Gene Expression Data Analysis and Visualization
410.671
HW #2

For this assignment, we will be evaluating different normalization methods on 2-channel arrays in which 4 biological samples were run. The study is from GEO and the description of the experiment is provided as follows.

Series GSE12050: Subcutaneous adipose tissue from lean and obese subjects

Obtaining adipose tissue samples are paramount to the understanding of human obesity. We have examined the impact of needle-aspirated and surgical biopsy techniques on the study of subcutaneous adipose tissue (scAT) gene expression in both obese and lean subjects. Biopsy sampling methods have a significant impact on data interpretation and revealed that gene expression profiles derived from surgical tissue biopsies better capture the significant changes in molecular pathways associated with obesity. We hypothesize that this is because needle biopsies do not aspirate the fibrotic fraction of scAT; which subsequently results in an under-representation of the inflammatory and metabolic changes that coincide with obesity. This analysis revealed that the biopsy technique influences the gene expression underlying the biological themes commonly discussed in obesity (e.g. inflammation, extracellular matrix, metabolism, etc), and is therefore a caveat to consider when designing microarray experiments. These results have crucial implications for the clinical and physiopathological understanding of human obesity and therapeutic approaches.

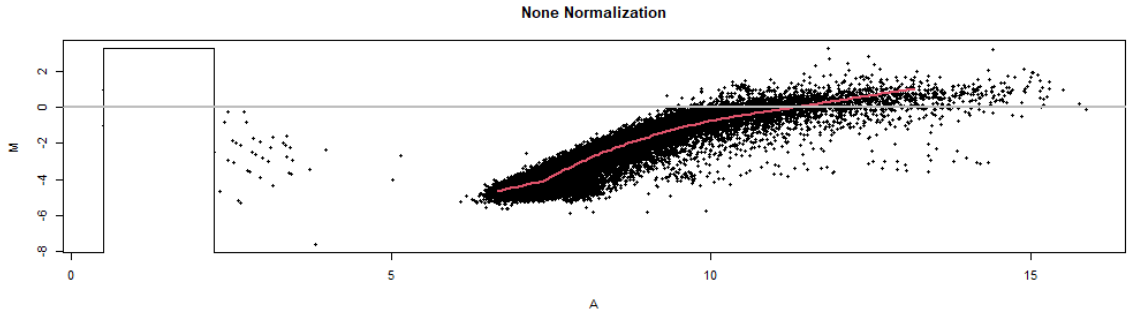
We will be working with 4 lean subjects from which a needle biopsy was taken.

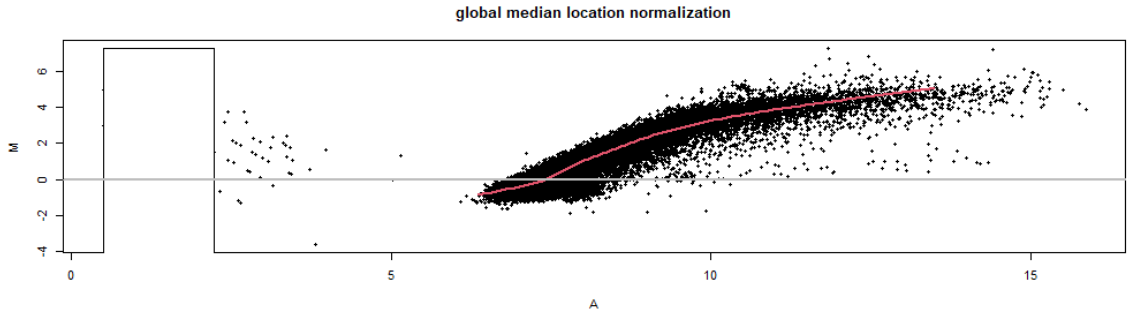
1.) First load the marray library, then load the 4 GenePix files, making sure to extract the foreground and background median values from the Cy5 and Cy3 channels. (2.5 pts)

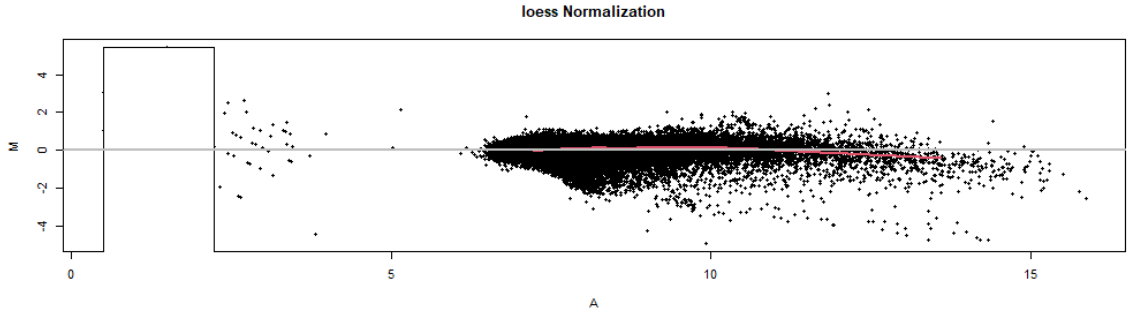
```
BiocManager::install('marray')
library(dplyr)
library(limma)
library(dplyr)
library(marray)
BiocManager::install("arrayQuality")
library(arrayQuality)
data("swirl")
if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install(version = "3.15")

datadir <- system.file("Samsfish", package = "marray")
gprfile <- read.GenePix (path='C:/study/JHU',skip=33)
cy3.f <- gprfile@maGf
cy3.b <- gprfile@maGb
cy5.f <- gprfile@maRf
cy5.b <- gprfile@maRb
```

