# Untitled

*Justine Guégan*

*26 septembre 2016*

```
library(DESeq2)
```

```
## 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 'S4Vectors'
## Loading required package: IRanges
## Loading required package: GenomicRanges
## Loading required package: GenomeInfoDb
## Loading required package: Rcpp
## Loading required package: RcppArmadillo
```

```
## Warning: replacing previous import by 'ggplot2::Position' when loading
## 'DESeq2'
```

```
library(ggplot2)
library(pheatmap)
library(RColorBrewer)
comptage = read.delim("counts.txt",sep="\t", header=T,row.names=1)
countData = comptage[,-c(1:5)]
design = read.delim("annot_sample.csv",sep="\t", header=T,row.names=1)
colnames(countData) = rownames(design)
```

```
dds <- DESeqDataSetFromMatrix(countData = countData,
    colData = design,
    design = ~ condition)
```

How many genes have only 0 or 1 reads in all samples ? –> Remove those genes

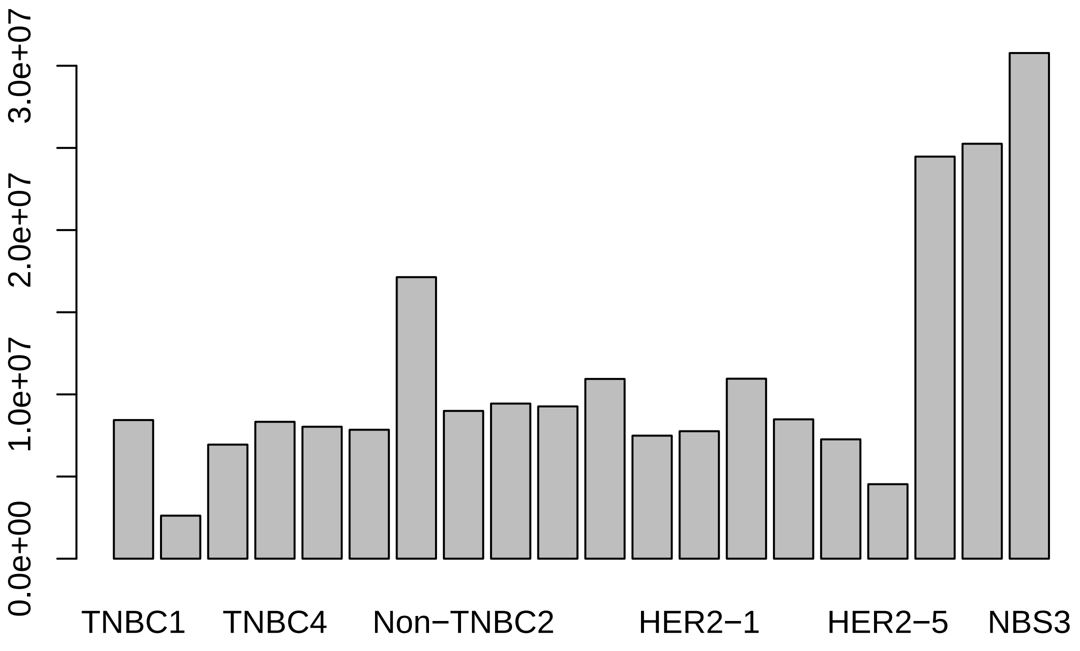
```
dds <- dds[ rowSums(counts(dds)) > 1, ]
```

```
dds$condition <- relevel(dds$condition, ref="NBS")
dds <- DESeq(dds)
```

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

Est- ce que toutes les librairires font la même taille ? Repésentation barplot des comptages, boxplot avant/après normalisation

