

# PM 579: Statistical Analysis of High-Dimensional Data

## | Homework 2

Saket Choudhary skchoudh@usc.edu

6/9/2016

### Reading data

```
library(limma)
library(knitr)
library(sva)

## Loading required package: mgcv

## Loading required package: nlme

## This is mgcv 1.8-12. For overview type 'help("mgcv-package")'.

## Loading required package: genefilter

##
## Attaching package: 'genefilter'

## The following object is masked from 'package:base':
## 
##     anyNA

sfiles <- c('../data/Stallcup/mySample Probe Profile FinalReport.txt')
cfiles <- c('../data/Stallcup/myControl Probe Profile FinalReport.txt')
rawdata = read.ilmn(files=sfiles, ctrlfiles=cfiles)

## Reading file ../data/Stallcup/mySample Probe Profile FinalReport.txt . . .
## Reading file ../data/Stallcup/myControl Probe Profile FinalReport.txt . . .
```

### Sanity Checks

```
dim(rawdata$E)

## [1] 25209    48

(table(rawdata$genes$status))

##
##          BIOTIN        CY3_HYB      HOUSEKEEPING
##             2              6                  1
##          LABELING_LOW_STRINGENCY_HYB      NEGATIVE
##             2                 8                  664
##          regular
##             24526
```

## Read targets

```
## Read targets
targetfile <- './data/Stallcup/SampleChars.csv'
targets <- read.csv(targetfile, row.names = 1)
## Assign targets to raw data
rawdata$targets=targets[colnames(rawdata),]
## Sanity Check
head(rawdata$targets)

##          treatment time
## 4849554032_A Control2   1
## 4849554032_B      X   2
## 4849554032_C      X   3
## 4849554032_D      X   1
## 4849554032_E Control2   1
## 4849554032_F      X   2

kable(table(rawdata$targets$treatment, rawdata$targets$time) )
```

	1	2	3
Control1	4	4	4
Control2	4	4	4
X	4	4	4
Y	4	4	4

## QC Figures

```
stemplot <- function(x,y,xlabels,pch=16,linecol=1,clinecol=1,...){
if (missing(y)){
  y = x
  x = 1:length(x) }
  plot(x,y,xaxt="n", pch=pch,...)

  for (i in 1:length(x)){
    lines(c(x[i],x[i]), c(0,y[i]),col=linecol)
  }
  lines(c(x[1]-2,x[length(x)]+2), c(0,0),col=clinecol)
  axis(1, at=1:length(xlabels),
        labels=xlabels,
        cex.axis=0.6,
        las=2)
}
par(mar=c(5,6,4,2)+0.1)
par(mgp=c(5,1,0))

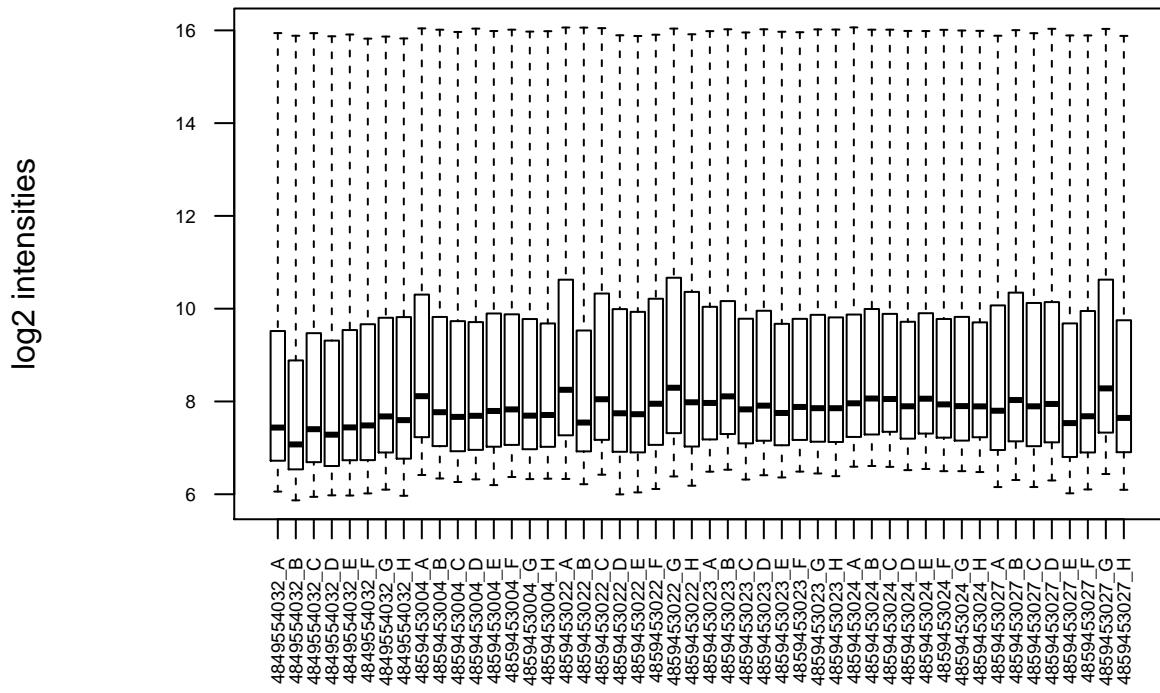
boxplot(log2(rawdata$E[rawdata$genes>Status=="regular",]),
        range=0,
```

```

xlab="Arrays",
ylab="log2 intensities",
main="Regular probes",
cex.axis=0.6,
las=2)

