Fall 5-2-2015

Transcription regulatory networks in medulloblastoma.

Yasmeen Hesham Howeedy
The American University in Cairo

```
#1#set Working directory
#2# Download packages:
source("http://Bioconductor.org/biocLite.R")
biocLite("affy")
library (affy)

#3# Read the CEL files:
Med <-ReadAffy()

#4# Download pallette:
install.packages("RColorBrewer")
library(RColorBrewer)
cols <- brewer.pal(8, "Set1")

#5# Loop to remove the last four digits of all filenames:
names = sampleNames(Med)
for(i in 1:length(names)){names[i]=substr(names[i], 0, nchar(names[i])-
4)}

#6# Data display:
#6.1# Test image quality:
par(mfrow=c(3,3))
image( Med[, 1], col = cols, axes=FALSE)
image( Med[, 2], col = cols, axes=FALSE)
image( Med[, 3], col = cols, axes=FALSE)
image( Med[, 4], col = cols, axes=FALSE)
image( Med[, 5], col = cols, axes=FALSE)
image( Med[, 6], col = cols, axes=FALSE)
image( Med[, 7], col = cols, axes=FALSE)
image( Med[, 8], col = cols, axes=FALSE)
image( Med[, 9], col = cols, axes=FALSE)
image( Med[, 10], col = cols, axes=FALSE)
image( Med[, 11], col = cols, axes=FALSE)
image( Med[, 12], col = cols, axes=FALSE)
image( Med[, 13], col = cols, axes=FALSE)
image( Med[, 14], col = cols, axes=FALSE)
image( Med[, 15], col = cols, axes=FALSE)
image( Med[, 16], col = cols, axes=FALSE)
image( Med[, 17], col = cols, axes=FALSE)
image( Med[, 18], col = cols, axes=FALSE)
image( Med[, 19], col = cols, axes=FALSE)
image( Med[, 20], col = cols, axes=FALSE)
image( Med[, 21], col = cols, axes=FALSE)
image( Med[, 22], col = cols, axes=FALSE)
image( Med[, 23], col = cols, axes=FALSE)
image( Med[, 24], col = cols, axes=FALSE)
image( Med[, 25], col = cols, axes=FALSE)
image( Med[, 26], col = cols, axes=FALSE)
image( Med[, 27], col = cols, axes=FALSE)
image( Med[, 28], col = cols, axes=FALSE)
image( Med[, 29], col = cols, axes=FALSE)
image( Med[, 30], col = cols, axes=FALSE)
image( Med[, 31], col = cols, axes=FALSE)
image( Med[, 32], col = cols, axes=FALSE)
```

```
image( Med[, 33], col = cols, axes=FALSE)
image( Med[, 34], col = cols, axes=FALSE)
image( Med[, 35], col = cols, axes=FALSE)
image( Med[, 36], col = cols, axes=FALSE)
image( Med[, 37], col = cols, axes=FALSE)
image( Med[, 38], col = cols, axes=FALSE)
image( Med[, 39], col = cols, axes=FALSE)
image( Med[, 40], col = cols, axes=FALSE)
image( Med[, 41], col = cols, axes=FALSE)
image( Med[, 42], col = cols, axes=FALSE)
image( Med[, 43], col = cols, axes=FALSE)
image( Med[, 44], col = cols, axes=FALSE)
image( Med[, 45], col = cols, axes=FALSE)
image( Med[, 46], col = cols, axes=FALSE)
image( Med[, 47], col = cols, axes=FALSE)
image( Med[, 48], col = cols, axes=FALSE)
image( Med[, 49], col = cols, axes=FALSE)
image( Med[, 50], col = cols, axes=FALSE)
image( Med[, 51], col = cols, axes=FALSE)
image( Med[, 52], col = cols, axes=FALSE)
image( Med[, 53], col = cols, axes=FALSE)
image( Med[, 54], col = cols, axes=FALSE)
