lab1.r

bauricenafack

2022-10-30

```
setwd("~/Documents/computational-genomics/kap1-experiment")
library(edgeR)
```

```
## Loading required package: limma
```

```
counts<-read.csv("data/GSE105128_genecount.csv")
head(counts)
```

```
##             geneName  WT1  WT2  WT3  KO1  KO2  KO3
## 1 ENSMUSG00000000001 2022 2141 2313 2311 2415 2748
## 2 ENSMUSG00000000003    0    0    0    0    0    0
## 3 ENSMUSG00000000028  157  128   97   98  112  145
## 4 ENSMUSG00000000031    0    0    0    0    0    0
## 5 ENSMUSG00000000037    5    5    8    7   10    8
## 6 ENSMUSG00000000049    0    0    0    0    0    2
```

```
dim(counts)
```

```
## [1] 37991     7
```

```
"WT: normal cell, KO: knockout cell. The counts has 7 columns, the first is the gene name.
It contains the id that identifies the different genes, every raw is related to a single
gene in the mouse genome.The column WT1 represent the wise type relative to the expression
level of every normal gene in the mice genome inreplication 1, the second WT2
contains the expression level of the normal T regulatory cell  and same for WT3.
KO1 contains expression level of the T-reg cell where KAP1 is shut down and
cannot produce the gene. ame interpretation for replicate 2 and 3. "
```

```
## [1] "WT: normal cell, KO: knockout cell. The counts has 7 columns, the first is the gene name. \nIt c
```

```
"This experiment/data aim to identify the genes that behave differently. We check
whether they have different average levels in Ko t-reg cells compared to With
type cells. In this experiment, we will be able to determine which portions of
a gene in the mouse genome in the T-reg cell are probably regulated by kap1.
It is important because we need to know all the different relationships between
cells, especially for a cell associated with a disease, to identify the type of
gene responsible for a specific disease. Kap1 off implies the autoimmunity."
```

```
## [1] "This experiment/data aim to identify the genes that behave differently. We check \nwhether they
```

```
# Let's explore the data
"Quality check using edgeR"
```

```
## [1] "Quality check using edgeR"
```

```
"1) Load data in edgeR
In this data, rows are gene and columns are relative to samples"
```

```
## [1] "1) Load data in edgeR\nIn this data, rows are gene and columns are relative to samples"
```

```
data<-DGEList(counts=counts[,2:7], genes = counts[,1])
# counts receive the matrix of sample

class(data)
```

```
## [1] "DGEList"
## attr(,"package")
## [1] "edgeR"
```

```
data
```

```
## An object of class "DGEList"
## $counts
##    WT1  WT2  WT3  KO1  KO2  KO3
## 1 2022 2141 2313 2311 2415 2748
## 2    0    0    0    0    0    0
## 3  157  128   97   98  112  145
## 4    0    0    0    0    0    0
## 5    5    5    8    7   10    8
## 37986 more rows ...
##
## $samples
##     group lib.size norm.factors
## WT1     1 10103117            1
## WT2     1 10180281            1
## WT3     1 10300626            1
## KO1     1 10080881            1
## KO2     1  9954344            1
## KO3     1 10994745            1
##
## $genes
##                genes
## 1 ENSMUSG00000000001
## 2 ENSMUSG00000000003
## 3 ENSMUSG00000000028
## 4 ENSMUSG00000000031
## 5 ENSMUSG00000000037
## 37986 more rows ...
```

```
"a) Let's apply the PCA"
```

```
## [1] "a) Let's apply the PCA"
```

```
plotMDS(data)
"Interpretation:
We see in the x-axis that there are two groups, but this is not what we expected.
This is telling us that this data may have an issue. We just need to say there
is something wrong as an assumtion. "
```

```
## [1] "Interpretation:\nWe see in the x-axis that there are two groups, but this is not what we expecte
```

```
"Let's check the expression level of kap1 to see what happen,
(why do we decide to check the kap1 gene expression?)"