```

## Regular probes

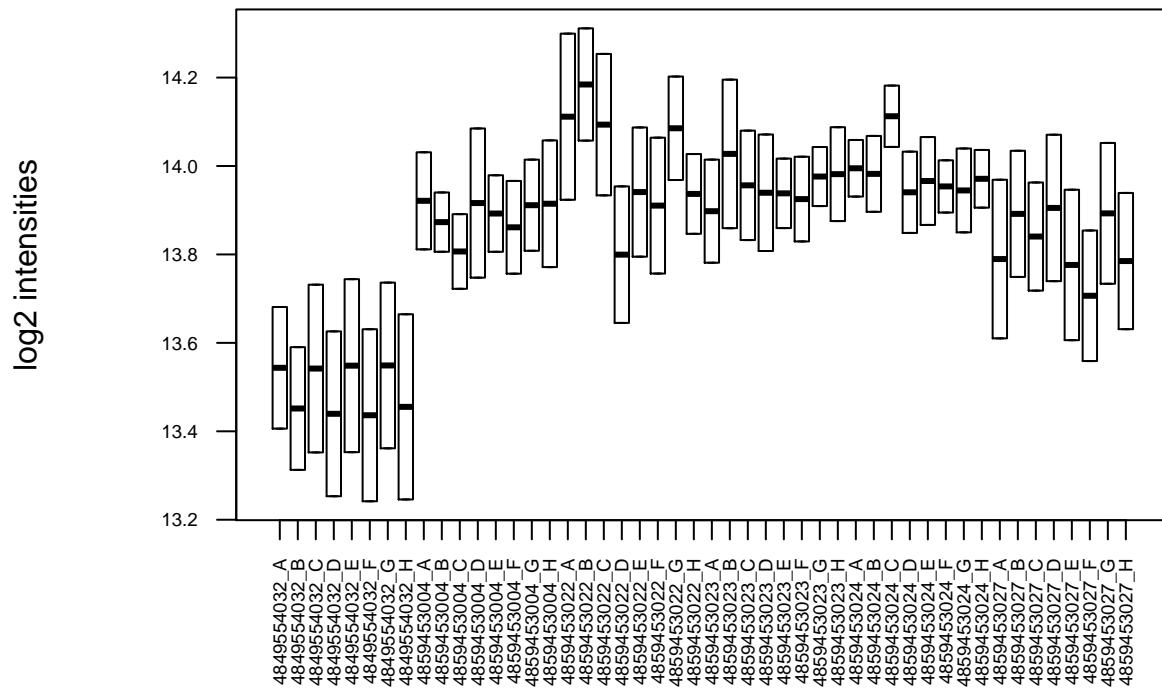


```

boxplot(log2(rawdata$E[rawdata$genes$status=="BIOTIN",]),
        range=0,
        xlab="Arrays",
        ylab="log2 intensities",
        main="BIOTIN probes",
        cex.axis=0.6,
        las=2)

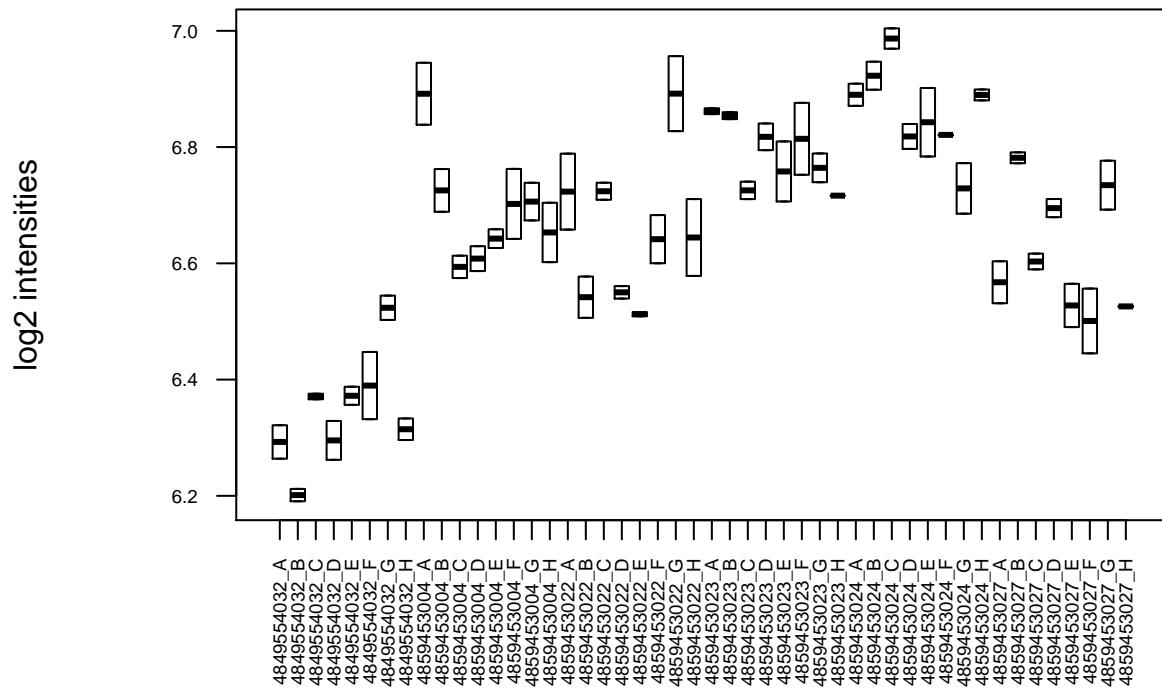
```