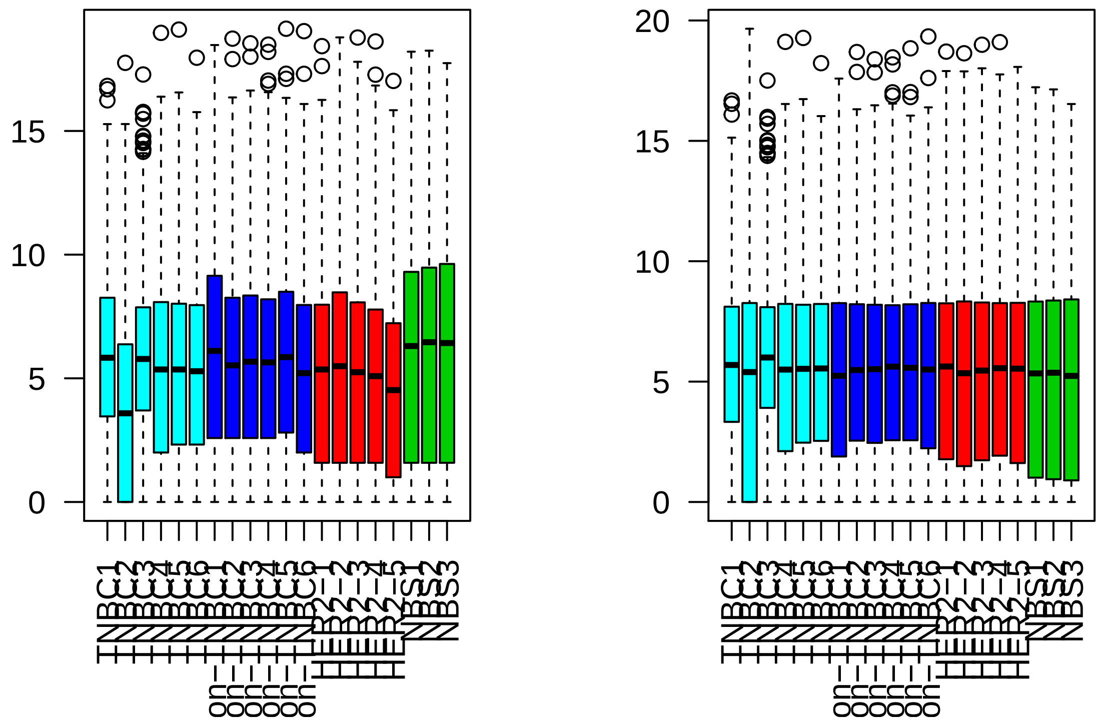
```
barplot(colSums(counts(dds)))
```



```
sizeFactors(dds) # divide each column by his size factor --> normalized counts
```

