title: “DiffExpAnalysis”
author: “Mohammed ElBelbesy”

- ## Load FeatureCountsTable into R

Analyzing RNA-seq data with DESeq2 , the analysis of count data from RNA-seq is the detection of differentially expressed genes. As input, the DESeq2 package expects count data in the form of a matrix.

```
library (DESeq2)
library (Biobase)
countdata<- read.table ("/home/mlsi/RNASeq/countTable/featureCounts.txt", header=T, row.names = 1)
head (countdata)#In this step we will use the Head command to view the data  ( first 6 by default)
##                                   Chr     Start
## U2                                  22  10736171
## FRG1FP     22;22;22;22;22;22;22;22;22  10939388;10940597;10941691;10944967;10947304;10949212;10950049
## CU104787.1                      22;22  11065974;11067335
## BAGE5                           22;22  11066501;11067985
## ACTR3BP6                        22;22  11124337;11124508
## 5_8S_rRNA                          22  11249809
```

- ## Editing FeatureCountsTable

```
countdata<- countdata [ ,6:ncol(countdata)] # remove the first 6 columns
colnames(countdata) <- gsub ("\\X.home.mlsi.RNASeq.mapping.","",colnames(countdata)) #gsub is tool to perform replacement/edit
colnames(countdata)<- gsub("\\.UHR_[123].bam","",colnames(countdata)) #remove .UHR1\2\3
colnames(countdata)<- gsub("\\.HBR_[123].bam","",colnames(countdata))
head(countdata)#view the first 6 rows
##            HBR_1 HBR_2 HBR_3 UHR_1 UHR_2 UHR_3
## U2             0     0     0     0     0     0
## FRG1FP         0     0     0     0     0     0
## CU104787.1     0     0     0     0     0     0
## BAGE5          0     0     0     0     0     0
## ACTR3BP6       0     0     0     0     0     0
## 5_8S_rRNA      0     0     0     0     0     0
```

- ## Make class labels

```
class(countdata)    #class function used to determine the datatype of a variable
## [1] "data.frame"
countdata<- as.matrix(countdata) #convert data frame to a matrix
group<- factor(c(rep("HBR",3),rep("UHR",3)))
con<- factor (c(rep("cancer",3),rep("normal",3)))
coldata <- data.frame (row.names = colnames (countdata),group,con) #create a coldata frame
```

- ## Preparing colors for plots

```
library(RColorBrewer)
mycols<- brewer.pal(11, "Set3")[1:length(unique(group))]
#display.brewer.all()
mycols #choose the 11 colors
## [1] "#8DD3C7" "#FFFFB3"
```

- ## Create the DESeq DataSet

The DESeqDataSet class enforces non-negative integer values in the "counts" matrix stored as the first element in the assay list,

DESeqDataSet has an associated design formula which expresses how the counts for each gene depend on the variables in colData .

https://rdrr.io/bioc/DESeq2/man/DESeqDataSet.html

```
dds<- DESeqDataSetFromMatrix (countData = countdata,colData = coldata, design = ~ con)
design(dds)
## ~con
head(counts(dds))
##            HBR_1 HBR_2 HBR_3 UHR_1 UHR_2 UHR_3
## U2             0     0     0     0     0     0
## FRG1FP         0     0     0     0     0     0
## CU104787.1     0     0     0     0     0     0
```

- ## Creating the estimateSizeFactors

  Estimate size factors, which are scaling factors used as "offsets" by the statistical model to make the different samples comparable. This is necessary because the different samples may have been sequenced to slightly different depths. Additionally, the presence of differentially expressed genes may cause the apparent depth of many genes to appear different.

  http://hartleys.github.io/JunctionSeq/Rhtml/estimateJunctionSeqSizeFactors.html

```
dds<- estimateSizeFactors(dds)
sF<- sizeFactors(dds)
sF
##     HBR_1     HBR_2     HBR_3     UHR_1     UHR_2     UHR_3
## 0.7793659 0.9703951 0.8423092 1.4913253 0.9805609 1.1620297
norm_dds <- counts(dds, normalized=TRUE)
head (norm_dds)
##            HBR_1 HBR_2 HBR_3 UHR_1 UHR_2 UHR_3
## U2             0     0     0     0     0     0
## FRG1FP         0     0     0     0     0     0
## CU104787.1     0     0     0     0     0     0
## BAGE5          0     0     0     0     0     0
## ACTR3BP6       0     0     0     0     0     0
## 5_8S_rRNA      0     0     0     0     0     0
write.table (norm_dds, file= "/home/mlsi/RNASeq/analysis/DESeq2/ddsNormSF.txt", sep=" ", col.names = NA)
```

