

Homework 6

Roberto Villegas-Diaz

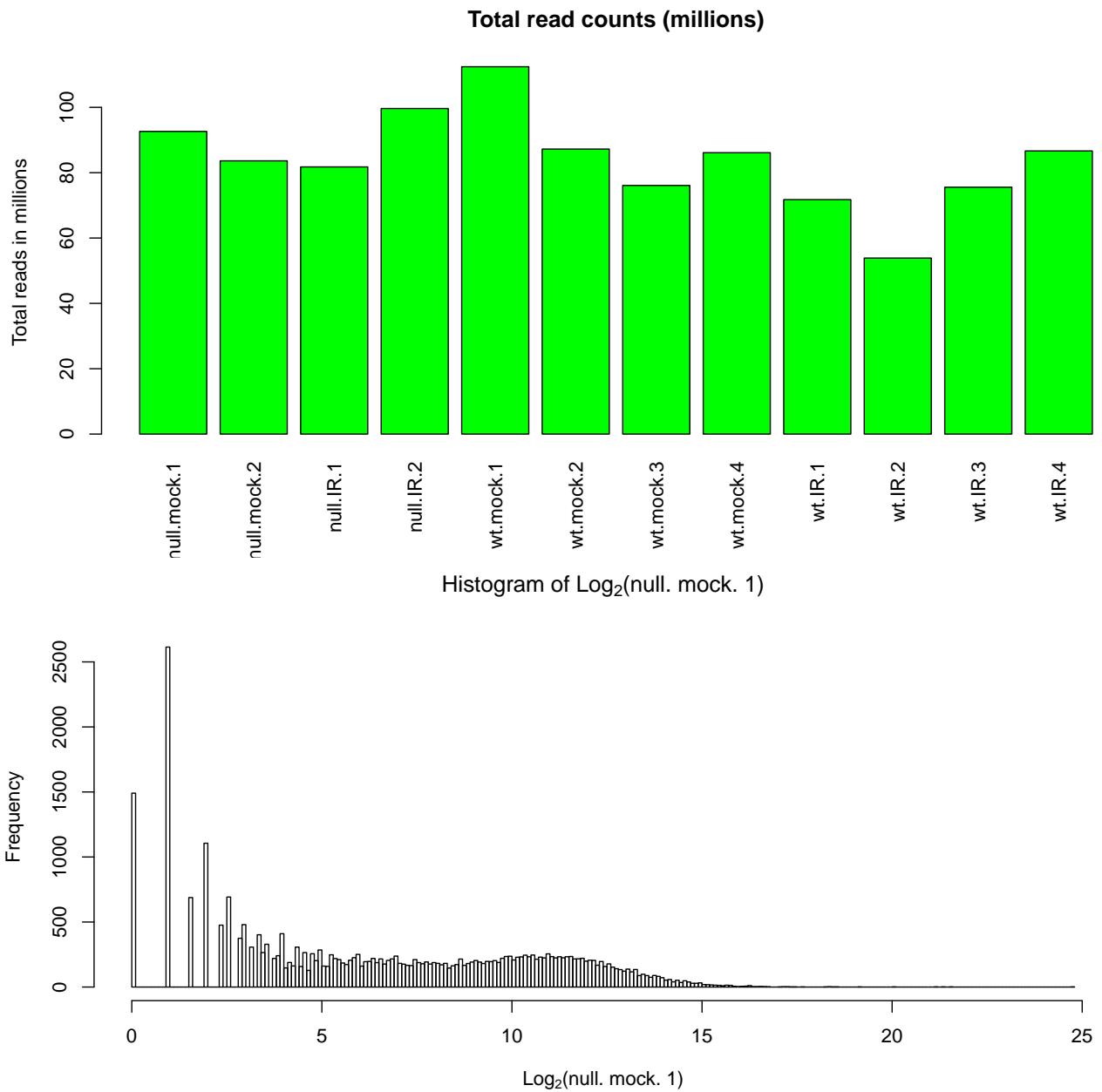
1. Reading data

First we read in counts data inside the `readCounts.txt`.

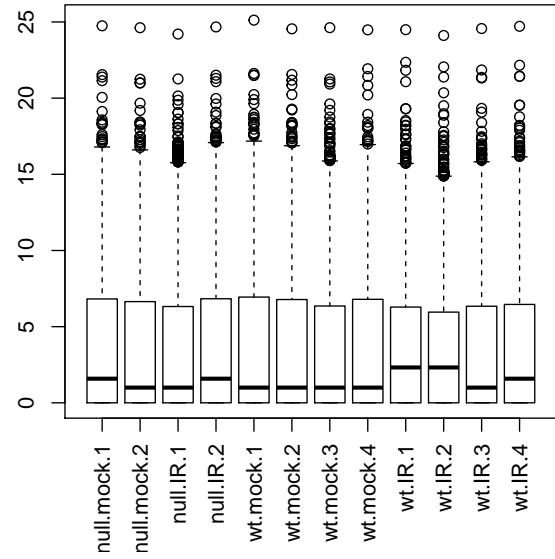
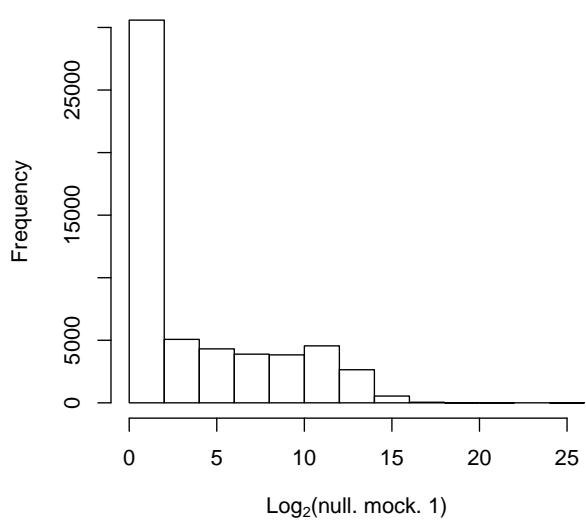
2. Simple exploration of the raw data

```
## [1] 55487    12

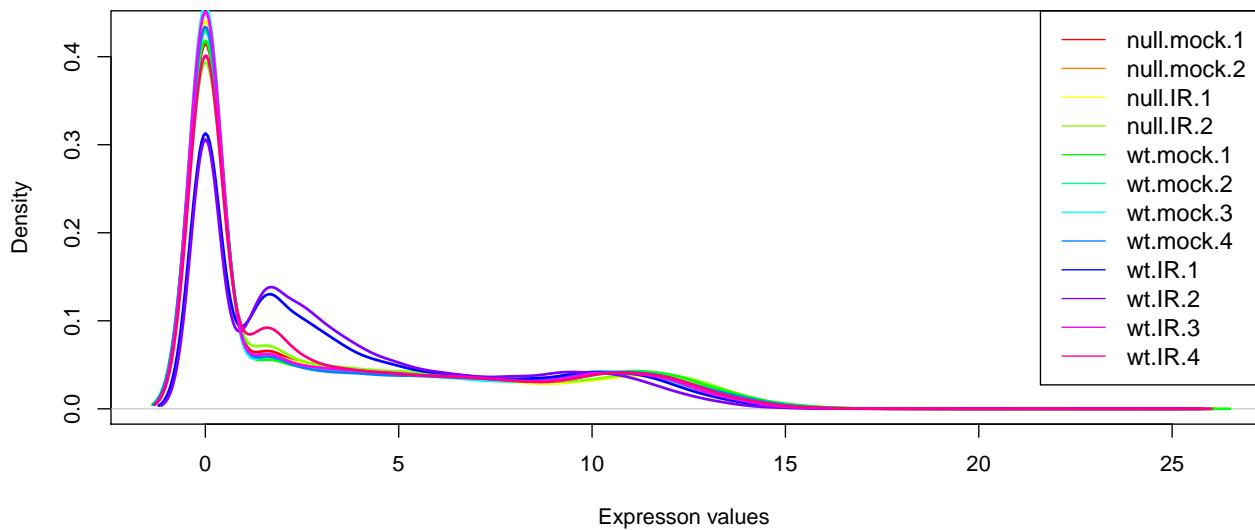
##   null.mock.1      null.mock.2      null.IR.1
## Min. :     0  Min. :     0  Min. :     0
## 1st Qu.:     0  1st Qu.:     0  1st Qu.:     0
## Median :     2  Median :     1  Median :     1
## Mean  : 1669  Mean  : 1507  Mean  : 1474
## 3rd Qu.:   112  3rd Qu.:    99  3rd Qu.:    79
## Max.  :28184807  Max.  :25805512  Max.  :19350199
##   null.IR.2      wt.mock.1      wt.mock.2
## Min. :     0  Min. :     0  Min. :     0
## 1st Qu.:     0  1st Qu.:     0  1st Qu.:     0
## Median :     2  Median :     1  Median :     1
## Mean  : 1796  Mean  : 2026  Mean  : 1572
## 3rd Qu.:   113  3rd Qu.:   122  3rd Qu.:   109
## Max.  :26735598  Max.  :36449709  Max.  :24558145
##   wt.mock.3      wt.mock.4      wt.IR.1
## Min. :     0  Min. :     0  Min. :     0
## 1st Qu.:     0  1st Qu.:     0  1st Qu.:     0
## Median :     1  Median :     1  Median :     4
## Mean  : 1371  Mean  : 1552  Mean  : 1293
## 3rd Qu.:    81  3rd Qu.:   110  3rd Qu.:    77
## Max.  :25857899  Max.  :23413269  Max.  :23616031
##   wt.IR.2      wt.IR.3      wt.IR.4
## Min. :     0  Min. :     0  Min. :     0
## 1st Qu.:     0  1st Qu.:     0  1st Qu.:     0
## Median :     4  Median :     1  Median :     2
## Mean  :  971  Mean  : 1362  Mean  : 1562
## 3rd Qu.:    61  3rd Qu.:    80  3rd Qu.:    87
## Max.  :18142468  Max.  :24872956  Max.  :27525525
```

