Differential expression using DESeq2

The first thing we did was to load the packages needed for using DESeq2.

In [10]:
```
library("DESeq2")
library(scales)
```

```
Loading required package: S4Vectors
Loading required package: stats4
Loading required package: BiocGenerics
Loading required package: parallel

Attaching package: 'BiocGenerics'

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

    clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
    clusterExport, clusterMap, parApply, parCapply, parLapply,
    parLapplyLB, parRapply, parSapply, parSapplyLB

The following object is masked from 'package:stats':

    xtabs

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

    anyDuplicated, append, as.data.frame, as.vector, cbind, colnames,
    do.call, duplicated, eval, evalq, Filter, Find, get, intersect,
    is.unsorted, lapply, Map, mapply, match, mget, order, paste, pmax,
    pmax.int, pmin, pmin.int, Position, rank, rbind, Reduce, rep.int,
    rownames, sapply, setdiff, sort, table, tapply, union, unique,
    unlist, unsplit

Creating a generic function for 'nchar' from package 'base' in package 'S4Vector
s'
Loading required package: IRanges
Loading required package: GenomicRanges
Loading required package: GenomeInfoDb
Loading required package: Rcpp
Loading required package: RcppArmadillo
Warning message:
: replacing previous import 'S4Vectors::Position' by 'ggplot2::Position' when lo
ading 'DESeq2'
```

We first generate a path to the file that contains the counts.

In [43]:
```
```

In [44]:
```
countData <- as.matrix(read.csv(matrixFile, sep="\t", row.names="Geneid"))
```

	SRR2479801.bam	SRR2479802.bam	SRR2479803.bam	SRR2479804.bam	SRR2479
Xkr4	47	68	76	118	66
Gm39585	49	4	4	6	36
LOC108167595	0	0	0	0	0
Gm38717	0	8	1	0	3
Rp1	4	4	3	9	0
Sox17	37	95	71	88	95

We then changed the name of the samples to the corresponding condition. The first four samples are the WT samples while the last four are the SOX9KO samples

In [45]:
```
colnames(countData) <- c("WT1", "WT2", "WT3", "WT4", "X91", "X92", "X93", "X94")
```

	WT1	WT2	WT3	WT4	X91	X92	X93	X94
Xkr4	47	68	76	118	66	24	36	32
Gm39585	49	4	4	6	36	0	2	0
LOC108167595	0	0	0	0	0	0	0	0
Gm38717	0	8	1	0	3	0	0	1
Rp1	4	4	3	9	0	2	8	2
Sox17	37	95	71	88	95	78	94	103

Next we defined the experimental conditions. We also checked the order of the levels. In this case WT comes first meaning we will be looking at the ratio of X9/WT. Meaning that negative fold changes will be downregulated genes and vice versa in the sox9 ko.

In [46]:
```
condition <- factor(c("WT", "WT", "WT", "WT", "X9", "X9", "X9", "X9"))
```

```
[1] WT WT WT WT X9 X9 X9 X9
Levels: WT X9
```

The next step is to make a frame in which the names and conditions are placed.

In [47]:
```
colData <- data.frame(sampleName = colnames(countData),
                      condition = condition)
```

sampleName	condition
WT1	WT
WT2	WT
WT3	WT
WT4	WT
X91	X9
X92	X9
X93	X9
X94	X9

Then we create a DESeq `DESeqDataSet` object for analysis.

In [48]:
```
dds <- DESeqDataSetFromMatrix(countData = countData,
                              colData = colData,
                              design = ~ condition)
```

In [49]:

```
class: DESeqDataSet
dim: 41575 8
exptData(0):
assays(1): counts
rownames(41575): Xkr4 Gm39585 ... TrnT TrnP
rowRanges metadata column names(0):
colnames(8): WT1 WT2 ... X93 X94
colData names(2): sampleName condition
```

With the following command, we make sure that the order is still correct.

In [50]:

```
WT  WT  WT  WT  X9  X9  X9  X9
```

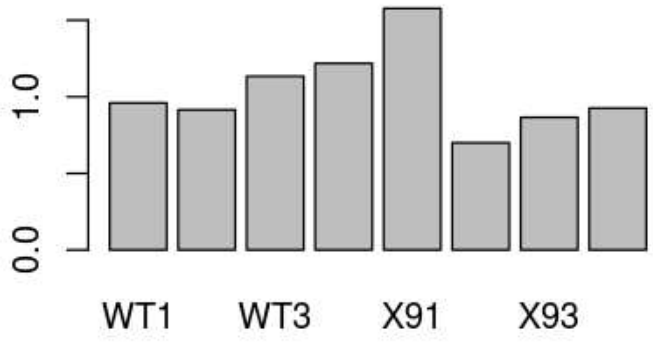
Then we ran the differential expression using the following command.

