

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 (Link #36)

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.

```
> library(marray)
```

```
> sample1<-"GSM304445.gpr"  
> sample2<-"GSM304446.gpr"  
> sample3<-"GSM304447.gpr"  
> sample4<-"GSM304448.gpr"
```

```
> samples<-c(sample1,sample2,sample3,sample4)
```

```
> dir.path <- "C:/Users/fermi/Documents/Fall 2020/HW2/"
```

```
> a.cdna <- read.GenePix(path=dir.path,name.Gf = "F532 Median",name.Gb = "B532 Median",  
name.Rf = "F635 Median", name.Rb = "B635 Median",name.W ="Flags")
```

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.

```
#Median
```

```
> mnorm45<-maNorm(a.cdna[,1], norm="median", echo=TRUE)  
> mnorm46<-maNorm(a.cdna[,2], norm="median", echo=TRUE)  
> mnorm47<-maNorm(a.cdna[,3], norm="median", echo=TRUE)  
> mnorm48<-maNorm(a.cdna[,4], norm="median", echo=TRUE)
```

```
#Loess
```

```
> lnorm45<-maNorm(a.cdna[,1], norm="loess", echo=TRUE)  
> lnorm46<-maNorm(a.cdna[,2], norm="loess", echo=TRUE)  
> lnorm47<-maNorm(a.cdna[,3], norm="loess", echo=TRUE)  
> lnorm48<-maNorm(a.cdna[,4], norm="loess", echo=TRUE)
```

```
#PrintTipLoess
```

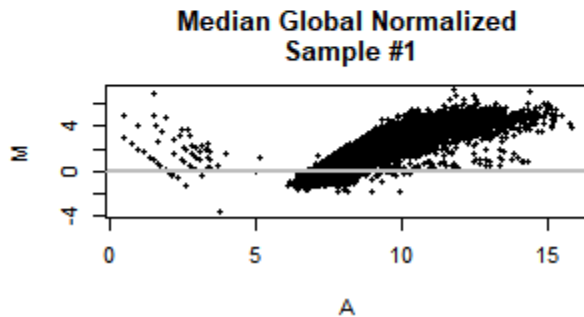
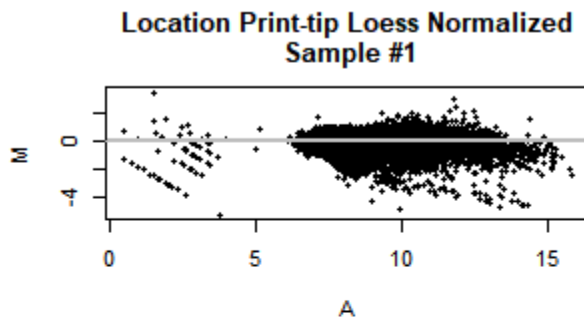
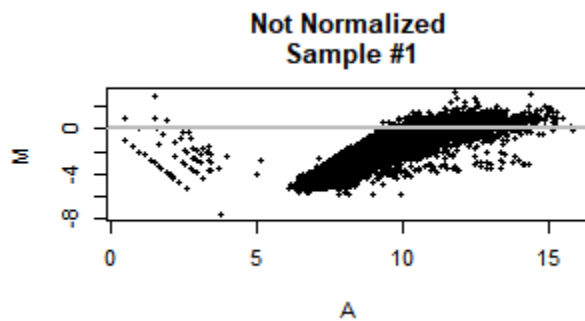
```
> PTnorm45<-maNorm(a.cdna[,1], norm="printTipLoess", echo=TRUE)
```

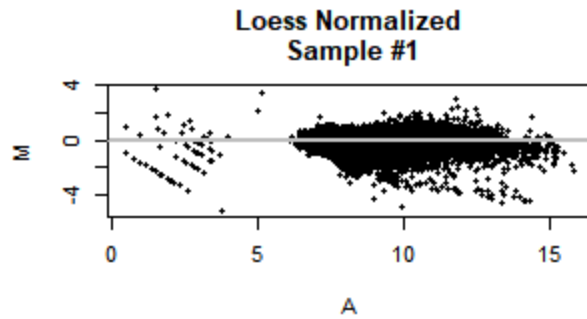
```
> PTnorm46<-maNorm(a.cdna[,2], norm="printTipLoess", echo=TRUE)
> PTnorm47<-maNorm(a.cdna[,3], norm="printTipLoess", echo=TRUE)
> PTnorm48<-maNorm(a.cdna[,4], norm="printTipLoess", echo=TRUE)
```