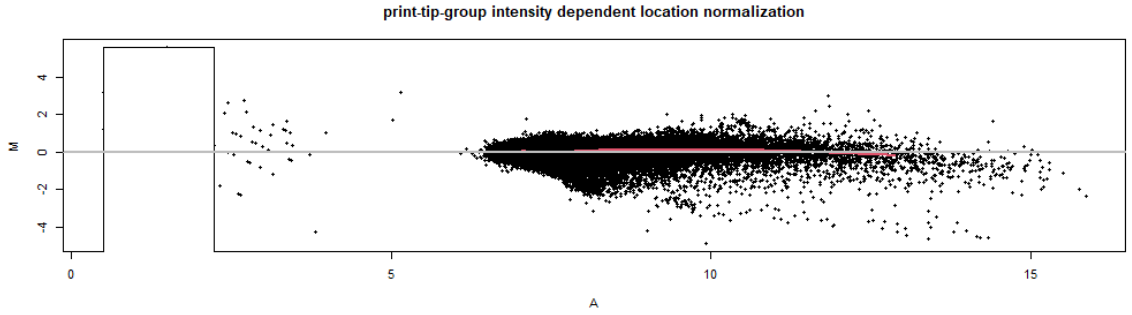
2.) Normalize each array using median global, loess, and print-tip-group loess methods. Then plot MvA plots of all 4 arrays comparing no normalization to the other 3 normalization approaches. (2 pts)

```
gprnormMedian <- maNorm(gprfile[,1:4],norm = 'median',span=0.45)
gprnormLoess <- maNorm(gprfile[,1:4],norm = 'loess',span=0.45)
gprnormPTGloess <- maNorm(gprfile[,1:4],norm = 'printTipLoess',span=0.45)
gprnormNon <- maNorm(gprfile[,1:4],norm = 'none',span=0.45)

par(mfrow=c(4,1))
maPlot(gprnormNon,main='None Normalization')

maPlot(gprnormMedian,main='global median location normalization')
maPlot(gprnormLoess,main='loess Normalization')
maPlot(gprnormPTGloess,main='print-tip-group intensity dependent location normalization')
```