```

```
## [1] "Let's check the expression level of kap1 to see what happen, \n(why do we decide to check the ka
```

```
"Because the authors of the paper did the following experiment: they took two
mice strains and turned off the expression of kap1 in one of them. The group
with the kap1 expression turned off is called KO, the other WT. Since the mice
don't cluster according to the experimental condition, The suggestion is to check
whether the experiment was truly going as expected, meaning that the expression
of kap1 was lower in the KO group"
```

```
## [1] "Because the authors of the paper did the following experiment: they took two\nmice strains and
```

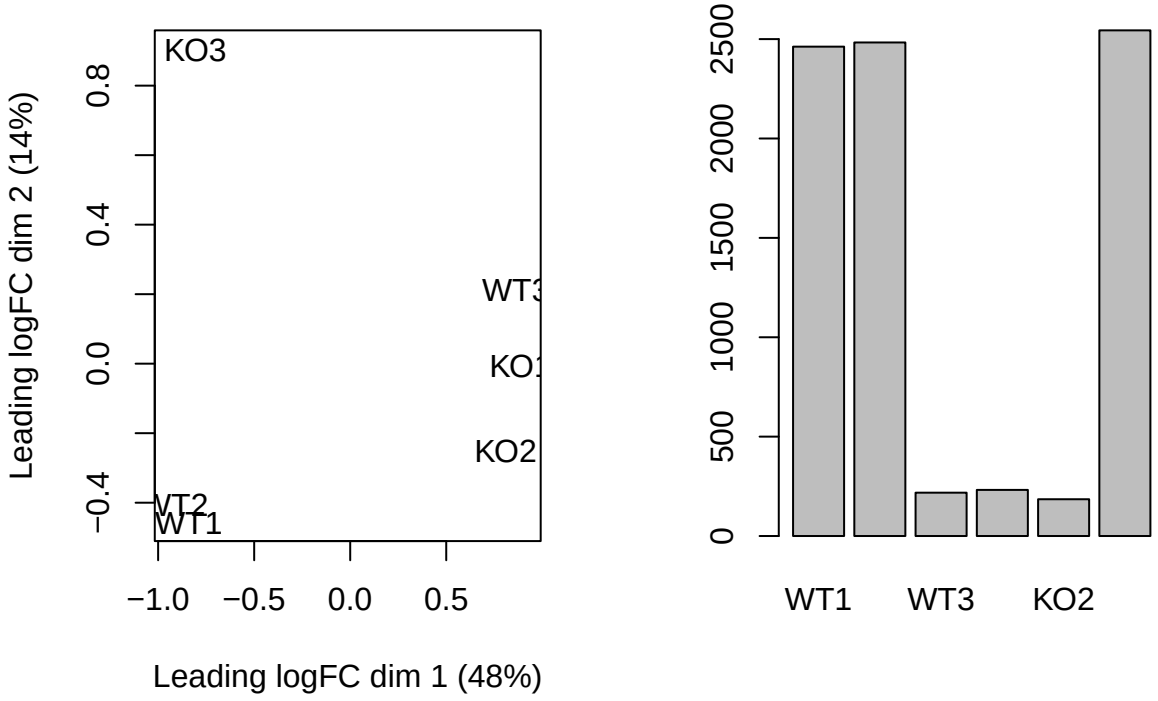
```
"How to get acces to kap1? kap1 is call Trim28, to retried it id gene, let's look
at mart_export.txt using grep Trim28 mart_export.tx , we got
ENSMUSG00000005566      Trim28"
```

```
## [1] "How to get acces to kap1? kap1 is call Trim28, to retried it id gene, let's look\nat mart_expor
```

```
"We have to retrieve the row that contain ENSMUSG00000005566"
```

```
## [1] "We have to retrieve the row that contain ENSMUSG00000005566"
```

```
su<-subset(counts,geneName =="ENSMUSG00000005566")
par(mfrow=c(1,2))
plotMDS(data)
barplot(as.numeric(su[,2:7]), names=colnames(counts[,2:7]))
```




```
"Interpretation:
WT3 is behaving similarly to KO1 and k02, and KO3 is behaving similarly to WT1
and WT2. In this case, one of the most likely interpretations is that one sample
was mislabeled. This could happen when we are working with multiple samples at time.