- ## Pre-Filtering

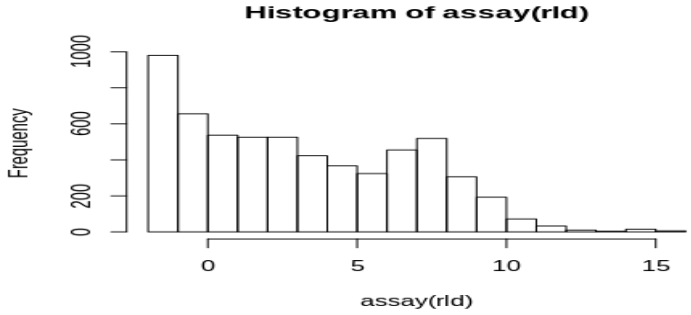
  filter out the lowly expressed genes which are represented in the table by zeros , by removing rows in which there are very few reads, we reduce the memory size of the dds data object, and we increase the speed of the transformation and testing functions within DESeq2.

```
dds<- dds [rowSums(counts(dds))>1, ]
dim(dds)
## [1] 992   6
dds<- estimateSizeFactors(dds)
sF<- sizeFactors(dds)
sF
##     HBR_1     HBR_2     HBR_3     UHR_1     UHR_2     UHR_3
## 0.7793659 0.9703951 0.8423092 1.4913253 0.9805609 1.1620297
norm_dds <- counts(dds, normalized=TRUE)
head (norm_dds)
##               HBR_1 HBR_2 HBR_3    UHR_1    UHR_2    UHR_3
## LA16c-60D12.1     0     0     0 4.023267 4.079298 0.000000
## LA16c-60D12.2     0     0     0 0.000000 8.158596 1.721126
## ZNF72P            0     0     0 0.000000 2.039649 0.000000
## BNIP3P2           0     0     0 0.000000 2.039649 0.000000
## LA16c-60G3.6      0     0     0 0.000000 2.039649 0.000000
## ARHGAP42P3        0     0     0 0.000000 4.079298 0.000000
write.table (norm_dds, file= "/home/mlsi/RNASeq/analysis/DESeq2/ddsNormSF.txt", sep=" ", col.names = NA)
colData(dds)
## DataFrame with 6 rows and 3 columns
##          group      con sizeFactor
##       <factor> <factor>  <numeric>
## HBR_1      HBR   cancer  0.7793659
## HBR_2      HBR   cancer  0.9703951
## HBR_3      HBR   cancer  0.8423092
## UHR_1      UHR   normal  1.4913253
## UHR_2      UHR   normal  0.9805609
## UHR_3      UHR   normal  1.1620297
```

- ## Scaling down the differences

  rlog function transforms the count data to the log2 scale in a way which minimizes differences between samples for rows with small counts, and which normalizes with respect to library size. https://www.rdocumentation.org/packages/DESeq2/versions/1.12.3/topics/rlog

```
rld<- rlogTransformation(dds)
hist(assay(rld)) #we make a histogram using "hist" to see the distribution of the values of the whole data set
```



```
head(assay(rld)) #To view the (rld) data we use the term assay.
##                    HBR_1      HBR_2      HBR_3      UHR_1      UHR_2   UHR_3
## LA16c-60D12.1 -0.4166441 -0.4389221 -0.4247142  0.6349742  0.6205201   0.1420802
## LA16c-60D12.2 -0.4904459 -0.5182887 -0.5004650 -0.5680166  1.2020601  -1.5899494
## ZNF72P        -1.5452155 -1.5697664 -1.5539134 -1.6178908 -1.1229702  -1.5899494
## BNIP3P2       -1.5452155 -1.5697664 -1.5539134 -1.6178908 -1.1229702  -1.5899494
## LA16c-60G3.6  -1.5452155 -1.5697664 -1.5539134 -1.6178908 -1.1229702  -1.5899494
## ARHGAP42P3    -1.1915067 -1.2311987 -1.2055688 -1.2804190 -0.2088327  -1.2638289
```

All the working steps before is just a data preparation. we have already calculated size factors, so we will use this pre-existing size factors .

- ## Differential Expression Analysis via DESeq2

```
dds_con<- DESeq(dds)
## using pre-existing size factors
## estimating dispersions
```

- ## Extract Results From A DESeq Analysis

  - results extracts a result table from a DESeq analysis giving base means across samples, log2 fold changes, standard errors, test statistics, p-values and adjusted p-values; resultsNames returns the names of the estimated effects of the model.

  - **Finding factor setting and possible comparisons**

    one possibility , if we have more than two factors variable liver cancer, breast cancer..., we can calculate the common ones or make comparisons "con normal_vs cancer for example . a comparison happened with at least two.

```
resultsNames(dds_con)
## [1] "Intercept"           "con_normal_vs_cancer"
res_con<- results(dds_con, contrast = c("con", "normal", "cancer"))
summary(res_con)
## out of 992 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)     : 199, 20%
## LFC < 0 (down)   : 182, 18%
## outliers [1]     : 0, 0%
## low counts [2]   : 250, 25%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

- **Exploring the result table:**

- Adjusted Values smaller than 0.1 statistical (all other genes are ignored)value gives an idea if the significant or not, this threshold is by default.
- Gives an idea if genes are up or down regulated.
- The significance and adjusted pvalue are decreasing, the higher the counts and the bigger the difference.
- The lower pvalue the higher the signinfance, when it comes to NGS adjusted pvalue is used.