#MvA Plots*:**

#Array1 (-45)

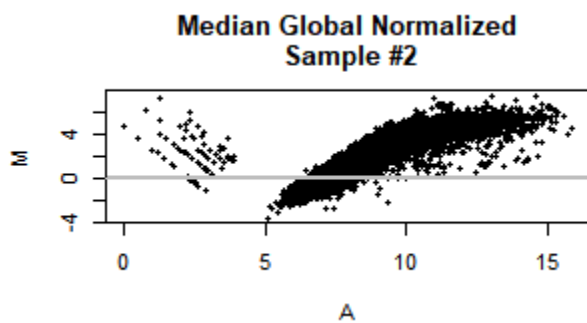
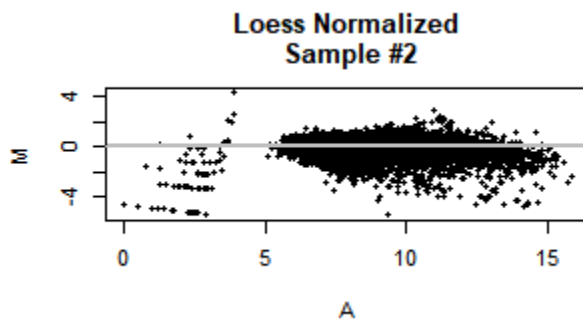
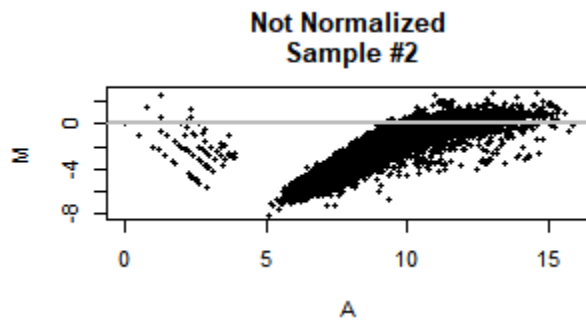
```
> par(mfrow=c(3,1),oma=c(0.1,10,0.1,10))
> maPlot(a.cdna.no.norm[,1],lines.func=NULL,legend.func=NULL,main='Not Normalized\n Sample #1')
> maPlot(mnorm45,lines.func=NULL,legend.func=NULL,main='Median Global Normalized\n Sample #1')
> maPlot(PTnorm45,lines.func=NULL,legend.func=NULL,main='Location Print-tip Loess Normalized\n Sample #1')
> maPlot(lnorm45,lines.func=NULL,legend.func=NULL,main='Loess Normalized\n Sample #1')
```

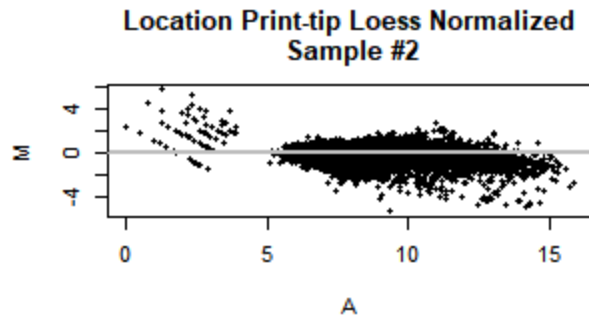




#Array2 (-46)