"
```

```
## [1] "Interpretation:\nWT3 is behaving similarly to KO1 and k02, and KO3 is behaving similarly to WT1
```

```
"Now, we want to see what will happen during the analysis and compare the result
to the one on the paper by doing a differential expression analysis. We will do
two types of analysis, in the first one, we will keep the label, and in the second one,
```

```
we will correct the label and then perform the analysis.  "
```

```
## [1] "Now, we want to see what will happen during the analysis and compare the result\nto the one on t
```

```
"For the analysis, we need to specify the group."
```

```
## [1] "For the analysis, we need to specify the group."
```

```
group<- factor(c("WT","WT","WT","KO","KO","KO"))
group
```

```
## [1] WT WT WT KO KO KO
## Levels: KO WT
```

```
group <-relevel(group,ref = "WT")
group
```

```
## [1] WT WT WT KO KO KO
## Levels: WT KO
```

```
data<-DGEList(counts=counts[,2:7], genes = counts[,1], group = group)
data
```

```
## An object of class "DGEList"
## $counts
##    WT1  WT2  WT3  KO1  KO2  KO3
## 1 2022 2141 2313 2311 2415 2748
## 2    0    0    0    0    0    0
## 3  157  128   97   98  112  145
## 4    0    0    0    0    0    0
## 5    5    5    8    7   10    8
## 37986 more rows ...
##
## $samples
##     group lib.size norm.factors
## WT1    WT 10103117            1
## WT2    WT 10180281            1
## WT3    WT 10300626            1
## KO1    KO 10080881            1
## KO2    KO  9954344            1
## KO3    KO 10994745            1
##
## $genes
##                genes
## 1 ENSMUSG00000000001
## 2 ENSMUSG00000000003
## 3 ENSMUSG00000000028
## 4 ENSMUSG00000000031
## 5 ENSMUSG00000000037
## 37986 more rows ...
```

```
"Let's follow the pipeline for analysis"
```

```
## [1] "Let's follow the pipeline for analysis"
```

```
"1) Nomalization factor"
```

```
## [1] "1) Nomalization factor"
```

```
data<-calcNormFactors(data)
data
```

```
## An object of class "DGEList"
## $counts
##    WT1  WT2  WT3  KO1  KO2  KO3
## 1 2022 2141 2313 2311 2415 2748
## 2    0    0    0    0    0    0
## 3  157  128   97   98  112  145
## 4    0    0    0    0    0    0
## 5    5    5    8    7   10    8
## 37986 more rows ...
##
## $samples
##     group lib.size norm.factors
## WT1    WT 10103117    0.9970501
## WT2    WT 10180281    1.0049992
## WT3    WT 10300626    0.9985815
## KO1    KO 10080881    0.9944356
## KO2    KO  9954344    0.9991544
## KO3    KO 10994745    1.0058299
##
## $genes
##                genes
## 1 ENSMUSG00000000001
## 2 ENSMUSG00000000003
## 3 ENSMUSG00000000028
## 4 ENSMUSG00000000031
## 5 ENSMUSG00000000037
## 37986 more rows ...
```

```
" Sample show the nowmalization factors that was calculated"
```

```
## [1] " Sample show the nowmalization factors that was calculated"
```

```
"In the next step, we are going to apply the generalized linear model to perform
the statistical analysis. Usually, the test can be applied. We need to estimate
the dispersion between the average of the whole data. It is done with a specific
function. Strong signals have high dispersion and lower movement have low signals.
The first step is to use the model matrix to calculate the expression."