```
mcols(res_con, use.names = T)
## DataFrame with 6 rows and 2 columns
##                         type                                 description
##                  <character>                                 <character>
## baseMean       intermediate    mean of normalized counts for all samples
## log2FoldChange      results log2 fold change (MLE): con normal vs cancer
## lfcSE               results         standard error: con normal vs cancer
## stat                results         Wald statistic: con normal vs cancer
## pvalue              results      Wald test p-value: con normal vs cancer
## padj                results                         BH adjusted p-values
```

- Gives you the meta/phenodata, another possibility of showing information.
- We have not many metadata, we have just a description of our comparison, we can analyze the intermediate, condition of cancer versus normal, and then we got a mean of normalized counts ,log2 fold change, standard error and adjusted p values.

```
head(res_con)
## log2 fold change (MLE): con normal vs cancer
## Wald test p-value: con normal vs cancer
## DataFrame with 6 rows and 6 columns
##                baseMean log2FoldChange     lfcSE      stat     pvalue   padj
##               <numeric>      <numeric> <numeric> <numeric>  <numeric>   <numeric>
## LA16c-60D12.1 1.3504275       3.659970  2.474500 1.4790747 0.13912036   0.2278749
## LA16c-60D12.2 1.6466203       3.925637  2.380037 1.6494014 0.09906544   0.1745999
## ZNF72P        0.3399415       1.636374  4.018398 0.4072204 0.68384612   NA
```

```
## BNIP3P2       0.3399415       1.636374  4.018398 0.4072204 0.68384612   NA
## LA16c-60G3.6  0.3399415       1.636374  4.018398 0.4072204 0.68384612   NA
## ARHGAP42P3    0.6798830       2.647426  3.555817 0.7445339 0.45655354   NA
```

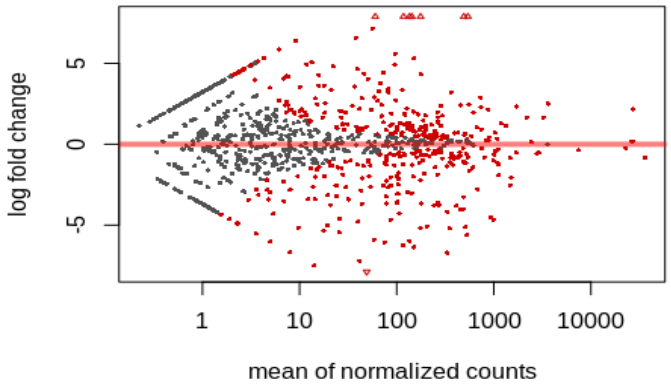
- **Change the design of the DESeqDataSet and start a new analysis.**

we shouldn't overwrite the dds, we should give it a name because we have another design dds_con from DESeq Pipeline , not to perform the all steps again . Designs can be complicated in multifunctional analyzes which means how the gene expressed is changing during time ?

```
design(dds)<- ~group
design(dds)
## ~group
dds_group<- DESeq(dds)
resultsNames(dds_group)
## [1] "Intercept"        "group_UHR_vs_HBR"
res_group<- results(dds_group)
```

- **plotMA Scatter plot.**

```
plotMA (res_con)
```

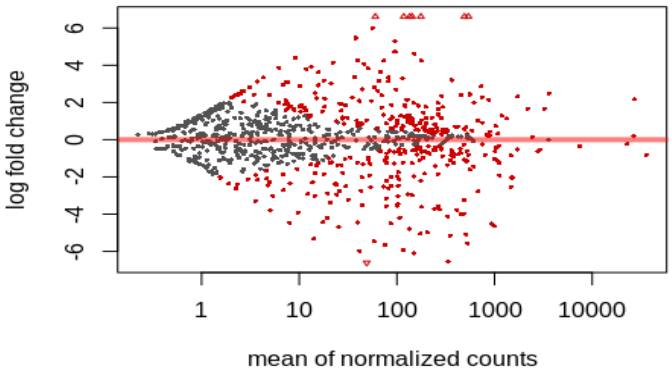


-Visualize these differences which means they need statistical values to make plot (log2fold changes vs the Mean of normalized counts).
-Lowly expressed genes are away from the zero line .
-Black dots these are the genes that are not differentially regulated between the sample cancer vs normal,
- The red dots which are closer to the zero line (adjusted pvalue is smaller than 0.1) are Significantly differentially regulated genes .

- **Shrinkage**

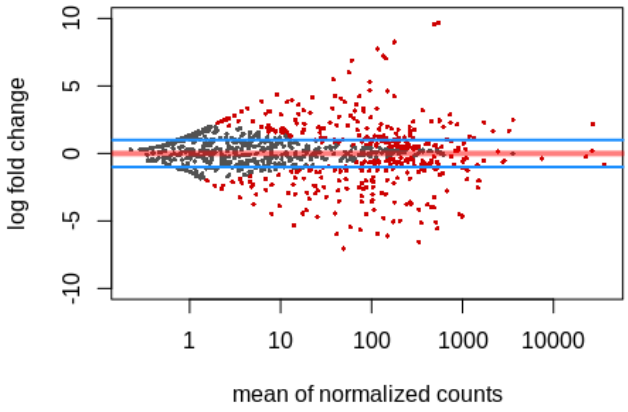
Shrinkage of effect size (LFC estimates) is useful for visualization and ranking of genes.