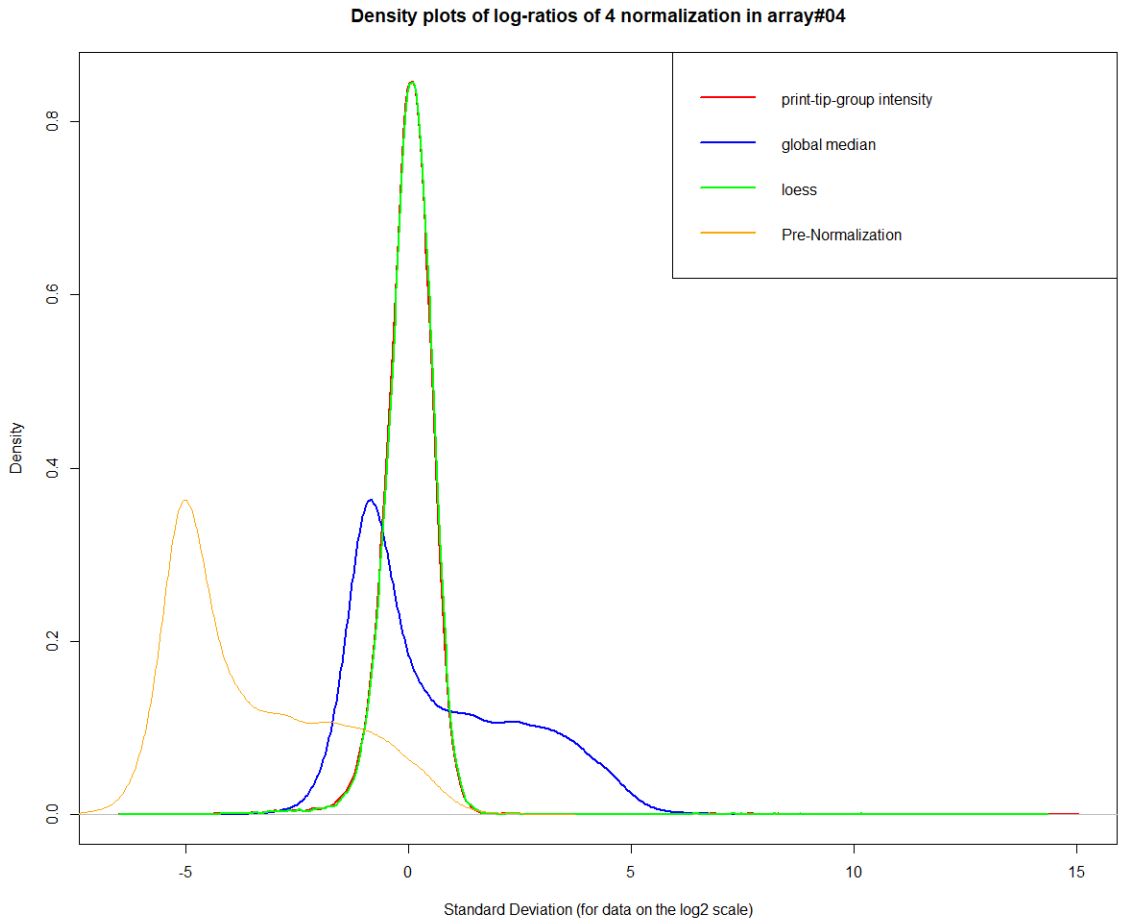






3.) Plot density plots of the log ratio values for each normalization (and pre normalization) for only array #4. Put them all on the same plot. Make sure to label the axes and provide a legend. (2 pts)

```
plot(density(rm.na(maM(gprnorm04PTGloess[,1]))),lwd=2, col='red',
main="Density plots of log-ratios of 4 normalization in array#04",
xlab = "Standard Deviation (for data on the log2 scale)",
ylab = "Density")
lines(density(rm.na(maM(gprnorm04Median[,1]))),lwd=2,col='blue')
lines(density(rm.na(maM(gprnorm04Loess[,1]))),lwd=2,col='green')
lines(density(rm.na(maM(gprfile[,4]))),col='orange')
legend("topright",
legend = c("print-tip-group intensity","global median","loess","Pre-Normalization"),
col=c('red','blue','green','orange'),lty=1,lwd=2)
```



4.) Based on the plots generated so far, which normalization do you think is most preferred for this dataset? (2 pts)

The print-tip and loess normalization seem more suit for the normal distribution so I prefer these 2 most.

5.) Research has demonstrated that often a single channel, background subtracted provides as good a normalization as using both channels. To test this, we will be utilizing the fact that these 4 samples are replicates and calculate the correlation between them. So, first extract the Cy5 foreground and background values for each of the 4 arrays and subtract the background from the foreground values, then $\log_2$ transform these values. Then calculate global median normalization on these 4 arrays using these background

subtracted Cy5 values. Hint, you need to use the median of each array to scale, such that after normalization, all arrays will have a median of 1. (4 pts)

```
redB <- gprfile@maRb[,1:4]
redF <- gprfile@maRf[,1:4]
subRed <- redF-redB
subRed[subRed<0]=NA
logred <- log2(subRed)
redmedian<- apply(logred,2,median,na.rm=T)
redNor <- sweep(logred,2,redmedian)
2**median(redNor[,1],na.rm=T)
2**median(redNor[,2],na.rm=T)
2**median(redNor[,3],na.rm=T)
2**median(redNor[,4],na.rm=T)
```

