HIV RNASeq

Guannan Shen

November 21, 2018

Contents


```
###################### Set up workspace
rm(list = ls())
library(edgeR)
library(EDASeq)
library(DESeq2)
library(knitr)
library(tidyverse)
library(magrittr)
library(stats)
library("BiocParallel")

options(stringsAsFactors = F)
options(dplyr.width = Inf)
getwd()
```

```
## [1] "/home/guanshim/Documents/gitlab/Cario_RNASeq_Microbiom_Inte/DataRaw"
```

```
## not in function
"%nin%" <- Negate("%in%")

# ######## clean memory ###################### rm(list =
# ls()) gc()
```

1 HIV patients, health control RNASeq data Normalization and QC (quality control)

Start with the counts table, and compare different normalization methods. DESeq2, TPM...

DESeq2 is inter-sample comparison normalization method, assuming the majority of the genes are not differentially expressed.

The top commonly used methods are DESeq (median-of-ratios) and TMM (Trimmed Mean of M values)-edgeR. DESeq and TMM-edgeR were reported to have overall better performance, based on the false positive rate and detection power.

```
# import unnormalized counts table
cnts.raw <- read.delim("All_Sample_geneCounts_raw_counts.txt",
    header = TRUE, sep = "\t")
head(cnts.raw)
```

```
##              Gene_ID   Symbol Length C138 C178 C255 C278 C361 C404 C493
## 1 ENSG00000000003.14   TSPAN6   4535  360  485 1862 2225 2550 2559 1158
```

```
## 2  ENSG00000000005.5     TNMD   1610    2    8   20   21   18   39   17
## 3 ENSG00000000419.12     DPM1   1207   34   45   95  259  383  247   98
## 4 ENSG00000000457.13    SCYL3   6883   14   16   57  102  114  141   53
## 5 ENSG00000000460.16 C1orf112   5967    3    6   12   21   15   25    9
## 6 ENSG00000000938.12      FGR   3474   17   12  113   35   71  186  104
##   C582 C708 C716 C914 C947 C972 H124 H132 H154 H188 H217 H286 H307 H323
## 1 2612 1592 6849 2481  786 3379 1596 1227 1648 1740 1116  564 1591 1170
## 2   50   11   84   24    5   36   43   14   20    5    5    9   23   12
## 3  233  273 1196  398   52 1086   79   75   80  324  119   63   64   99
## 4   95   80  235  142   20  156   46   61   46   86   50   22   51   60
## 5   28   23   47   28   11   52    9   21    6   10   10    3   14   21
## 6   76   55  168   54   44   93   58  159   99  396   85   43   99  306
##   H391 H428 H594 H622 H648 H683 H819 H825 H839 H965 H998
## 1 1060  216 6647 1112  921  274  719 1396  954  194 7622
## 2   20    2   34   17   25   11   10    7   18    4   54
## 3   64    7 1017  171   41   15   35   80   18   11 1573
## 4   47    9  249   44   69    8   35   60   39   16  361
## 5    4    3   44    8    6    9   10    7   20    0  107
## 6   93  232  342  104  131   54  292   31  366   36  913
```

```
ncol(cnts.raw)
```

```
## [1] 35
```

```
## explore the gene Symbol 'unique'
anyNA(cnts.raw$Symbol)
```

```
## [1] FALSE
```

```
length(unique(cnts.raw$Symbol)) == nrow(cnts.raw)  # there is duplication
```

```
## [1] FALSE
```

```
gene_sym_sum <- table(cnts.raw$Symbol)
typeof(gene_sym_sum)
```

```
## [1] "integer"
```

```
sum(gene_sym_sum[gene_sym_sum >= 2])  #  1035 >= 2 symbols
```

```
## [1] 1035
```

```
range(gene_sym_sum)  # Y_RNA has been used for 490 times
```

```
## [1]   1 490
```

```
# check unique Gene_ID
length(unique(cnts.raw$Gene_ID)) == nrow(cnts.raw)  # Gene_ID is unique
```

```
## [1] TRUE
```

```
# generate the common counts table
cnts <- cnts.raw %>% dplyr::select(-c(Symbol, Length)) %>% tibble::column_to_rownames("Gene_ID")
head(cnts)
```

```
##                    C138 C178 C255 C278 C361 C404 C493 C582 C708 C716 C914
## ENSG00000000003.14  360  485 1862 2225 2550 2559 1158 2612 1592 6849 2481
## ENSG00000000005.5     2    8   20   21   18   39   17   50   11   84   24
## ENSG00000000419.12   34   45   95  259  383  247   98  233  273 1196  398
## ENSG00000000457.13   14   16   57  102  114  141   53   95   80  235  142
```