```

```
## [1] "In the next step, we are going to apply the generalized linear model to perform\nthe statistica
```

```
design <-model.matrix(~group)
design
```

```
##   (Intercept) groupKO
## 1           1       0
## 2           1       0
## 3           1       0
## 4           1       1
## 5           1       1
## 6           1       1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
```

```
## attr(,"contrasts")$group
## [1] "contr.treatment"
```

```
" 1 correspond to the groupe of ko"
```

```
## [1] " 1 correspond to the groupe of ko"
```

```
data<-estimateDisp(data,design)
data
```

```
## An object of class "DGEList"
## $counts
##    WT1  WT2  WT3  KO1  KO2  KO3
## 1 2022 2141 2313 2311 2415 2748
## 2    0    0    0    0    0    0
## 3  157  128   97   98  112  145
## 4    0    0    0    0    0    0
## 5    5    5    8    7   10    8
## 37986 more rows ...
##
## $samples
##     group lib.size norm.factors
## WT1    WT 10103117    0.9970501
## WT2    WT 10180281    1.0049992
## WT3    WT 10300626    0.9985815
## KO1    KO 10080881    0.9944356
## KO2    KO  9954344    0.9991544
## KO3    KO 10994745    1.0058299
##
## $genes
##                genes
## 1 ENSMUSG00000000001
## 2 ENSMUSG00000000003
## 3 ENSMUSG00000000028
## 4 ENSMUSG00000000031
## 5 ENSMUSG00000000037
## 37986 more rows ...
##
## $design
##   (Intercept) groupKO
## 1           1       0
## 2           1       0
## 3           1       0
## 4           1       1
## 5           1       1
## 6           1       1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
##
##
## $common.dispersion
## [1] 0.01290625
##
```

```
## $trended.dispersion
## [1] 0.009439764 0.081151682 0.022322250 0.081151682 0.070753443
## 37986 more elements ...
##
## $tagwise.dispersion
## [1] 0.005889131 0.081151682 0.027635319 0.081151682 0.013151598
## 37986 more elements ...
##
## $AveLogCPM
## [1]  7.8222056 -2.3603662  3.6024009 -2.3603662 -0.1638938
## 37986 more elements ...
##
## $trend.method
## [1] "locfit"
##
## $prior.df
## [1] 5.019259
##
## $prior.n
## [1] 1.254815
##
## $span
## [1] 0.2926343
```

```
"Dispersion is information that will be used for linear modeling we are going
to do "
```

```
## [1] "Dispersion is information that will be used for linear modeling we are going \nto do "
```

```
"Let's fit the linear model to our data"
```

```
## [1] "Let's fit the linear model to our data"
```

```
fit<-glmFit(data)
fit
```

```
## An object of class "DGEGLM"
## $coefficients
##   (Intercept)    groupKO
## 1   -8.460634  0.1284183
## 2  -18.224180  0.0000000
## 3  -11.288474 -0.0927059
## 4  -18.224180  0.0000000
## 5  -14.325455  0.3092921
## 37986 more rows ...
##
## $fitted.values
##          WT1         WT2         WT3         KO1         KO2         KO3
## 1 2131.77092 2165.178273 2176.783797 2412.224344 2393.248421 2661.045287
## 2    0.00000    0.000000    0.000000    0.000000    0.000000    0.000000
## 3  125.95863  127.932548  128.618276  114.242718  113.344020  126.026855
## 4    0.00000    0.000000    0.000000    0.000000    0.000000    0.000000
## 5    5.92672    6.019599    6.051865    8.080185    8.016622    8.913657
## 37986 more rows ...
##
## $deviance
##        1        2        3        4        5
```