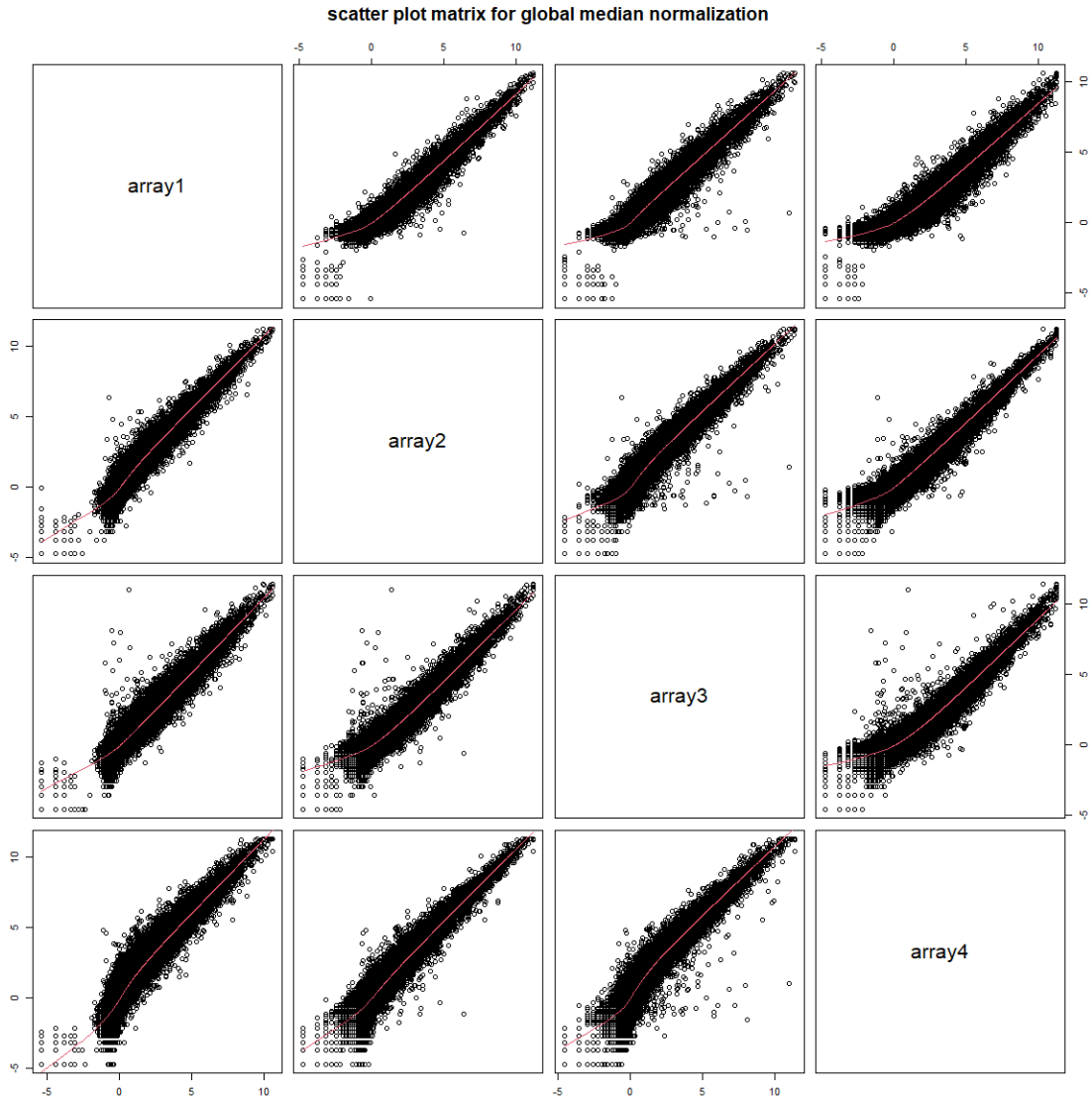
```
> 2**median(redNor[,1],na.rm=T)
[1] 1
> 2**median(redNor[,2],na.rm=T)
[1] 1
> 2**median(redNor[,3],na.rm=T)
[1] 1
> 2**median(redNor[,4],na.rm=T)
[1] 1
```