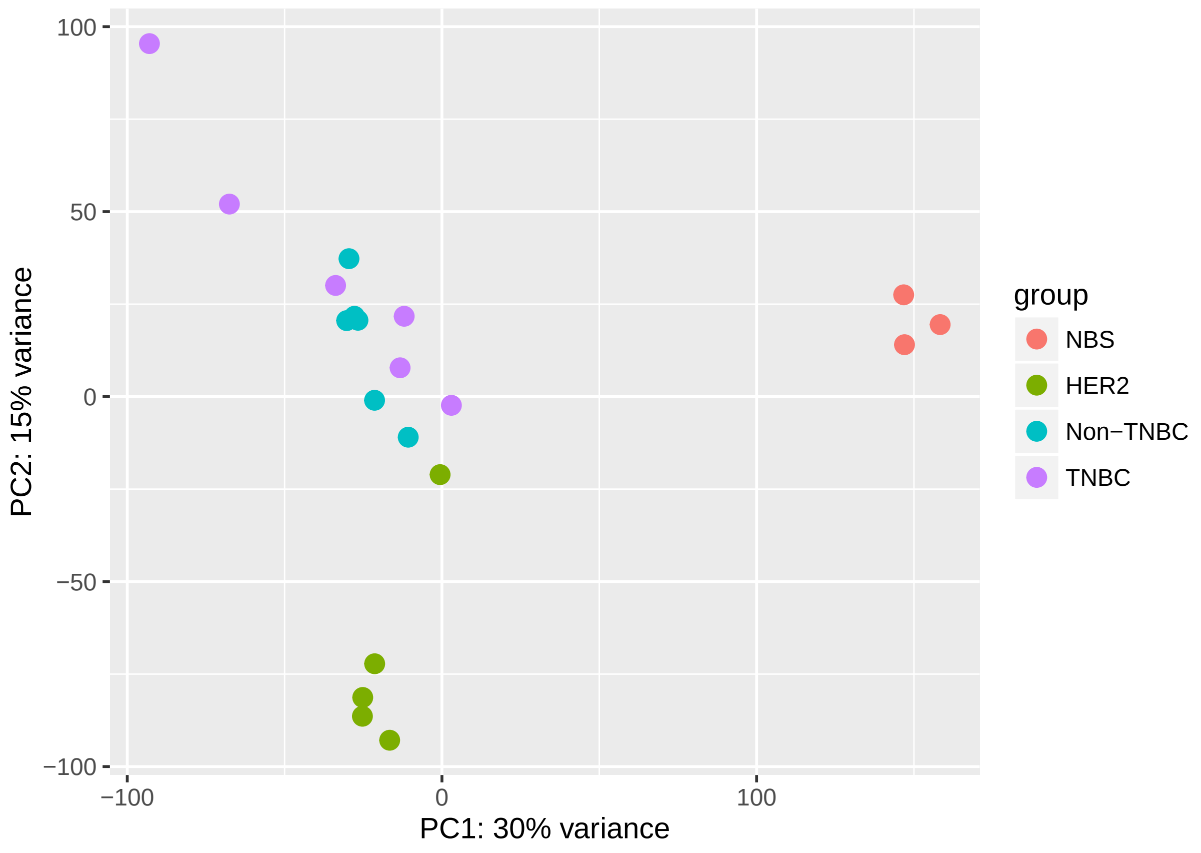
```
##     TNBC1     TNBC2     TNBC3     TNBC4     TNBC5     TNBC6 Non-TNBC1
## 1.1061054 0.2676300 0.8567024 0.9036194 0.8848666 0.8316407 1.8482558
## Non-TNBC2 Non-TNBC3 Non-TNBC4 Non-TNBC5 Non-TNBC6    HER2-1    HER2-2
## 1.0308634 1.1157055 1.0159386 1.2218178 0.8114948 0.8257115 1.1063464
##    HER2-3    HER2-4    HER2-5      NBS1      NBS2      NBS3
## 0.8611049 0.7160106 0.4839434 1.9725085 2.1573269 2.3162630
```

```r
par(mfrow=c(1,2))
boxplot(log2(counts(dds) +1 ), las=2, col=as.numeric(design$condition)+1)
boxplot(log2(counts(dds, normalized=TRUE) +1 ),las=2, col=as.numeric(design$condition)+1)
```