In [51]:

```
estimating size factors
estimating dispersions
gene-wise dispersion estimates
mean-dispersion relationship
final dispersion estimates
fitting model and testing
```

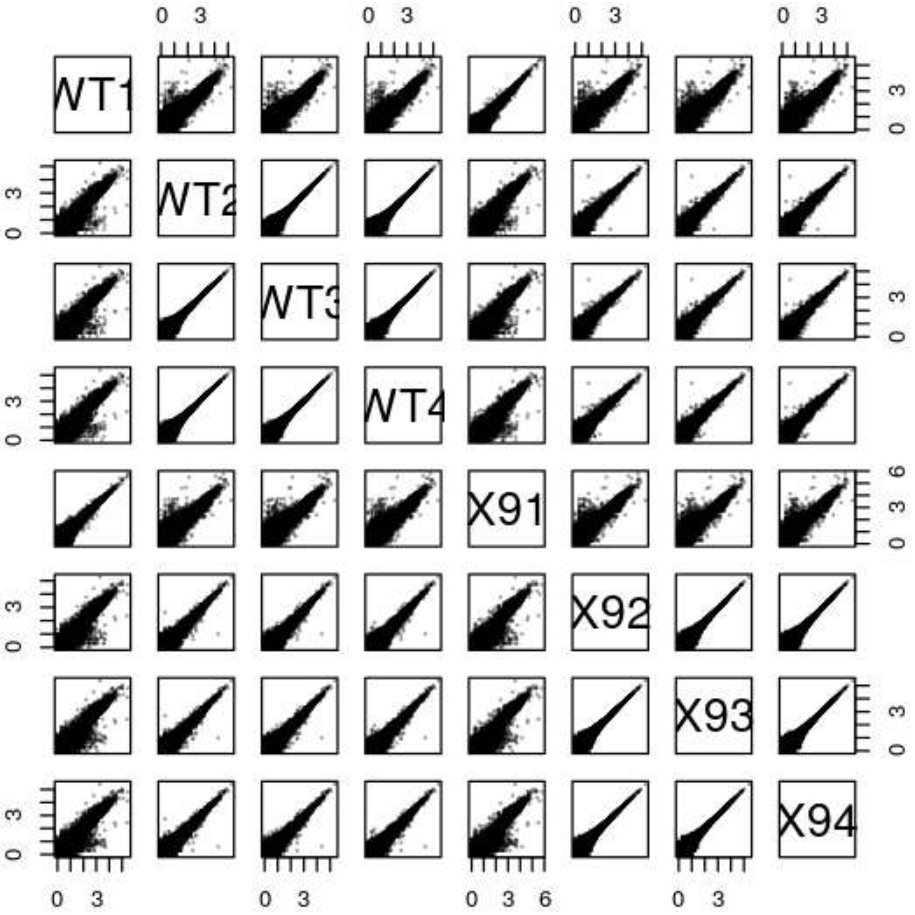
In [52]:

```
options(repr.plot.width=4, repr.plot.height=3)
```



Scatter plots allow us to compare samples. As you can see in WT1, WT1 seem to have big differences compared to the other WT samples that are more or less similar. The same can be said about X91. X91 seems far less similar compared to the other SX9 samples. This is something that we had already picked up earlier when we looked at the raw count data. It was already clear that the WT1 and SX91 differed from their other respective condition samples.