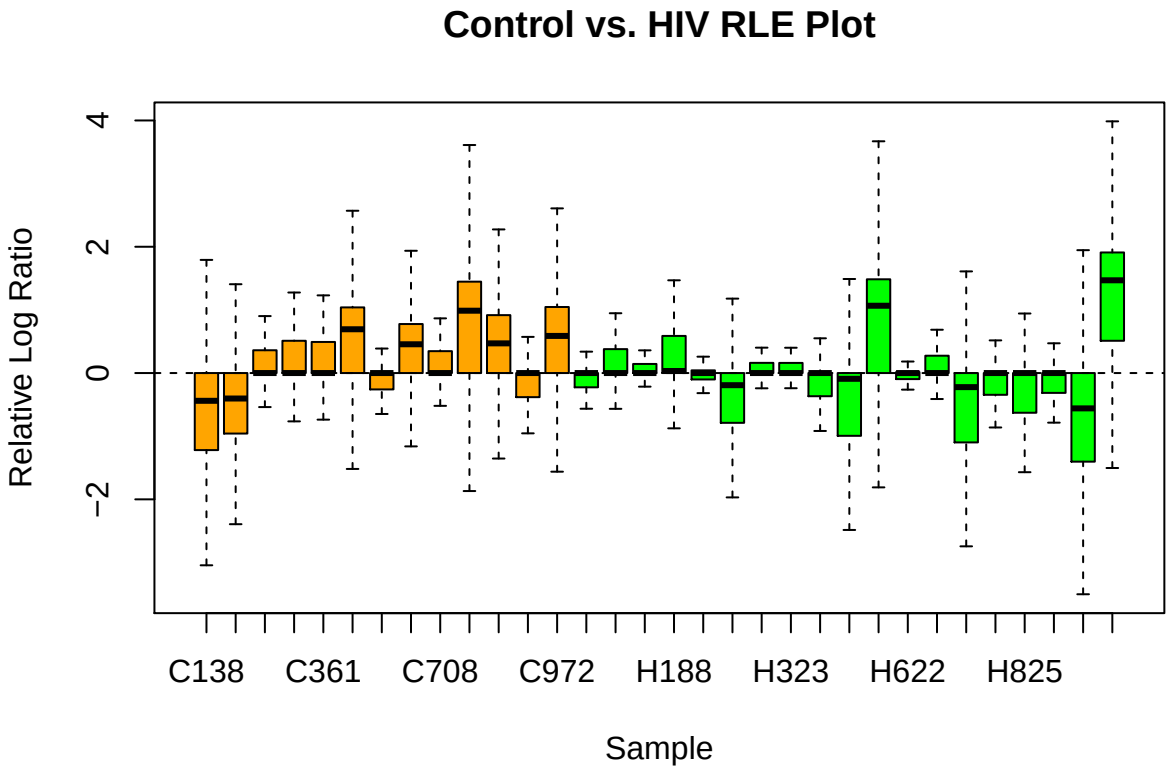
```
## ENSG00000000460.16    3    6   12   21   15   25    9   28   23   47   28
## ENSG00000000938.12   17   12  113   35   71  186  104   76   55  168   54
##                    C947 C972 H124 H132 H154 H188 H217 H286 H307 H323 H391
## ENSG00000000003.14  786 3379 1596 1227 1648 1740 1116  564 1591 1170 1060
## ENSG00000000005.5     5   36   43   14   20    5    5    9   23   12   20
## ENSG00000000419.12   52 1086   79   75   80  324  119   63   64   99   64
## ENSG00000000457.13   20  156   46   61   46   86   50   22   51   60   47
## ENSG00000000460.16   11   52    9   21    6   10   10    3   14   21    4
## ENSG00000000938.12   44   93   58  159   99  396   85   43   99  306   93
##                    H428 H594 H622 H648 H683 H819 H825 H839 H965 H998
## ENSG00000000003.14  216 6647 1112  921  274  719 1396  954  194 7622
## ENSG00000000005.5     2   34   17   25   11   10    7   18    4   54
## ENSG00000000419.12    7 1017  171   41   15   35   80   18   11 1573
## ENSG00000000457.13    9  249   44   69    8   35   60   39   16  361
## ENSG00000000460.16    3   44    8    6    9   10    7   20    0  107
## ENSG00000000938.12  232  342  104  131   54  292   31  366   36  913
```

```
dim(cnts)
```

```
## [1] 43297    32
```