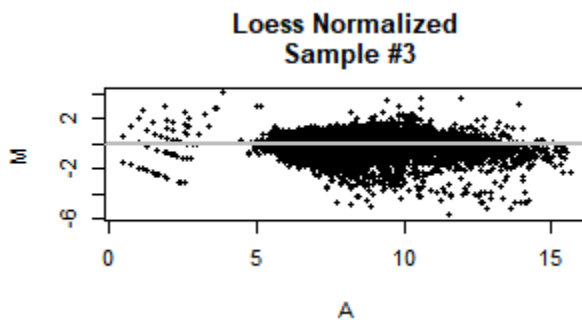
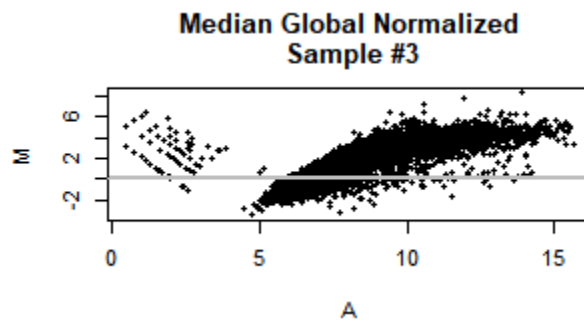
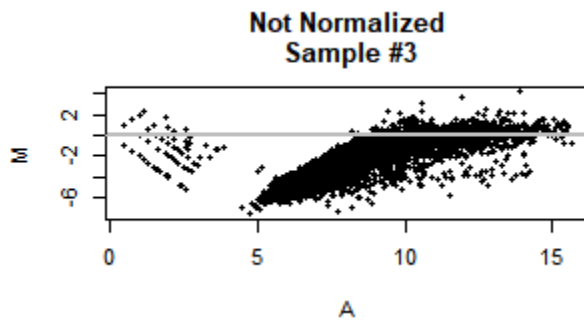
```
> maPlot(a.cdna.no.norm[,2],lines.func=NULL,legend.func=NULL,main='Not Normalized\n Sample #2')
> maPlot(mnorm46,lines.func=NULL,legend.func=NULL,main='Median Global Normalized\n Sample #2')
> maPlot(PTnorm46,lines.func=NULL,legend.func=NULL,main='Location Print-tip Loess Normalized\n Sample #2')
> maPlot(lnorm46,lines.func=NULL,legend.func=NULL,main='Loess Normalized\n Sample #2')
```

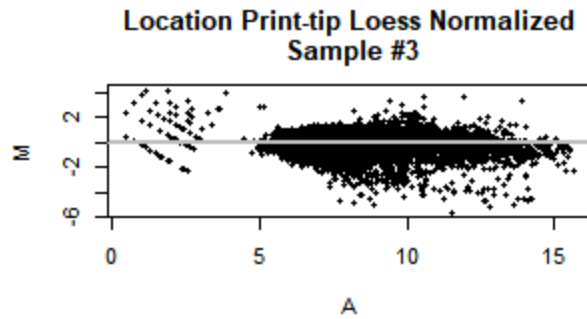




#Array3 (-47)

```
> maPlot(a.cdna.no.norm[,3],lines.func=NULL,legend.func=NULL,main='Not Normalized\n Sample #3')
> maPlot(mnorm47,lines.func=NULL,legend.func=NULL,main='Median Global Normalized\n Sample #3')
> maPlot(PTnorm47,lines.func=NULL,legend.func=NULL,main='Location Print-tip Loess Normalized\n Sample #3')
> maPlot(lnorm47,lines.func=NULL,legend.func=NULL,main='Loess Normalized\n Sample #3')
```





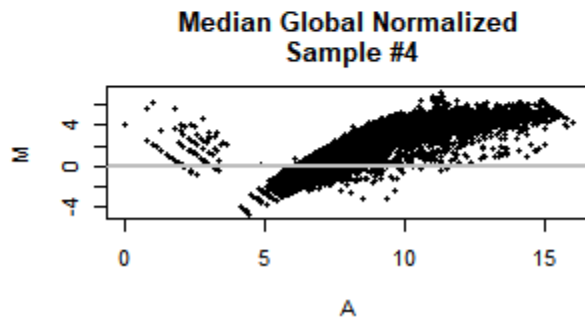
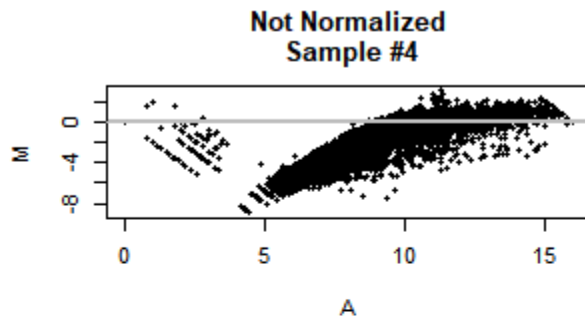
#Array4 (-48)