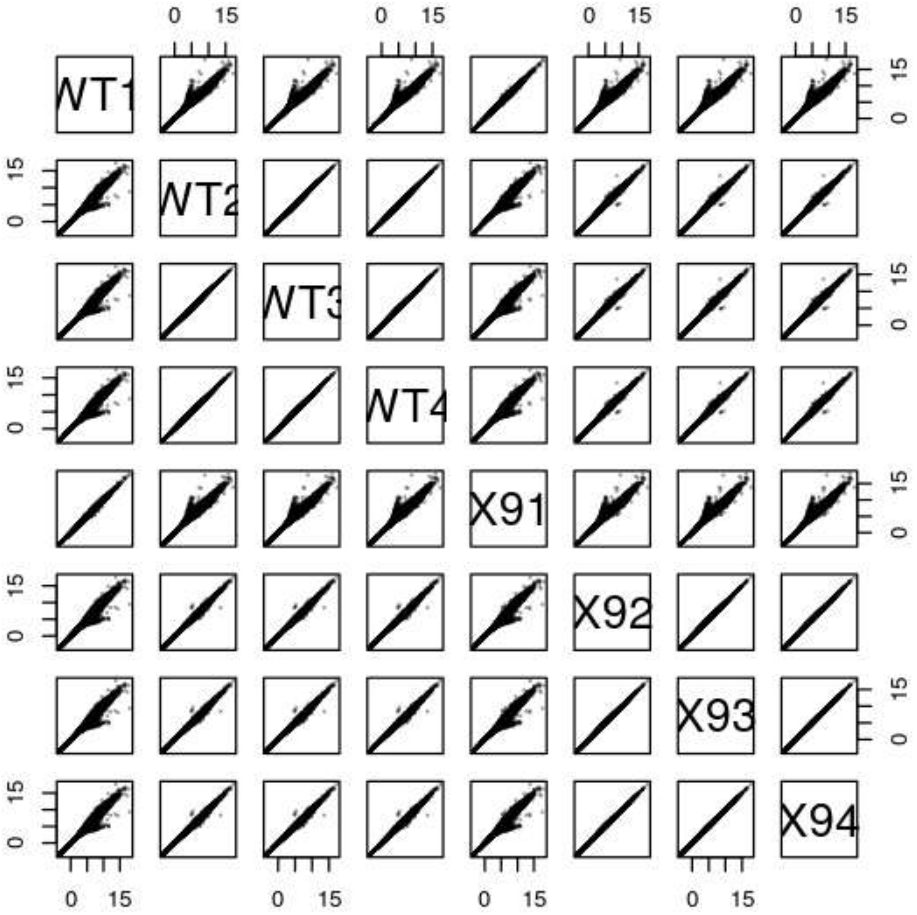
In [53]:
```
options(repr.plot.width=5, repr.plot.height=5)
```



Next, we minimized the variation of the genes that have low expression, normalized for library size and performed a log transformation on the data.

