48. '83549' 49. '8824' 50. '8833' 51. '9' 52. '978'

[40]:
```
foldchanges = res$log2FoldChange
names(foldchanges) = res$entrez
head(foldchanges)
```

7105 -0.350703020686574 **64102** <NA> **8813** 0.206107766417853 **57147** 0.0245269479387485 **55732** -0.147142049222146 **2268** -1.73228897394308

[41]:
```
# Get the results
keggres = gage(foldchanges, gsets=kegg.sets.hs)
attributes(keggres)
```

$names = 1. 'greater' 2. 'less' 3. 'stats'

[42]:
```
head(keggres$less, 3)
```

A matrix: 3 × 6 of type dbl

	p.geomean	stat.mean	p.val	q.va
hsa05332 Graft-versus-host disease	0.0004250461	-3.473346	0.0004250461	0.09
hsa04940 Type I diabetes mellitus	0.0017820293	-3.002352	0.0017820293	0.14
hsa05310 Asthma	0.0020045888	-3.009050	0.0020045888	0.14

[43]:
```
write.csv(res, file="DESeq2_results.csv")
```