image( Med[, 55], col = cols, axes=FALSE)
image( Med[, 56], col = cols, axes=FALSE)
image( Med[, 57], col = cols, axes=FALSE)
image( Med[, 58], col = cols, axes=FALSE)
image( Med[, 59], col = cols, axes=FALSE)
image( Med[, 60], col = cols, axes=FALSE)
image( Med[, 61], col = cols, axes=FALSE)
image( Med[, 62], col = cols, axes=FALSE)
image( Med[, 63], col = cols, axes=FALSE)
image( Med[, 64], col = cols, axes=FALSE)
image( Med[, 65], col = cols, axes=FALSE)
image( Med[, 66], col = cols, axes=FALSE)
image( Med[, 67], col = cols, axes=FALSE)
image( Med[, 68], col = cols, axes=FALSE)
image( Med[, 69], col = cols, axes=FALSE)
image( Med[, 70], col = cols, axes=FALSE)
image( Med[, 71], col = cols, axes=FALSE)
image( Med[, 72], col = cols, axes=FALSE)
image( Med[, 73], col = cols, axes=FALSE)
image( Med[, 74], col = cols, axes=FALSE)
image( Med[, 75], col = cols, axes=FALSE)
image( Med[, 76], col = cols, axes=FALSE)

#6.2# Boxplot:
boxplot(Med, main = "A boxplot for GSE37418 samples' expression
intensities",
        cex.main=0.7, cex.lab=0.6, cex.axis=0.45, ylab = "Expression
intensity variation",
        las= 2, col = cols, names = names, las = 2)

#6.3# Histogram:
```

```
hist(Med, col = cols, lty = 1, main= "Histogram for samples GEO
ID:GSE37418",
     xlab = "Intensities", cex.lab=0.7, cex.axis=0.7, cex.main=0.8)

#6.4# RNA degradation plots:
biocLite("AmpAffyExample")
library(AmpAffyExample)
data(AmpData)
RNAdeg <- AffyRNAdeg(Med)
plotAffyRNAdeg(RNAdeg)
RNAdeg_summary<-summaryAffyRNAdeg(RNAdeg) #summary of the slopes

#6.5# prenormalization MA-plot:

windows(10,10)
mva.pairs(exprs(Med[,1:10]), main="GSE37418: MA plots for prenormalized
data.", cex.main=1, cex=0.5, col="blue")
mva.pairs(exprs(Med[,11:20]), main="GSE37418: MA plots for prenormalized
data.", cex.main=1, cex=0.5, col="blue")
mva.pairs(exprs(Med[,21:30]), main="GSE37418: MA plots for prenormalized
data.", cex.main=1, cex=0.5, col="blue")
mva.pairs(exprs(Med[,31:40]), main="GSE37418: MA plots for prenormalized
data.", cex.main=1, cex=0.5, col="blue")
mva.pairs(exprs(Med[,41:50]), main="GSE37418: MA plots for prenormalized
data.", cex.main=1, cex=0.5, col="blue")
mva.pairs(exprs(Med[,51:60]), main="GSE37418: MA plots for prenormalized
data.", cex.main=1, cex=0.5, col="blue")
mva.pairs(exprs(Med[,61:70]), main="GSE37418: MA plots for prenormalized
data.", cex.main=1, cex=0.5, col="blue")
mva.pairs(exprs(Med[,71:76]), main="GSE37418: MA plots for prenormalized
data.", cex.main=1, cex=0.5, col="blue")




#####################################
#7# QC prenormalization:
#7.1# Quality of affymetrix batch:
biocLite("simpleaffy")
library("simpleaffy")

Med.qc <- qc(Med)
#7.2# Background data:

ratios<-ratios(Med.qc)
Bkgd<-avbg(Med.qc)
Bkgd_max<-maxbg(Med.qc)
Bkgd_min<-minbg(Med.qc)
spike<-spikeInProbes(Med.qc)
probes<-qcProbes(Med.qc)
PP<-percent.present(Med.qc)
sfs<-sfs(Med.qc)
```

```