```
resLFC_con<- lfcShrink(dds_con,coef=2) #it removes the noise associated with log2 fold changes
plotMA(resLFC_con)
```

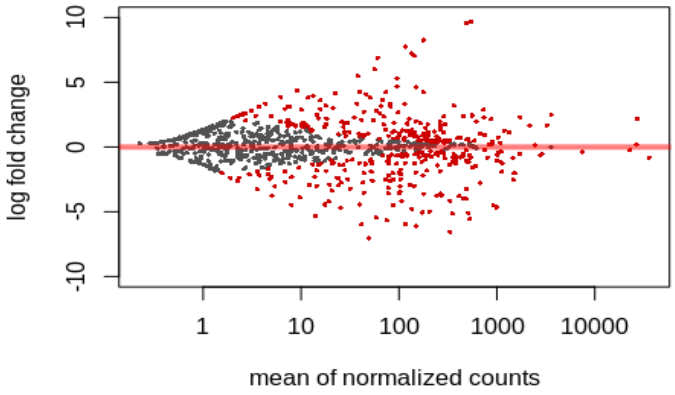


- Add an abline to the MAplot

```
plotMA(resLFC_con, ylim = c(-10,10))
abline (h=c(-1,1), col="dodgerblue" , lwd=2)
```

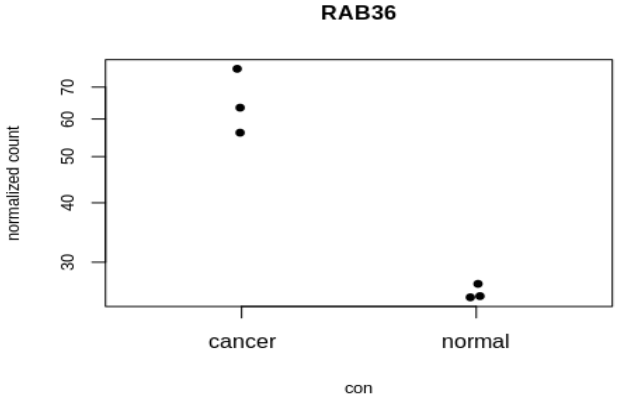



```
plotMA(resLFC_con, ylim = c(-10,10))
idx <- identify(resLFC_con$baseMean, resLFC_con$log2FoldChange)
```



- Interactively detect the row number of individual genes by clicking on the plot: (run plotMA and identify in console)

```
rownames(resLFC_con)[idx]
## character(0)
plotCounts (dds_con,gene = "RAB36", intgroup = "con", xlab= "con" , cex =0.8, pch=19, cex.lab=0.8, cex.sub=0.8,
cex.axis =0.8, cex.main=1  )
```



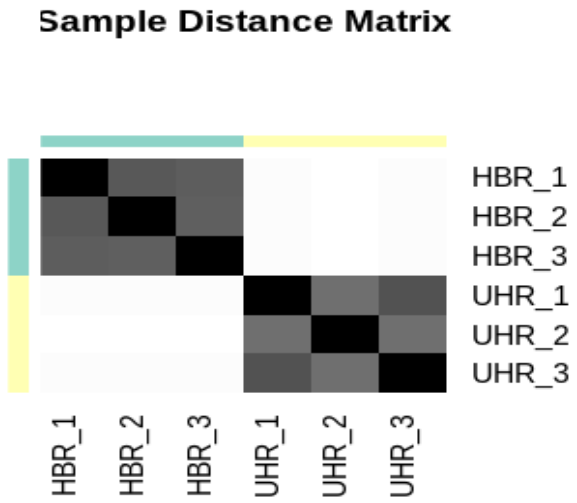
- **Sample distance heatmap**

  A heatmap of this distance matrix gives us an overview over similarities and dissimilarities between samples.