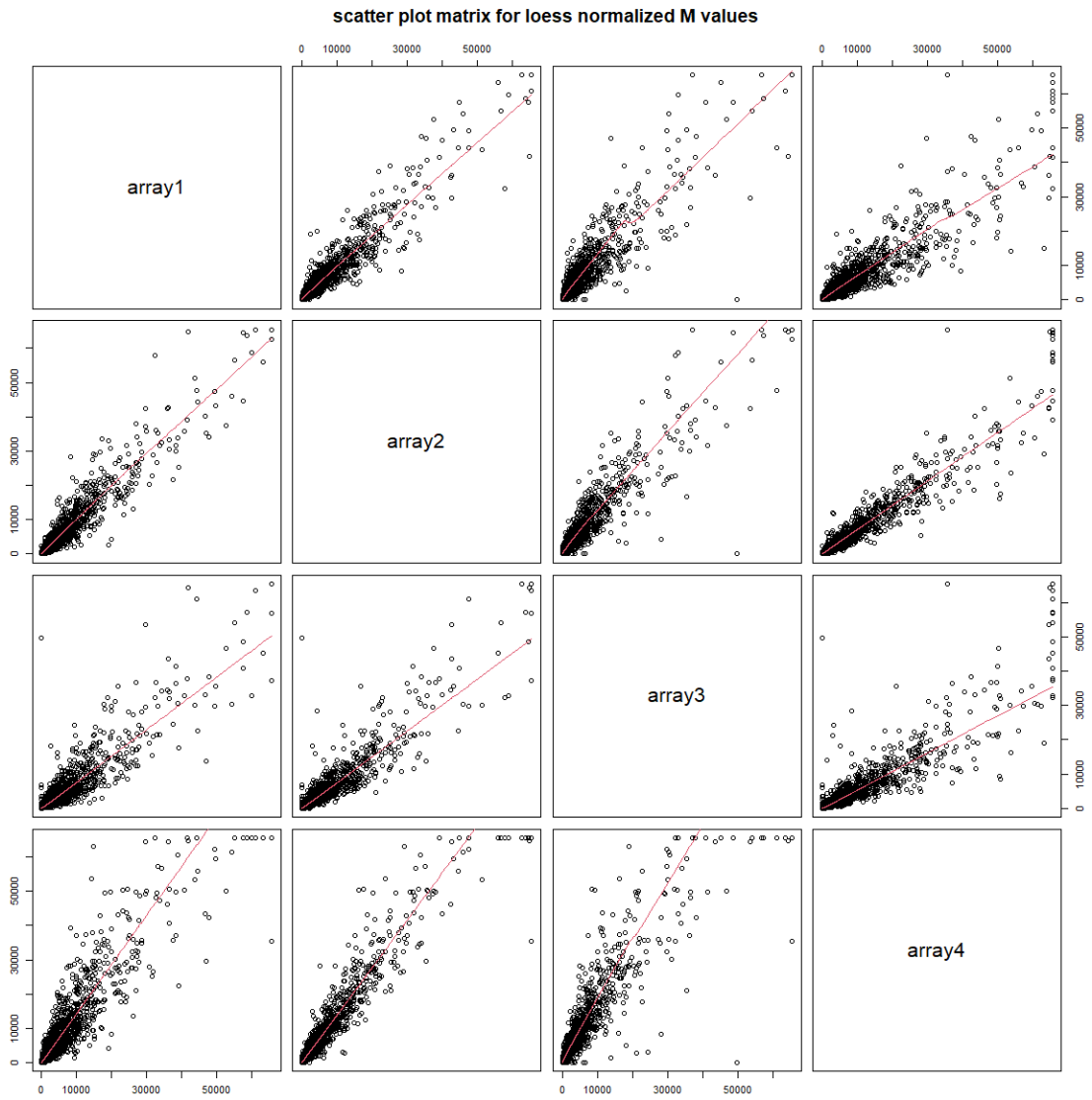
6.) Next calculate a Spearman's rank correlation between all 4 arrays that you normalized in #5 and do the same with the M values from loess normalized data that you generated in #2. Plot a scatter plot matrix for each of the two normalizations (pairs() function), and be sure to label the arrays and title the plot. Print the correlation coefficients to the screen. (4 pts)

```
colnames(redNor)<- c("array1","array2","array3","array4")
cor(redNor,method='spearman',use='complete.obs')
lossM <- gprnormLoess@maM
colnames(lossM)<- c("array1","array2","array3","array4")
cor(lossM,method='spearman',use='complete.obs')
```

```
> cor(redNor,method='spearman',use='complete.obs')
          array1    array2    array3    array4
array1 1.0000000 0.8957946 0.8784760 0.8985979
array2 0.8957946 1.0000000 0.8758774 0.9075121
array3 0.8784760 0.8758774 1.0000000 0.8848059
array4 0.8985979 0.9075121 0.8848059 1.0000000
> lossM <- gprnormLoess@maM
> colnames(lossM)<- c("array1","array2","array3","array4")
> cor(lossM,method='spearman',use='complete.obs')
          array1    array2    array3    array4
array1 1.0000000 0.6972744 0.7576741 0.6966909
array2 0.6972744 1.0000000 0.7324231 0.7131769
array3 0.7576741 0.7324231 1.0000000 0.7496356
array4 0.6966909 0.7131769 0.7496356 1.0000000
```

```
pairs(~array1+array2+array3+array4,panel = panel.smooth,data=redNor,main="scatter plot matrix for global median normalization")
pairs(~array1+array2+array3+array4,panel = panel.smooth,data=subRed,main="scatter plot matrix for loess normalized M values")
```


scatter plot matrix for global median normalization
array1
array2
array3
array4
-5
0
5
10



7.) Now we want to compare these normalizations to quantile normalized data to see if we gain anything by leveraging the distributions across all 4 arrays. Carry out the steps in the lecture or use the paper from Bolstad *et al*. entitled: "A comparison of normalization methods for high density oligonucleotide array data based on variance and bias" (on the course website), but we are only going to conduct this on the Cy5 channel. The basic steps are as follows (these 6 steps are calculated on non-logged data; the data is logged after these steps are carried out): (8 pts)

1. Subtract the foreground – background for each of the 4 chips for only the Cy5 channel. This should all be on the linear or raw scale (no logging yet).

```
redB <- gprfile@maRb[,1:4]
redF <- gprfile@maRf[,1:4]
subRed <- redF-redB
```

2. Sort each column independently in this new matrix.

```
sorted.sub.red <- apply(subRed, 2, sort)
```

3. Calculate row means for the sorted matrix

```
sorted.sub.red.mean <- rowMeans(sorted.sub.red)
```

4. Create a new matrix with each row having the same values as the sorted row mean vectors from step #3 (you should have a new R matrix)

```
red.row.mean <- matrix(data = c(sorted.sub.red.mean,sorted.sub.red.mean,sorted.sub.red.mean,sorted.sub.red.mean), ncol = 4)
```

5. Rank the columns independently on the original background subtracted matrix (from step #1)
Hint: use the rank() function with the argument ties="first" or order()

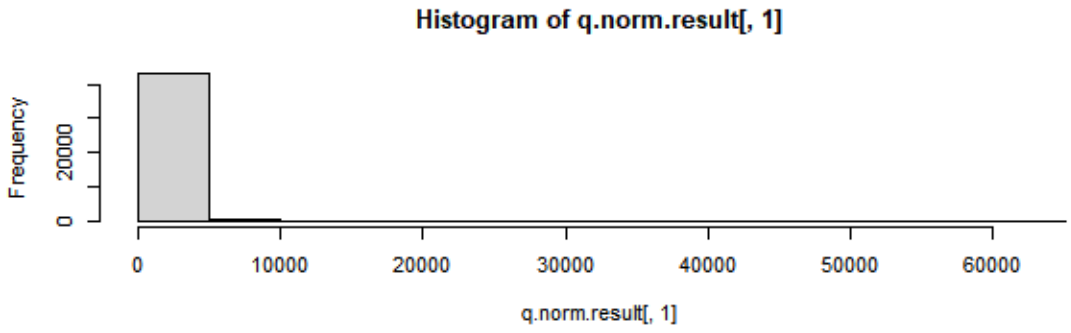
```
sub.red.rank <- apply(subRed,2,rank,ties='first')
```

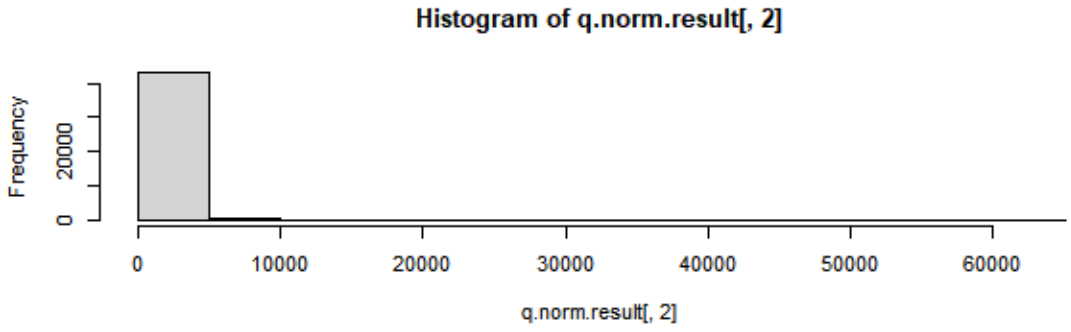
6. Reorder the columns in the new matrix from step #4 using the ranks from step #5

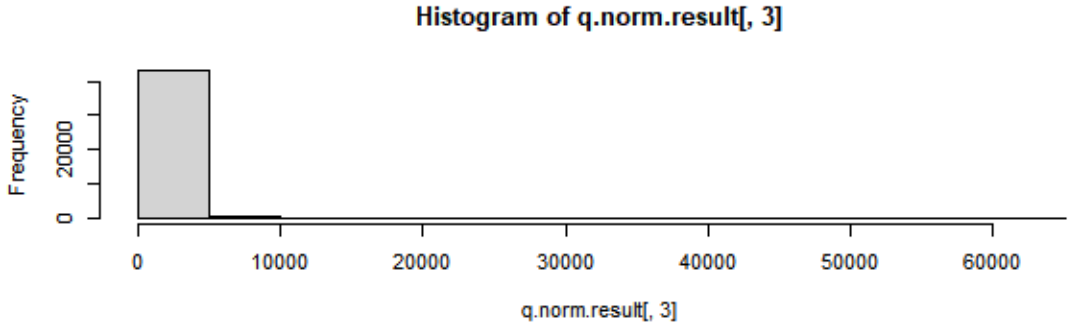
```
sub.red.rank <- apply(subRed,2,rank,ties.method='first')

index_to_mean <- function(input.rank, input.mean){
  return(input.mean[input.rank])
}

q.norm.result <- apply(sub.red.rank, 2, index_to_mean, input.mean=sorted.sub.red.mean)
q.norm.result
```

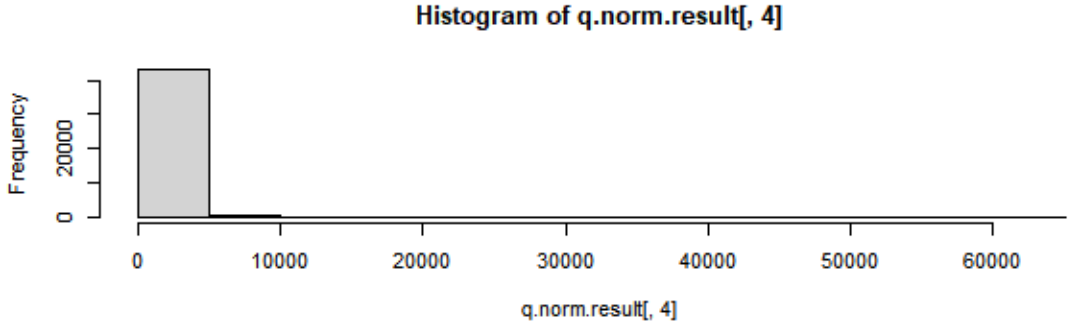
To verify that each array has the same distribution, use the hist() function to look at various arrays (e.g., hist(c5.norm[,1]); hist(c5.norm[,2]); etc.). Slight differences in distributions are a result of the ties in the ranking

```
par(mfrow=c(4,1))
hist(q.norm.result[,1])
hist(q.norm.result[,2])
hist(q.norm.result[,3])
hist(q.norm.result[,4])
```





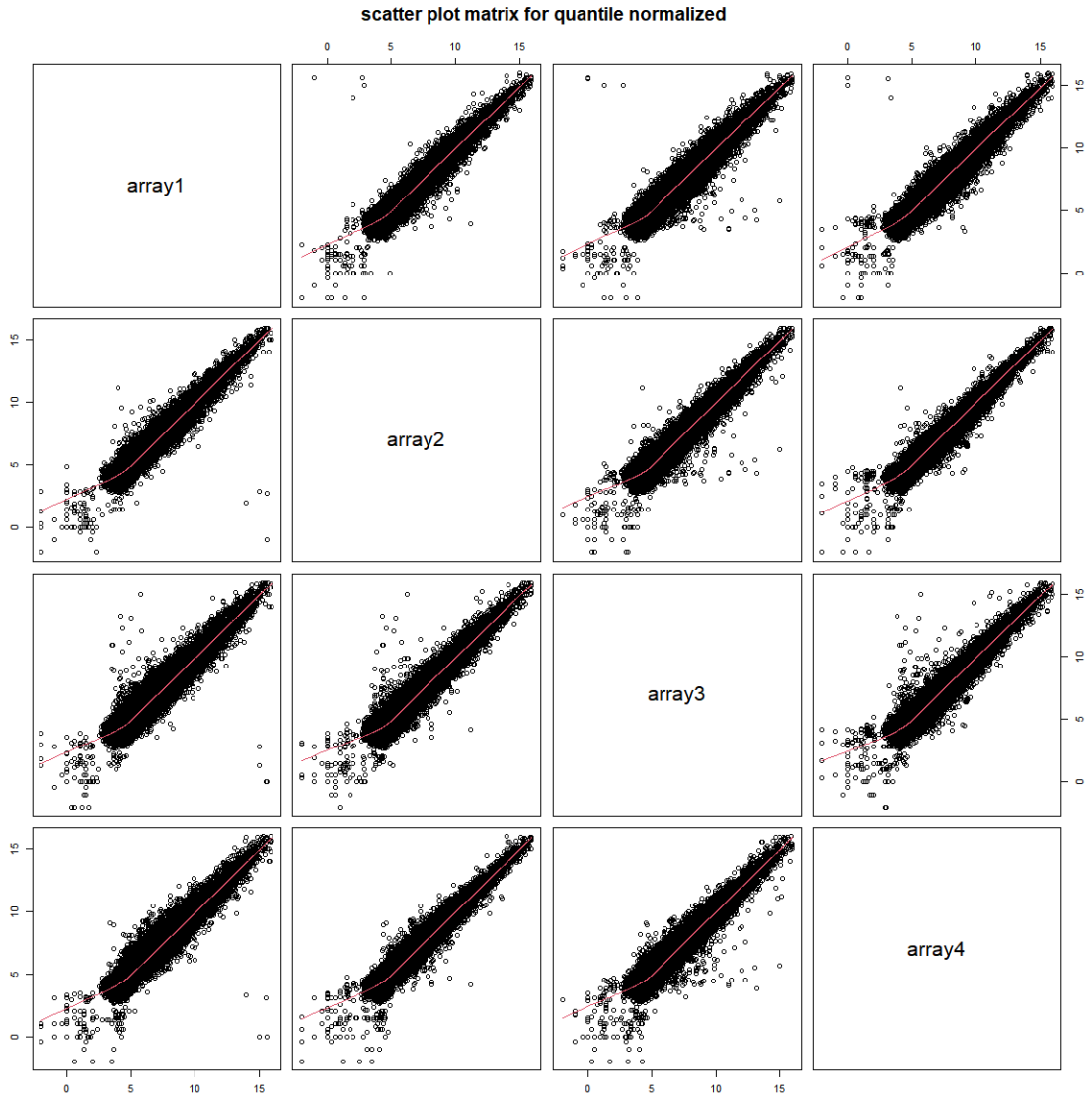




8.) Now log (base 2) the new R matrix you created from step 6 (question #7) and calculate a Spearman's rank correlation between the 4 arrays and plot a scatter plot matrix as you did before. Print the correlation coefficients to the screen. (5 pts)

```
q.norm.result.log <- log2(q.norm.result)
cor(q.norm.result.log,method='spearman',use='complete.obs')
pairs(~array1+array2+array3+array4,panel = panel.smooth,data=q.norm.result.log,main="scatter plot matrix for quantile normalized ")
```

```
> cor(q.norm.result.log,method='spearman',use='complete.obs')
           array1    array2    array3    array4
array1 1.0000000 0.8947797 0.8771078 0.8974478
array2 0.8947797 1.0000000 0.8762304 0.9071054
array3 0.8771078 0.8762304 1.0000000 0.8836381
array4 0.8974478 0.9071054 0.8836381 1.0000000
```



9.) Of the 4 normalization methods, which do you suggest as optimal and why? (2.5 pts)

quantile normalized is the optimal because it has the highest correlation.

10.) Now we want to work with a qRT-PCR dataset from patients with an inflammatory disease. The genes measured for this experiment included a set of proinflammatory chemokines and cytokines that are related to the disease.

Run the normalization script and output a data matrix of fold change values.

```
f.parse <- function(path=pa,file=fi,out=out.fi) {
  d <- read.table(paste(path,file,sep=""),skip=11,sep=",",header=T)
  u <- as.character(unique(d$Name))
  u <- u[u!=""]; u <- u[!is.na(u)];
  ref <- unique(as.character(d$Name[d$Type=="Reference"]))
  u <- unique(c(ref,u))
  hg <- c("B-ACTIN","GAPDH","18S")
  hg <- toupper(hg)
  p <- unique(toupper(as.character(d$Name.1)))
  p <- sort(setdiff(p,c("",hg)))
  mat <- matrix(0,nrow=length(u),ncol=length(p))
  dimnames(mat) <- list(u,p)
  for (i in 1:length(u)) {
    print(paste(i,": ",u[i],sep=""))
    tmp <- d[d$Name %in% u[i],c(1:3,6,9)]
    g <- toupper(unique(as.character(tmp$Name.1)))
    g <- sort(setdiff(g,c("",hg)))
    for (j in 1:length(g)) {
      v <- tmp[toupper(as.character(tmp$Name.1)) %in% g[j],5]
      v <- v[v!=999]
      v <- v[((v/mean(v))<1.5) & ((v/mean(v))>0.67)] #gene j vector
      hv3 <- NULL
      for (k in 1:length(hg)) { #housekeeping gene vector (each filtered by reps)
        hv <- tmp[toupper(as.character(tmp$Name.1)) %in% hg[k],5]
        hv <- hv[hv!=999]
        hv3 <- c(hv3,hv[((hv/mean(hv))<1.5) & ((hv/mean(hv))>0.67)])
      }
      # qRT-PCR file formatting and calculation of fold changes (cont)
      sv <- mean(as.numeric(v)) - mean(as.numeric(hv3)) #scaled value for gene j
      if(i==1) { #reference sample only
        mat[u[i],g[j]] <- sv
        next
      }
      mat[u[i],g[j]] <- sv - mat[u[1],g[j]]
    }
  }
  mat[1,][!is.na(mat[1,])] <- 0
  fc <- 2^(-1 * mat)
  write.table(t(c("Subject",dimnames(mat)[[2]])),paste(path,out,sep=""),quote=F,sep="\t",co
              =F)
  write.table(round(fc,3),paste(path,out,sep=""),quote=F,sep="\t",append=T,col.names=F)
}
# run function
pa <- "C:/Users/QiHY/Downloads/"
fi <- "Inflammation_qRT-PCR.csv"
out.fi <- 'fold_chg_matrix.txt'
f.parse(pa,fi,out.fi)
```

11.) Read the normalized qRT-PCR data matrix into R, using a Spearman's rank correlation, which two patients are most correlated? Plot these two patients against each other in a scatter plot. (3 pts)

```
foldchange <- read.table('C:/Users/QiHY/Downloads/fold_chg_matrix.txt',
                         sep='\t',header=T,row.names = 1)
#foldchange <- t(foldchange)
foldchange[foldchange<0]=NA
sp.fc<- cor(foldchange,method='spearman',use='pairwise.complete.obs')
colnames(foldchange) <- paste('patient',colnames(foldchange),sep='.')

which(sp.fc==max(rm.na(sp.fc[sp.fc!=max(rm.na(sp.fc))])),arr.ind = T)
```

```
> which(sp.fc==max(rm.na(sp.fc[sp.fc!=max(rm.na(sp.fc))])),arr.ind = T)
      row col
MX1    22  19
IRF7   19  22
> sp.fc[19,22]
[1] 0.9821429
```

MX1 and IRF7 has the most correlation with 0.9821429

```
datf.foldchange <- data.frame(foldchange)
plot(x=datf.foldchange$MX1,y=datf.foldchange$IRF7,col='red',
     xlab='patient.434_3',ylab='patient.434_8',
     main='scatter plot among the patient 434_3 and 434_8')
```