## BIOTIN probes



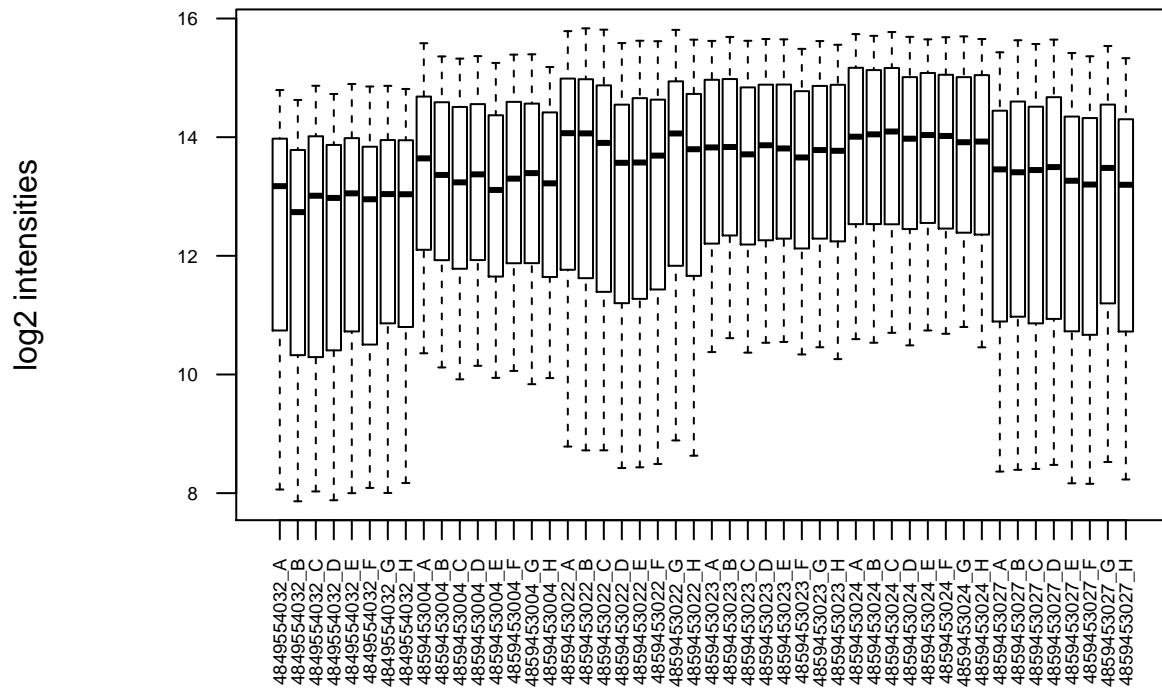
```
boxplot(log2(rawdata$E[rawdata$genes$status=="LABELING",]),  
        range=0,  
        xlab="Arrays",  
        ylab="log2 intensities",  
        main="LABELING probes",  
        cex.axis=0.6,  
        las=2)
```

## LABELING probes



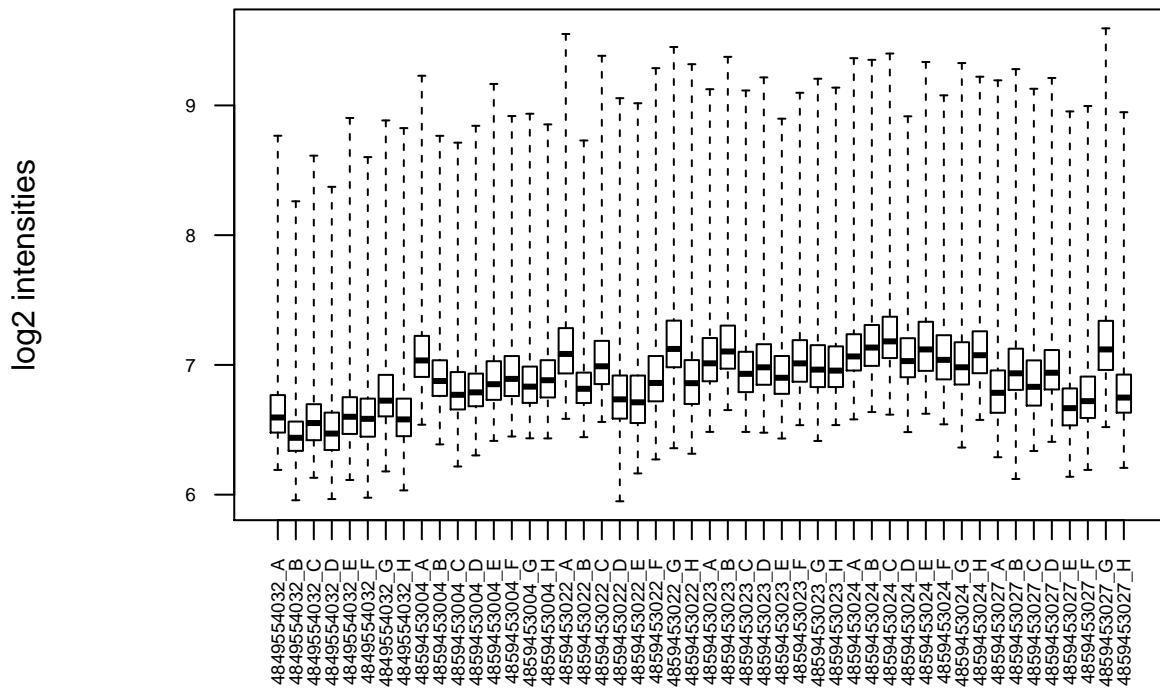
```
boxplot(log2(rawdata$E[rawdata$genes$status=="LOW_STRINGENCY_HYB",]),  
        range=0,  
        xlab="Arrays",  
        ylab="log2 intensities",  
        main="LOW_STRINGENCY_HYB probes",  
        cex.axis=0.6,  
        las=2)
```

## LOW\_STRINGENCY\_HYB probes



```
boxplot(log2(rawdata$E[rawdata$genes$status=="NEGATIVE",]),
        range=0,
        xlab="Arrays",
        ylab="log2 intensities",
        main="NEGATIVE probes",
        cex.axis=0.6,
        las=2)
```

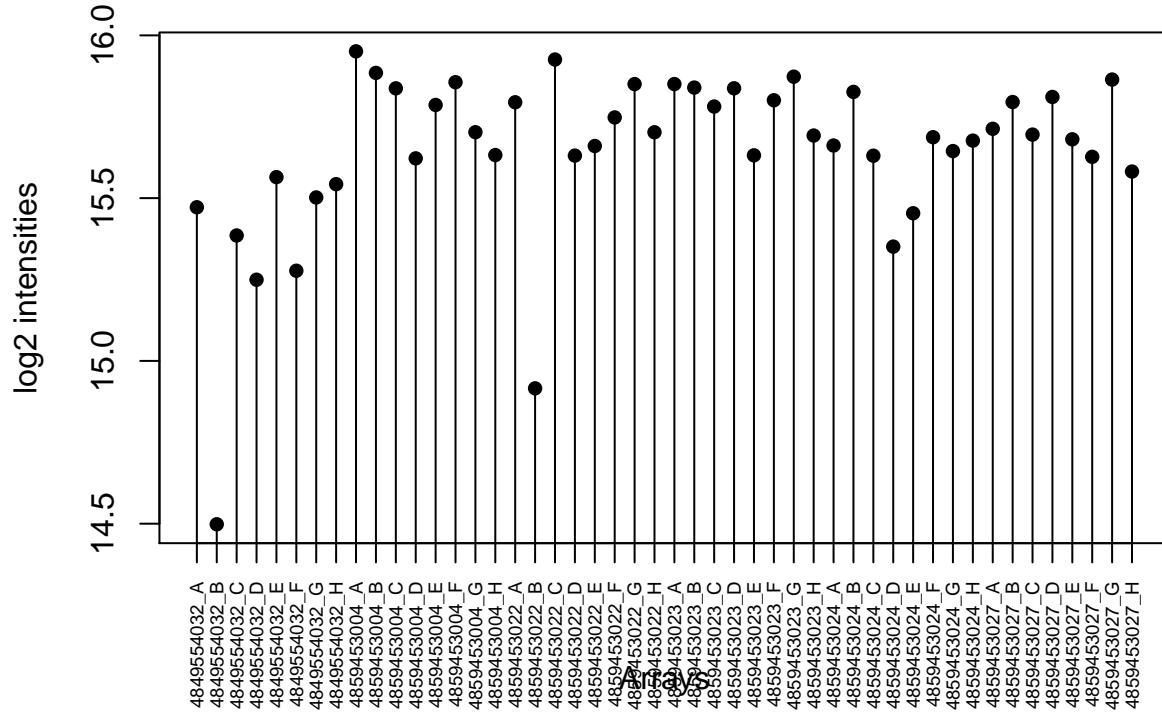
## NEGATIVE probes



There is only one housekeeping gene for every array, and hence we use a stem plot to look at the distribution

```
stemplot(log2(rawdata$E[rawdata$genes$status=="HOUSEKEEPING",]),
         xlab="Arrays",
         ylab="log2 intensities",
         xlabel = rownames(rawdata$targets),
         main="HOUSEKEEPING probes")
```

## HOUSEKEEPING probes



From the QC plots, we observe that regular probes have similar distribution across arrays, the BIOTIN probes exhibit high intensities across arrays and also seem to exhibit array specific bias. Array specific bias is also evident from the LABELLING and LOW\_STRINGENCY\_HYB probes. There are 2 labelling probes on each array and they seem to have moderate intensities while exhibiting array specific bias. The LOW\_STRINGENCY\_HYB probes show high intensity throughout which is expected but also exhibit array specific bias with lower intensities in the first and last 8 arrays. Negative probes also have array specific bias, but otherwise have low values throughout as compared to the regular probes which is expected.

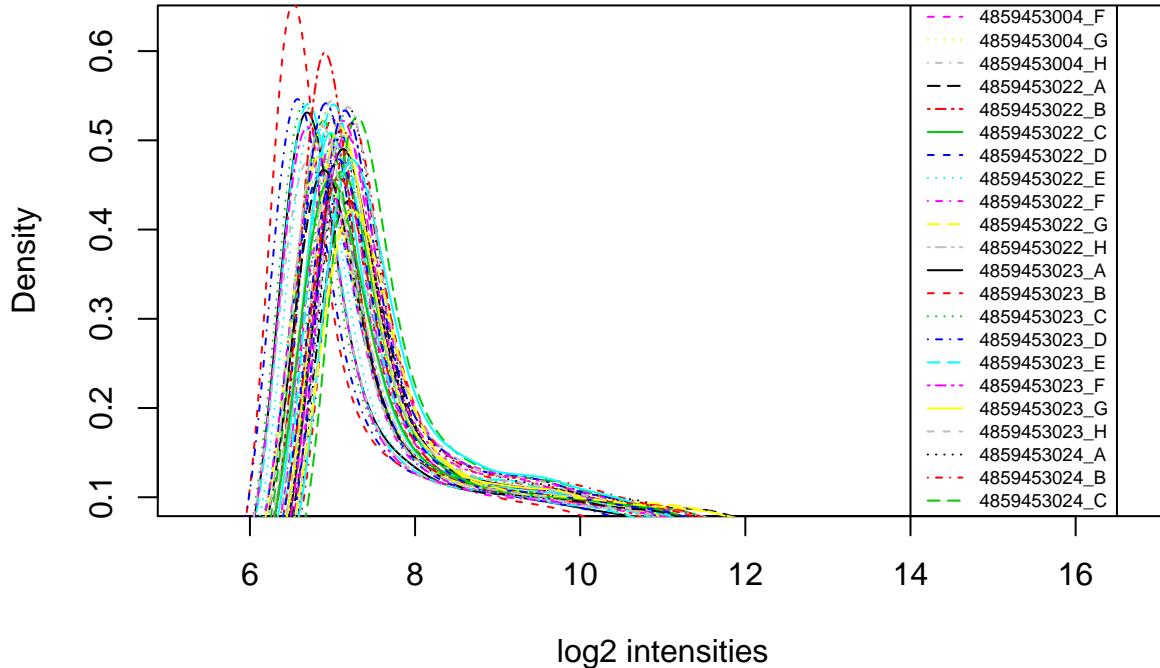
Housekeeping probes have high expression throughout all arrays but exhibit array specific bias. A particular case is of 4849554032\_B array which has low value here but also looks like a consistent outlier in all QC analysis.

```

plot(density(log2(rawdata$E[rawdata$genes$status=="regular",1])),
      xlab="log2 intensities",
      ylab="Density",
      main="Density Plot of Intensities", ylim=c(0.1,0.63))
na=length(rownames(rawdata$target))
for (i in 2:na)
  lines(density(log2(rawdata$E[rawdata$genes$status=="regular",i])), col=i, lty=i)
legend(14,1,rownames(rawdata$target),lty=1:48,col=1:48,cex=.6)

```

## Density Plot of Intensities



Also evident from the density plot 4849554032\_B looks like an outlier with right skewed distribution of intensities.

### Background correction

```
rawdata.bgcorrected <- neqc(rawdata)
dim(rawdata.bgcorrected)
```

```
## [1] 24526     48
```

Hence 683 probes were removed, this includes all control probes, BIOTIN(2), CY3\_HYB(6), HOUSEKEEPING(1), LABELING(2), LOW\_STRINGENCY\_HYB(8), NEGATIVE(664)

### Checking for batch effects

```
rawdata.bgcorrected <- rawdata.bgcorrected[, order(rawdata.bgcorrected$targets$treatment,
batch           <- unclass(rawdata.bgcorrected$targets$time)
treatment       <- unclass(rawdata.bgcorrected$targets$treatment)

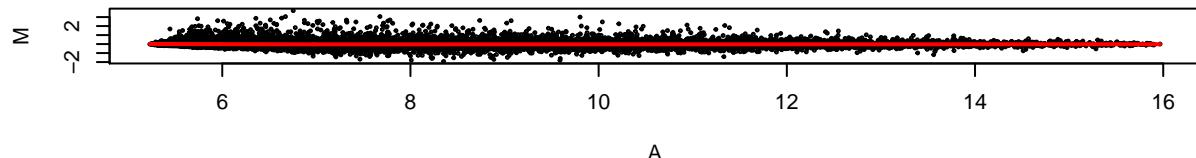
#plot(A,M[,1],ylab="M",cex=.5)
#lines(lowess(A,M[,1]),col=2,lwd=4)
par(mfrow=c(3,1))
for (i in 1:3) {
  ## Since the arrays are processed like 16 in a batch
  ## I use only samples from the same batch to draw MA plots
```

```

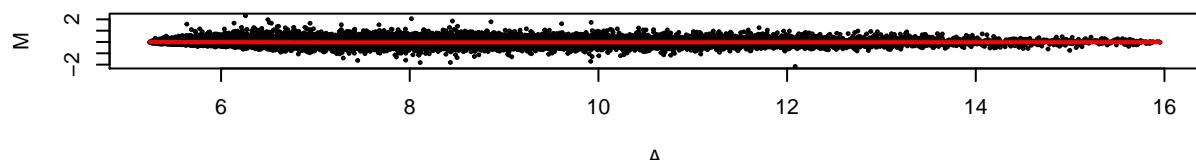
## Each MA plot is of the 1st array in that batch to avoid drawing 16 x 3 plots
idx <- rawdata.bgcorrected$targets$time==i
batchdata <- rawdata.bgcorrected[, idx]
A <- apply(batchdata$E, 1, median)
M <- batchdata$E-A
plot(A, M[, 1], ylab="M", main=paste("Batch ", i, " 1st Array vs Overall Avg"), cex=.3)
lines(lowess(A, M[, 1]), col=2, lwd=2)
}

```

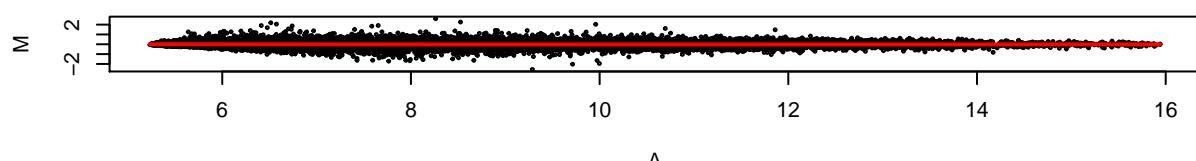
**Batch 1 1st Array vs Overall Avg**



**Batch 2 1st Array vs Overall Avg**



**Batch 3 1st Array vs Overall Avg**



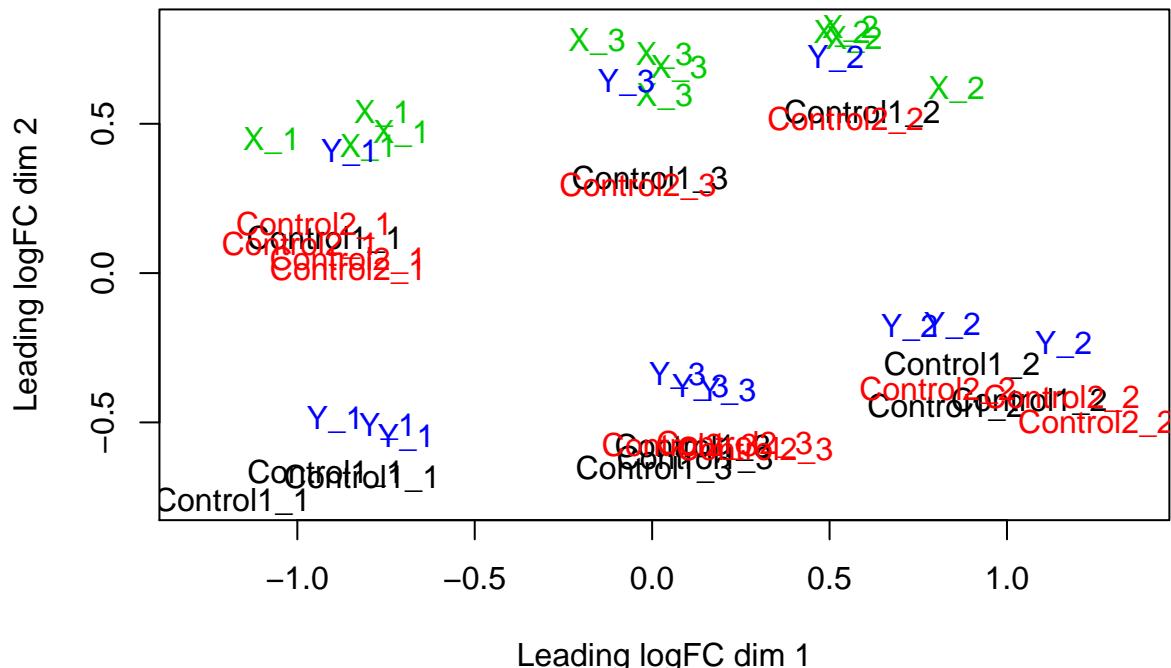
## Before Batch Correction

```

plotMDS(rawdata.bgcorrected,
        labels=paste(rawdata.bgcorrected$targets$treatment, unclass(rawdata.bgcorrected$targets$time), ),
        col=unclass(rawdata.bgcorrected$targets$treatment), main="MDS plot before batch correction colored by treatment")

```

## MDS plot before batch correction colored by treatment

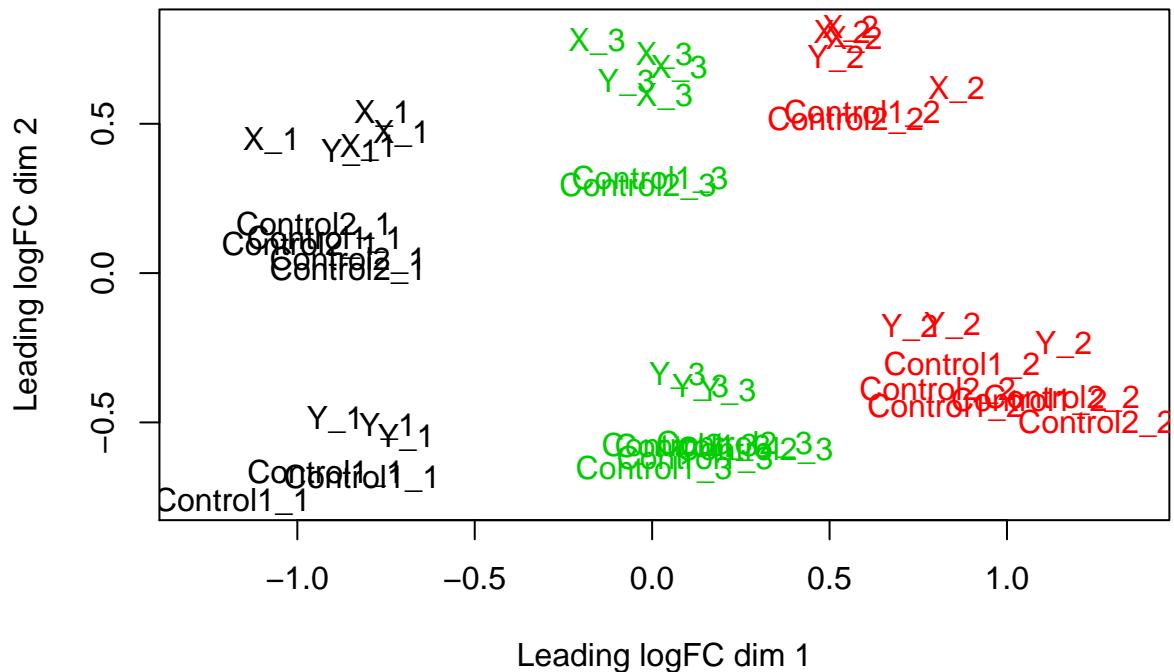


Clearly there is a batch effect. The MDS plot can be partitioned vertically into 3 partitions such that Batch 1 samples are fall along the left most partition, the Batch 2 samples fall along the middle partition and Batch 2 samples fall along the right most partition.

We replot it by coloring by batches:

```
plotMDS(rawdata.bgcorrected,
        labels= paste(rawdata.bgcorrected$targets$treatment, unclass(rawdata.bgcorrected$targets$time),
        col=unclass(rawdata.bgcorrected$targets$time), main="MDS plot before batch correction colored by
```

## MDS plot before batch correction colored by batch



## Batch Correction

```

model <- model.matrix(~as.factor(treatment), data=as.data.frame(treatment))
rawdata.bgcorrected.bc <- ComBat(dat=rawdata.bgcorrected$E, batch=batch, mod=model)

## Found 3 batches
## Adjusting for 3 covariate(s) or covariate level(s)
## Standardizing Data across genes
## Fitting L/S model and finding priors
## Finding parametric adjustments
## Adjusting the Data

plotMDS(rawdata.bgcorrected.bc,
        labels=paste(rawdata.bgcorrected$targets$treatment, unclass(rawdata.bgcorrected$targets$time),
                     col=unclass(rawdata.bgcorrected$targets$treatment), main="MDS Plot post batch correction")
    )

```

### MDS Plot post batch correction