```
sampleDists <- as.matrix(dist(t(assay(rld))))
```

  - Creating the plot via gplot

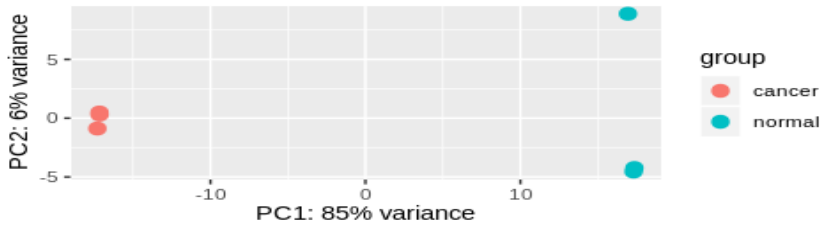
```
library(gplots)
heatmap.2 (as.matrix(sampleDists), key=F, trace="none",Colv = c("normal", "cancer"),Rowv =
"Colv",dendrogram="none",col=colorpanel(100,"black","white"),ColSideColors=mycols[con],
RowSideColors=mycols[con],margin=c(10,10),main="Sample Distance Matrix")
```



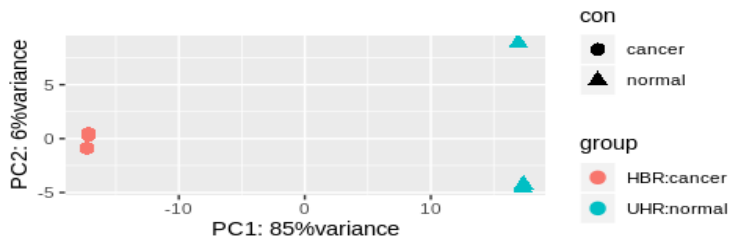
- **Principal component plot (PCA)**

This type of plot is useful for visualizing the overall effect of experimental covariates and batch effects.

```
plotPCA(rld, intgroup="con") #reload to the distance matrix is the PCA plot
```

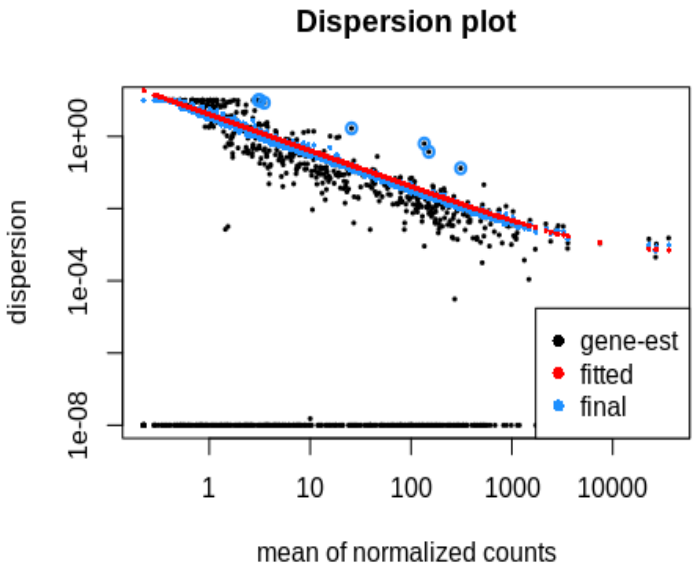



```
library(ggplot2)
pcaData <- plotPCA(rld, intgroup=c("group","con"), returnData=TRUE)
percentVar <- round(100 * attr(pcaData, "percentVar"))
ggplot(pcaData, aes(PC1, PC2, color=group, shape=con)) + geom_point(size=3)+
xlab(paste0("PC1: ", percentVar[1],"%variance")) +
ylab(paste0("PC2: ", percentVar[2],"%variance")) +
coord_fixed()
```



- **Plot dispersion**
  Note : the ground line represents the spike-in

```
plotDispEsts(dds_con, main= "Dispersion plot")
```



  - Checking some p-values in the results data

```
table(res_con$padj<0.5)
##
## FALSE  TRUE
##   191   551
res_con <- res_con[order(res_con$padj), ]
res_con
## log2 fold change (MLE): con normal vs cancer
## Wald test p-value: con normal vs cancer
## DataFrame with 992 rows and 6 columns
##              baseMean log2FoldChange      lfcSE       stat        pvalue       padj
##             <numeric>      <numeric>  <numeric>  <numeric>     <numeric>  <numeric>
## SYNGR1       986.7215      -4.662950 0.11914480  -39.13683  0.000000e+00 0.00000e+00
## SEPT3        926.8501      -4.527019 0.12032694  -37.62266  0.000000e+00 0.00000e+00
## ERCC-00004  3589.9516       2.501269 0.05482358   45.62396  0.000000e+00 0.00000e+00
## ERCC-00130 27062.3042       2.175087 0.03596987   60.46970  0.000000e+00 0.00000e+00
## YWHAH       1474.2754      -2.530765 0.07315383  -34.59512 2.991523e-262 4.43942e-260
## ...               ...            ...        ...        ...           ...          ...
## ERCC-00134  0.3957375     -2.3568352   4.035717 -0.5839942     0.5592242           NA
## ERCC-00137  0.9033978      3.0663660   2.866876  1.0695844     0.2848064           NA
## ERCC-00138  0.7676396     -0.3567371   3.126116 -0.1141151     0.9091465           NA
## ERCC-00142  0.3399415      1.6363736   4.018398  0.4072204     0.6838461           NA
## ERCC-00164  0.3435027     -2.1801430   4.052199 -0.5380148     0.5905668           NA
```

- how many genes smaller than adjusted p Value

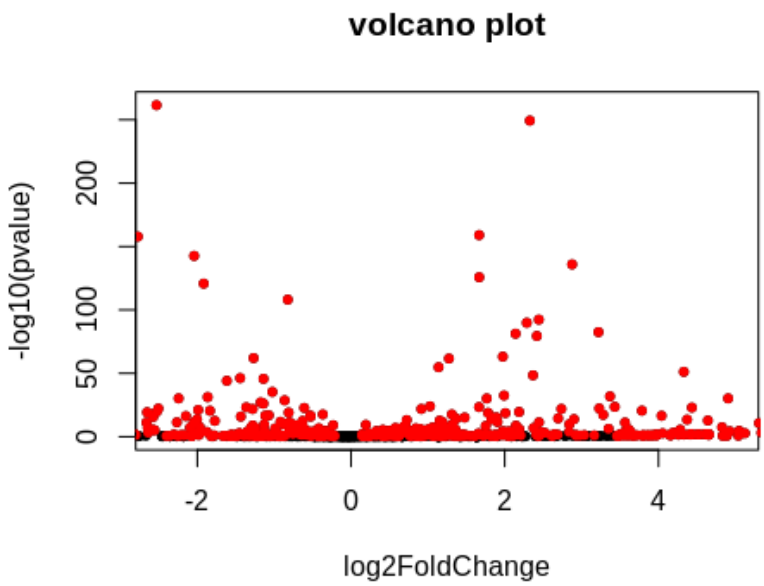
```
table(res_con$pvalue>0.001)
##
## FALSE  TRUE
##   251   741