```
## 1.511284 0.000000 4.694782 0.000000 1.469003
## 37986 more elements ...
##
## $method
## [1] "oneway"
##
## $counts
##    WT1  WT2  WT3  KO1  KO2  KO3
## 1 2022 2141 2313 2311 2415 2748
## 2    0    0    0    0    0    0
## 3  157  128   97   98  112  145
## 4    0    0    0    0    0    0
## 5    5    5    8    7   10    8
## 37986 more rows ...
##
## $unshrunk.coefficients
##     (Intercept)     groupKO
## 1 -8.460692e+00  0.12842524
## 2 -1.000000e+08  0.00000000
## 3 -1.128945e+01 -0.09279921
## 4 -1.000000e+08  0.00000000
## 5 -1.434593e+01  0.31477281
## 37986 more rows ...
##
## $df.residual
## [1] 4 4 4 4 4
## 37986 more elements ...
##
## $design
##   (Intercept) groupKO
## 1           1       0
## 2           1       0
## 3           1       0
## 4           1       1
## 5           1       1
## 6           1       1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
##
##
## $offset
##         [,1]     [,2]    [,3]     [,4]     [,5]     [,6]
## [1,] 16.1254 16.14095 16.1463 16.12057 16.11267 16.21874
## attr(,"class")
## [1] "CompressedMatrix"
## attr(,"Dims")
## [1] 5 6
## attr(,"repeat.row")
## [1] TRUE
## attr(,"repeat.col")
## [1] FALSE
```

```
## 37986 more rows ...
##
## $dispersion
## [1] 0.005889131 0.081151682 0.027635319 0.081151682 0.013151598
## 37986 more elements ...
##
## $prior.count
## [1] 0.125
##
## $samples
##     group lib.size norm.factors
## WT1    WT 10103117    0.9970501
## WT2    WT 10180281    1.0049992
## WT3    WT 10300626    0.9985815
## KO1    KO 10080881    0.9944356
## KO2    KO  9954344    0.9991544
## KO3    KO 10994745    1.0058299
##
## $genes
##                genes
## 1 ENSMUSG00000000001
## 2 ENSMUSG00000000003
## 3 ENSMUSG00000000028
## 4 ENSMUSG00000000031
## 5 ENSMUSG00000000037
## 37986 more rows ...
##
## $prior.df
## [1] 5.019259
##
## $AveLogCPM
## [1]  7.8222056 -2.3603662  3.6024009 -2.3603662 -0.1638938
## 37986 more elements ...
```

```
"In every step, information is going to be added. Fitted. values are the normalized
expression"
```

```
## [1] "In every step, information is going to be added. Fitted. values are the normalized \nexpression
```

```
"Let's perform the differential expression analysis using glmlrt to perform
gene-wise statistical test"
```

```
## [1] "Let's perform the differential expression analysis using glmlrt to perform \ngene-wise statistic
```

```
lrt<-glmLRT(fit)
lrt
```

```
## An object of class "DGELRT"
## $coefficients
##   (Intercept)    groupKO
## 1   -8.460634  0.1284183
## 2  -18.224180  0.0000000
## 3  -11.288474 -0.0927059
## 4  -18.224180  0.0000000
## 5  -14.325455  0.3092921
## 37986 more rows ...
```