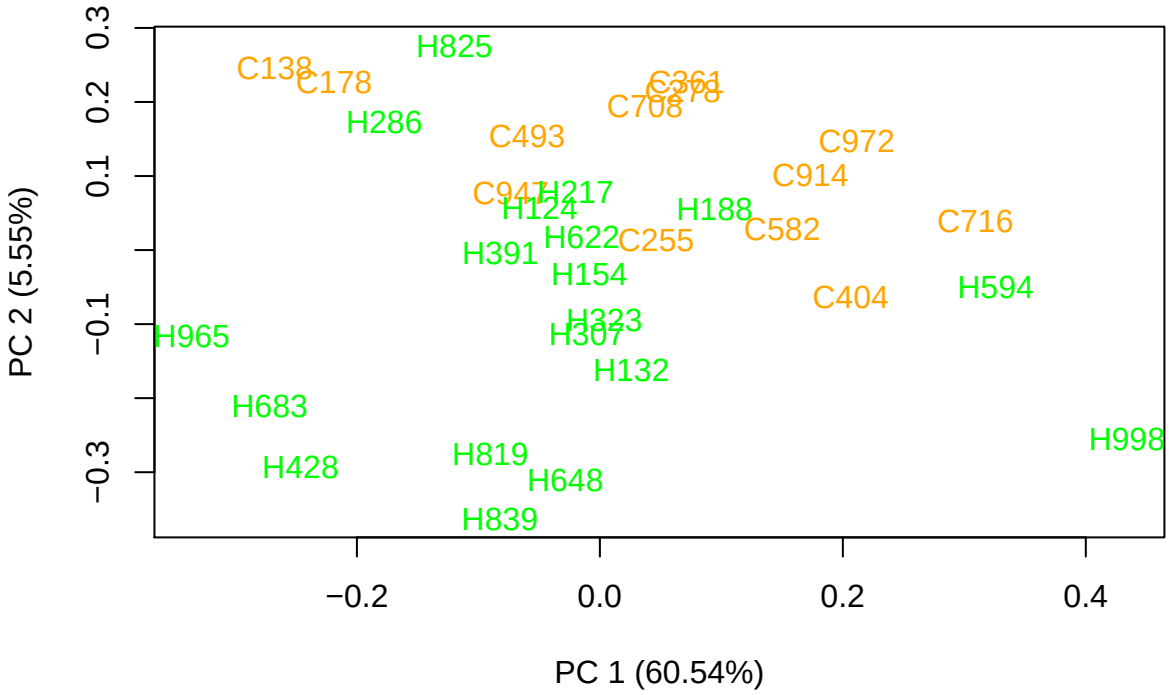
```
rna.pid <- colnames(cnts)
# now we have the common counts table pheno
ctrl.id <- colnames(cnts)[1:13]
hiv.id <- colnames(cnts)[14:32]
## from dim() we know there are 32 samples
pheno <- data.frame(pid = rna.pid, txt = as.factor(c(rep("Control",
    13), rep("HIV", 19))))
pheno$txt %<>% relevel("Control")
## This is an important step so that DESeq will know to treat
## the control group as the reference


## without filering using the function from EDASeq using group
## here
set <- newSeqExpressionSet(as.matrix(round(cnts)), phenoData = data.frame(group = as.factor(pheno$txt),
    row.names = colnames(cnts)))
## general QC images ## plotRLE from EDASeq
plotRLE(set, outline = FALSE, col = c(rep("Orange", 13), rep("Green",
    19)), main = "Control vs. HIV RLE Plot", xlab = "Sample",
    ylab = "Relative Log Ratio")
```




```
## PCA plot to show clustering plotPCA from EDASeq package

plotPCA(set, col = c(rep("Orange", 13), rep("Green", 19)))
```




```
## filter the raw data and check dim
cnts_f <- cnts[rowSums(cnts) >= (5 * ncol(cnts)), ]
## should end up around 15 - 20K genes
ngenes <- nrow(cnts_f)
```