table(res_con$log2FoldChange>5)
##
## FALSE  TRUE
##   975    17
```

- **Volcano Plot**

A volcano plot is a type of scatterplot that shows statistical significance (P value) versus magnitude of change (fold change). It enables quick visual identification of genes with large fold changes that are also statistically significant. In a volcano plot, the most upregulated genes are towards the right, the most downregulated genes are towards the left, and the most statistically significant genes are towards the top.

```
with(res_con, plot(log2FoldChange, -log10(pvalue), pch=20, main="volcano plot", xlim=c(-2.5,5)))
with(subset(res_con, padj<0.5 ), points(log2FoldChange, -log10(pvalue), pch=20, col="red"))
```



- **Merge results with normalized count data**

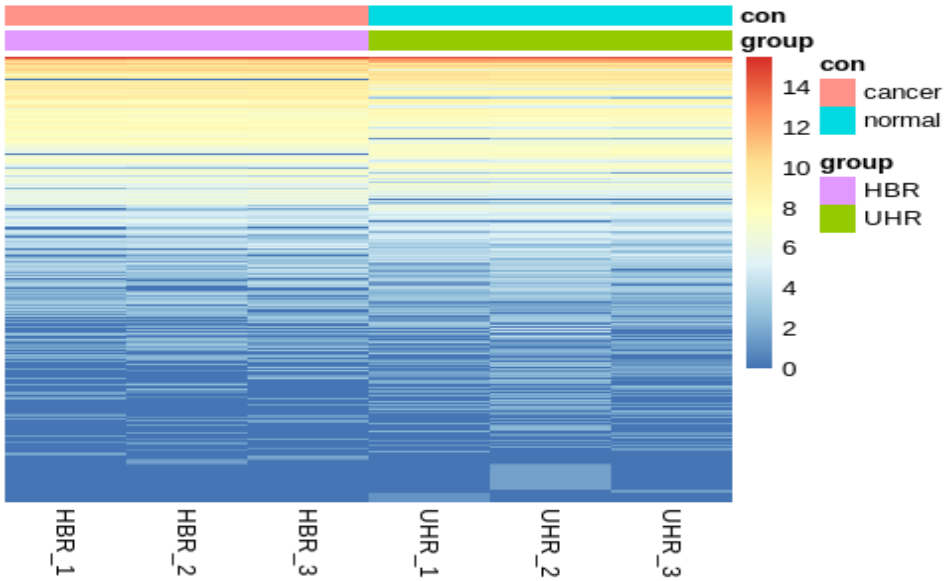
```
resdata <- merge(as.data.frame(res_con), as.data.frame(counts(dds_con, normalized =T)), by="row.names", sort=F)
names(resdata)[1] <- "Gene"
head(resdata)
##         Gene   baseMean log2FoldChange      lfcSE      stat        pvalue
## 1     SYNGR1   986.7215      -4.662950 0.11914480 -39.13683  0.000000e+00
## 2      SEPT3   926.8501      -4.527019 0.12032694 -37.62266  0.000000e+00
## 3 ERCC-00004  3589.9516       2.501269 0.05482358  45.62396  0.000000e+00
## 4 ERCC-00130 27062.3042       2.175087 0.03596987  60.46970  0.000000e+00
## 5      YWHAH  1474.2754      -2.530765 0.07315383 -34.59512 2.991523e-262
## 6 ERCC-00136  1727.3486       2.326855 0.06889317  33.77483 4.619251e-250
##            padj     HBR_1      HBR_2     HBR_3       UHR_1       UHR_2    UHR_3
## 1  0.000000e+00 1852.7882  1902.3180 1937.5307    71.74826    94.84367   61.09999
## 2  0.000000e+00 1710.3648  1728.1621 1890.0422    67.72500    74.44718   90.35914
## 3  0.000000e+00 1098.3288  1032.5691 1104.1076  5942.36562  6377.98212   5984.35667
## 4  0.000000e+00 9713.0242 10113.4068 9606.9230 44973.42155 42345.15087   45621.89887
## 5 4.439420e-260 2478.9383  2572.1483 2490.7717   431.16013   407.92978   464.70414
## 6 5.712474e-248  603.0543   585.3286  534.2456  2865.90732  2884.06355   2891.49244
```

- **Heatmap of normalized dds counts**

To explore a count matrix, it is often instructive to look at it as a heatmap.

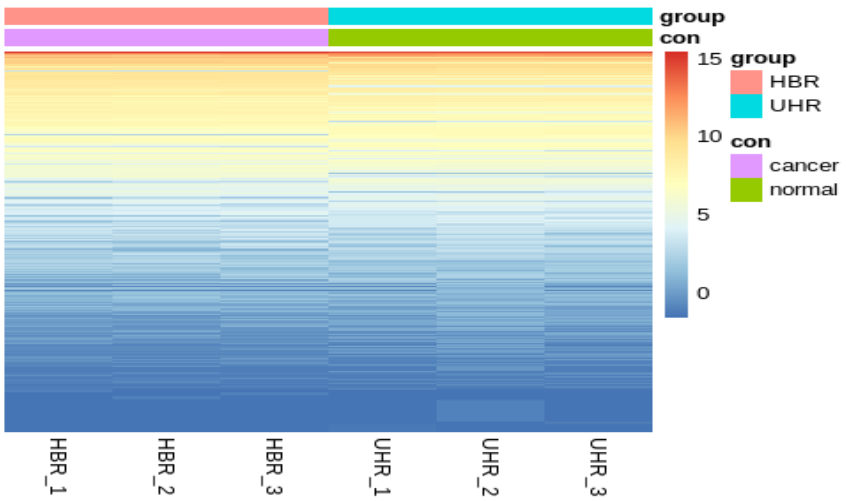
```
library(pheatmap)
select <- order(rowMeans(counts(dds,normalized=TRUE)),decreasing=TRUE) [1:992]
nt <- normTransform(dds) # defaults to log2(x+1)
log2.norm.counts <- assay(nt)[select,]
df <- as.data.frame(colData(dds)[,c("group","con")]) #for annotation