```
##
## $fitted.values
##          WT1         WT2         WT3         KO1         KO2         KO3
## 1 2131.77092 2165.178273 2176.783797 2412.224344 2393.248421 2661.045287
## 2    0.00000    0.000000    0.000000    0.000000    0.000000    0.000000
## 3  125.95863  127.932548  128.618276  114.242718  113.344020  126.026855
## 4    0.00000    0.000000    0.000000    0.000000    0.000000    0.000000
## 5    5.92672    6.019599    6.051865    8.080185    8.016622    8.913657
## 37986 more rows ...
##
## $deviance
##        1        2        3        4        5
## 1.511284 0.000000 4.694782 0.000000 1.469003
## 37986 more elements ...
##
## $method
## [1] "oneway"
##
## $unshrunk.coefficients
##     (Intercept)     groupKO
## 1 -8.460692e+00  0.12842524
## 2 -1.000000e+08  0.00000000
## 3 -1.128945e+01 -0.09279921
## 4 -1.000000e+08  0.00000000
## 5 -1.434593e+01  0.31477281
## 37986 more rows ...
##
## $df.residual
## [1] 4 4 4 4 4
## 37986 more elements ...
##
## $design
##   (Intercept) groupKO
## 1           1       0
## 2           1       0
## 3           1       0
## 4           1       1
## 5           1       1
## 6           1       1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
##
##
## $offset
##         [,1]     [,2]    [,3]     [,4]     [,5]     [,6]
## [1,] 16.1254 16.14095 16.1463 16.12057 16.11267 16.21874
## attr(,"class")
## [1] "CompressedMatrix"
## attr(,"Dims")
## [1] 5 6
## attr(,"repeat.row")
```

```
## [1] TRUE
## attr(,"repeat.col")
## [1] FALSE
## 37986 more rows ...
##
## $dispersion
## [1] 0.005889131 0.081151682 0.027635319 0.081151682 0.013151598
## 37986 more elements ...
##
## $prior.count
## [1] 0.125
##
## $samples
##     group lib.size norm.factors
## WT1    WT 10103117    0.9970501
## WT2    WT 10180281    1.0049992
## WT3    WT 10300626    0.9985815
## KO1    KO 10080881    0.9944356
## KO2    KO  9954344    0.9991544
## KO3    KO 10994745    1.0058299
##
## $genes
##                genes
## 1 ENSMUSG00000000001
## 2 ENSMUSG00000000003
## 3 ENSMUSG00000000028
## 4 ENSMUSG00000000031
## 5 ENSMUSG00000000037
## 37986 more rows ...
##
## $prior.df
## [1] 5.019259
##
## $AveLogCPM
## [1]  7.8222056 -2.3603662  3.6024009 -2.3603662 -0.1638938
## 37986 more elements ...
##
## $table
##        logFC     logCPM        LR     PValue
## 1  0.1852684  7.8222056 3.9111858 0.04796574
## 2  0.0000000 -2.3603662 0.0000000 1.00000000
## 3 -0.1337463  3.6024009 0.3608294 0.54804597
## 4  0.0000000 -2.3603662 0.0000000 1.00000000
## 5  0.4462141 -0.1638938 0.9603861 0.32708963
## 37986 more rows ...
##
## $comparison
## [1] "groupKO"
##
## $df.test
## [1] 1 1 1 1 1
## 37986 more elements ...
```

```
"lrt contains information about statistical analysis. The table reports the logFC,
logCPM, LR, and p-value. Let's use the function toptags to extract the most
expressed genes"
```

```
## [1] "lrt contains information about statistical analysis. The table reports the logFC, \nlogCPM, LR,
```

```
res<- topTags(lrt, n=nrow(counts))
head(res)
```

```
## Coefficient:  groupKO
##                      genes      logFC   logCPM        LR       PValue
## 34350 ENSMUSG00000090015  5.1544613 2.001929 157.80822 3.408363e-36
## 29563 ENSMUSG00000084796 -1.2122359 4.797591  55.83054 7.899499e-14
## 35139 ENSMUSG00000090877  3.1356989 2.256526  48.72510 2.944735e-12
## 15418 ENSMUSG00000052146  0.6709838 4.390101  24.62596 6.960751e-07
## 9998  ENSMUSG00000035202  1.0240864 8.623115  24.01969 9.535542e-07
## 37287 ENSMUSG00000093077  0.7627732 7.717906  21.54891 3.449183e-06
##                 FDR
## 34350 1.294871e-31
## 29563 1.500549e-09
## 35139 3.729115e-08
## 15418 6.611147e-03
## 9998  7.245296e-03
## 37287 2.183965e-02
```

```
"How many genes are significantly differentially expressed?"
```

