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.

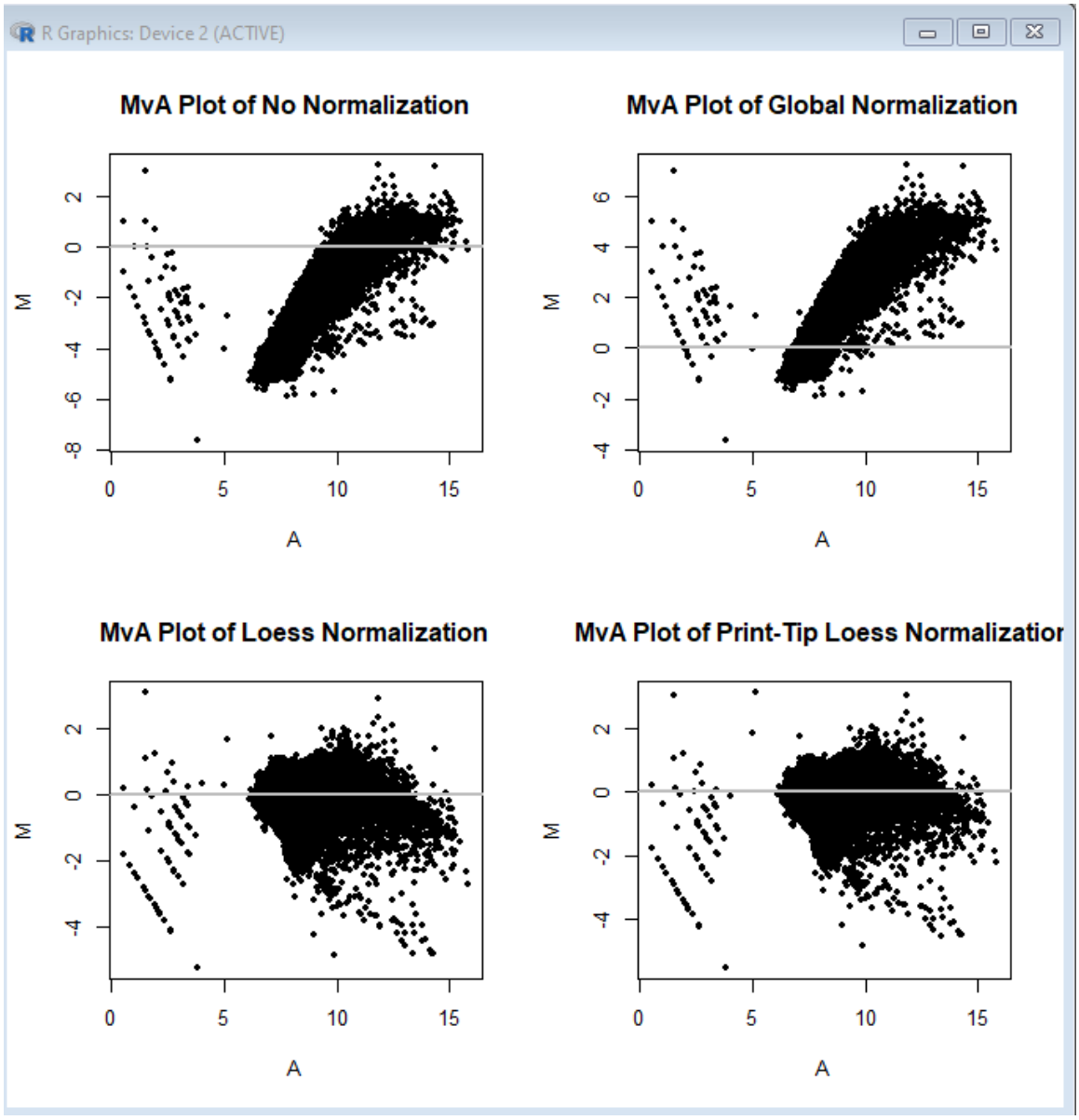
We will be working with 4 lean subjects from which a needle biopsy was taken.

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)

```
> library(marray)
> a.cdna = read.GenePix(path='E:\\', name.Gf = "F532 Median",
+ name.Gb ="B532 Median", name.Rf = "F635 Median", name.Rb = "B635 Median",
+ name.W ="Flags")
Reading ...  E:\/GSM304445.gpr
Reading ...  E:\/GSM304446.gpr
Reading ...  E:\/GSM304447.gpr
Reading ...  E:\/GSM304448.gpr
```

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)