pheatmap(log2.norm.counts, cluster_rows=FALSE, show_rownames=FALSE,
cluster_cols=FALSE, annotation_col=df, fontsize_row = 5)
```



- **Heatmap of regularized log transformed dds counts**

```
df <- as.data.frame(colData(rld)[,c("con","group")])
pheatmap(assay(rld)[select,], cluster_rows=FALSE, show_rownames=FALSE,
cluster_cols=FALSE, annotation_col=df, fontsize_row =8)
```




```
resdata <- merge(as.data.frame(res_con), as.data.frame(counts(dds_con, normalized =T)), by="row.names", sort=F)
names(resdata)[1] <- "Gene"
head(resdata)
##         Gene   baseMean log2FoldChange      lfcSE      stat        pvalue
## 1     SYNGR1   986.7215      -4.662950 0.11914480 -39.13683  0.000000e+00
## 2      SEPT3   926.8501      -4.527019 0.12032694 -37.62266  0.000000e+00
## 3 ERCC-00004  3589.9516       2.501269 0.05482358  45.62396  0.000000e+00
## 4 ERCC-00130 27062.3042       2.175087 0.03596987  60.46970  0.000000e+00
## 5      YWHAH  1474.2754      -2.530765 0.07315383 -34.59512 2.991523e-262
## 6 ERCC-00136  1727.3486       2.326855 0.06889317  33.77483 4.619251e-250
##            padj     HBR_1      HBR_2     HBR_3       UHR_1       UHR_2    UHR_3
## 1  0.000000e+00 1852.7882  1902.3180 1937.5307    71.74826    94.84367    61.09999
## 2  0.000000e+00 1710.3648  1728.1621 1890.0422    67.72500    74.44718    90.35914
## 3  0.000000e+00 1098.3288  1032.5691 1104.1076  5942.36562  6377.98212    5984.35667
## 4  0.000000e+00 9713.0242 10113.4068 9606.9230 44973.42155 42345.15087    45621.89887
```

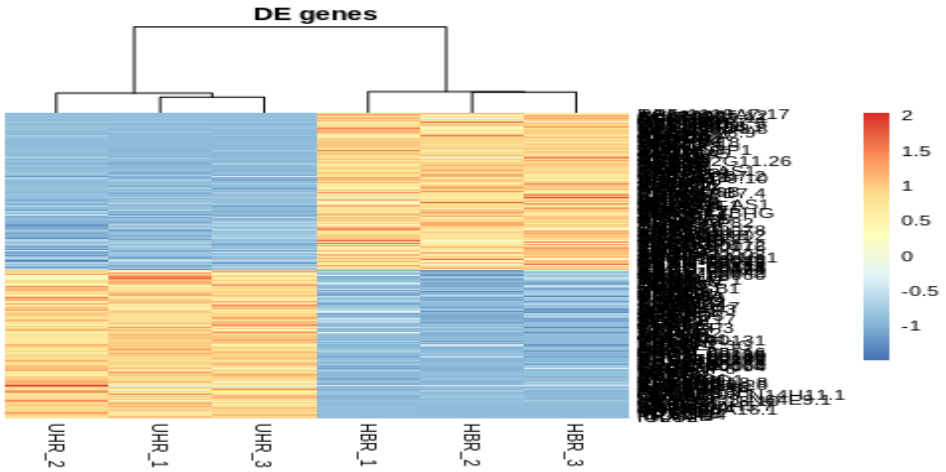
- **Heatmaps of the results tables:**