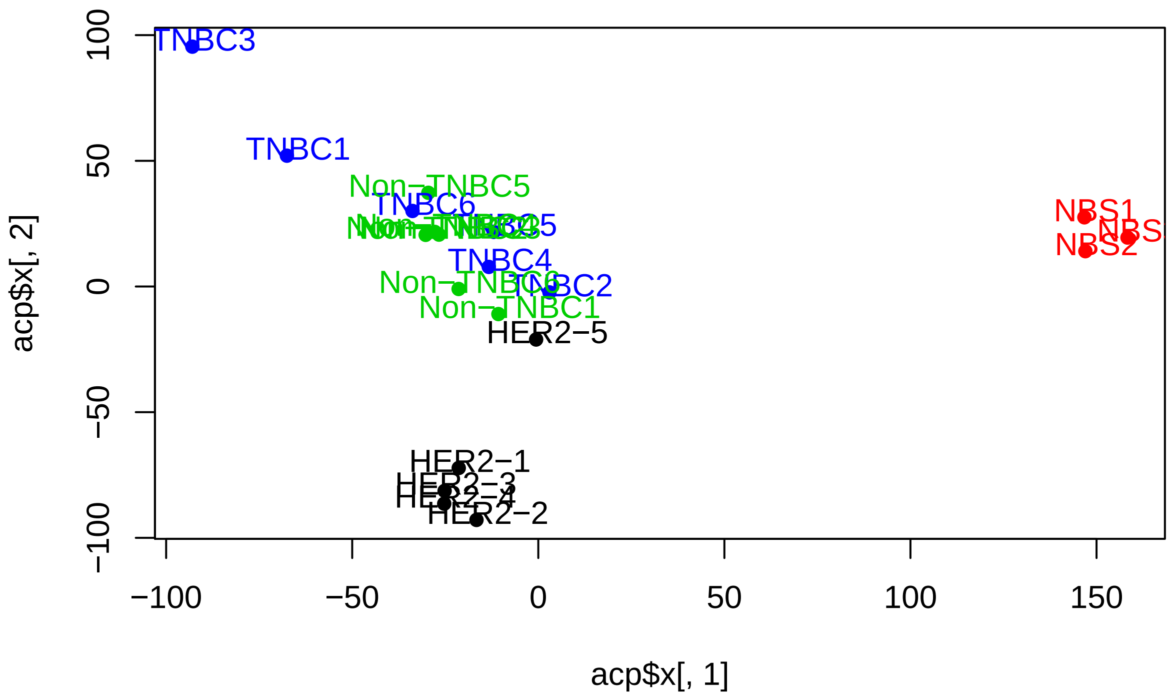


PCA :

```r
rld = rlog(dds, blind=FALSE)
plotPCA(rld, intgroup=c("condition"),ntop=nrow(rld))
```

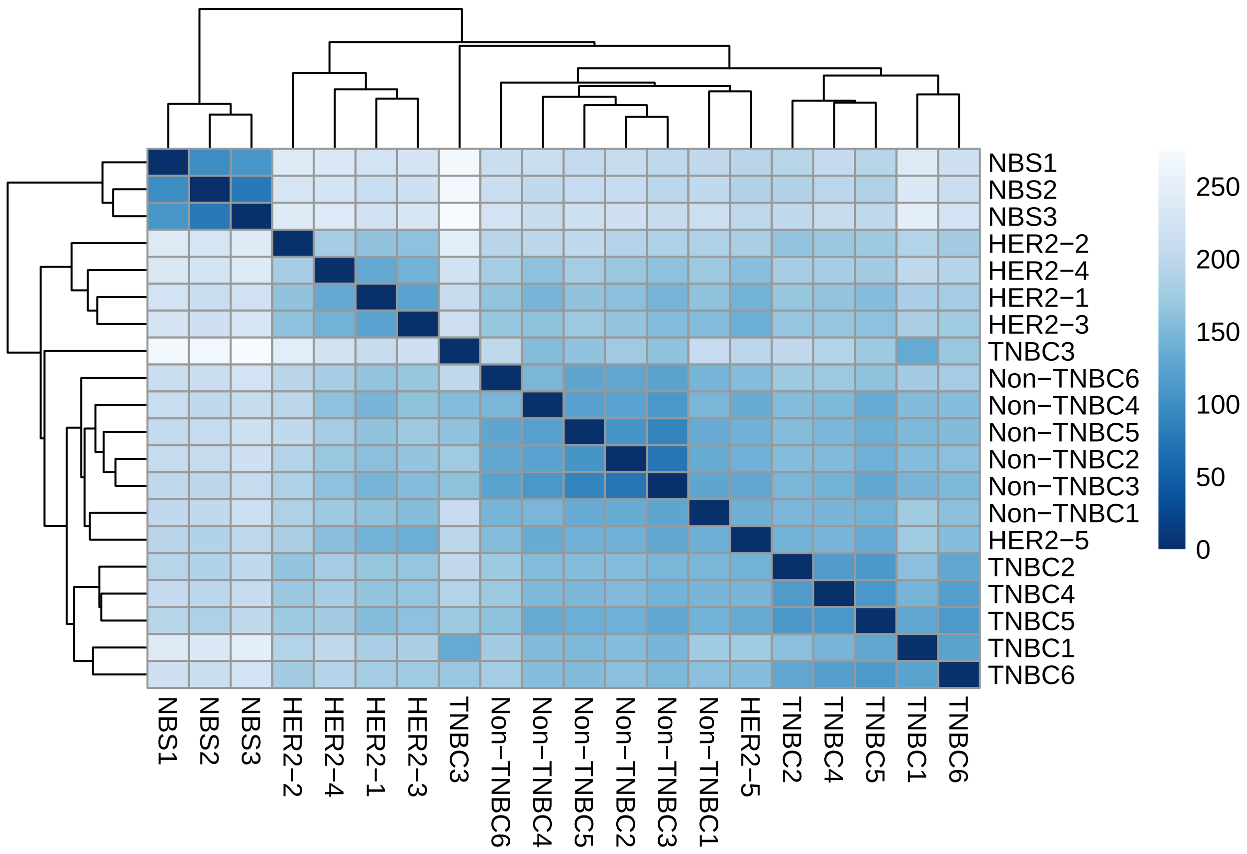


```
acp = prcomp(t(assay(rld)),center=TRUE)
plot(acp$x[,1], acp$x[,2], col=as.numeric(design$condition), pch=16)
text(acp$x[,1]+3, acp$x[,2]+3, col=as.numeric(design$condition),labels = colnames(rld))
```