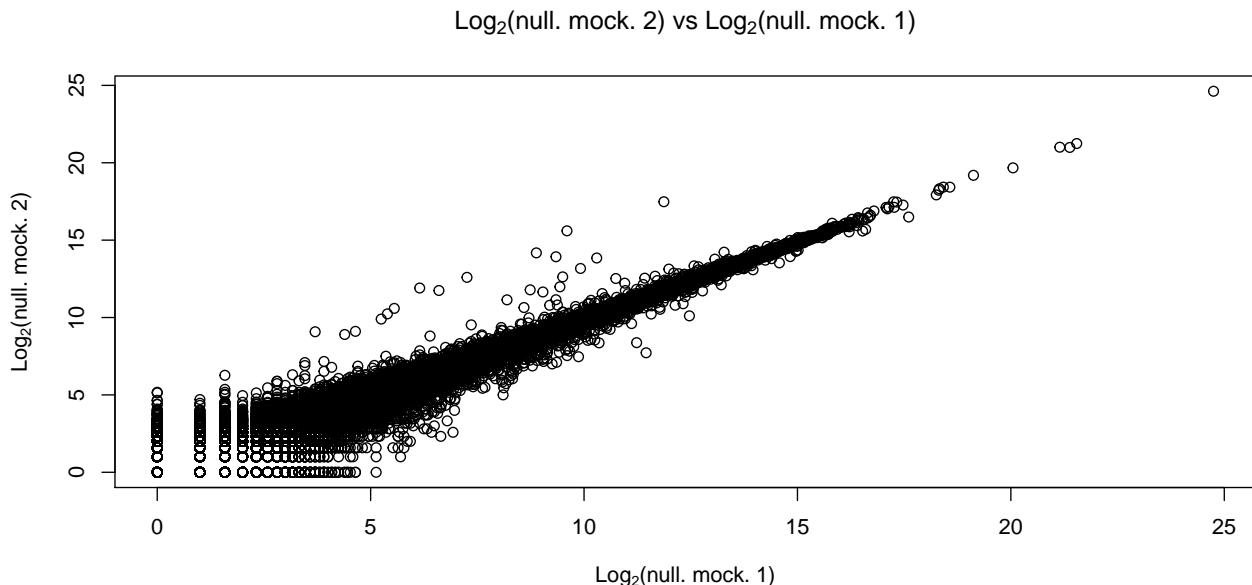


Simple log transformation on unnormalized data

Histogram of $\text{Log}_2(\text{null. mock. } 1)$ 

Distribution of transformed data





3. Filtering, normalization, and transformation using DESeq2

Define sample groups

First we define a function for parsing samples into groups. Define sample groups based on column names

Now we need to add sample IDs, as this is very likely a paired situation. We add a new column this is according to this post

```
##           groups p53 treatment
## null.mock.1 null.mock null      mock
## null.mock.2 null.mock null      mock
## null.IR.1   null.IR  null      IR
## null.IR.2   null.IR  null      IR
## wt.mock.1   wt.mock  wt      mock
## wt.mock.2   wt.mock  wt      mock
## wt.mock.3   wt.mock  wt      mock
## wt.mock.4   wt.mock  wt      mock
## wt.IR.1     wt.IR    wt      IR
## wt.IR.2     wt.IR    wt      IR
## wt.IR.3     wt.IR    wt      IR
## wt.IR.4     wt.IR    wt      IR

## 'data.frame': 12 obs. of 3 variables:
## $ groups : Factor w/ 4 levels "null.IR","null.mock",...: 2 2 1 1 4 4 4 4 3 3 ...
## $ p53   : Factor w/ 2 levels "null","wt": 1 1 1 1 2 2 2 2 2 ...
## ..- attr(*, "names")= chr "null.mock.1" "null.mock.2" "null.IR.1" "null.IR.2" ...
## $ treatment: Factor w/ 2 levels "IR","mock": 2 2 1 1 2 2 2 2 1 1 ...
## ..- attr(*, "names")= chr "null.mock.1" "null.mock.2" "null.IR.1" "null.IR.2" ...
```

Set up the DESeqDataSet Object and run the DESeq pipeline

```
## [1] 55487
```

Filtering

```
## [1] 36579
```

rlog transformation

```
##           null.mock.1 null.mock.2 null.IR.1 null.IR.2
## ENSMUSG00000051951.5  0.09379245  0.09484254  0.09527086  0.09310978
## ENSMUSG00000103377.1 -0.57297871 -0.57239922 -0.57216237 -0.57335454
## ENSMUSG00000104017.1 -0.28881375 -0.28816033 -0.28789303 -0.28923708
##           wt.mock.1 wt.mock.2 wt.mock.3 wt.mock.4
## ENSMUSG00000051951.5  0.0928600   0.09407265  0.09673701  0.09410612
## ENSMUSG00000103377.1 -0.5734919   -0.57282425 -0.57134954 -0.57280579
## ENSMUSG00000104017.1 -0.2893917   -0.28863966 -0.28697462 -0.28861885
##           wt.IR.1    wt.IR.2    wt.IR.3    wt.IR.4
## ENSMUSG00000051951.5  0.26809062  0.58508811  0.09716842  0.09591506
## ENSMUSG00000103377.1 -0.19151931 -0.56245087 -0.57110974 -0.57180563
## ENSMUSG00000104017.1 -0.08419834 -0.03388829 -0.28670337 -0.28749015
```