```
## [1] "How many genes are significantly differentially expressed?"
```

```
sum(res$table$FDR<0.05)
```

```
## [1] 7
```

```
res_tab<-as.data.frame(res)
res_tab[(res$table$FDR<0.05),]
```

```
##                      genes      logFC   logCPM        LR       PValue
## 34350 ENSMUSG00000090015  5.1544613 2.001929 157.80822 3.408363e-36
## 29563 ENSMUSG00000084796 -1.2122359 4.797591  55.83054 7.899499e-14
## 35139 ENSMUSG00000090877  3.1356989 2.256526  48.72510 2.944735e-12
## 15418 ENSMUSG00000052146  0.6709838 4.390101  24.62596 6.960751e-07
## 9998  ENSMUSG00000035202  1.0240864 8.623115  24.01969 9.535542e-07
## 37287 ENSMUSG00000093077  0.7627732 7.717906  21.54891 3.449183e-06
## 31025 ENSMUSG00000086324  1.2955337 6.981972  21.08096 4.402813e-06
##                 FDR
## 34350 1.294871e-31
## 29563 1.500549e-09
## 35139 3.729115e-08
## 15418 6.611147e-03
## 9998  7.245296e-03
## 37287 2.183965e-02
## 31025 2.389533e-02
```

```
" This result is very different to what was published"
```

```
## [1] " This result is very different to what was published"
```

```
"Let's look at the value of kap1"
```

```
## [1] "Let's look at the value of kap1"
```

```
subset(res_tab, genes=="ENSMUSG00000005566")
```

```
##                   genes      logFC   logCPM        LR   PValue FDR
## 870 ENSMUSG00000005566 -0.9029714 7.025737 0.8895695 0.345594   1
```

```
"FDR is 1, so kap1 is not differentialy express."
```

```
## [1] "FDR is 1, so kap1 is not differentialy express."
```

```
"Let's do the same analysis by  swapping the label"
```

```
## [1] "Let's do the same analysis by  swapping the label"
```

```
"Let's change the group"
```

```
## [1] "Let's change the group"
```

```
colnames(counts)<-c("geneNames","WT1","WT2", 'KO3',"KO1","KO2","WT3")
head(counts)
```

```
##            geneNames  WT1  WT2  KO3  KO1  KO2  WT3
## 1 ENSMUSG00000000001 2022 2141 2313 2311 2415 2748
## 2 ENSMUSG00000000003    0    0    0    0    0    0
## 3 ENSMUSG00000000028  157  128   97   98  112  145
## 4 ENSMUSG00000000031    0    0    0    0    0    0
## 5 ENSMUSG00000000037    5    5    8    7   10    8
## 6 ENSMUSG00000000049    0    0    0    0    0    2
```

```
group<-factor(c("WT","WT", 'KO',"KO","KO","WT"))
group<-relevel(group, ref="WT")
group
```

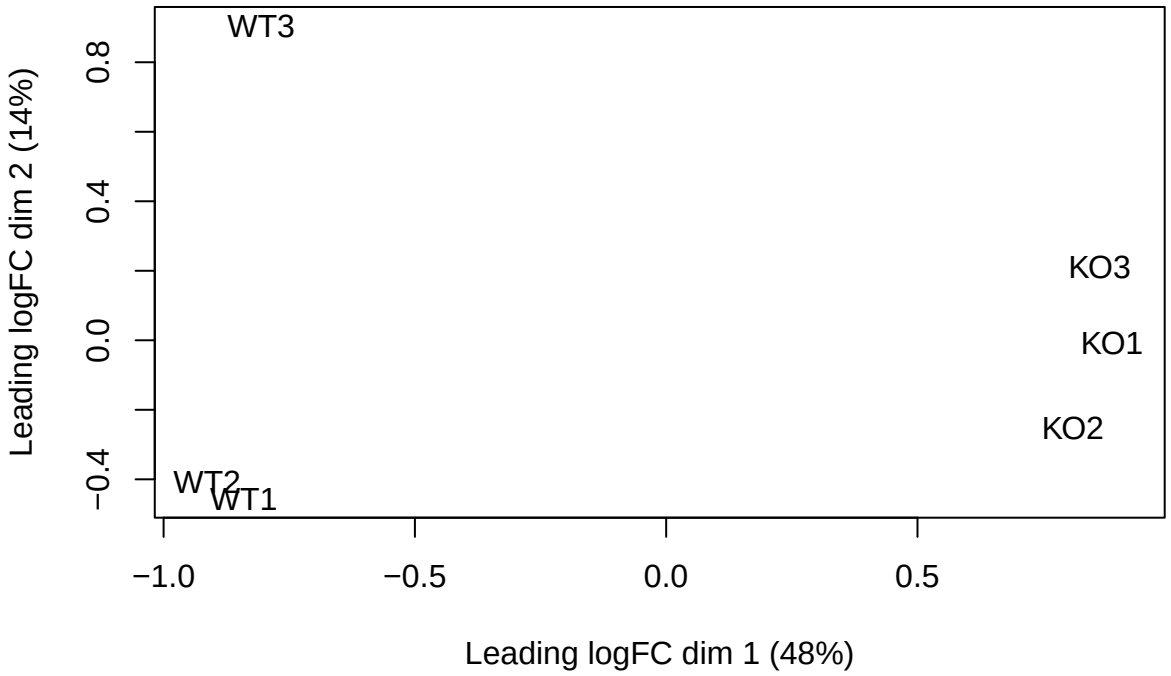
```
## [1] WT WT KO KO KO WT
## Levels: WT KO
```

```
data<-DGEList(counts = counts[,2:7],genes =counts[,1], group = group)
data
```

```