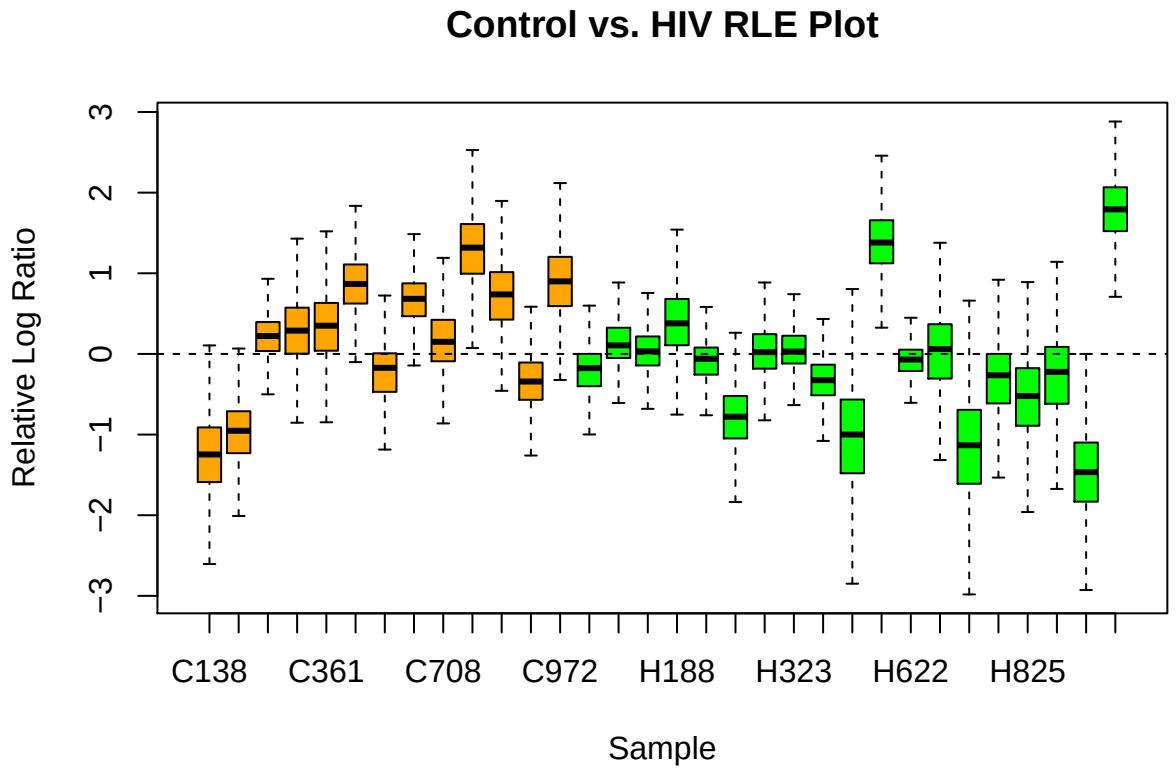
```r
paste("The number of remaining genes: ", ngenes, sep = "")
```

```
## [1] "The number of remaining genes: 19890"
```

```r
# all integer in cnts_f
typeof(as.matrix(cnts_f))
```