Heatmap sample sample

```
sampledist = dist(t(assay(rld)))
sampledistMatrix =as.matrix(sampledist)
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampledistMatrix,col=colors)
```



Analyse Diff

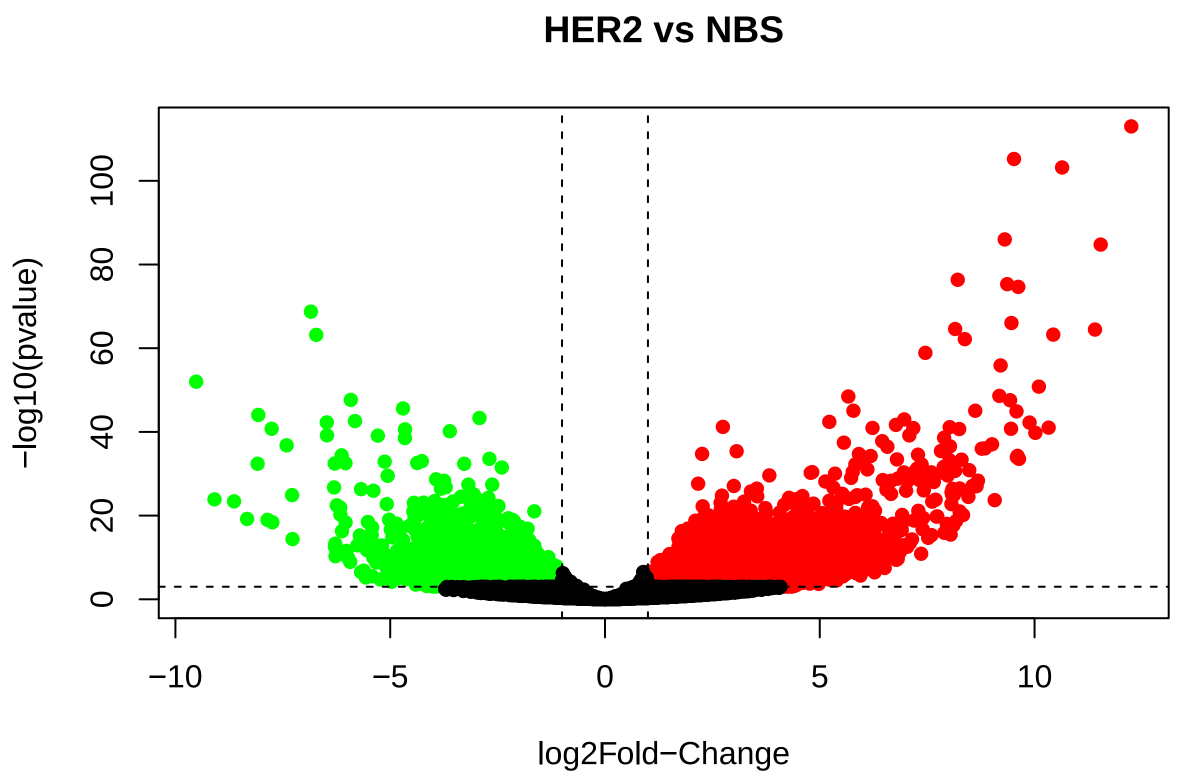
```
resHER2 <- results(dds, contrast=c("condition","HER2","NBS"),alpha = 0.001)
resHER2Ordered <- resHER2[order(resHER2$padj),]
summary(resHER2)
```

```
##
## out of 25346 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 6189, 24%
## LFC < 0 (down)     : 4603, 18%
## outliers [1]       : 307, 1.2%
## low counts [2]     : 3758, 15%
## (mean count < 2.4)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
up = which(resHER2$pvalue <= 0.001 & resHER2$log2FoldChange >= 1)
down = which(resHER2$pvalue <= 0.001 & resHER2$log2FoldChange <= -1)
```