target<-target(Med.qc)
plot(Med.qc)



#8# Normalization:
#8.1# TRying the algorithm, RMA:
normalized<-rma(Med)
rmax<-exprs(normalized)
png("rmax_boxplot.png", width = 1000, height = 1000)
boxplot(rmax, main = "A boxplot for samples normalized by rma alg.",
        cex.main=0.7, cex.lab=0.6, cex.axis=0.45, ylab = "Expression
intensity variation",
        las= 2, col = cols, names = names, las = 2)
dev.off()

#8.2# Using mas.5 algorithm:
m<-mas5(Med)
xm<-exprs(m)
log_xm<-log2(xm)

#9# QC after normalization:
#9.1# Boxplot:
png("mas5x_boxplot.png", width = 1000, height = 1000)
boxplot(log_xm, main = "A boxplot for samples normalized by mas5.0
algorithm",
        cex.main=0.7, cex.lab=0.6, cex.axis=0.45, ylab = "Expression
intensity variation",
        las= 2, col = cols, names = names, las = 2)
dev.off()

png("hist_post.png", width=1000, height=1000)
hist(log_xm, cex.main=1, lty=1,main="Histogram for samples logged and
normalized by MAS5.0 algorithm")
dev.off()
#9.2# MA plots after normalization:
windows(10,10)
mva.pairs(log_xm[,1:10], log=F, main="GSE37418: MA plots for
postnormalized data.", cex.main=1, cex=0.6, col="brown")
mva.pairs(log_xm[,11:20], log=F, main="GSE37418: MA plots for
postnormalized data.", cex.main=1, cex=0.6, col="brown")
mva.pairs(log_xm[,21:30], log=F, main="GSE37418: MA plots for
postnormalized data.", cex.main=1, cex=0.6, col="brown")
mva.pairs(log_xm[,31:40], log=F, main="GSE37418: MA plots for
postnormalized data.", cex.main=1, cex=0.6, col="brown")
mva.pairs(log_xm[,41:50], log=F, main="GSE37418: MA plots for
postnormalized data.", cex.main=1, cex=0.6, col="brown")
mva.pairs(log_xm[,51:60], log=F, main="GSE37418: MA plots for
postnormalized data.", cex.main=1, cex=0.6, col="brown")
mva.pairs(log_xm[,61:70], log=F, main="GSE37418: MA plots for
postnormalized data.", cex.main=1, cex=0.6, col="brown")
```

```
mva.pairs(log_xm[,71:76], log=F, main="GSE37418: MA plots for
postnormalized data.", cex.main=1, cex=0.6, col="brown")

#10# Array selection:#correlation, cv vs mean and dedrogram:
#10.1# Dendrogram:
x=t(log_xm)
distance_tree <- dist(x,method="euclidean") # calculate distance
clusters_tree <- hclust(distance_tree,method="complete") # calculate
clusters
windows(9,9)
plot(clusters_tree,labels=names,cex=0.6, hang=0.06,
     main="Cluster Dendrogram for Samples' Similarity",
     cex.lab=0.8, cex.main=0.8)
##################################
#10.2: the correlation:
# average correlation plot:
correlation<-cor(log_xm, method="pearson", use="pairwise.complete.obs")
correlation_mean<- apply(correlation,1,mean)
par(oma=c(3,0.5,2,0.5))
plot(c(1,length(correlation_mean)),range(correlation_mean),type="n",
     xlab="Arrays",ylab="Avg r",main="Average correlation plot for
medulloblastoma samples",
     cex.main=0.8,axes=F)
points(correlation_mean,bg="red",col=1,pch=21,cex=1.25)
axis(1,at=c(1:length(correlation_mean)),labels=names,las=2,