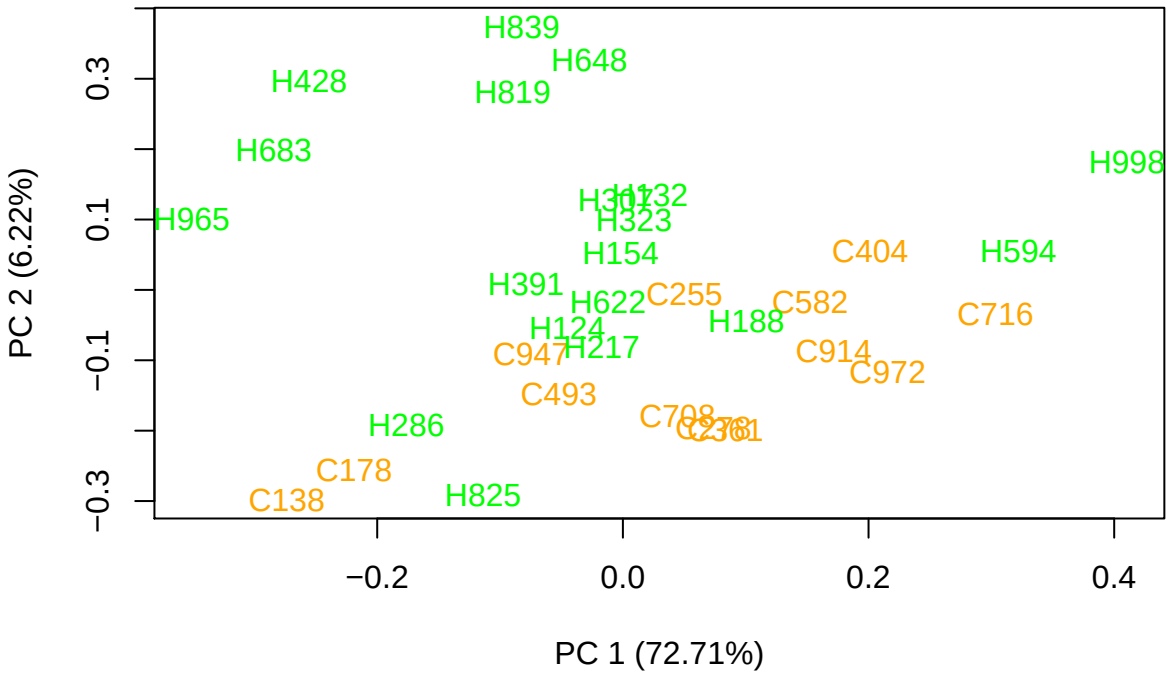
```
## [1] "integer"
```

```r
# using the function from EDASeq
set <- newSeqExpressionSet(as.matrix(round(cnts_f)), phenoData = data.frame(group = as.factor(pheno$txt)
    row.names = colnames(cnts_f)))
## general QC images ## plotRLE from EDASeq
plotRLE(set, outline = FALSE, col = c(rep("Orange", 13), rep("Green",
    19)), main = "Control vs. HIV RLE Plot", xlab = "Sample",
    ylab = "Relative Log Ratio")
```




```r
## PCA plot to show clustering plotPCA from EDASeq package

plotPCA(set, col = c(rep("Orange", 13), rep("Green", 19)))
```




```
# counts from EDASeq (DESeq2) pData is phenoData from Biobase
countData <- counts(set)  #Matrix with transcripts IDs as rows and sample IDs as columns
colData <- pData(set)  #Vector of type list in which the group column is the treat/control identfier, a

# Run DESeq function using above objects
print("this is a single factor: group, and 2 groups design (2 levels)")
```

```
## [1] "this is a single factor: group, and 2 groups design (2 levels)"
```

```
## now using deseq2
dds <- DESeqDataSetFromMatrix(countData = counts(set), colData = pData(set),
    design = ~group)

## deseq normalization and DE analysis
register(MulticoreParam(6))
dds <- DESeq(dds)
plotMA(dds)

res <- results(dds)
summary(res)
```

```
##
## out of 19890 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 3705, 19%
## LFC < 0 (down)     : 2880, 14%
## outliers [1]       : 0, 0%
## low counts [2]     : 0, 0%
## (mean count < 0)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
## res is the result
resOrdered <- res[order(res$pvalue), ]
```