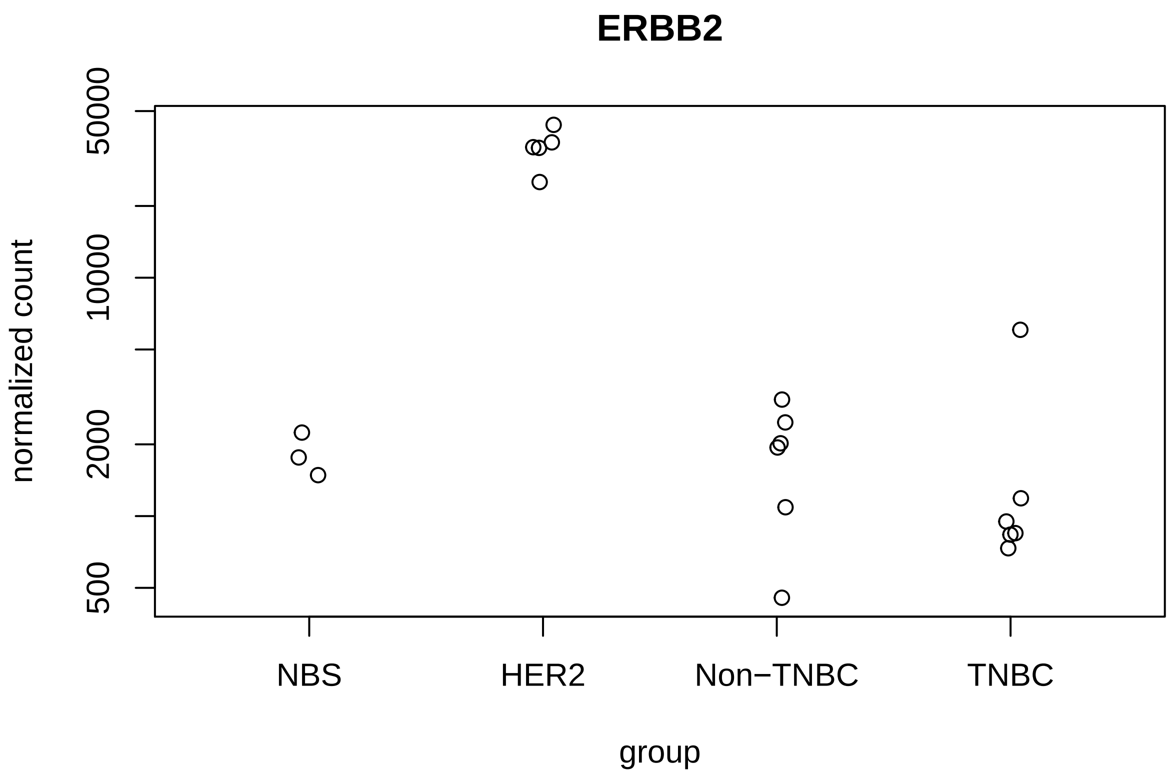
```r
plot(resHER2$log2FoldChange, -log10(resHER2$pvalue), type="n", xlab="log2Fold-Change", ylab="-log10(pval
points(resHER2$log2FoldChange[up], -log10(resHER2$pvalue[up]),col="red",pch=16)
points(resHER2$log2FoldChange[down], -log10(resHER2$pvalue[down]),col="green",pch=16)
points(resHER2$log2FoldChange[-c(up,down)], -log10(resHER2$pvalue[-c(up,down)]), pch=16)

abline(v=c(-1,1), lty=2)
abline(h=-log10(0.001), lty=2)
```



Dessiner le profil du gène clef dans les cancers HER2 (ERBB2) comparé aux autres types de cancer et aux éch NBS

```r
plotCounts(dds, gene="ERBB2", intgroup="condition")
```


ERBB2
normalized count
50000
10000
2000
500
NBS
HER2
Non−TNBC
TNBC
group