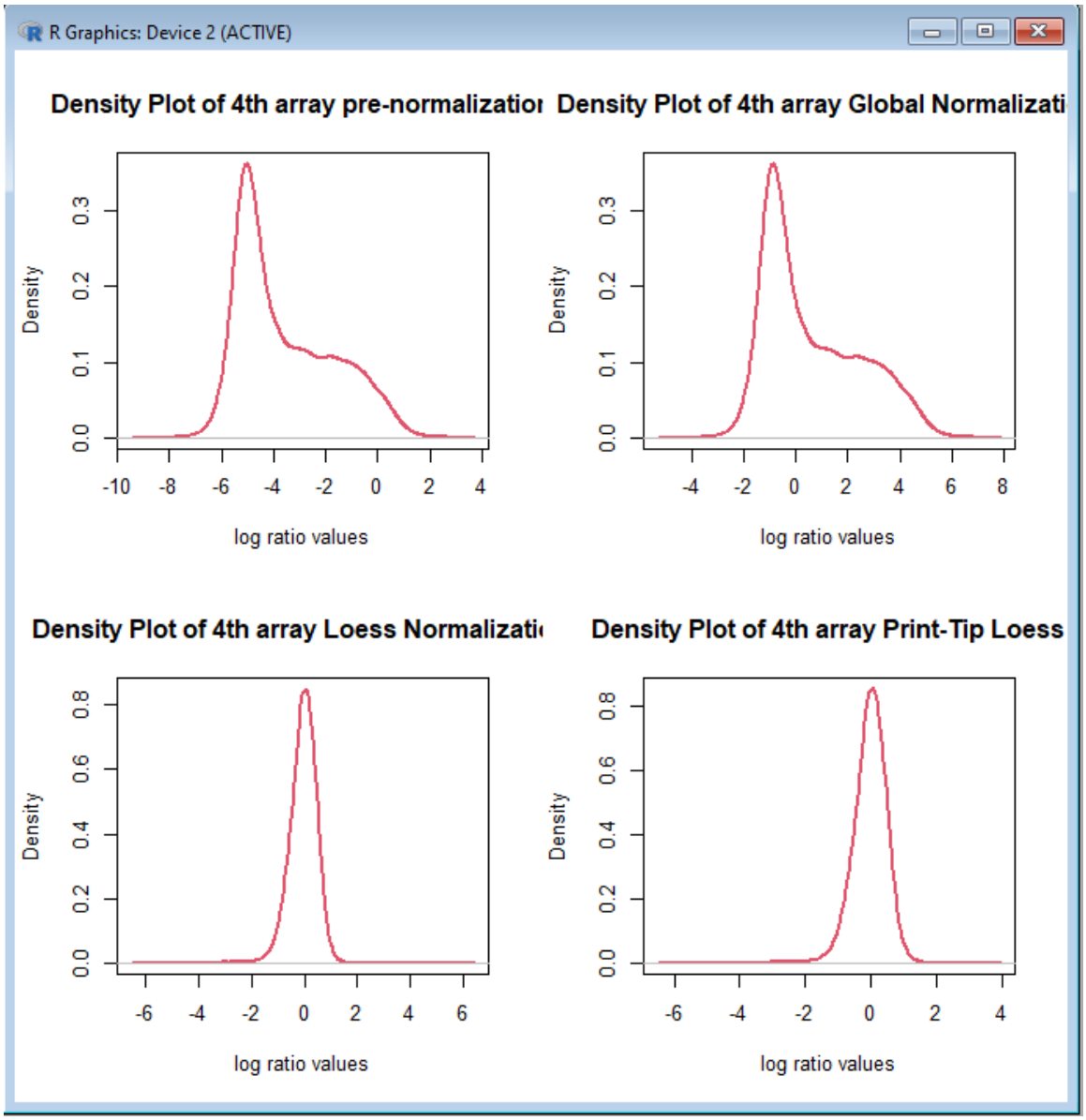
```
> a.cdna.no.norm = maNorm(a.cdna,norm='none')
> a.cdna.gl.norm=maNorm(a.cdna,norm='median')
> a.cdna.lo.norm=maNorm(a.cdna,norm='loess')
> a.cdna.pt.norm=maNorm(a.cdna,norm='printTipLoess')

> par(mfrow=c(2,2))
> maPlot(a.cdna.no.norm,lines.func=NULL,legend.func=NULL,main='MvA Plot of No Normalization')
> maPlot(a.cdna.gl.norm,lines.func=NULL,legend.func=NULL,main='MvA Plot of Global Normalization')
> maPlot(a.cdna.lo.norm,lines.func=NULL,legend.func=NULL,main='MvA Plot of Loess Normalization')
> maPlot(a.cdna.pt.norm,lines.func=NULL,legend.func=NULL,main='MvA Plot of Print-Tip Loess
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)

```
> a.cdna.no.mat = as.matrix(a.cdna.no.norm[,4])
> a.cdna.no.mat = na.omit(a.cdna.no.mat)
> a.cdna.gl.mat = as.matrix(a.cdna.gl.norm[,4])
> a.cdna.gl.mat = na.omit(a.cdna.gl.mat)
> a.cdna.lo.mat = as.matrix(a.cdna.lo.norm[,4])
> a.cdna.lo.mat = na.omit(a.cdna.lo.mat)
> a.cdna.pt.mat = as.matrix(a.cdna.pt.norm[,4])
> a.cdna.pt.mat = na.omit(a.cdna.pt.mat)
> par(mfrow=c(2,2))
> plot(density(a.cdna.no.mat),lwd=2,col=2,xlab = 'log ratio values', main = 'Density Plot of 4th array pre-normalization')
> plot(density(a.cdna.gl.mat),lwd=2,col=2,xlab = 'log ratio values', main = 'Density Plot of 4th array Global Normalization')
```

> plot(density(a.cdna.lo.mat),lwd=2,col=2,xlab = 'log ratio values', main = 'Density Plot of 4th array Loess Normalization')
> plot(density(a.cdna.pt.mat),lwd=2,col=2,xlab = 'log ratio values', main = 'Density Plot of 4th array Print-Tip Loess')



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

Based on the plots, I would say that the loess normalization has provided a more average organization of the data than shown by the other normalization methods/pre-normalization data.

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)

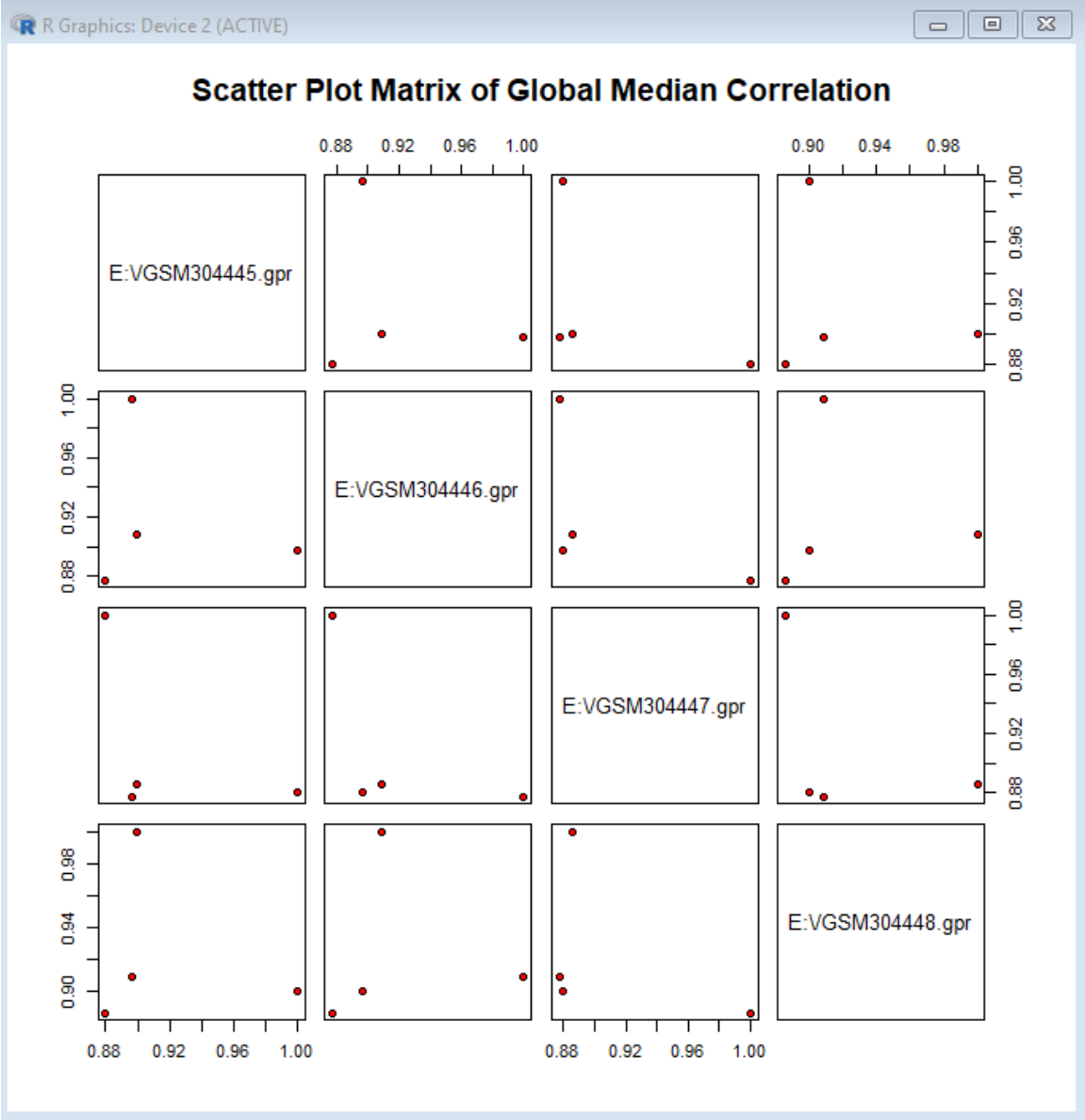
```
> rf = slot(a.cdna,'maRf')
> rb = slot(a.cdna,'maRb')
> q5 = rf – rb
> q5 = abs(q5)
> q5.log2 = log2(q5)
> summary(q5.log2)
 E:\\/GSM304445.gpr E:\\/GSM304446.gpr E:\\/GSM304447.gpr E:\\/GSM304448.gpr
 Min.   :  -Inf     Min.   :  -Inf     Min.   :  -Inf     Min.   :  -Inf
 1st Qu.: 4.907     1st Qu.: 4.087     1st Qu.: 4.087     1st Qu.: 3.807
 Median : 5.392     Median : 4.700     Median : 4.585     Median : 4.755
 Mean   :  -Inf     Mean   :  -Inf     Mean   :  -Inf     Mean   :  -Inf
 3rd Qu.: 6.977     3rd Qu.: 6.768     3rd Qu.: 6.150     3rd Qu.: 7.077
 Max.   :15.999     Max.   :15.999     Max.   :15.999     Max.   :15.999
> q5.log2 = q5.log2[!is.infinite(rowSums(q5.log2)),]
> summary(q5.log2)
E:\\/GSM304445.gpr E:\\/GSM304446.gpr E:\\/GSM304447.gpr E:\\/GSM304448.gpr
Min.   : 0.000     Min.   : 0.000     Min.   : 0.000     Min.   : 0.000
1st Qu.: 4.907     1st Qu.: 4.087     1st Qu.: 4.087     1st Qu.: 3.807
Median : 5.392     Median : 4.755     Median : 4.585     Median : 4.755
Mean   : 6.203     Mean   : 5.642     Mean   : 5.364     Mean   : 5.659
3rd Qu.: 6.989     3rd Qu.: 6.781     3rd Qu.: 6.150     3rd Qu.: 7.087
Max.   :15.999     Max.   :15.999     Max.   :15.999     Max.   :15.999
> q5.scale = scale(q5.log2, center = FALSE, scale = apply(q5.log2,2,median,na.rm=TRUE))
> summary(q5.scale)
 E:\\/GSM304445.gpr E:\\/GSM304446.gpr E:\\/GSM304447.gpr E:\\/GSM304448.gpr
 Min.   :0.000      Min.   :0.0000     Min.   :0.0000     Min.   :0.0000
 1st Qu.:0.910      1st Qu.:0.8596     1st Qu.:0.8915     1st Qu.:0.8007
 Median :1.000      Median :1.0000     Median :1.0000     Median :1.0000
 Mean   :1.150      Mean   :1.1866     Mean   :1.1700     Mean   :1.1901
 3rd Qu.:1.296      3rd Qu.:1.4262     3rd Qu.:1.3413     3rd Qu.:1.4906
 Max.   :2.967      Max.   :3.3647     Max.   :3.4894     Max.   :3.3647
```

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)

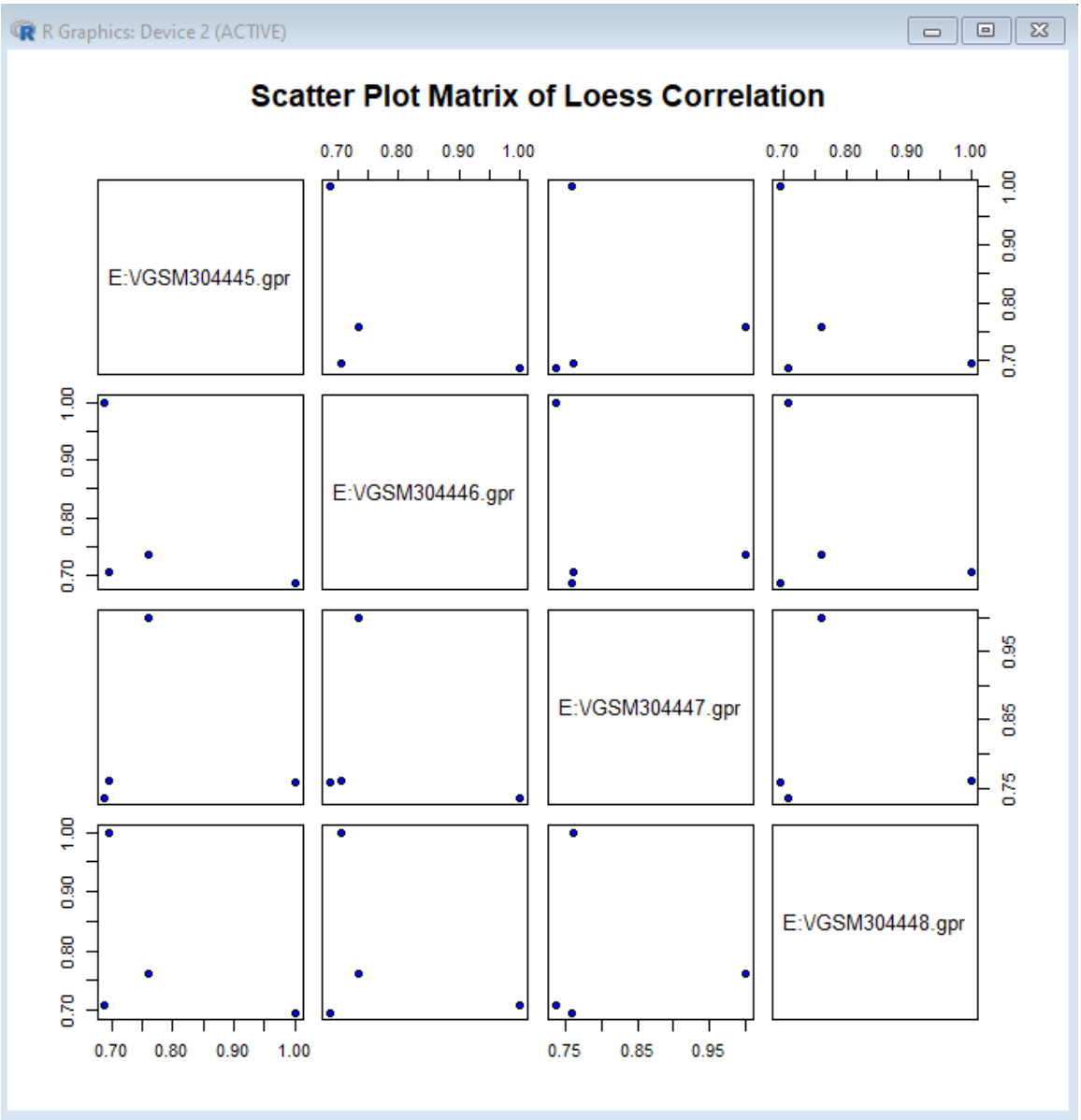
```
> cor.q5 = cor(q5.scale, method = 'spearman')
                    E:\\/GSM304445.gpr E:\\/GSM304446.gpr E:\\/GSM304447.gpr E:\\/GSM304448.gpr
E:\\/GSM304445.gpr          1.0000000          0.8969572          0.8798755          0.8997086
E:\\/GSM304446.gpr          0.8969572          1.0000000          0.8772387          0.9085362
E:\\/GSM304447.gpr          0.8798755          0.8772387          1.0000000          0.8860463
E:\\/GSM304448.gpr          0.8997086          0.9085362          0.8860463          1.0000000
> a.cdna.lo.mat = as.matrix(a.cdna.lo.norm)
> a.cdna.lo.mat = na.omit(a.cdna.lo.mat)
> cor.q2 = cor(a.cdna.lo.mat, method = 'spearman')
```

```
                    E:\\/GSM304445.gpr E:\\/GSM304446.gpr E:\\/GSM304447.gpr E:\\/GSM304448.gpr
E:\\/GSM304445.gpr          1.0000000          0.6882504          0.7593636          0.6952448
E:\\/GSM304446.gpr          0.6882504          1.0000000          0.7358808          0.7076792
E:\\/GSM304447.gpr          0.7593636          0.7358808          1.0000000          0.7610899
E:\\/GSM304448.gpr          0.6952448          0.7076792          0.7610899          1.0000000
```

> pairs(cor.q5, labels = colnames(cor.q5), main = 'Scatter Plot Matrix of Global Median Correlation', pch = 21, bg = ‘red’)



> pairs(cor.q2, labels = colnames(cor.q2), main = 'Scatter Plot Matrix of Loess Correlation', pch = 21, bg = 'blue')



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).

```
> rf = slot(a.cdna,'maRf')
> rb = slot(a.cdna,'maRb')
> q7 = rf-rb
```

2. Sort each column independently in this new matrix

```
> q7.sort = apply(q7,2,sort)
```

3. Calculate row means for the sorted matrix

```
> q7.mean = rowMeans(q7.sort)
```

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)

```
> q7.mat = replace(q7.sort,values = q7.mean)
```

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()

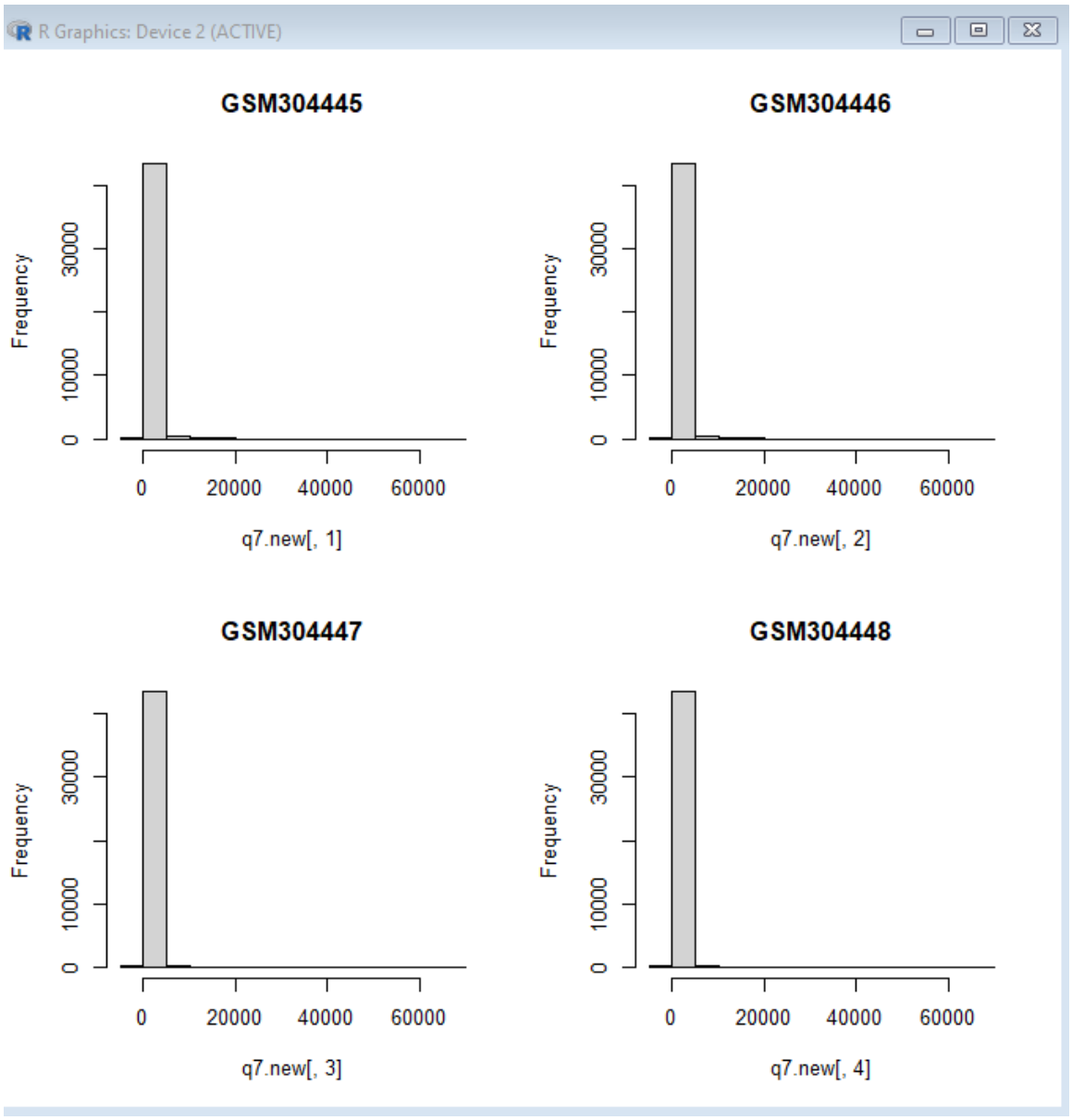
```
> q7.rank = apply(q7,2,rank)
```

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

```
> q7.new = sapply(1:ncol(q7.mat),function(i){q7.mat[q7.rank[,i],i]})
```

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.

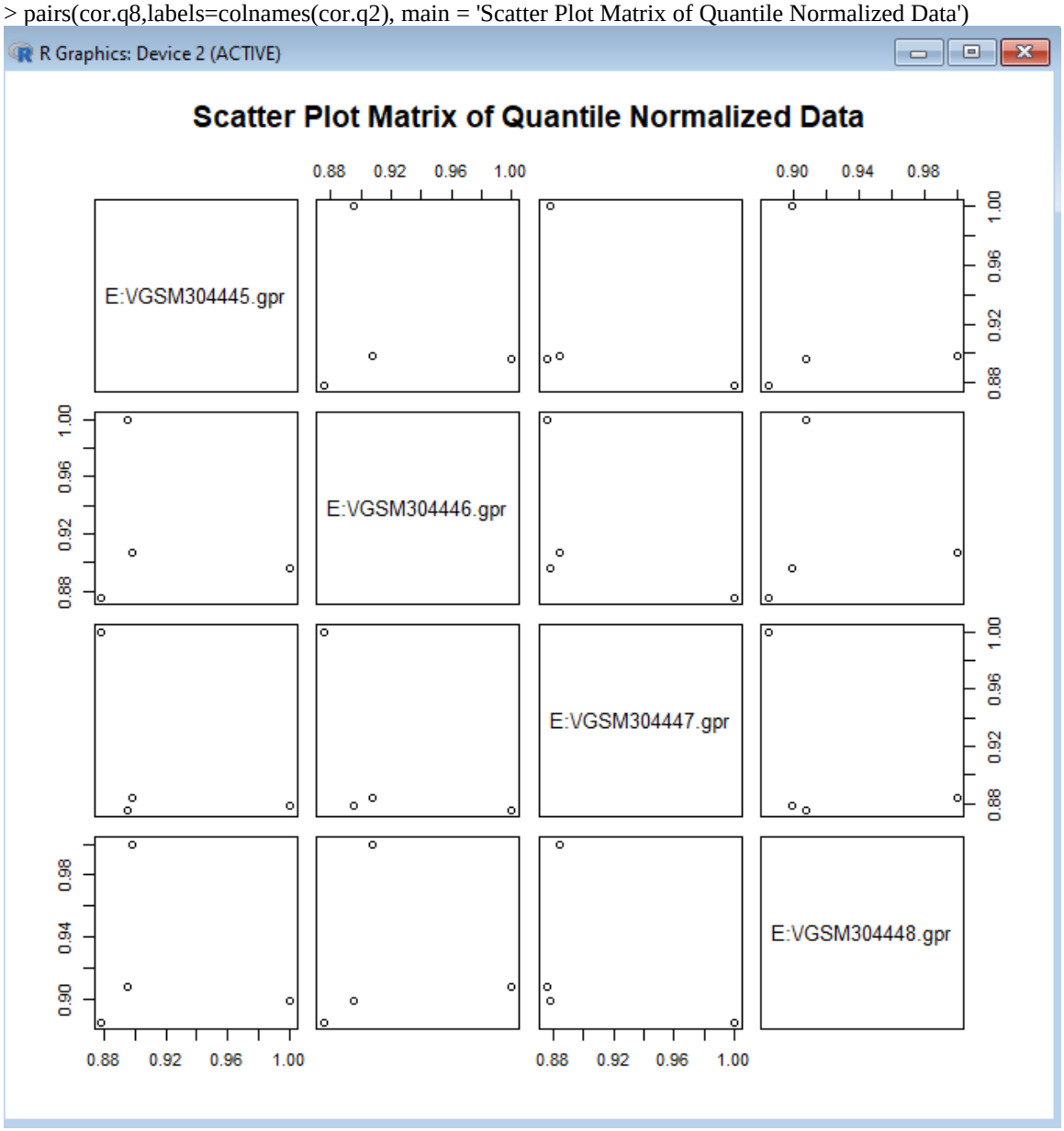
```
> par(mfrow = c(2,2))
> hist(q7.new[,1],main='GSM304445')
> hist(q7.new[,2],main='GSM304446')
> hist(q7.new[,3],main='GSM304447')
> hist(q7.new[,4],main='GSM304448')
```



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)

```
> q8 = apply(q7.new,2,log2)
> q8 = q8[!is.infinite(rowSums(q8)),]
> q8 = q8[!is.na(rowSums(q8)),]
> cor.q8 = cor(q8, method = 'spearman')
          [,1]      [,2]      [,3]      [,4]
[1,] 1.0000000 0.8956444 0.8783062 0.8984446
[2,] 0.8956444 1.0000000 0.8756825 0.9073969
[3,] 0.8783062 0.8756825 1.0000000 0.8846299
[4,] 0.8984446 0.9073969 0.8846299 1.0000000
```

> pairs(cor.q8,labels=colnames(cor.q2), main = 'Scatter Plot Matrix of Quantile Normalized Data')



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

By examining each of the correlation coefficients and scatter plot matrices created for the 4 normalization methods shown, I would determine that for this data, the global median normalization method seemed to be the most average and show the most correlation amongst all the data points compared to the others. It seems the method of using the median value as the scale for the data seemed to bolster the global median normalization methods ability to normalize the data and determine correlation amongst the many data points, something that wasn't done with the loess and quantile normalization methods.

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. Download the raw qRT-PCR file called Inflammation_qRT-PCR.csv. Then change the normalization script from the lecture notes to include the housekeeping genes beta actin, GAPDH, and 18S. Look at the file to make sure the housekeepers are spelled correctly.

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$Type=="Reference"]))
        u <- unique(c(ref,u))
        hg <- c("B-actin","GAPDH","18S")  #changed from B-ACTIN since file uses lowercase
        hg <- toupper(hg)
        p <- unique(toupper(as.character(d$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)])
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",col.names
=F,row.names=F)
        write.table(round(fc,3),paste(path,out,sep=""),quote=F,sep="\t",append=T,col.names=F)
}

pa = "E:\\"
fi = "Inflammation_qRT-PCR.csv"
out.fi = "fold_chg_matrix.txt"
```

f.parse(pa,fi,out.fi) #file saved to E drive as fold_chg_matrix.txt!

```
[1] "1: 434_1"
[1] "2: 434_15"
[1] "3: 434_2"
[1] "4: 434_14"
[1] "5: 434_3"
[1] "6: 434_13"
[1] "7: 434_4"
[1] "8: 434_12"
[1] "9: 434_5"
[1] "10: 434_11"
[1] "11: 434_6"
[1] "12: 434_10"
[1] "13: 434_7"
[1] "14: 434_9"
[1] "15: 434_8"
```

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)

```
> qrt.norm = read.table("E:\\fold_chg_matrix.txt", header = T, row.names = 1, fill = TRUE)
> qrt.norm[,c("IL.6","ALPHA")] = NULL                    #removed two columns with just NA values
> qrt.t = as.data.frame(t(qrt.norm)                      #transpose so that patients are columns
> qrt.t[,"434_1"] =NULL                                  #remove column with all "1" value
```

	434_15	434_2	434_14	434_3	434_13	434_4	434_12	434_5	434_11	434_6	434_10	434_7	434_9	434_8
APOBEC3B	0.031	0.010	0.007	0.014	0.004	0.018	0.024	0.014	0.015	0.010	0.013	0.011	0.010	0.012
CCL.24	0.632	2.623	0.794	0.576	0.287	0.136	0.385	0.383	0.777	0.104	0.205	0.069	0.530	0.444
CCL2	0.468	2.828	4.131	48.019	1.062	109.116	51.044	21.315	5.681	21.782	4.393	26.353	0.534	51.552
CCL20	0.064	1.201	1.450	0.545	0.577	0.923	3.236	2.198	0.995	5.223	0.508	3.412	0.076	0.454
CCL22	0.466	1.440	0.401	0.609	1.687	1.074	0.513	1.093	0.549	0.635	0.656	0.669	0.437	0.908
DDX58.RIG.1	0.944	0.191	0.139	1.445	0.935	4.236	4.597	0.295	1.382	0.890	0.634	1.430	0.147	1.943
G1P2.ISG15	0.054	0.067	0.076	3.799	0.234	12.262	16.617	0.505	3.810	2.035	1.179	4.082	0.078	3.359
G1P3	0.303	0.391	0.366	11.485	0.208	36.661	25.049	1.849	4.161	7.373	3.878	19.901	0.367	16.176
IFI44	0.051	0.103	0.066	2.112	0.437	6.769	4.091	0.608	0.409	1.583	0.733	2.062	0.100	2.929
IFIT1	0.127	0.097	0.069	3.667	0.159	16.843	25.805	0.507	8.037	2.446	1.126	5.010	0.084	4.807
IFIT4	0.130	0.024	0.018	1.123	0.063	3.352	10.930	0.161	2.034	1.131	0.586	1.797	0.040	1.002
IL.10	0.148	0.012	0.019	0.017	0.022	0.223	0.091	0.081	0.011	0.119	0.095	0.073	0.026	0.041
IL1A	0.314	0.631	1.299	0.777	0.206	0.679	0.534	0.994	0.943	1.240	0.112	1.146	0.139	0.372
IL1B	17.557	25.541	55.104	23.573	9.031	25.882	29.180	55.066	26.312	63.914	12.782	67.494	5.280	21.138
INFA5	0.002	0.000	0.000	0.001	0.000	0.001	0.828	0.000	0.270	0.000	0.000	0.000	0.000	0.002
INFB1	0.001	0.001	0.000	0.003	0.003	0.006	1.256	0.000	0.533	0.001	0.001	0.001	0.001	0.006
IRF5	0.075	0.116	0.055	0.182	0.093	0.301	0.191	0.137	0.139	0.126	0.131	0.147	0.090	0.215
IRF7	0.766	0.740	0.540	6.609	0.846	11.710	10.616	1.782	3.966	4.579	3.384	7.289	0.826	9.134
LPL	0.047	0.102	0.029	0.099	0.034	0.028	0.174	0.058	0.084	0.038	0.030	0.014	0.034	0.035
LY6E	0.314	0.381	0.256	4.734	3.033	11.874	5.505	1.470	3.181	2.181	1.575	6.982	0.437	5.390
MX1	0.818	1.230	0.676	23.399	1.939	62.468	59.271	4.693	19.250	16.550	9.911	22.728	0.833	34.636
NK4.IL.32.	2.979	2.952	3.257	4.041	3.354	4.896	3.153	4.975	3.756	2.609	3.913	2.818	3.117	4.149
OAS3	0.038	0.037	0.038	0.511	0.344	1.631	1.668	0.088	1.655	0.295	0.145	1.025	0.049	0.495
OASL	1.320	0.898	0.810	4.526	1.015	11.583	16.578	4.216	6.367	3.439	2.994	4.830	1.193	7.458
OPN.SPP1.	1.161	10.212	2.490	6.557	0.970	3.902	7.513	6.428	6.827	2.076	0.939	2.806	1.524	4.916
PBEF	7.200	1.636	4.317	2.681	0.500	3.382	4.614	4.749	6.534	7.136	4.695	4.932	3.823	2.988
PI3	0.609	0.390	0.579	0.718	3.230	1.279	1.509	1.761	0.730	0.752	0.902	0.405	0.409	0.718
PRKR	0.230	0.235	0.176	2.377	0.798	5.412	6.944	1.032	2.050	2.373	1.421	2.304	0.279	2.362
TNF	0.299	0.308	0.266	0.856	1.428	0.823	2.076	0.495	0.746	0.737	0.402	0.657	0.316	0.823

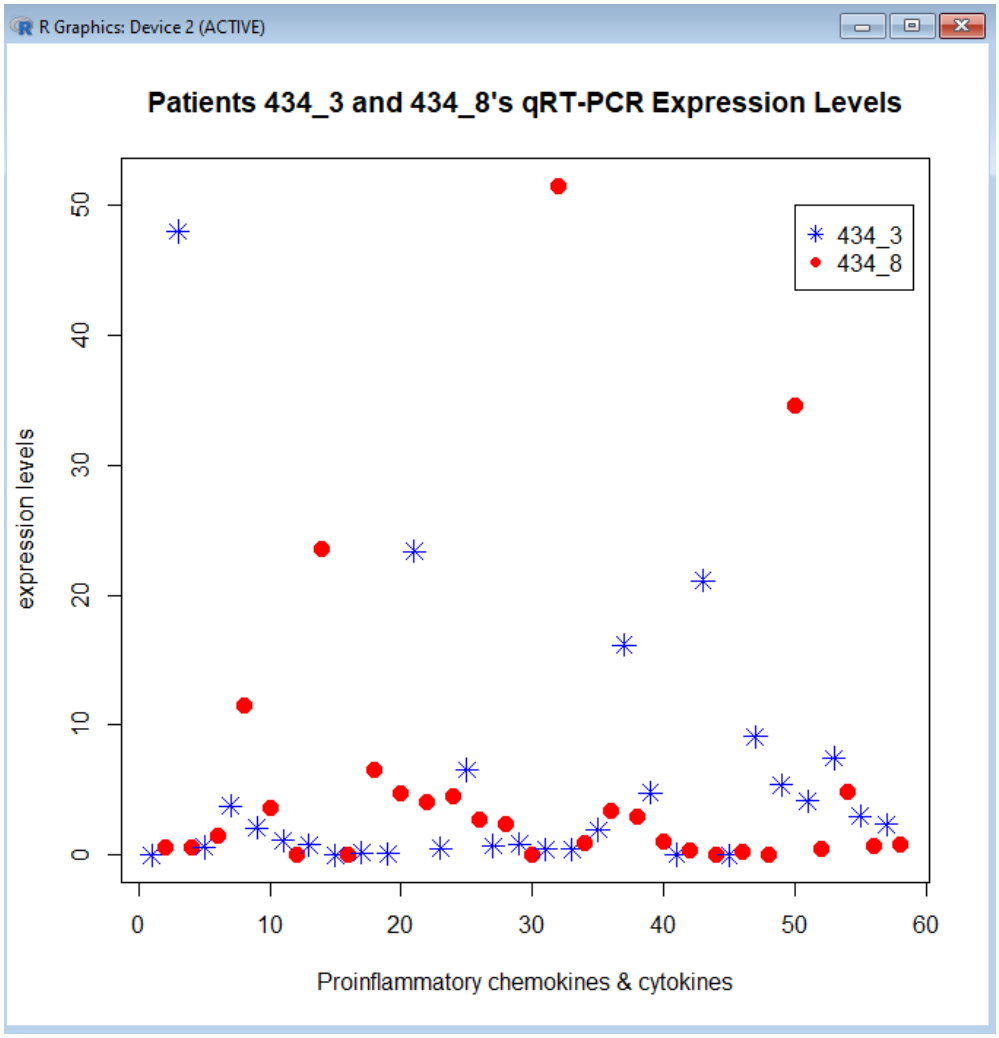
```
> qrt.cor = cor(qrt.t)
```

	434_15	434_2	434_14	434_3	434_13	434_4	434_12	434_5	434_11	434_6	434_10	434_7
434_15	1.00000000	0.8807497	0.9425256	0.3236402	0.8170938	0.09840297	0.2177526	0.8812882	0.7563178	0.8707414	0.7599334	0.8028158
434_2	0.88074968	1.0000000	0.9387604	0.4099549	0.8170999	0.16260600	0.2659011	0.9179042	0.7574390	0.8721179	0.6803922	0.8193755
434_14	0.94252557	0.9387604	1.0000000	0.3992096	0.8551498	0.16741500	0.2733095	0.9518888	0.7609618	0.9371863	0.7275257	0.8810418
434_3	0.32364021	0.4099549	0.3992096	1.0000000	0.3782748	0.95948242	0.8636572	0.6435753	0.6021304	0.6680119	0.6747786	0.7226739
434_13	0.81709378	0.8170999	0.8551498	0.3782748	1.0000000	0.17604094	0.2677527	0.8291986	0.7145077	0.8115187	0.7299358	0.7752754
434_4	0.09840297	0.1626060	0.1674150	0.9594824	0.1760409	1.00000000	0.9021375	0.4275966	0.4783358	0.4830635	0.5629108	0.5709495
434_12	0.21775263	0.2659011	0.2733095	0.8636572	0.2677527	0.90213754	1.0000000	0.4622627	0.7188840	0.5685730	0.7243989	0.6573264
434_5	0.88128817	0.9179042	0.9518888	0.6435753	0.8291986	0.42759659	0.4622627	1.0000000	0.7872247	0.9742118	0.7801526	0.9345183
434_11	0.75631778	0.7574390	0.7609618	0.6021304	0.7145077	0.47833580	0.7188840	0.7872247	1.0000000	0.8634423	0.9375416	0.8733258
434_6	0.87074140	0.8721179	0.9371863	0.6680119	0.8115187	0.48306348	0.5685730	0.9742118	0.8634423	1.0000000	0.8670166	0.9800767
434_10	0.75993341	0.6803922	0.7275257	0.6747786	0.7299358	0.56291083	0.7243989	0.7801526	0.9375416	0.8670166	1.0000000	0.8891029
434_7	0.80281579	0.8193755	0.8810418	0.7226739	0.7752754	0.57094948	0.6573264	0.9345183	0.8733258	0.9800767	0.8891029	1.0000000
434_9	0.90921335	0.7704727	0.7623390	0.3119222	0.7415225	0.10915307	0.2056642	0.7397098	0.6967624	0.7113393	0.7495944	0.6516869
434_8	0.23728345	0.3043579	0.2997673	0.9823637	0.3086837	0.98145668	0.9160431	0.5436860	0.6007600	0.6001792	0.6857223	0.6744924

	434_9	434_8
434_15	0.9092133	0.2372835
434_2	0.7704727	0.3043579
434_14	0.7623390	0.2997673
434_3	0.3119222	0.9823637
434_13	0.7415225	0.3086837
434_4	0.1091531	0.9814567
434_12	0.2056642	0.9160431
434_5	0.7397098	0.5436860
434_11	0.6967624	0.6007600
434_6	0.7113393	0.6001792
434_10	0.7495944	0.6857223
434_7	0.6516869	0.6744924
434_9	1.0000000	0.2464719
434_8	0.2464719	1.0000000

Patients 434_8 and 434_3 were the most related in the table with a correlation coeffient of .9823637

```
>qrt3 = qrt[,"434_8"]
>qrt8 = qrt[,"434_8"]
>plot(c(qrt3,qrt8), xlab = 'Proinflammatory chemokines & cytokines', ylab = "expression levels",
+main = "Patients 434_3 and 434_8's qRT-PCR Expression Levels"
+color = c('blue','red'), pch = (8,16))
>legend(x=50,y=50, legend=c("434_3","434_8"),col=c('blue','red'), pch=c(8,16))
```


R Graphics: Device 2 (ACTIVE)
Patients 434_3 and 434_8's qRT-PCR Expression Levels
434_3
434_8
expression levels
0
10
20
30
40
50
0
10
20
30
40
50
60
Proinflammatory chemokines & cytokines