VSD transformation

```
##           null.mock.1 null.mock.2 null.IR.1 null.IR.2 wt.mock.1
## ENSMUSG00000051951.5      5.236118     5.236118  5.236118  5.236118  5.236118
## ENSMUSG00000103377.1      5.236118     5.236118  5.236118  5.236118  5.236118
## ENSMUSG00000104017.1      5.236118     5.236118  5.236118  5.236118  5.236118
##           wt.mock.2 wt.mock.3 wt.mock.4 wt.IR.1    wt.IR.2
## ENSMUSG00000051951.5      5.236118     5.236118  5.236118  5.696882  6.102231
## ENSMUSG00000103377.1      5.236118     5.236118  5.236118  5.983363  5.236118
## ENSMUSG00000104017.1      5.236118     5.236118  5.236118  5.767419  5.853079
##           wt.IR.3    wt.IR.4
## ENSMUSG00000051951.5 5.236118 5.236118
## ENSMUSG00000103377.1 5.236118 5.236118
## ENSMUSG00000104017.1 5.236118 5.236118
```

For the log2 approach, we need to first estimate size factors to account for sequencing depth, and then specify normalized=TRUE. Sequencing depth correction is done automatically for the rlog and the vst.

```
## null.mock.1 null.mock.2 null.IR.1 null.IR.2 wt.mock.1    wt.mock.2
##  1.2031419  1.0714551  1.0244459  1.3040657  1.3446410  1.1654621
## wt.mock.3  wt.mock.4    wt.IR.1    wt.IR.2    wt.IR.3    wt.IR.4
##  0.8864428  1.1610935  0.7737430  0.5715530  0.8514457  0.9598173
```

Started log on scaled data

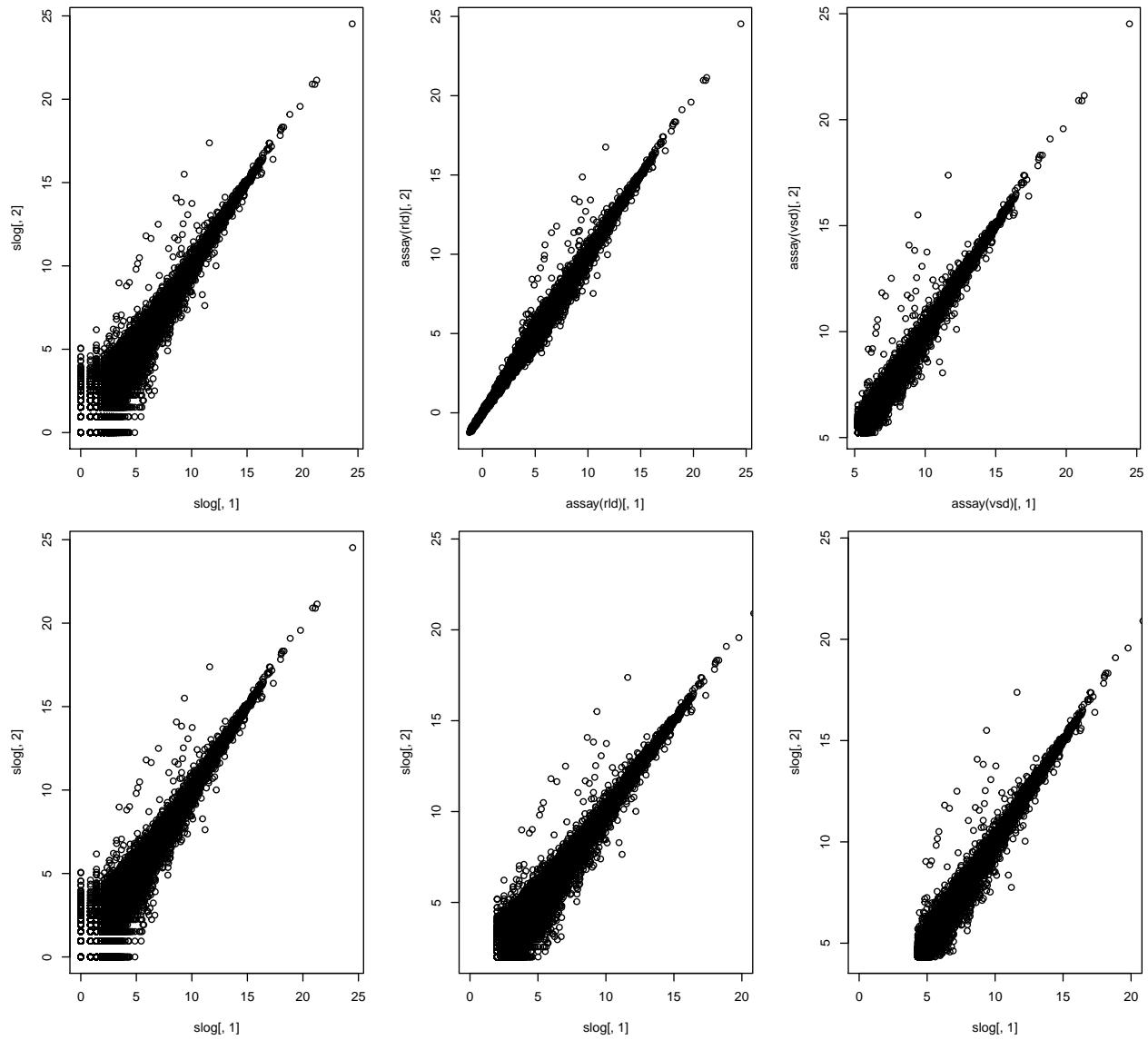
Using the normalized=TRUE option in the counts() method of DESeq2, we adjust for different library sizes.

```
##           null.mock.1 null.mock.2 null.IR.1 null.IR.2 wt.mock.1
## ENSMUSG00000051951.5          0          0          0          0          0
## ENSMUSG00000103377.1          0          0          0          0          0
## ENSMUSG00000104017.1          0          0          0          0          0
## ENSMUSG00000102331.1          0          0          0          0          0
## ENSMUSG00000102592.1          0          0          0          0          0
## ENSMUSG00000102343.1          0          0          0          0          0
##           wt.mock.2 wt.mock.3 wt.mock.4 wt.IR.1    wt.IR.2
## ENSMUSG00000051951.5          0          0          0 2.286070 3.906597
## ENSMUSG00000103377.1          0          0          0 3.503266 0.000000
## ENSMUSG00000104017.1          0          0          0 2.625194 2.999725
## ENSMUSG00000102331.1          0          0          0 1.841908 2.999725
## ENSMUSG00000102592.1          0          0          0 4.045183 2.999725
## ENSMUSG00000102343.1          0          0          0 2.899580 0.000000
##           wt.IR.3    wt.IR.4
```

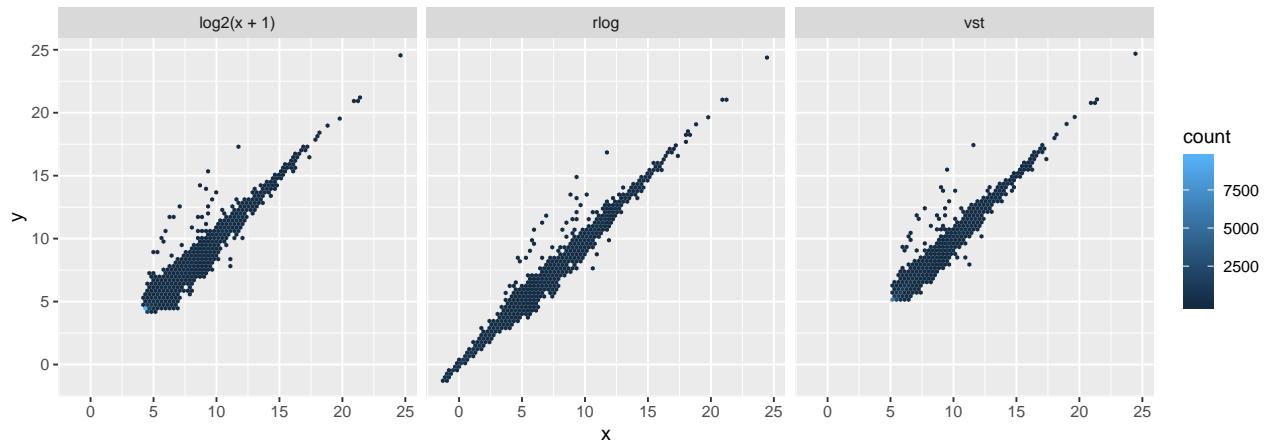
```

## ENSMUSG00000051951.5      0  0.000000
## ENSMUSG00000103377.1      0  0.000000
## ENSMUSG00000104017.1      0  0.000000
## ENSMUSG00000102331.1      0  1.029887
## ENSMUSG00000102592.1      0  0.000000
## ENSMUSG00000102343.1      0  2.369455

```

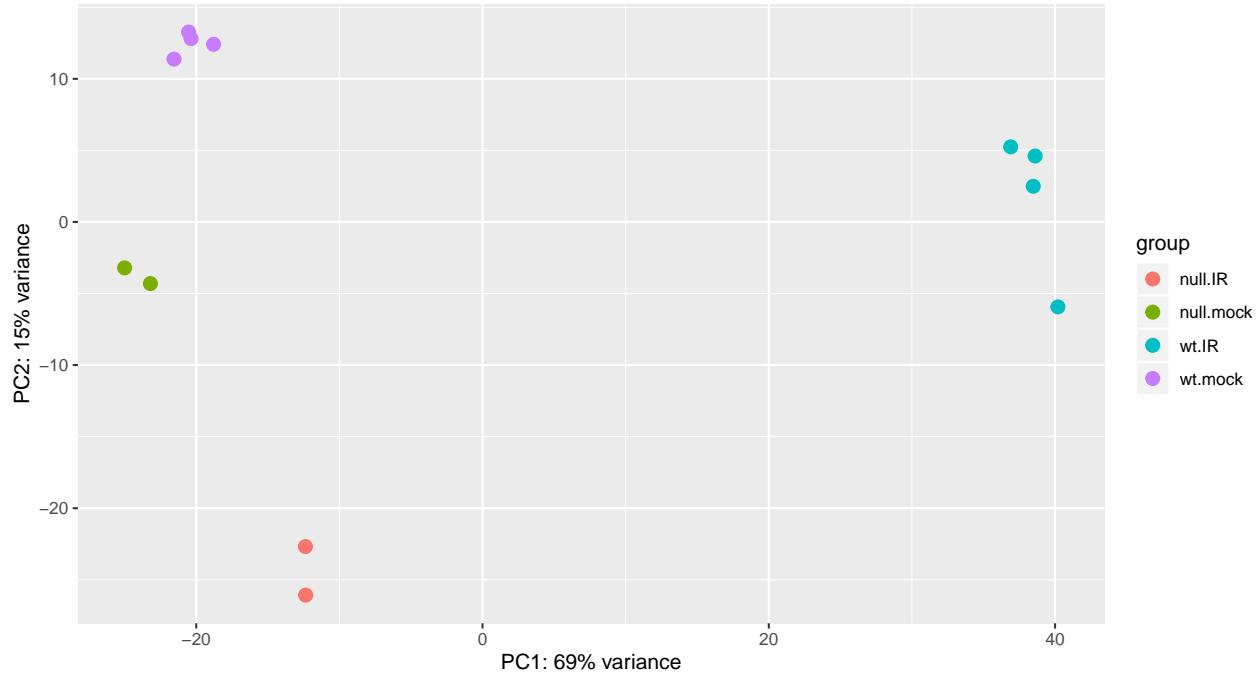


A more elegant plot using ggplot2 according to DESeq2 tutorial:

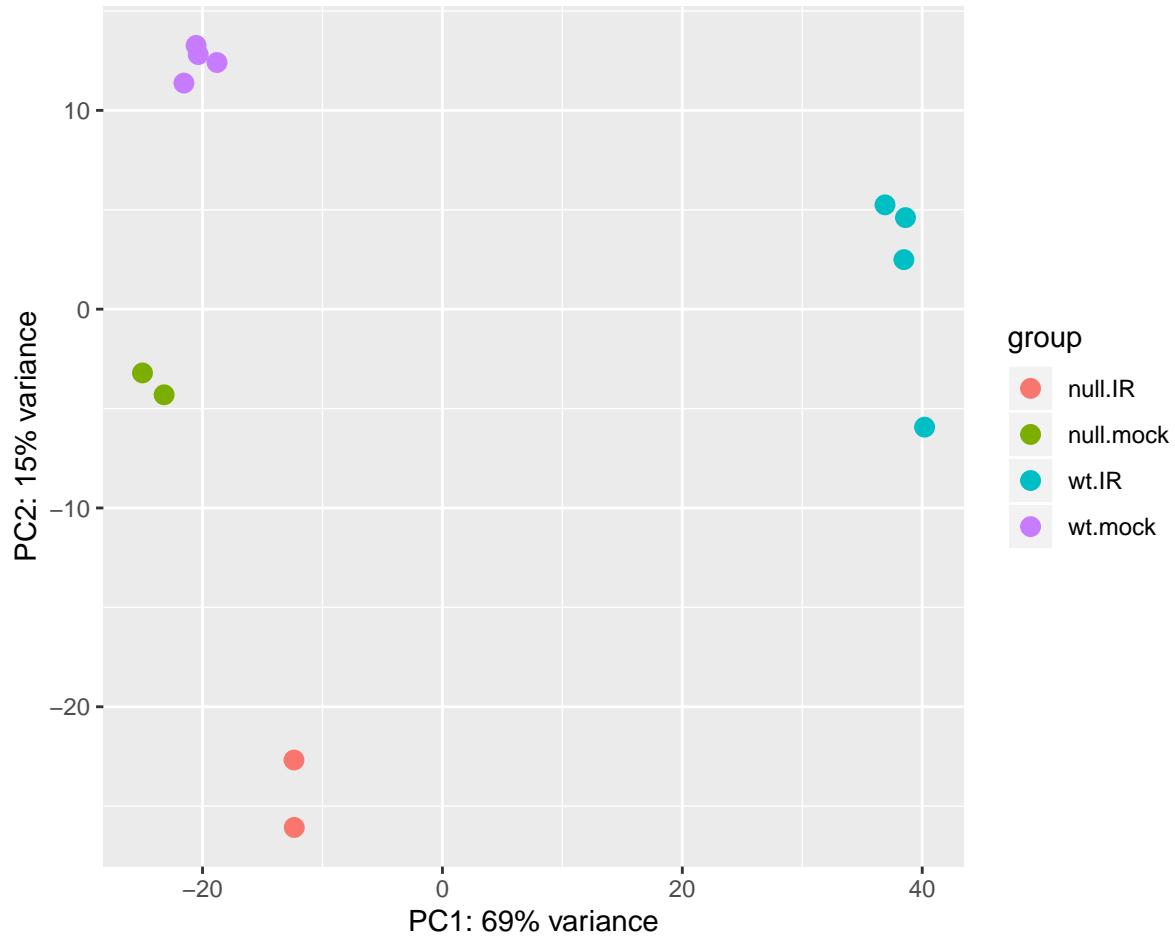


4. Exploratory Data Analysis

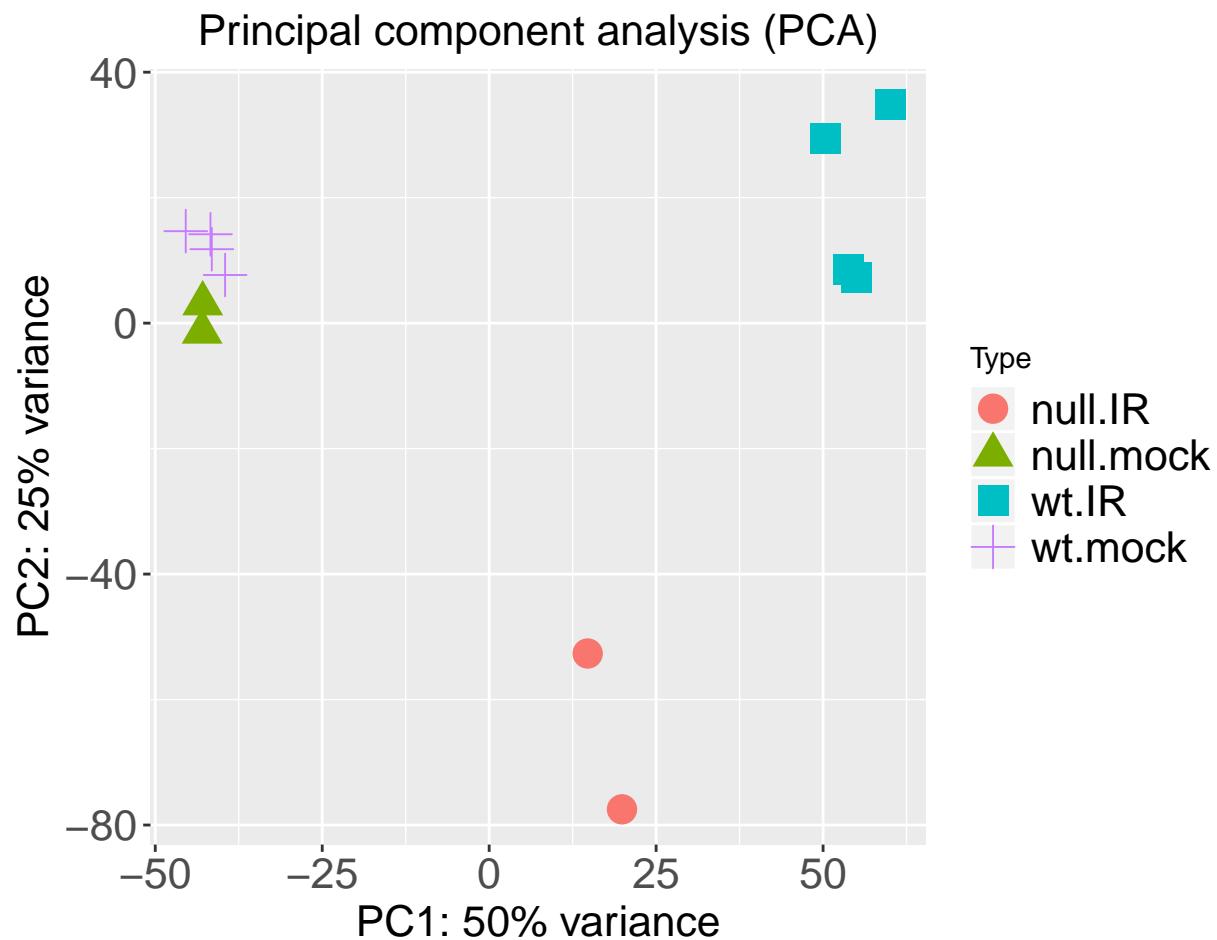
PCA plot



The figure looks odd. What we need is to make x and y aspect ratios independent of the data ranges.

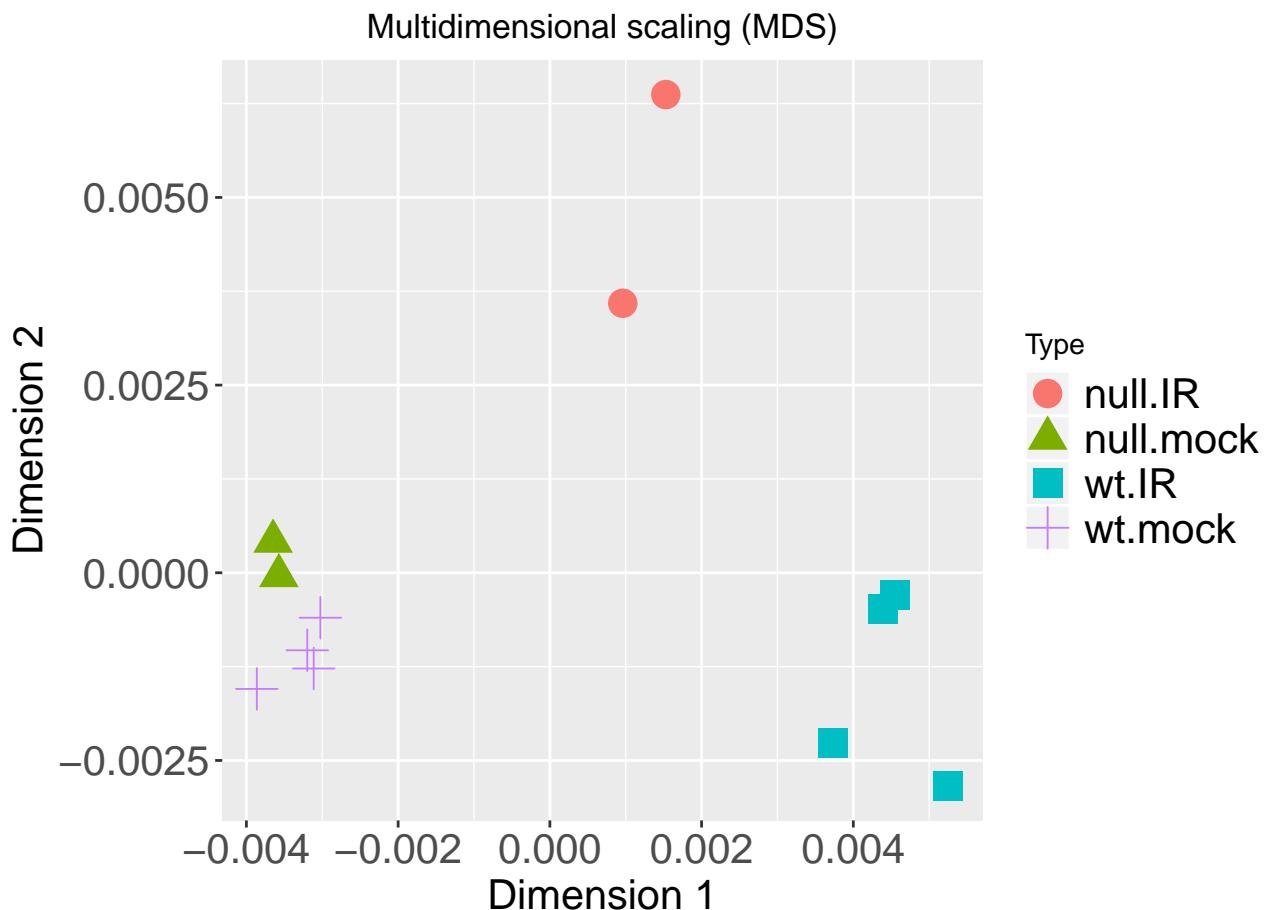


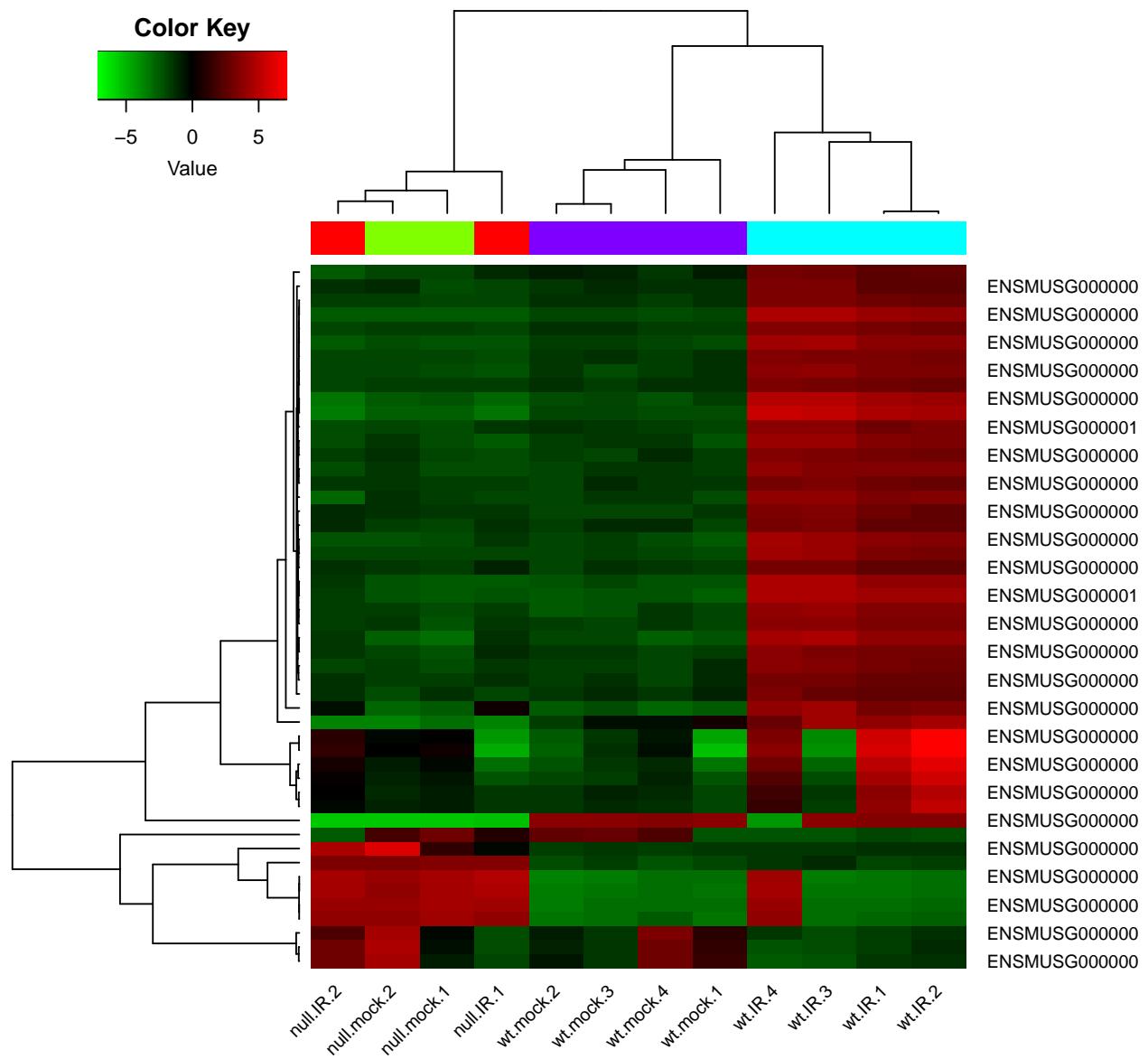
You can do the PCA plot yourself by building a plot using ggplot2, step by step



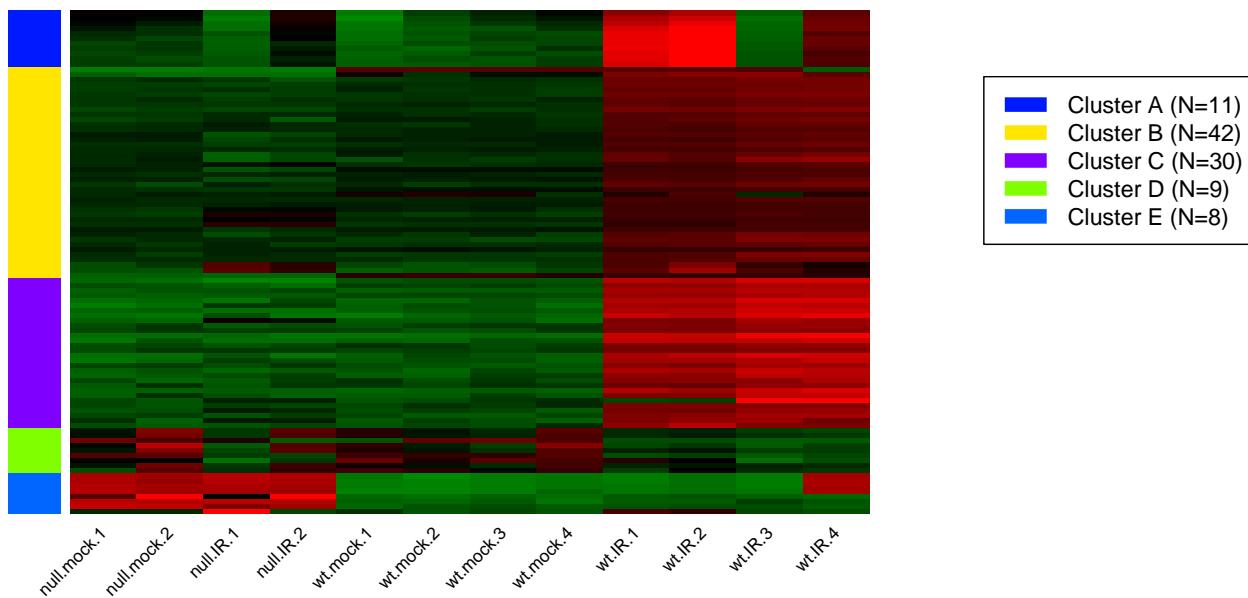
MDS plot

Similarly, we can produce MDS plots. As you can see from the code below, I recycled the plotting code and didn't even change some of the variable names.



Hierarchical clustering with heatmap

K-means clustering of genes



5. Differential expression analysis using DESeq2

```
## log2 fold change (MLE): treatment mock vs IR
## Wald test p-value: treatment mock vs IR
## DataFrame with 36579 rows and 6 columns
##                                     baseMean      log2FoldChange
##                                     <numeric>      <numeric>
## ENSMUSG00000051951.5  1.48951739302888 -3.45769667767622
## ENSMUSG00000103377.1  0.861612478553997 -2.78547912179598
## ENSMUSG00000104017.1  1.01401259606256 -2.95855679102816
## ENSMUSG00000102331.1  0.885431555118192 -2.80787322518581
## ENSMUSG00000102592.1  1.87562507461656 -3.77830056153258
## ...
##                                     ...
## ENSMUSG00000064368.1   48761.3379686471 -0.36069784835848
## ENSMUSG00000064369.1   3148.20650066707  0.248995813844674
## ENSMUSG00000064370.1   229153.247939432 -0.463149607155602
## ENSMUSG00000064371.1   54.7512721213404 -0.61944345152823
## ENSMUSG00000064372.1   942.383128434213 -0.870667263888078
##                                     lfcSE          stat
##                                     <numeric>      <numeric>
## ENSMUSG00000051951.5   3.04592047170739 -1.13518941475777
## ENSMUSG00000103377.1   3.06610664118698 -0.908474312138615
## ENSMUSG00000104017.1   3.0612051051142 -0.966468005062925
## ENSMUSG00000102331.1   2.67830291137901 -1.0483777668524
## ENSMUSG00000102592.1   2.93580169068832 -1.28697403966912
## ...
##                                     ...
## ENSMUSG00000064368.1   0.238976299912887 -1.50934569030471
## ENSMUSG00000064369.1   0.159271605309944  1.56334089406663
## ENSMUSG00000064370.1   0.213827759595239 -2.16599382620999
## ENSMUSG00000064371.1   0.3060705555512 -2.0238583564913
## ENSMUSG00000064372.1   0.202240636016501 -4.30510544783413
##                                     pvalue      padj
##                                     <numeric>      <numeric>
```

```

##                                     <numeric>          <numeric>
## ENSMUSG0000051951.5    0.256295963671824  0.376836207666776
## ENSMUSG00000103377.1    0.363627679571275      NA
## ENSMUSG00000104017.1    0.333810049356437      NA
## ENSMUSG00000102331.1    0.294464592857858      NA
## ENSMUSG00000102592.1    0.198103336743242  0.312377384457178
## ...
##                                     ...
## ENSMUSG0000064368.1    0.131210464193762  0.228775700116307
## ENSMUSG0000064369.1    0.117972434432238  0.211583081591477
## ENSMUSG0000064370.1    0.0303116587152136  0.072138804635945
## ENSMUSG0000064371.1    0.0429847277621139  0.0962055462138653
## ENSMUSG0000064372.1    1.66906214208969e-05 9.45648783641496e-05

```

Applying cutoff

DESeq2 uses the Benjamini-Hochberg (BH) adjustment (Benjamini and Hochberg 1995) as implemented in the base R p.adjust function

```

##                                     <numeric>          <numeric>
## out of 36579 with nonzero total read count
## adjusted p-value < 0.5
## LFC > 0.01 (up)      : 8857, 24%
## LFC < -0.01 (down)   : 15557, 43%
## outliers [1]           : 22, 0.06%
## low counts [2]         : 1334, 3.6%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

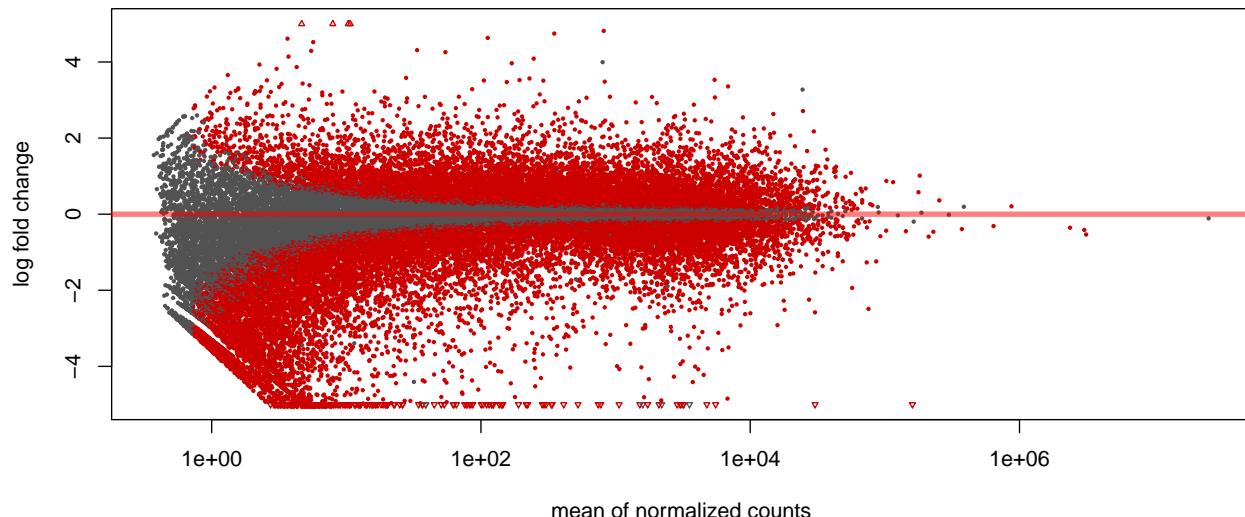
##Sort genes by fold change

## log2 fold change (MLE): treatment mock vs IR
## Wald test p-value: treatment mock vs IR
## DataFrame with 6 rows and 6 columns
##                                     baseMean      log2FoldChange       lfcSE
##                                     <numeric>      <numeric>      <numeric>
## ENSMUSG0000026831.16 294.434890082783 -10.8986209218927 2.39452946470452
## ENSMUSG0000000308.14 115.338446153961 -9.2173382850945 1.54573375190591
## ENSMUSG00000087132.8  50.8350280621933 -8.25552375037134 1.93318328778446
## ENSMUSG00000114192.1  26.2235736013177 -8.17831689044707 1.08940650090745
## ENSMUSG0000034818.16 525.897102370369 -7.80808045725016 0.917981250446441
## ENSMUSG00000069049.11 2201.07598176852 -7.36932902833692 2.5869262636059
##                                     stat          pvalue
##                                     <numeric>      <numeric>
## ENSMUSG0000026831.16 -4.54729043112287 5.43409860442426e-06
## ENSMUSG0000000308.14 -5.95661333896718 2.57518541118292e-09
## ENSMUSG00000087132.8 -4.26525710338681 1.99672135724724e-05
## ENSMUSG00000114192.1 -7.4979513006789 6.48229732567845e-14
## ENSMUSG0000034818.16 -8.49481452203706 1.98249242800415e-17
## ENSMUSG00000069049.11 -2.84481592377464      NA
##                                     padj
##                                     <numeric>
## ENSMUSG0000026831.16 3.92086476779822e-05
## ENSMUSG0000000308.14 3.15059936568587e-08
## ENSMUSG00000087132.8 0.000129641504822709
## ENSMUSG00000114192.1 1.3680404955205e-12

```

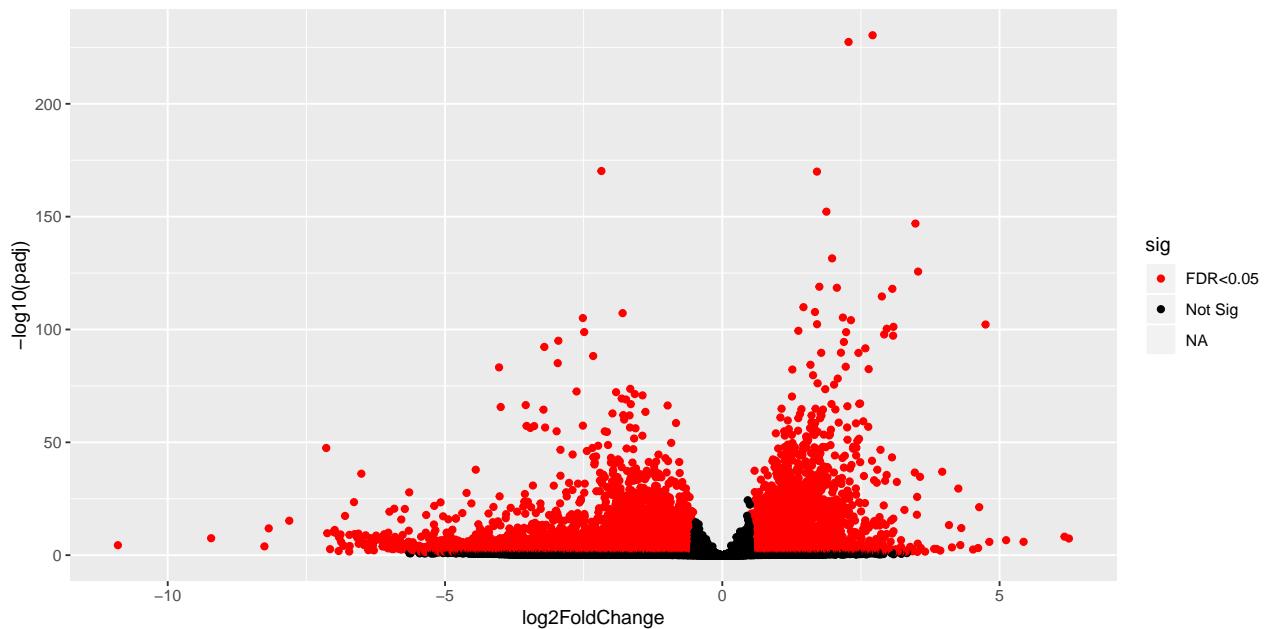
```
## ENSMUSG00000034818.16 5.67718136516993e-16
## ENSMUSG00000069049.11 NA
```

MA plot

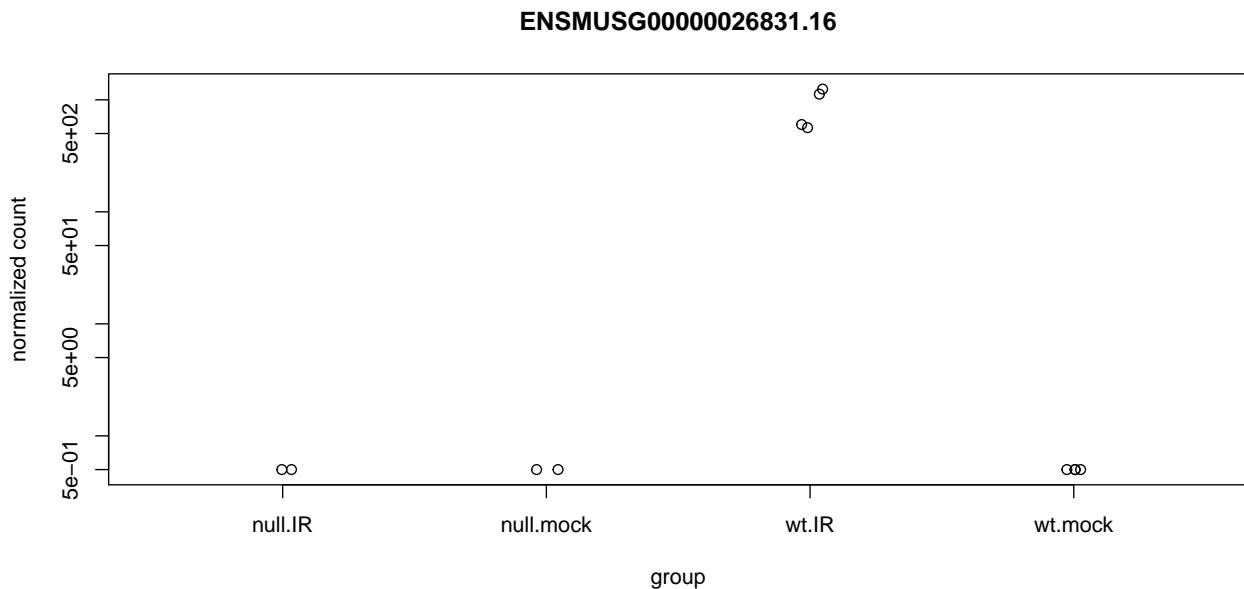


DESeq2 also provides shrinked log fold changes:

Basic volcano plot



Plotting counts of selected genes



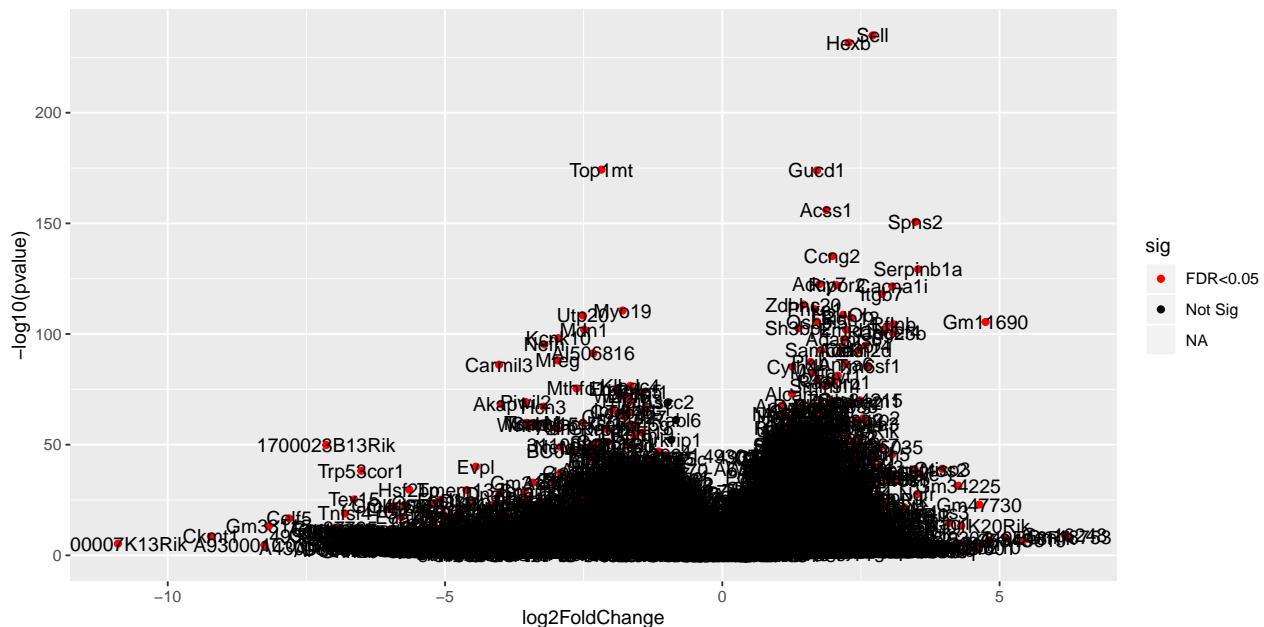
6. Annotating genes using Bioconductor

First we need to install Bioconductor base and some Bioconductor packages.

That file contains all genes passed filter. Let's output only the significant genes. We also sorted the genes by fold change.

```
##           symbol  baseMean log2FoldChange
## ENSMUSG00000107262.1 ENSMUSG00000107262    7.938182   6.250145
## ENSMUSG00000086137.1 ENSMUSG00000086137   10.699320   6.171791
## ENSMUSG00000116939.1 ENSMUSG00000116939    4.660927   5.433744
## ENSMUSG00000117684.1 ENSMUSG00000117684   10.347561   5.118549
## ENSMUSG00000080440.1 ENSMUSG00000080440   816.154550   4.816159
## ENSMUSG00000085844.1 ENSMUSG00000085844  350.345774   4.747714
##          lfcSE      stat      pvalue      padj
## ENSMUSG00000107262.1 1.0575439  5.900601 3.621792e-09 4.337653e-08
## ENSMUSG00000086137.1 0.9900713  6.223583 4.859269e-10 6.567845e-09
## ENSMUSG00000116939.1 1.0280289  5.275867 1.321299e-07 1.254112e-06
## ENSMUSG00000117684.1 0.9128756  5.596106 2.192192e-08 2.351982e-07
## ENSMUSG00000080440.1 0.9125607  5.266673 1.389185e-07 1.315003e-06
## ENSMUSG00000085844.1 0.2164780 21.885434 3.575948e-106 6.297780e-103
##          entrez
## ENSMUSG00000107262.1      Gm18753
## ENSMUSG00000086137.1      Gm16248
## ENSMUSG00000116939.1      Gm49619
## ENSMUSG00000117684.1 4930449E18Rik
## ENSMUSG00000080440.1      Gm25848
## ENSMUSG00000085844.1      Gm11690
```

Enhanced volcano plot with gene symbols



Further improved volcano plot with ggrepel