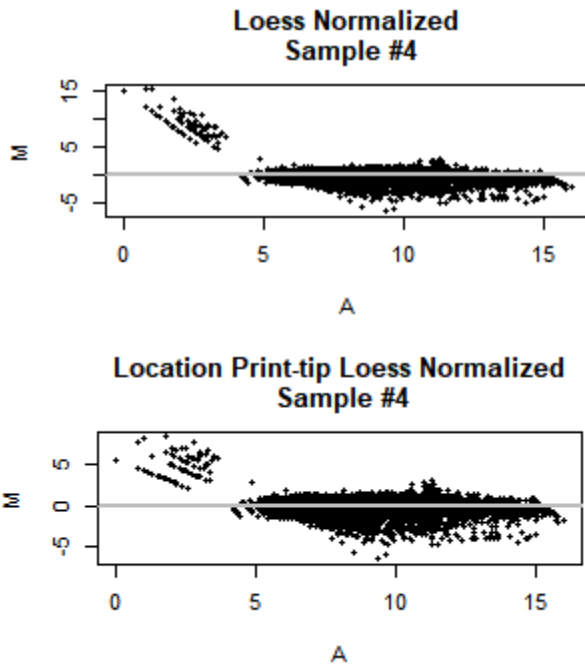
```
> maPlot(a.cdna.no.norm[,4],lines.func=NULL,legend.func=NULL,main='Not Normalized\n Sample #4')
```

```
> maPlot(mnorm48,lines.func=NULL,legend.func=NULL,main='Median Global Normalized\n Sample #4')
```

```
> maPlot(PTnorm48,lines.func=NULL,legend.func=NULL,main='Location Print-tip Loess  
Normalized\n Sample #4')
```

```
> maPlot(lnorm48,lines.func=NULL,legend.func=NULL,main='Loess Normalized\n Sample #4')
```





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.

```
#maM(a.cdna[,4])
#maM(mnorm48)
#maM(PTnorm48)
#maM(lnorm48)

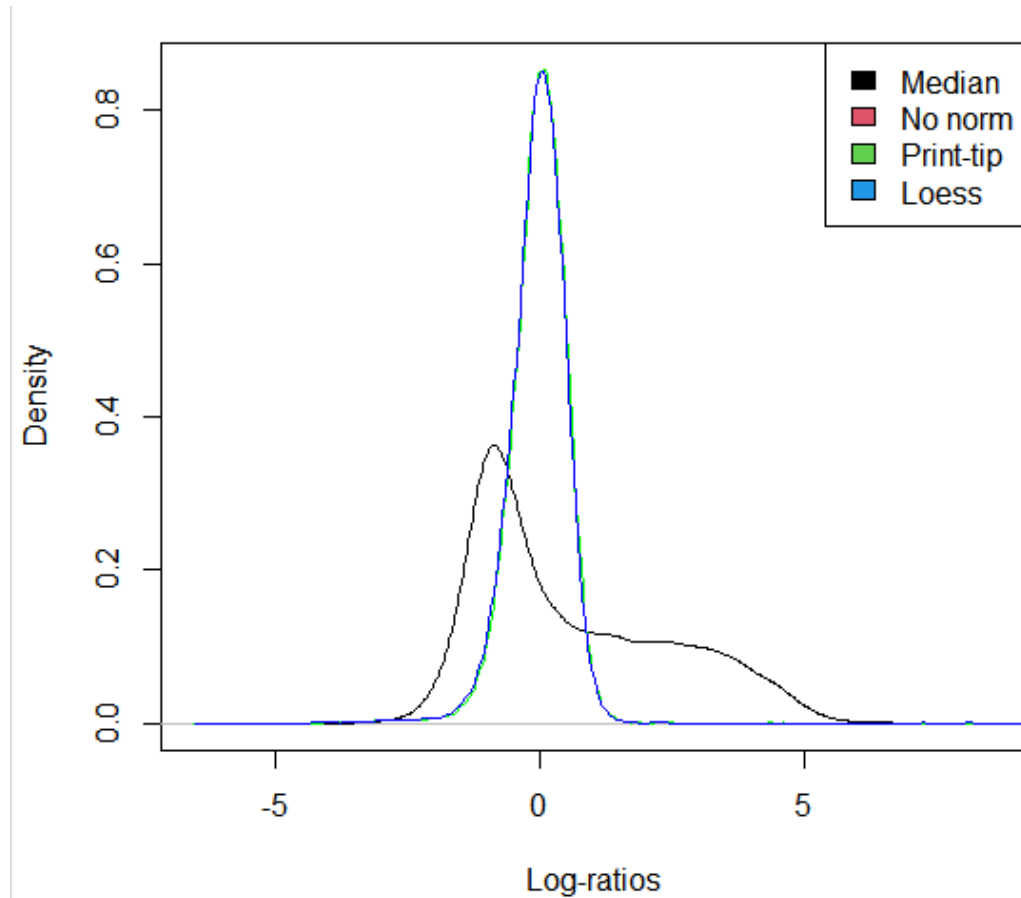
>norm48dens<-as.numeric(maM(mnorm48))
>density(norm48dens, na.rm=TRUE)
> plot(density(norm48dens))

>nonorm48dens<-na.omit((as.numeric(maM(a.cdna[,4]))))
> lines(density(nonorm48dens))

> PTnorm48dens<-na.omit((as.numeric(maM(PTnorm48))))
> lines(density(PTnorm48dens))

lnorm48dens<-na.omit((as.numeric(maM(lnorm48))))
lines(density(lnorm48dens))

> densnames=c("Median","No norm", "Print-tip","Loess")
> legend("topright",legend=densnames, fill=1:length(densnames))
```



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

#Location print-tip loess normalized is the normalization most preferred for this dataset.

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.

#subtract the background from the foreground values = maRf-maRb

```
>BS45<-a.cdna[,1]@maRf-(a.cdna[,1]@maRb)
>BS46<-a.cdna[,2]@maRf-(a.cdna[,2]@maRb)
>BS47<-a.cdna[,3]@maRf-(a.cdna[,3]@maRb)
>BS48<-a.cdna[,4]@maRf-(a.cdna[,4]@maRb)
```

```
#log2
#log2(BSXX)
```

```
>BS45log2<-log2(BS45)
>BS46log2<-log2(BS46)
```

```
>BS47log2<-log2(BS47)
>BS48log2<-log2(BS48)
```

#calculate global median normalization

```
> BS45log2M<-BS45log2/median(BS45log2,na.rm=TRUE) #constant = 1/5.392317
> BS46log2M<-BS46log2/median(BS46log2,na.rm=TRUE) #constant = 1/4.754888
> BS47log2M<-BS47log2/median(BS47log2,na.rm=TRUE) #constant = 1/4.584963
> BS48log2M<-BS48log2/median(BS48log2,na.rm=TRUE) #constant = 1/4.754888
```

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.

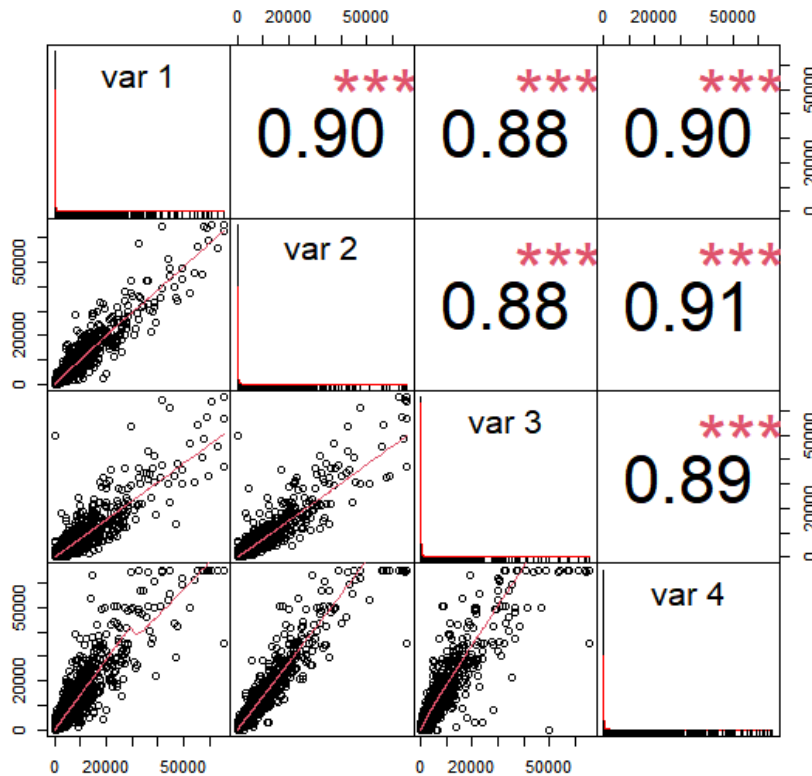
#spearman with data from #5 (all 4 arrays in 1), scatter plot matrix

```
> AllBS<-matrix(nrow=44290, ncol=4)
> AllBS[,1]<-BS45[,1]
> AllBS[,2]<-BS46[,1]
> AllBS[,3]<-BS47[,1]
> AllBS[,4]<-BS48[,1]

> cor(AllBS, method="spearman")
> BSSpearman<-cor(AllBS, method="spearman")
```

```
> chart.Correlation(AllBS, method="spearman")
```

#Spearman's rank correlation between all 4 arrays using background subtracted.



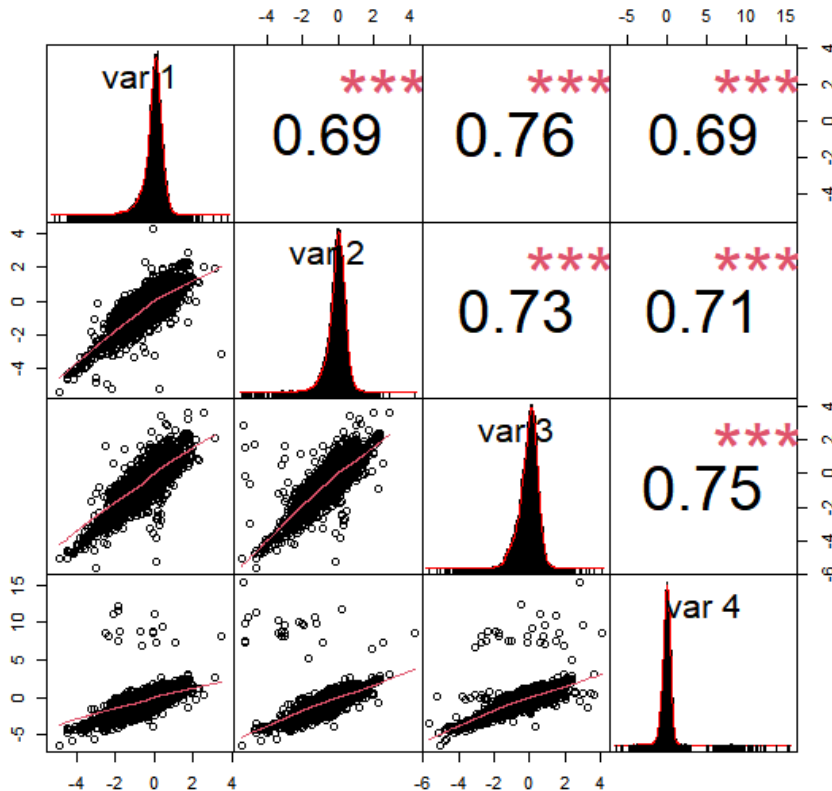
#spearman M values from loess (all 4 in 1), scatter plot matrix

```
> AllLM<-matrix(nrow=44290, ncol=4)
```



```
> AILM[,1]=maM(lnorm45)
> AILM[,2]=maM(lnorm46)
> AILM[,3]=maM(lnorm47)
> AILM[,4]=maM(lnorm48)
```

```
> chart.Correlation(AILM, method="spearman")
```



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):

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).

#from 5

```
> BS45<-a.cdna[,1]@maRf-(a.cdna[,1]@maRb)
> BS46<-a.cdna[,2]@maRf-(a.cdna[,2]@maRb)
> BS47<-a.cdna[,3]@maRf-(a.cdna[,3]@maRb)
> BS48<-a.cdna[,4]@maRf-(a.cdna[,4]@maRb)
```

```
> x<-matrix(nrow=44290, ncol=4)
```

```
> x[,1]=BS45
> x[,2]=BS46
```

```
>x[,3]=BS47  
>x[,4]=BS48
```

2. Sort each column independently in this new matrix

```
>xsort<-apply(x,2,sort)
```

3. Calculate row means for the sorted matrix

```
>rowMeans(xsort)
```

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)

```
>xsortRmeans<-matrix(nrow=44290, ncol=4)  
xsortRmeans[,1]=rowMeans(xsort)  
xsortRmeans[,2]=rowMeans(xsort)  
xsortRmeans[,3]=rowMeans(xsort)  
xsortRmeans[,4]=rowMeans(xsort)
```

5. Rank the columns independently on the original background subtracted matrix (from step #1)

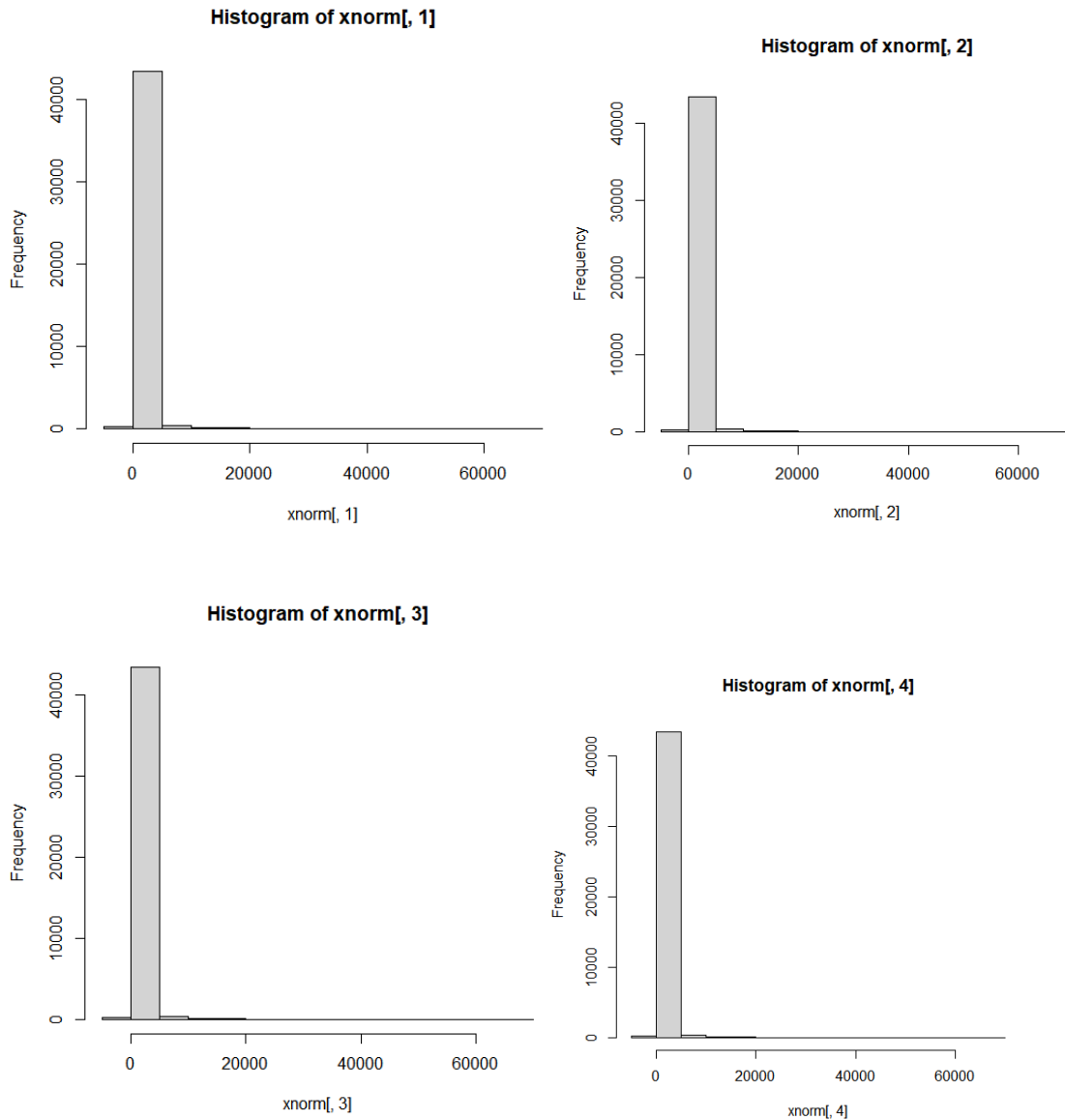
Hint: use the rank() function with the argument ties="first" or order()

```
>x45rank<-rank(x[,1],ties.method="first")  
>x46rank<-rank(x[,2],ties.method="first")  
>x47rank<-rank(x[,3],ties.method="first")  
>x48rank<-rank(x[,4],ties.method="first")
```

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

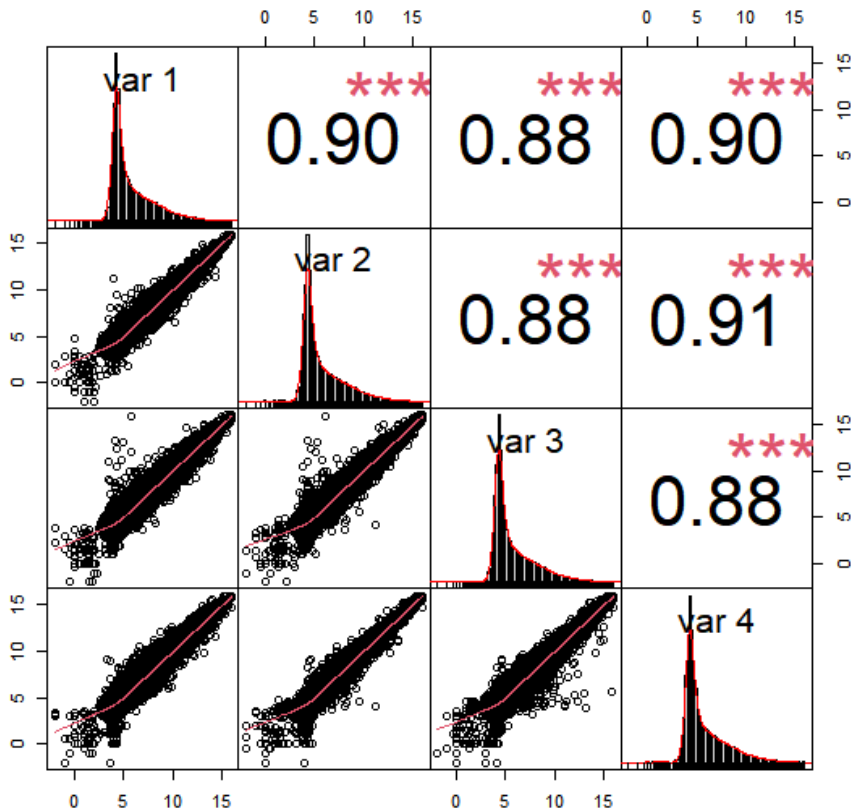
```
>xnorm<-matrix(nrow=44290, ncol=4)  
  
>xnorm[,1]<-xsortRmeans[,1][x45rank]  
>xnorm[,2]<-xsortRmeans[,2][x46rank]  
>xnorm[,3]<-xsortRmeans[,3][x47rank]  
>xnorm[,4]<-xsortRmeans[,4][x48rank]
```

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.



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.

```
>xnormlog2<-log2(xnorm)
> chart.Correlation(xnormlog2, method="spearman")
```



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

Out of the 4 normalization methods, the quantile normalization method is the most optimal. It is the most optimal because it has the higher correlation between the 4 arrays with a similar distribution between the 4. The background subtracted normalization method is also optimal as a second choice.