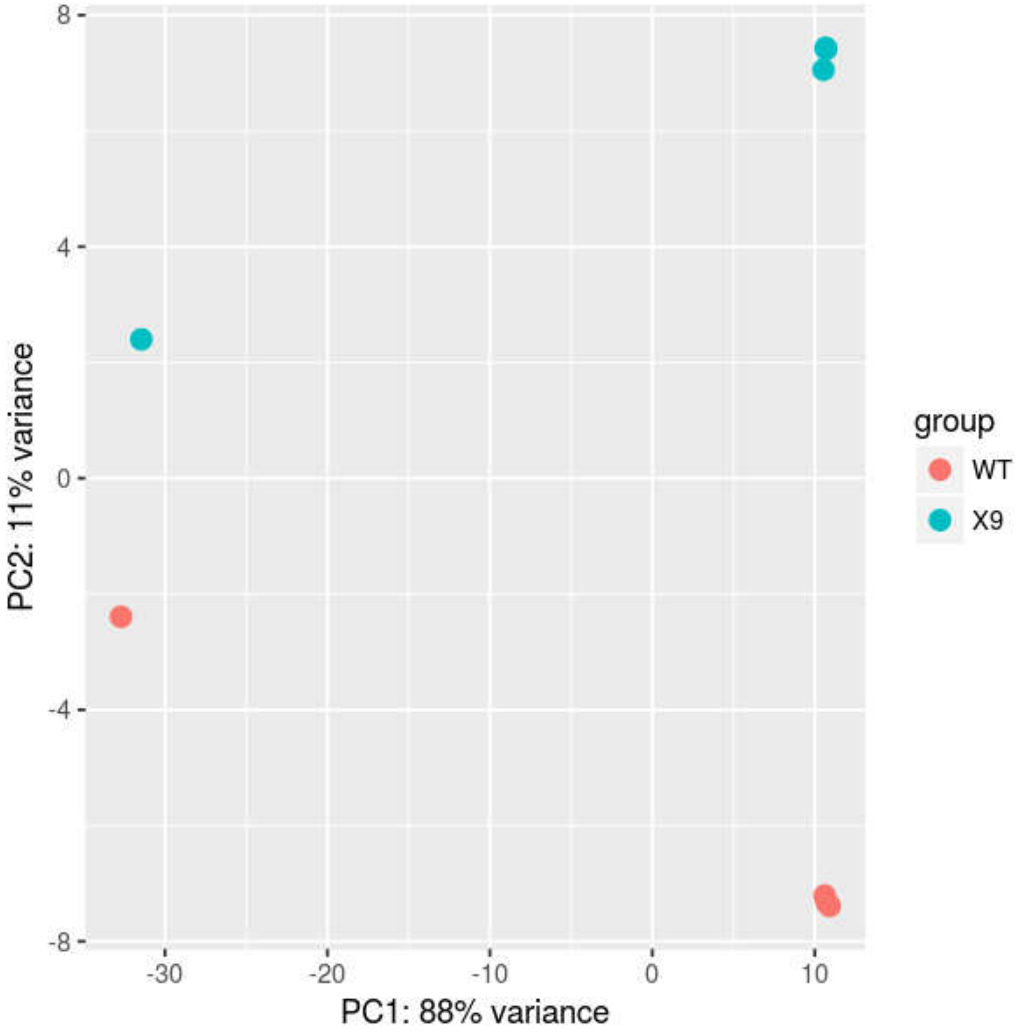
In [54]:
```

```

In [55]:
```
options(repr.plot.width=5, repr.plot.height=5)
```



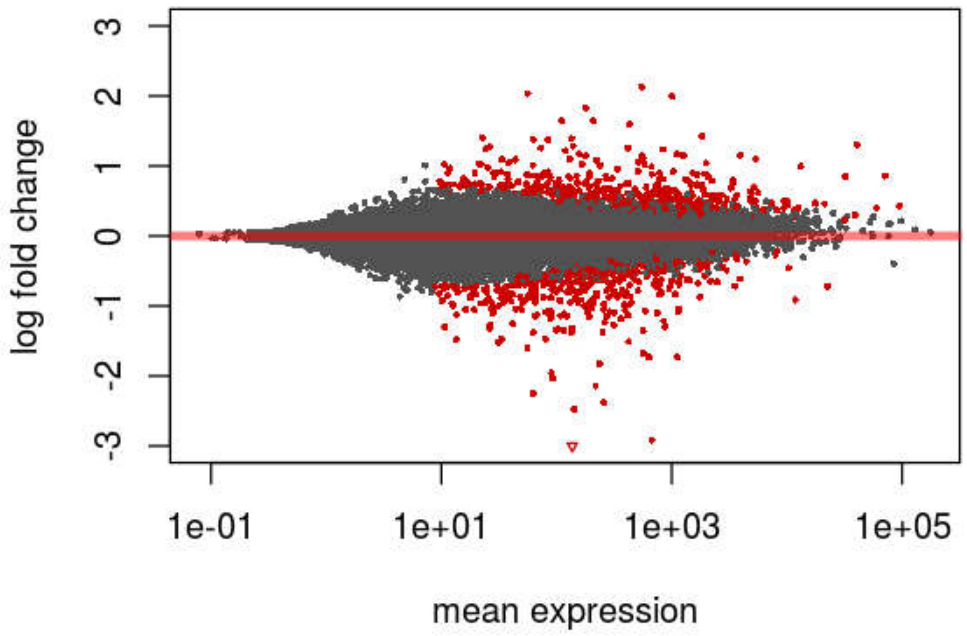
This PCA plot again confirms that in each condition we have has one sample that has great variance compared to the other samples of the respective condition.

In [56]:



This command allows us to show the mean expression against the fold log change. The red dots represent differentially expressed genes.

In [57]:
```
options(repr.plot.width=5, repr.plot.height=4)
```



The final steps are taken so we can export the results to our computer.

In [58]:
```
res <- results(dds)
res <- res[order(res$padj),]
```

```
log2 fold change (MAP): condition X9 vs WT
Wald test p-value: condition X9 vs WT
DataFrame with 6 rows and 6 columns
        baseMean log2FoldChange      lfcSE      stat        pvalue
       <numeric>      <numeric>  <numeric> <numeric>     <numeric>
Papss2  670.2378      -2.917510 0.11210935 -26.02379 2.664810e-149
Btn1a1  136.9021      -3.550124 0.20724760 -17.12987  8.885567e-66
Ddit3   548.3712       2.127837 0.13926956  15.27855  1.062772e-52
Alas2  1985.4826       1.039248 0.08011212  12.97241  1.754165e-38
Lum    1119.9212      -1.727724 0.14182498 -12.18208  3.872668e-34
Shisa2  637.8850      -1.733755 0.14260049 -12.15813  5.193648e-34
                padj
           <numeric>
Papss2 4.144845e-145
Btn1a1  6.910306e-62
Ddit3   5.510118e-49
Alas2   6.821071e-35
Lum     1.204710e-30
Shisa2  1.346367e-30
```

We used the following command to see how many significant differentially expressed gene we have (with an adjusted P value lower that 0.05).

In [59]:
```
res.sign <- res[(!is.na(res$padj)) & (res$padj < 0.05),]
```

```
666  6
```

Finally, we write our results to a tsv file which we can download and open in Excel.

In [60]: