# Exploratory analysis

*Jeff Leek*



# Dependencies

This document depends on the following packages:

```
library(gplots)
library(devtools)
library(Biobase)
library(RSkittleBrewer)
library(org.Hs.eg.db)
library(AnnotationDbi)
```

To install these packages you can use the code (or if you are compiling the document, remove the `eval=FALSE` from the chunk.)

```
install.packages(c("devtools","gplots"))
source("http://www.bioconductor.org/biocLite.R")
biocLite(c("Biobase","org.Hs.eg.db","AnnotationDbi"))
biocLite("alyssafrazee/RSkittleBrewer")
```

# General principles

- Use plots as often as possible
- Use tables for phenotype data
- Look for
- Missing values
- Outlier values
- Mislabeled samples
- Naming consistency

# Make the plots pretty

Typically we will use color to explore data sets and label different values. There are a large number of color options in R. I like the RSkittleBrewer (http://alyssafrazee.com/RSkittleBrewer.html) package, but you can also check out Jenny Bryan's excellent lecture on colors in R (http://www.stat.ubc.ca/~jenny/STAT545A/block14_colors.html) for more information.

Load the library and set the color palette with the `palette` function. Now when I type `col = 1` it will look for the first color in the `trop` colors. We also set the character to be a filled dot with `par(pch=19)`.

```
library(RSkittleBrewer)
# Make the colors pretty
trop = RSkittleBrewer("tropical")
palette(trop)
par(pch=19)
```

# Load some data

We will use this expression set to look at how we use plots and tables to check for different characteristics

```
con = url("http://bowtie-bio.sourceforge.net/recount/ExpressionSets/bodymap_eset.RData")
load(file=con)
close(con)
bm = bodymap.eset
pdata=pData(bm)
edata=exprs(bm)
fdata = fData(bm)
ls()
```

```
## [1] "bm"           "bodymap.eset" "con"          "edata"
## [5] "fdata"        "pdata"        "tropical"
```

## Tables for factor/character variables

Tables are good for looking at factor or character variables, especially in phenotype data

```
table(pdata$gender)
```

```
##
## F M
## 8 8
```

```
table(pdata$gender,pdata$race)
```

```
##
##    african_american caucasian
##   F                1         7
##   M                0         8
```

## Look for missing values

First check a summary of the distribution to look for scale, this is also one way to check for `NA` values.

```
summary(edata)
```

```
##    ERS025098           ERS025092          ERS025085
##  Min.   :      0.0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:      0.0   1st Qu.:     0.0   1st Qu.:     0.0
##  Median :      0.0   Median :     0.0   Median :     0.0
##  Mean   :    455.6   Mean   :   361.1   Mean   :   399.3
##  3rd Qu.:      0.0   3rd Qu.:     0.0   3rd Qu.:     0.0
##  Max.   :1584528.0   Max.   :499802.0   Max.   :808641.0
##    ERS025088           ERS025089         ERS025082
##  Min.   :      0.0   Min.   :      0   Min.   :      0
##  1st Qu.:      0.0   1st Qu.:      0   1st Qu.:      0
##  Median :      0.0   Median :      0   Median :      0
##  Mean   :    445.5   Mean   :    445   Mean   :    509
##  3rd Qu.:      0.0   3rd Qu.:      0   3rd Qu.:      0
##  Max.   :1014579.0   Max.   :1415741   Max.   :2484692
##    ERS025081           ERS025096         ERS025099
##  Min.   :      0.0   Min.   :      0   Min.   :     0.0
##  1st Qu.:      0.0   1st Qu.:      0   1st Qu.:     0.0
##  Median :      0.0   Median :      0   Median :     0.0
##  Mean   :    430.4   Mean   :    558   Mean   :   445.5
##  3rd Qu.:      0.0   3rd Qu.:      0   3rd Qu.:     0.0
##  Max.   :1356643.0   Max.   :3517964   Max.   :572374.0
##    ERS025086           ERS025084          ERS025087
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0
##  1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0
##  Median :     0.0   Median :     0.0   Median :     0
##  Mean   :   370.7   Mean   :   592.1   Mean   :  1097
##  3rd Qu.:     0.0   3rd Qu.:     0.0   3rd Qu.:     0
##  Max.   :458168.0   Max.   :287539.0   Max.   :519683
##    ERS025093           ERS025083          ERS025095
##  Min.   :      0.0   Min.   :     0.0   Min.   :      0.0
##  1st Qu.:      0.0   1st Qu.:     0.0   1st Qu.:      0.0
##  Median :      0.0   Median :     0.0   Median :      0.0
##  Mean   :    997.7   Mean   :   434.7   Mean   :    479.6
##  3rd Qu.:      0.0   3rd Qu.:     0.0   3rd Qu.:      0.0
##  Max.   :1332827.0   Max.   :626824.0   Max.   :1605570.0
##    ERS025097           ERS025094          ERS025090
##  Min.   :      0.0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:      0.0   1st Qu.:     0.0   1st Qu.:     0.0
##  Median :      0.0   Median :     0.0   Median :     0.0
```

```
##  Mean   :    540.1   Mean   :   518.5   Mean   :   465.3
##  3rd Qu.:      0.0   3rd Qu.:     0.0   3rd Qu.:     0.0
##  Max.   :1888083.0   Max.   :776623.0   Max.   :535410.0
##    ERS025091
##  Min.   :      0.0
##  1st Qu.:      0.0
##  Median :      0.0
##  Mean   :    530.1
##  3rd Qu.:      0.0
##  Max.   :1221921.0
```

`NA` is the most common character for missing values, but sometimes they are coded as spaces, 999, -1 or "missing". Check for missing values in a variety of ways

```
# Use option useNA to include NA's in table
table(pdata$age,useNA="ifany")
```

```
##
##   19   29   37   47   58   60   65   68   73   77   86 <NA>
##    1    1    1    1    1    3    1    1    2    3    1    3
```

```
# is.na checks for NA values
table(is.na(pdata$age))
```

```
##
## FALSE  TRUE
##    16     3
```

```
# Check for other common missing names
sum(pdata$age==" ")
```

```
## [1] NA
```

```
# Check genomic data for NAs
sum(is.na(edata))
```

```
## [1] 0
```

```
# Make the distribution of NA's by genes
gene_na = rowSums(is.na(edata))
table(gene_na)
```

```
## gene_na
##     0
## 52580
```

```
# Make the distribution of NA's by samples
sample_na = rowSums(is.na(edata))
table(sample_na)
```

```
## sample_na
##     0
## 52580
```

## Make sure dimensions match up

The number of rows of the feature data should match the number of rows of the genomic data (both are the number of genes). The number of rows of the phenotype data should match the number of columns of the genomic data (both are the number of samples).

```
dim(fdata)
```

```
## [1] 52580     1
```

```
dim(pdata)
```

```
## [1] 19  6
```

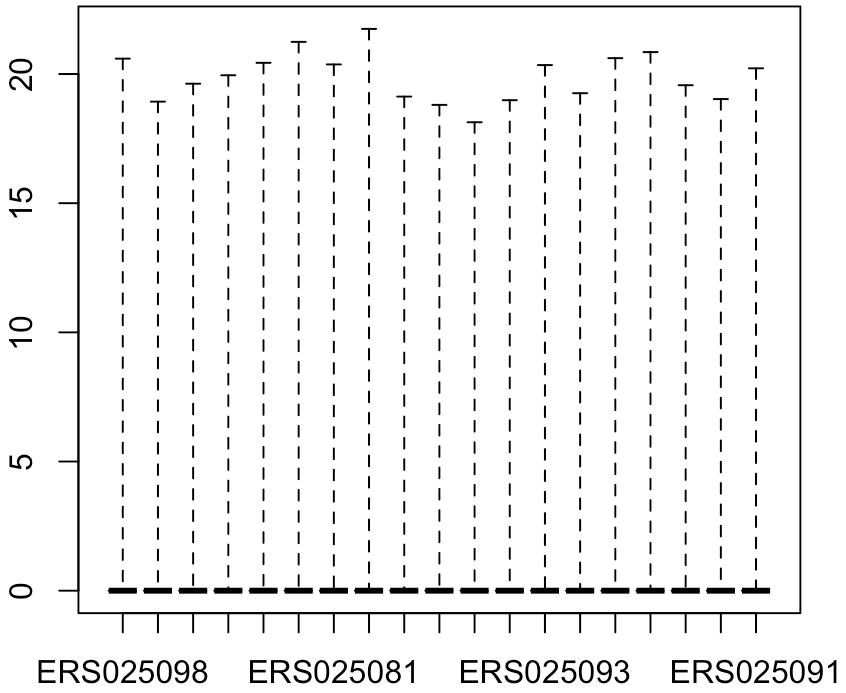
```
dim(edata)
```

```
## [1] 52580    19
```

## Look at overall distributions

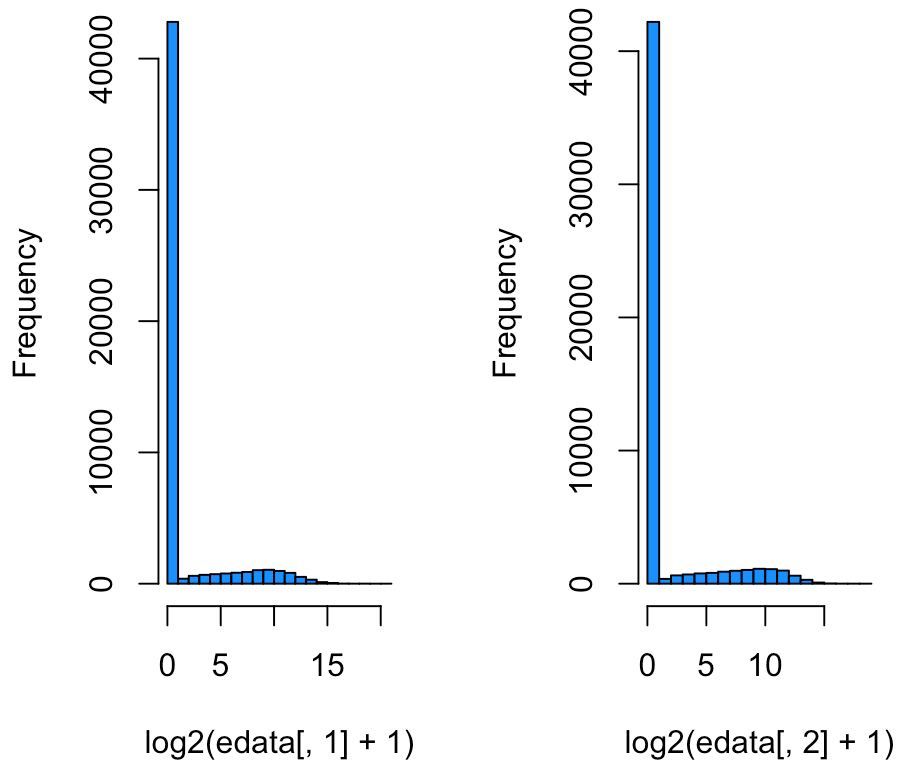
Here we see that there are a lot of outliers

```
boxplot(log2(edata+1),col=2,range=0)
```



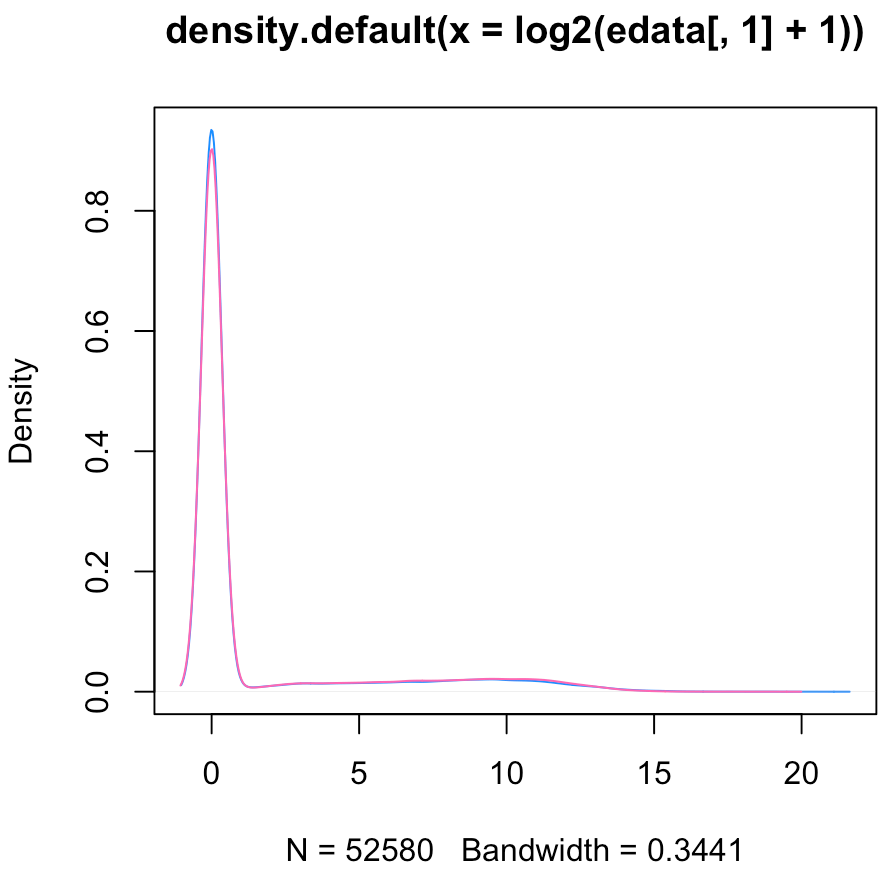

We can also look at this sample by sample with histograms

```
par(mfrow=c(1,2))
hist(log2(edata[,1]+1),col=2)
hist(log2(edata[,2]+1),col=2)
```



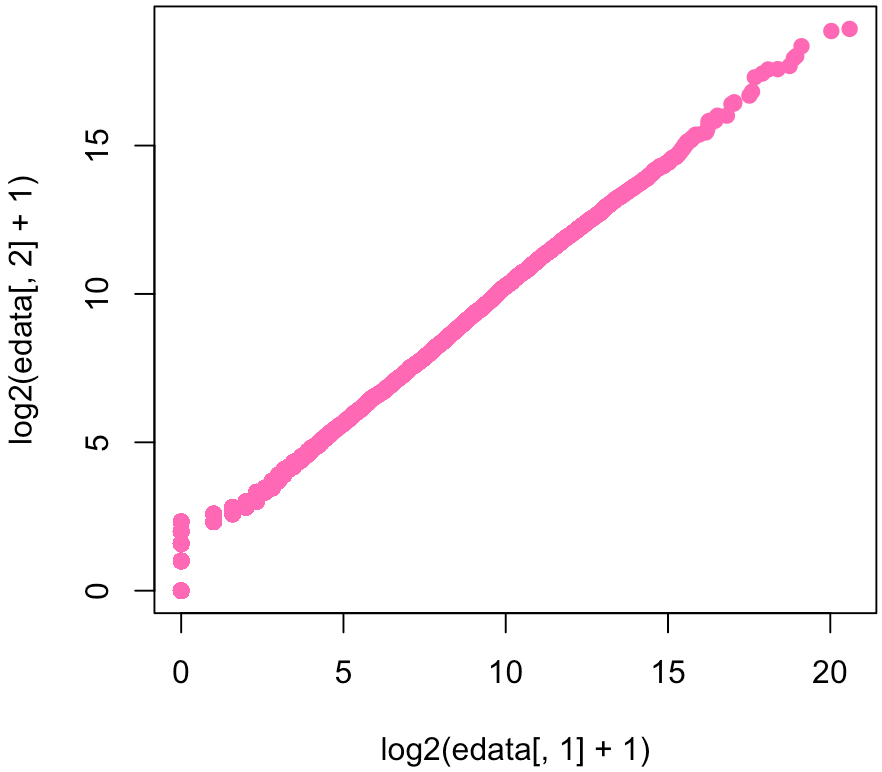

Or with density plots

```
plot(density(log2(edata[,1]+1)),col=2)
lines(density(log2(edata[,2]+1)),col=3)
```



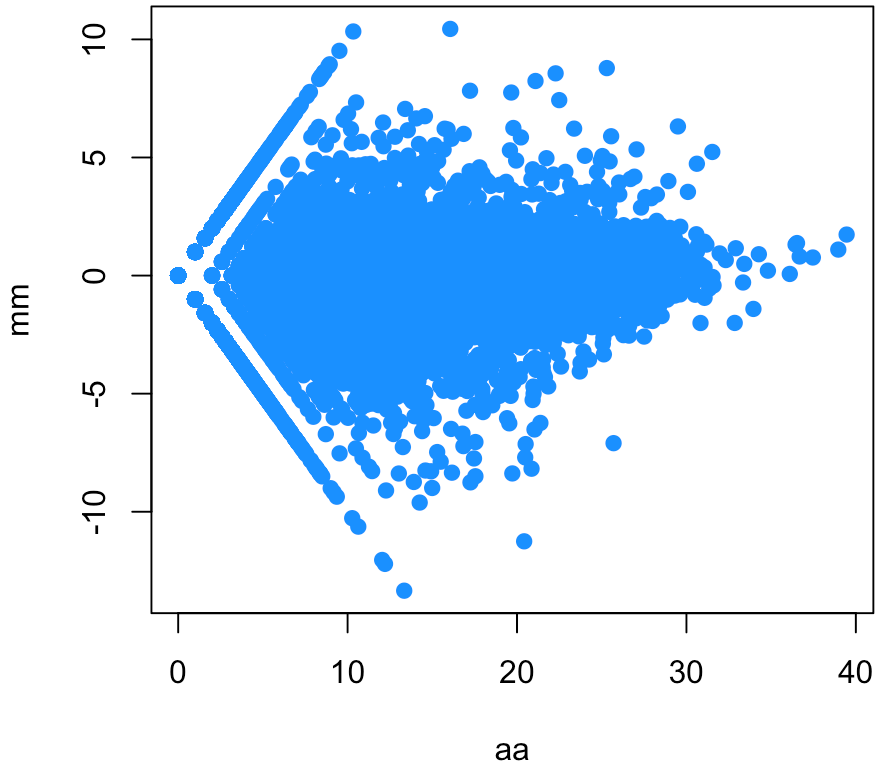

A very common task is to compare distributions of measurements (say before normalization). You can do this with a qq-plot

```
qqplot(log2(edata[,1]+1), log2(edata[,2]+1),col=3)
```



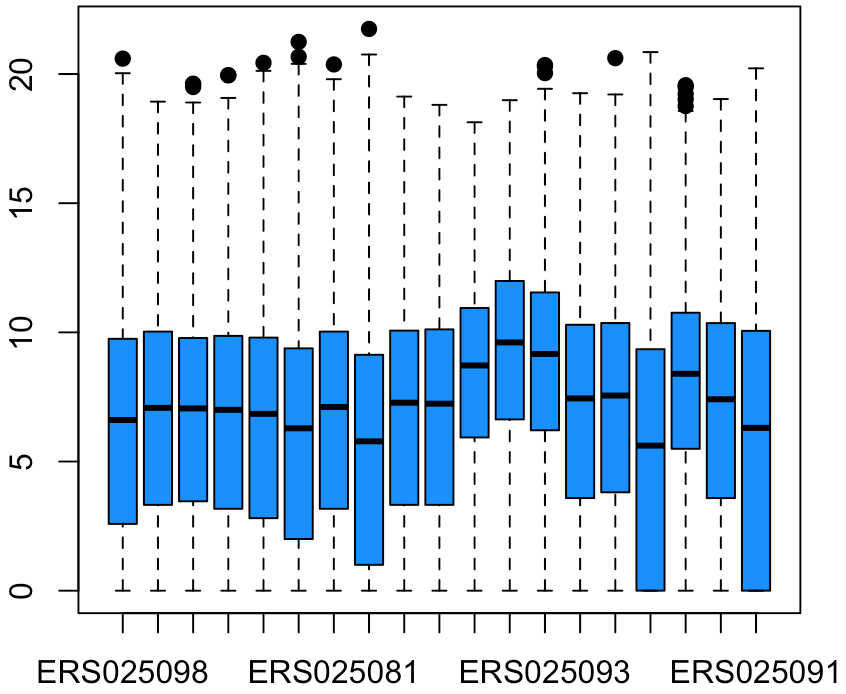

A very widely used plot is what is known as a M-A plot (https://en.wikipedia.org/wiki/MA_plot), sometimes called a Bland Altman plot (https://en.wikipedia.org/wiki/Bland%E2%80%93Altman_plot). The basic idea is to plot the sum of the two values on the x-axis and the difference on the y-axis. This can be used to see any difference between the (samples, averages, etc.) and to see if there is any intensity-specific biases.

```
mm = log2(edata[,1]+1) - log2(edata[,2]+1)
aa = log2(edata[,1]+1) + log2(edata[,2]+1)
plot(aa,mm,col=2)
```




We can remove rows that are mostly zero and notice any differences in the distributions across samples.

```
edata = as.data.frame(edata)
filt_edata = filter(edata,rowMeans(edata) > 1)
boxplot(as.matrix(log2(filt_edata+1)),col=2)
```




## Check for obvious data mixups

Here we are going to do a check to make sure that the men and women are correctly labeled by looking at expression on the Y chromosome. In general you might do several of this type of check to confirm the data are correctly labeled.

Get the chromosomes for each gene using the feature data.

```
aeid = as.character(fdata[,1])
chr = AnnotationDbi::select(org.Hs.eg.db,keys=aeid,keytype="ENSEMBL",columns="CHR")
head(chr)
```

```
##           ENSEMBL CHR
## 1 ENSG00000000003   X
## 2 ENSG00000000005   X
## 3 ENSG00000000419  20
## 4 ENSG00000000457   1
## 5 ENSG00000000460   1
## 6 ENSG00000000938   1
```

Filter to the data on chromsome Y and sum up all the counts. A tricky issue is that some genes are annotated to multiple chromsomes. Here we take the first chromsome each is annotated to.

```
dim(chr)
```

```
## [1] 52724     2
```

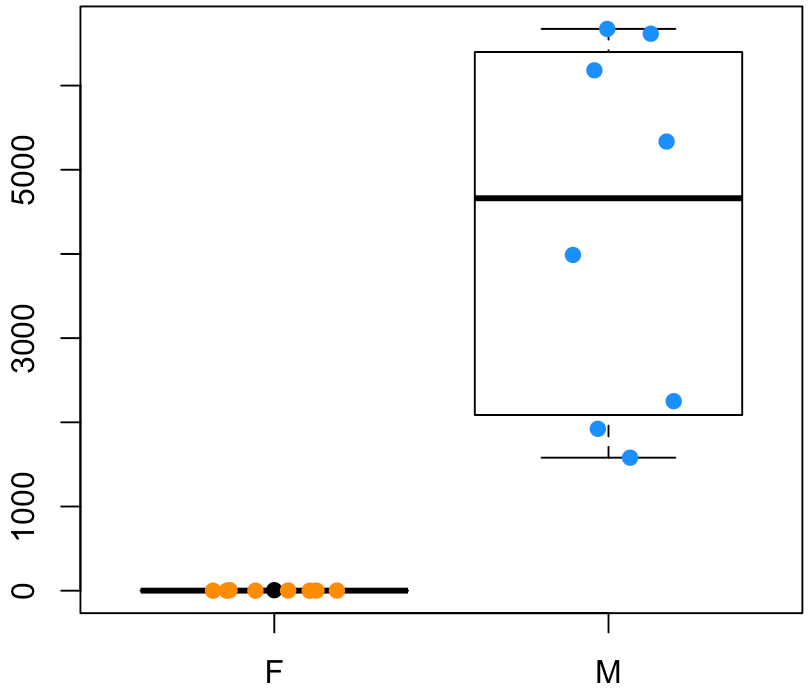
```
dim(edata)
```

```
## [1] 52580    19
```

```
# Take non-duplicated chromsomes
chr = chr[!duplicated(chr[,1]),]

# Confirm that the annotation still is in the right order
all(chr[,1] == rownames(edata))
```

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

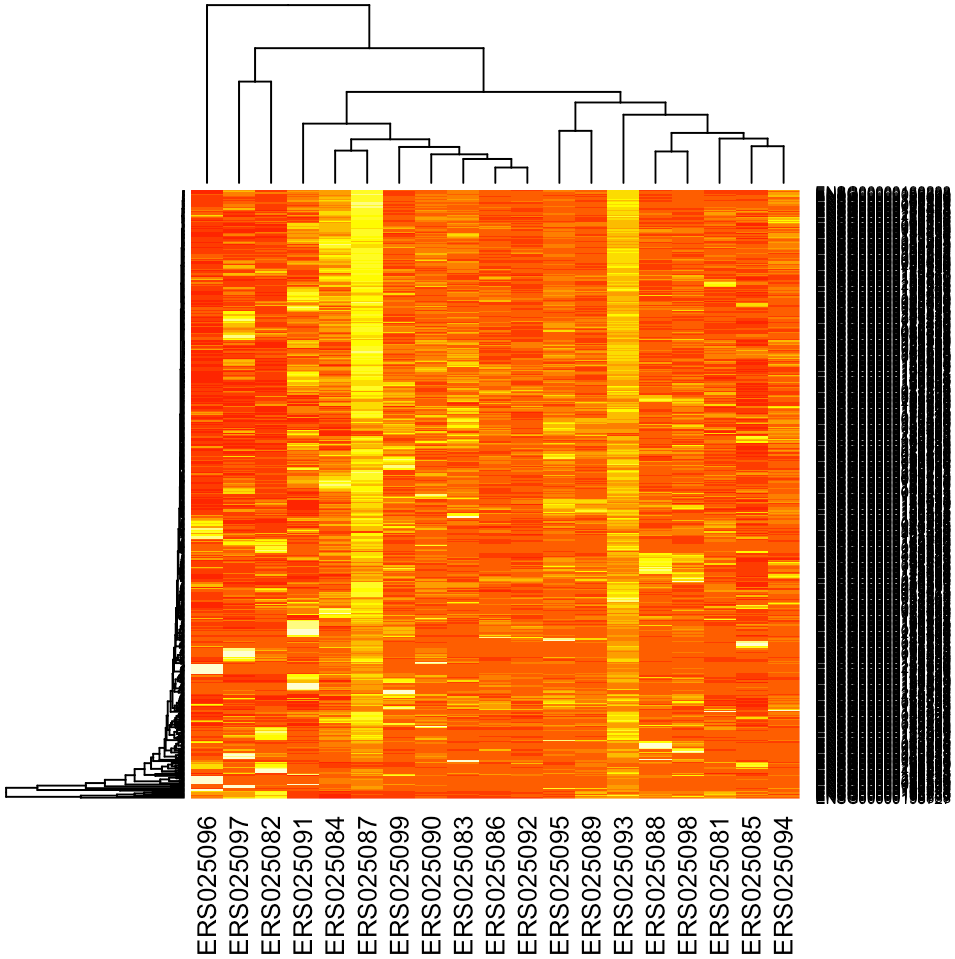
```
# Select the chromosome Y samples
edatay = dplyr::filter(edata,chr$CHR=="Y")

# Males have Y chromsome expression as expected
boxplot(colSums(edatay) ~ pdata$gender)
points(colSums(edatay) ~ jitter(as.numeric(pdata$gender)),
        col=as.numeric(pdata$gender),
        pch=19)
```



# Heatmaps

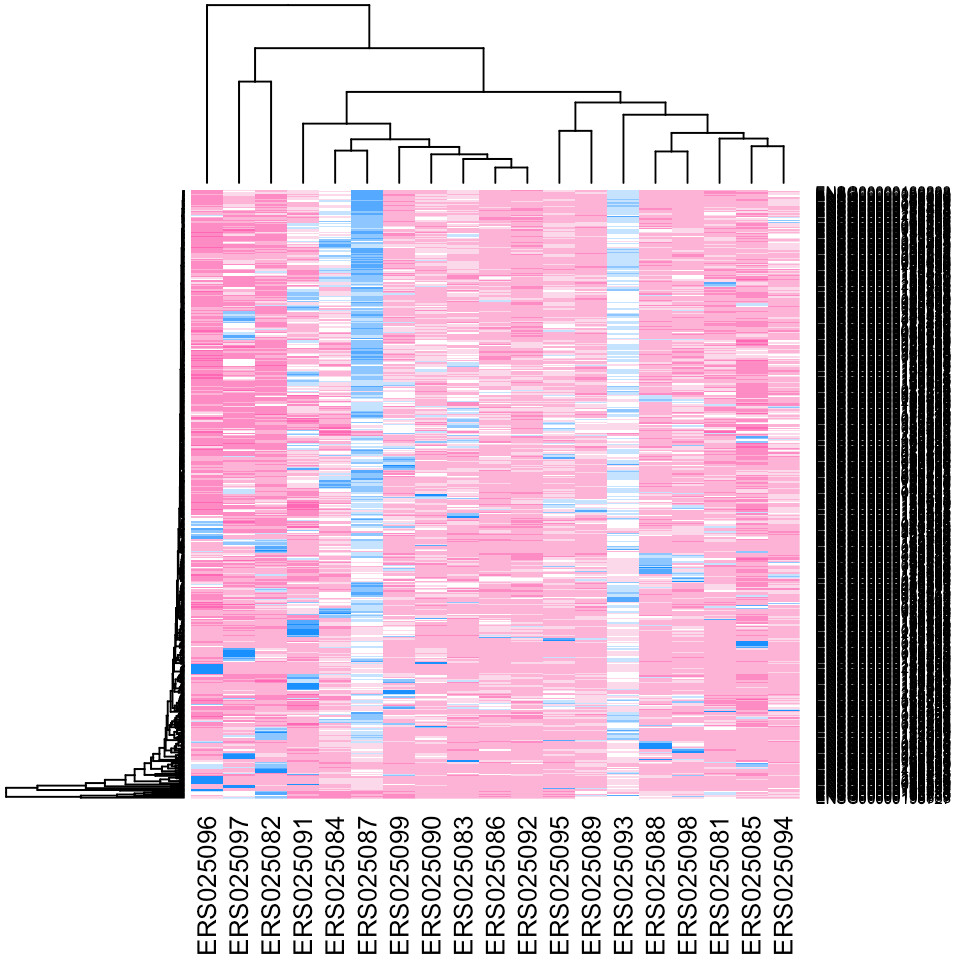
A common type of plot for genomics data is a heatmap. They are usually used for visualizing matrices. For example we can look at all genes with an average number of counts greater than 500:

```
ematrix = as.matrix(edata)[rowMeans(edata) > 10000,]
heatmap(ematrix)
```



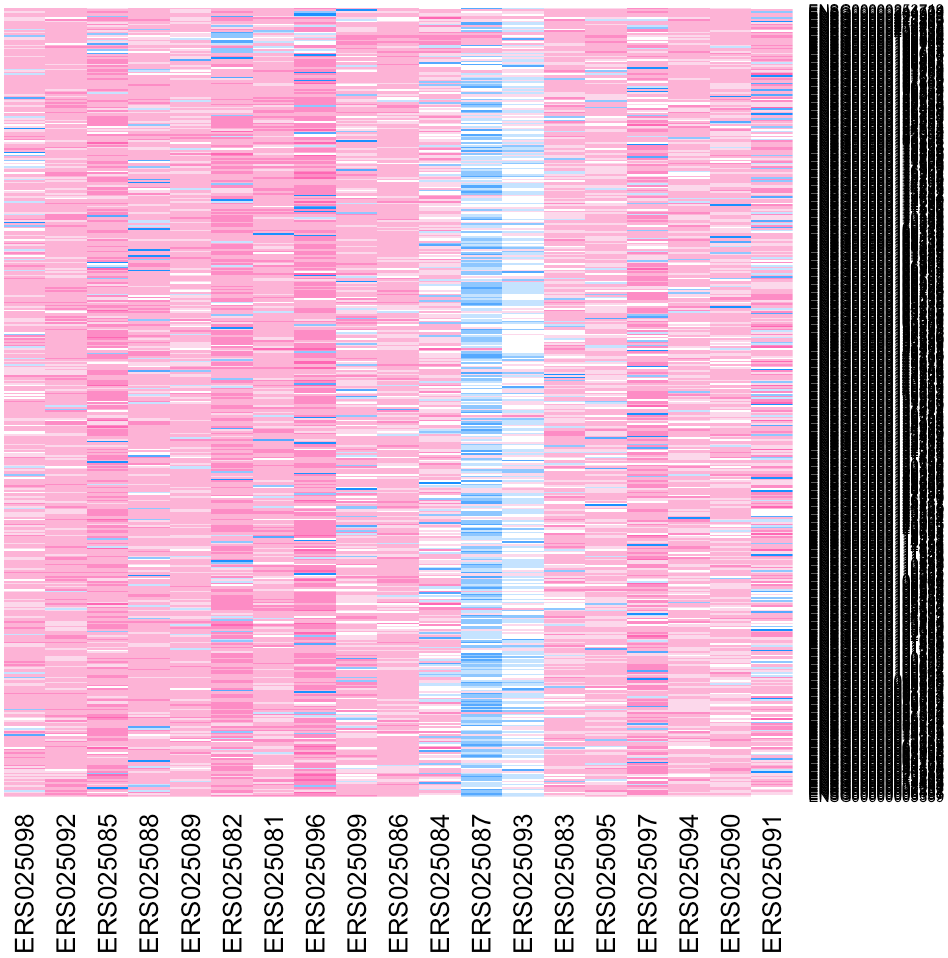

We might change the coloring since this one is a little hard to see. To do this you have to set up a color palette.

```
colramp = colorRampPalette(c(3,"white",2))(9)
heatmap(ematrix,col=colramp)
```



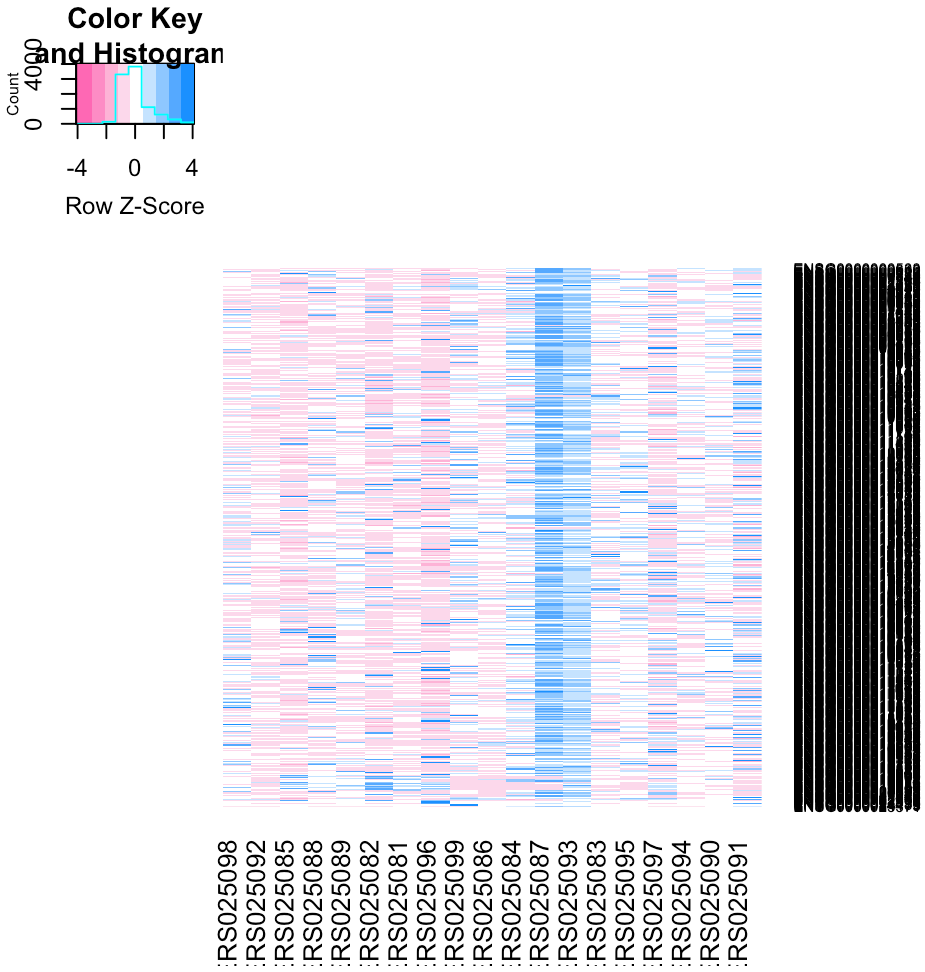

You might have noticed some automatic clustering here, you can turn that off (we'll learn more about it in a later lecture)

```
heatmap(ematrix,col=colramp,Rowv=NA,Colv=NA)
```




If you load the `gplots` package you can add a color scale with the `heatmap.2` package. Here we have to add some options to make the dendogram disappear, scale the data by rows, and remove a tracing plot.

```
heatmap.2(ematrix,col=colramp,Rowv=NA,Colv=NA,
          dendrogram="none", scale="row",trace="none")
```




# Session information

Here is the session information

```
devtools::session_info()
```

```
##  setting  value
##  version  R version 3.2.1 (2015-06-18)
##  system   x86_64, darwin10.8.0
##  ui       RStudio (0.99.447)
##  language (EN)
##  collate  en_US.UTF-8
##  tz       America/New_York
##
##  package           * version     date
##  acepack             1.3-3.3     2014-11-24
##  annotate            1.46.1      2015-07-11
##  AnnotationDbi     * 1.30.1      2015-04-26
##  assertthat          0.1         2013-12-06
##  BiasedUrn         * 1.06.1      2013-12-29
##  Biobase           * 2.28.0      2015-04-17
##  BiocGenerics      * 0.14.0      2015-04-17
##  BiocInstaller     * 1.18.4      2015-07-22
##  BiocParallel        1.2.20      2015-08-07
##  biomaRt             2.24.0      2015-04-17
##  Biostrings          2.36.3      2015-08-12
##  bitops              1.0-6       2013-08-17
##  bladderbatch      * 1.6.0       2015-08-26
##  broom             * 0.3.7       2015-05-06
##  caTools             1.17.1      2014-09-10
##  cluster             2.0.3       2015-07-21
##  colorspace          1.2-6       2015-03-11
##  corpcor             1.6.8       2015-07-08
##  curl                0.9.2       2015-08-08
##  DBI               * 0.3.1       2014-09-24
##  dendextend        * 1.1.0       2015-07-31
##  DESeq2            * 1.8.1       2015-05-02
##  devtools          * 1.8.0       2015-05-09
##  digest              0.6.8       2014-12-31
##  dplyr             * 0.4.3       2015-09-01
##  edge              * 2.1.0       2015-09-06
##  evaluate            0.7.2       2015-08-13
##  foreign             0.8-66      2015-08-19
##  formatR             1.2         2015-04-21
##  Formula           * 1.2-1       2015-04-07
```

```
##  futile.logger       1.4.1       2015-04-20
##  futile.options      1.0.0       2010-04-06
##  gdata               2.17.0      2015-07-04
##  genefilter        * 1.50.0      2015-04-17
##  geneLenDataBase   * 1.4.0       2015-09-06
##  geneplotter         1.46.0      2015-04-17
##  GenomeInfoDb      * 1.4.2       2015-08-15
##  GenomicAlignments   1.4.1       2015-04-24
##  GenomicFeatures     1.20.2      2015-08-14
##  GenomicRanges     * 1.20.5      2015-06-09
##  genstats          * 0.1.02      2015-09-05
##  ggplot2           * 1.0.1       2015-03-17
##  git2r               0.11.0      2015-08-12
##  GO.db               3.1.2       2015-09-06
##  goseq             * 1.20.0      2015-04-17
##  gplots            * 2.17.0      2015-05-02
##  gridExtra           2.0.0       2015-07-14
##  gtable              0.1.2       2012-12-05
##  gtools              3.5.0       2015-05-29
##  highr               0.5         2015-04-21
##  HistData          * 0.7-5       2014-04-26
##  Hmisc             * 3.16-0      2015-04-30
##  htmltools           0.2.6       2014-09-08
##  httr                1.0.0       2015-06-25
##  IRanges           * 2.2.7       2015-08-09
##  KernSmooth          2.23-15     2015-06-29
##  knitr             * 1.11        2015-08-14
##  lambda.r            1.1.7       2015-03-20
##  lattice           * 0.20-33     2015-07-14
##  latticeExtra        0.6-26      2013-08-15
##  lazyeval            0.1.10      2015-01-02
##  limma             * 3.24.15     2015-08-06
##  lme4                1.1-9       2015-08-20
##  locfit              1.5-9.1     2013-04-20
##  magrittr            1.5         2014-11-22
##  MASS              * 7.3-43      2015-07-16
##  Matrix            * 1.2-2       2015-07-08
##  MatrixEQTL        * 2.1.1       2015-02-03
##  memoise             0.2.1       2014-04-22
```

```
##  mgcv             * 1.8-7       2015-07-23
##  minqa              1.2.4       2014-10-09
##  mnormt             1.5-3       2015-05-25
##  munsell            0.4.2       2013-07-11
##  nlme             * 3.1-122     2015-08-19
##  nloptr             1.0.4       2014-08-04
##  nnet               7.3-10      2015-06-29
##  org.Hs.eg.db     * 3.1.2       2015-07-17
##  plyr               1.8.3       2015-06-12
##  preprocessCore   * 1.30.0      2015-04-17
##  proto              0.3-10      2012-12-22
##  psych              1.5.6       2015-07-08
##  qvalue           * 2.0.0       2015-04-17
##  R6                 2.1.1       2015-08-19
##  RColorBrewer       1.1-2       2014-12-07
##  Rcpp             * 0.12.0      2015-07-25
##  RcppArmadillo    * 0.5.400.2.0 2015-08-17
##  RCurl              1.95-4.7    2015-06-30
##  reshape2           1.4.1       2014-12-06
##  rmarkdown          0.7         2015-06-13
##  rpart              4.1-10      2015-06-29
##  Rsamtools          1.20.4      2015-06-01
##  RSkittleBrewer   * 1.1         2015-09-05
##  RSQLite          * 1.0.0       2014-10-25
##  rstudioapi         0.3.1       2015-04-07
##  rtracklayer        1.28.9      2015-08-19
##  rversions          1.0.2       2015-07-13
##  S4Vectors        * 0.6.5       2015-09-01
##  scales             0.3.0       2015-08-25
##  snm                1.16.0      2015-04-17
##  snpStats         * 1.18.0      2015-04-17
##  stringi            0.5-5       2015-06-29
##  stringr            1.0.0       2015-04-30
##  survival         * 2.38-3      2015-07-02
##  sva              * 3.14.0      2015-04-17
##  tidyr              0.2.0       2014-12-05
##  UsingR           * 2.0-5       2015-08-06
##  whisker            0.3-2       2013-04-28
##  XML                3.98-1.3    2015-06-30
```

```
##  xml2                0.1.2       2015-09-01
##  xtable              1.7-4       2014-09-12
##  XVector             0.8.0       2015-04-17
##  yaml                2.1.13      2014-06-12
##  zlibbioc            1.14.0      2015-04-17
##  source
##  CRAN (R 3.2.0)
##  Bioconductor
##  Bioconductor
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.0)
##  Bioconductor
##  Bioconductor
##  Bioconductor
##  Bioconductor
##  Bioconductor
##  Bioconductor
##  CRAN (R 3.2.0)
##  Bioconductor
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
##  Bioconductor
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.2)
##  Github (jdstorey/edge@a1947b5)
##  CRAN (R 3.2.2)
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
```

```
##  Bioconductor
##  Bioconductor
##  Bioconductor
##  Bioconductor
##  Bioconductor
##  Bioconductor
##  Bioconductor
##  local
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.2)
##  Bioconductor
##  Bioconductor
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
##  Bioconductor
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.2)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.0)
##  Bioconductor
##  CRAN (R 3.2.2)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
```

```
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
##  Bioconductor
##  CRAN (R 3.2.1)
##  Bioconductor
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
##  Bioconductor
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.1)
##  Bioconductor
##  Github (alyssafrazee/RSkittleBrewer@0a96a20)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.0)
##  Bioconductor
##  CRAN (R 3.2.1)
##  Bioconductor
##  CRAN (R 3.2.2)
##  Bioconductor
##  Bioconductor
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
##  Bioconductor
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.2)
##  CRAN (R 3.2.0)
##  CRAN (R 3.2.1)
##  CRAN (R 3.2.2)
##  CRAN (R 3.2.0)
##  Bioconductor
```

```
##  CRAN (R 3.2.0)
##  Bioconductor
```

It is also useful to compile the time the document was processed. This document was processed on: 2015-09-06.