     cex.lab=0.5,cex.axis=0.5)
axis(2)
#####################################
#10.3# correlation plot:
layout(matrix(c(1,1,1,1,1,1,1,1,2,2), 5, 2, byrow = TRUE))
par(oma=c(0,0,0,0))
install.packages("gplots")
library(gplots)
color<-rev(colorpanel(30,"yellow","black","blue"))
order<- seq(min(correlation,na.rm=T),max(correlation,na.rm=T),length=10)
image(correlation,main="Correlation plot for Medulloblastoma
samples",axes=F,col=color, xlab="", ylab="")
axis(1,at=seq(0,1,length=ncol(correlation)),label=names,cex.axis=0.6,las=
2)
axis(2,at=seq(0,1,length=ncol(correlation)),label=names,cex.axis=0.6,las=
2)
image(as.matrix(order),col=color,axes=F,oma=c(1,0,1,0))
order2 <- round(order,2)
axis(1,at=seq(0,1,length=length(order)),labels=order2,cex.axis=1)

#10.3# CV vs mean:
mean=apply(log_xm,2,mean)
variance= apply(log_xm, 2, var)
sd=sqrt(variance)
cv= sd/mean
plot(mean,cv,main=" Coefficient of variation vs. mean plot of logged
MAS5.0 normalized data",
     xlab="Mean",ylab="CV",col='blue',cex.main=0.9,type="n", cex.lab=0.9)
points(mean,cv,bg="darkblue",col=1,pch=21)
```

```
text(mean,cv,label=names,pos=1,cex=0.45)

##############
###PCA:###
pca<-prcomp(log_xm, center=T, scale.=T)
sdev<-as.data.frame(pca$sdev)
rot<-as.data.frame(pca$rotation)
cent<-as.data.frame(pca$center)
scale<-as.data.frame(pca$scale)
plot(pca$rotation, main= "PCA plot for 76 medulloblastoma arrays")
text(pca$rotation, labels=rownames(pca$rotation),
pos=3,col="blue",cex=0.5)


#11# Remove outlier array: GSM918628
med = m[,-51]
data <- exprs (med)
log_data <-log2(data)
newnames<-names[-51]

#to test if your CV method will be used for filteration,
#do it but on the rows (genes themselves and not the arrays)
mean_gene<-apply(log_data,1,mean)
var_gene<-apply(log_data,1,var)
sd_gene= sqrt(var_gene)
cv_gene= sd_gene/mean_gene
plot(mean_gene,cv_gene,main=" CV vs. mean plot for the genes on each
array ",
     xlab="Mean",ylab="CV",col='blue',cex.main=0.9,type="n", cex.lab=0.9)
points(mean_gene,cv_gene,bg="darkblue",col=2,pch=21)
text(mean_gene,cv_gene,label=newnames,pos=3,cex=0.5)
############################

#####################################################################
#12# Filteration:(using genefilter pkg)
source("http://bioconductor.org/biocLite.R")
biocLite("genefilter")
library("genefilter")

#12.1# PM calls:
datacalls<-mas5calls(Med[,-51])
Xdata<-exprs(datacalls)

#12.2# Remove genes "A" in all arrays:
absent<- rowSums(Xdata =='A')
absent2<-which(absent==ncol(Xdata))

#12.3# Filter them out:
log_filtered<-log_data[-absent2,]


#Now to visualize this, cv vs mean again:
Fmean_gene<-apply(Fil_datax,1,mean)
Fvar_gene<-apply(Fil_datax,1,var)
```

```
Fsd_gene= sqrt(Fvar_gene)
Fcv_gene= Fsd_gene/Fmean_gene
plot(Fmean_gene,Fcv_gene,main=" CV vs. mean plot for MAS5.0 calls
filtered genes on each array ",
     xlab="Mean",ylab="CV",col='green',cex.main=0.9,type="n",