## An object of class "DGEList"
## $counts
##    WT1  WT2  KO3  KO1  KO2  WT3
## 1 2022 2141 2313 2311 2415 2748
## 2    0    0    0    0    0    0
## 3  157  128   97   98  112  145
## 4    0    0    0    0    0    0
## 5    5    5    8    7   10    8
## 37986 more rows ...
##
## $samples
##     group lib.size norm.factors
## WT1    WT 10103117            1
## WT2    WT 10180281            1
## KO3    KO 10300626            1
## KO1    KO 10080881            1
## KO2    KO  9954344            1
```

```
## WT3    WT 10994745            1
##
## $genes
##                 genes
## 1 ENSMUSG00000000001
## 2 ENSMUSG00000000003
## 3 ENSMUSG00000000028
## 4 ENSMUSG00000000031
## 5 ENSMUSG00000000037
## 37986 more rows ...
```

```
par(mfrow=c(1,1))
plotMDS(data)
```




```
design<-model.matrix(~group)
data<-calcNormFactors(data)
data<-estimateDisp(data,design = design)
fit<-glmFit(data)
lrt<-glmLRT(fit)
res2<-topTags(lrt, n=nrow(counts))
head(res2)
```

```
## Coefficient:  groupKO
##                    genes     logFC   logCPM        LR        PValue
## 5260  ENSMUSG00000025885 10.330094 6.489680 1531.8848  0.000000e+00
## 15325 ENSMUSG00000051726  5.339843 6.756629 2102.7679  0.000000e+00
## 870   ENSMUSG00000005566 -3.509022 7.027317 1533.0188  0.000000e+00
## 7928  ENSMUSG00000030553  3.108967 7.105370 1402.2383 6.855433e-307
## 341   ENSMUSG00000001918 -2.212498 8.034318  999.1197 2.790071e-219
## 16665 ENSMUSG00000057409  1.851867 7.642062  710.2856 1.734042e-156
##                  FDR
## 5260   0.000000e+00
## 15325  0.000000e+00
```

```
## 870    0.000000e+00
## 7928  6.511119e-303
## 341   2.119952e-215
## 16665 1.097967e-152
```

```
"We get the very significant value"
```

```
## [1] "We get the very significant value"
```

```
res.tab2 <- as.data.frame(res2)

summary(decideTests(lrt))
```

```
##        groupKO
## Down       415
## NotSig   37055
## Up         521
```

```
"Let's check which gene is ENSMUSG00000025885. We use the following command :
grep ENSMUSG00000025885 mart_export.txt, the result is Myo5b which is not present
in figure c published by the paper. grep ENSMUSG00000051726 mart_export.txt,
the output is Kcnf1 which is one of the most expressed genes with kap1 in figure
c on the paper.
so they did the correct analysis, but the data was not mislabeled. During the
study of this kind of job, we need to be very critical and suspicious of public data."
```

```
## [1] "Let's check which gene is ENSMUSG00000025885. We use the following command :\ngrep ENSMUSG00000
```