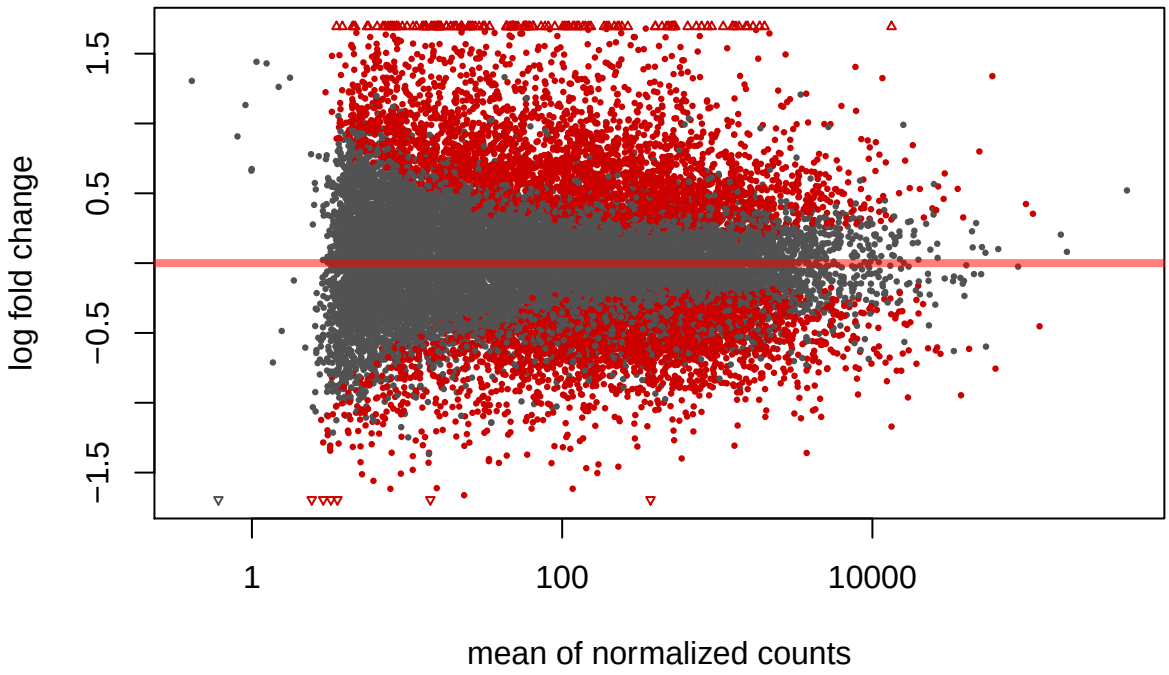
```r
head(resOrdered, 10)
```

```
## log2 fold change (MLE): group HIV vs Control
## Wald test p-value: group HIV vs Control
## DataFrame with 10 rows and 6 columns
##                            baseMean   log2FoldChange             lfcSE
##                           <numeric>        <numeric>         <numeric>
## ENSG00000163508.12 148.172487119719 3.50192736966002 0.265108926129653
## ENSG00000100450.12 58.9165102012857 3.90022933499764 0.317886006212177
## ENSG00000139187.9  153.151373167996 2.46990130046568  0.23015771230705
## ENSG00000105374.9  81.4074489628874 3.00419398588898 0.304679121745529
## ENSG00000153563.15 128.367367755952 2.18568001019696 0.225749044200061
## ENSG00000197057.8  22.5962808547272 3.18047962895702  0.32894563239503
## ENSG00000111801.15 201.904300380782 1.56853411161155 0.163155324139545
## ENSG00000140853.15 1529.20677515213 1.81911417622931 0.198685231273562
## ENSG00000271503.5  1091.49418438573 2.16985711515286 0.238080799765497
## ENSG00000168394.10 97.4956664659528  1.5446206860865 0.169498468195027
##                                stat               pvalue
##                           <numeric>            <numeric>
## ENSG00000163508.12 13.2093906485343 7.74488443955435e-40
## ENSG00000100450.12  12.269270300607 1.32441527920711e-34
## ENSG00000139187.9  10.7313427636551 7.25344237645123e-27
## ENSG00000105374.9  9.86018985704612 6.19321739671832e-23
## ENSG00000153563.15 9.68190150236023 3.59957833466214e-22
## ENSG00000197057.8  9.66870909882637 4.09510790386617e-22
## ENSG00000111801.15 9.61374763516757  6.9955430381441e-22
## ENSG00000140853.15 9.15575941185404 5.39768101836225e-20
## ENSG00000271503.5  9.11395256270188 7.94340448589183e-20
## ENSG00000168394.10 9.11288876256532 8.02170418976026e-20
##                                 padj
##                            <numeric>
## ENSG00000163508.12 1.54045751502736e-35
## ENSG00000100450.12 1.31713099517147e-30
## ENSG00000139187.9  4.80903229558716e-23
## ENSG00000105374.9  3.07957735051818e-19
## ENSG00000153563.15 1.35752827013163e-18
## ENSG00000197057.8  1.35752827013163e-18
## ENSG00000111801.15 1.98773358612409e-18
## ENSG00000140853.15 1.34199844319031e-16
## ENSG00000271503.5  1.59551696334332e-16
## ENSG00000168394.10 1.59551696334332e-16
```

```r
##
sum(res$padj < 0.1, na.rm = TRUE)
```

```
## [1] 6585
```

```r
# plots
plotMA(res)
```



2 Reference

1. A comparison of per sample global scaling and per gene normalization methods for differential expression analysis of RNA-seq data, PLos One. 2017.