```
library(pheatmap)
resSig1<- resdata [which(resdata$padj<0.01), ]
diff_genes<-resSig1 [order(resSig1$log2FoldChange), ]  #to sort the results
head(diff_genes)
##              Gene   baseMean log2FoldChange     lfcSE       stat
## 113 RP5-1119A7.17  49.100459      -9.316448 1.2209597  -7.630430
## 154   AC000095.12  14.119416      -7.513837 1.3241644  -5.674399
## 136        CACNG2  41.627655      -7.208934 1.0801789  -6.673833
## 22        SULT4A1 333.983679      -6.705632 0.3263016 -20.550413
## 193   AC000095.11   7.945515      -6.682305 1.4414552  -4.635805
## 40         MPPED1 150.976639      -6.368008 0.4391808 -14.499742
##           pvalue         padj     HBR_1      HBR_2     HBR_3    UHR_1     UHR_2    UHR_3
## 113 2.339725e-14 1.536350e-13  97.51517 109.233862  87.85372 0.000000   0.000000  0.000000
## 154 1.391762e-08 6.705765e-08  35.92664  22.671179  26.11867 0.000000   0.000000  0.000000
## 136 2.492066e-11 1.359642e-10  76.98566  63.891504 106.84912 0.000000   2.039649  0.000000
## 22  7.631181e-94 2.573789e-92 698.00332 634.793007 651.77963 6.705445   9.178420  3.442253
## 193 3.555516e-06 1.366939e-05  19.24642   8.244065  20.18261 0.000000   0.000000  0.000000
## 40  1.216058e-47 2.255788e-46 289.97932 297.816849 306.30081 1.341089   6.118947  4.302816
```

```
z<- (as.data.frame(diff_genes, row.names = diff_genes$Gene))
z<- subset (z,select=c("UHR_1","UHR_2","UHR_3", "HBR_1","HBR_2", "HBR_3"))
deRNA_scale <- t(z)
deRNA_scale<- scale(deRNA_scale)
deRNA_scale<- t (deRNA_scale)
head(deRNA_scale)
##                    UHR_1      UHR_2      UHR_3     HBR_1     HBR_2  HBR_3
## RP5-1119A7.17 -0.9057217 -0.9057217 -0.9057217 0.8930722 1.1092386  0.7148542
## AC000095.12   -0.8787832 -0.8787832 -0.8787832 1.3572674 0.5322561  0.7468259
## CACNG2        -0.8861908 -0.8427697 -0.8861908 0.7527192 0.4739642  1.3884678
## SULT4A1       -0.9106125 -0.9037318 -0.9196920 1.0128411 0.8369660  0.8842292
## AC000095.11   -0.8221985 -0.8221985 -0.8221985 1.1694123 0.0308938  1.2662895
## MPPED1        -0.9283676 -0.8987249 -0.9099925 0.8623993 0.9110248  0.9636609

dim(deRNA_scale)
## [1] 277   6

RNA<- head (deRNA_scale, n = 381L)
df <- as.data.frame (colData(rld)[,c("con")])
pheatmap (RNA,fontsize=8,main = "DE genes", margins= c(6,14), cexRow=.1, cluster_cols =
T, show_rownames=T,show_colnames= T, cluster_rows = F)
```



- **Prepare your results table for GSEA analysis to make enrichment analysis via GSEA.**

```
write.table(resdata, file = "/home/mlsi/RNASeq/analysis/DESeq2/diffecpr=results_RNASeq.txt" , sep =" ",
col.names = NA)
resdata_GSEA<- resdata[ ,-(2:7)]

write.table (resdata_GSEA, file = "/home/mlsi/RNASeq/analysis/DESeq2/diffexpr-results_RNASeq_GSEA.txt", sep =
"\t", col.names=NA)
resdata_GSEA<- read.table ("/home/mlsi/RNASeq/analysis/DESeq2/diffexpr-results_RNASeq_GSEA.txt", header= TRUE,
row.names=2)
resdata_GSEA<- resdata_GSEA [ ,-1]

write.table (resdata_GSEA, file = "/home/mlsi/RNASeq/analysis/DESeq2/diffexpr-results_RNASeq_GSEA.txt", sep =
"\t ", col.names=NA)
```