cex.lab=0.9)
points(Fmean_gene,Fcv_gene,bg="yellow",col=2,pch=21)
text(Fmean_gene,Fcv_gene,label=newnames,pos=3,cex=0.5)

#The filetring step number two is:
#Put your filtering threshold at 20%:


indsum <- apply(log_filtered [,1:75], 1, function(x)
  sum(x[c(3,15,25,47,59,62:64)] == "P") > 6 ||
                sum(x[c(5,29:32, 42:44,70,72)] =="P") > 7||
                sum(x[c(9:12,17,23:24,34:35,37,47, 51, 58:59,
61:62)]=="P") > 11  ||
                sum(x[c(1:2, 4, 6:8, 13:15, 18:22, 25, 27:28, 33:36,
38:40,
                        45:46, 49:50, 52:57, 67:70, 72, 74:75)]=="P")>
29 || sum(x[c(41,66)]=="P")>2)
sum(indsum, na.rm=TRUE)#To check for the presence of True results

#12.4# remove the indsum from your original:
Fil_data<-med[-indsum,]

#12.5# remove duplicated entrez ids, probe signals with no entrez id
#and probe signals below variance of 0.2 from mean signal value(control)
source("http://bioconductor.org/biocLite.R")
biocLite("genefilter")
library("genefilter")
##
getCV <- function(x){
  mymean <- mean(x)
  mydev <- sd(x)
  mycv <- mydev / mymean
  mycv}
filtered.cv<-apply(Fil_data, 1, function(x) getCV>0.2)
if(getCV>0.2){apply(Fil_data, 1, function(x))}
##

myFilteredData <- nsFilter(Fil_data, require.entrez=TRUE,
remove.dupEntrez=
TRUE,var.func=getCV,var.cutoff=0.2,filterByQuantile=FALSE)

Fdata<- exprs(myFilteredData$eset)
dim(Fdata)
nsfiltereddata<-log_data[rownames(Fdata),]
write.table(nsfiltereddata, "nsfiltereddata.txt", sep="\t", quote=F)

#13# ANOVA:
#at filteration cutoff=0.2:
library(limma)
```

```
design <- model.matrix(~-1+factor(schedule$Group))
View(design)
colnames(design)<- c("WNT", "SHH", "G3","G4", "Unknown")
dataset.fit <- lmFit(nsfiltereddata, design)
MATRIX<-c ("WNT-(SHH+G3+G4+Unknown)/4", "SHH-(WNT+G3+G4+Unknown)/4", "G3-
(WNT+SHH+G4+Unknown)/4", "G4-(WNT+SHH+G3+Unknown)/4", "Unknown-
(WNT+SHH+G3+G4)/4")
lev<-colnames(design)
contrast.matrix <- makeContrasts(contrasts=MATRIX,levels=lev)
FIT<-contrasts.fit(dataset.fit,contrast.matrix)
ebays.dataset<-eBayes(FIT)
qqt(ebays.dataset$t,df=ebays.dataset$df.prior+ebays.dataset$df.residual,m
ain="Moderated t-statistic",
    cex.main=0.8, cex.lab=0.8, cex.axis=0.8)
abline(0,1, col=2)

#14# FDR:
##
fdr1 <-
topTable(ebays.dataset,n=nrow(nsfiltereddata),adjust="fdr",coef=1,
                lfc=2, p.value=0.02, sort.by="logFC")
dim(fdr1)
coeff1<-nsfiltereddata[row.names(fdr1),]
dim(coeff1)
write.table(fdr1, "WNT_DIFF_GENES.txt", sep="\t", quote=F)
##
fdr2<- topTable(ebays.dataset, n=nrow(nsfiltereddata), adjust="fdr",
coef=2,
                lfc=2, p.value=0.02, sort.by="logFC")
dim(fdr2)
coeff2<-nsfiltereddata[row.names(fdr2),]
dim(coeff2)
write.table(fdr2, "SHH_DIFF_GENES.txt", sep="\t", quote=F)
##

fdr3<- topTable(ebays.dataset, n=nrow(nsfiltereddata), adjust="fdr",
coef=3,
                lfc=2, p.value=0.02, sort.by="logFC")
dim(fdr3)
coeff3<-nsfiltereddata[row.names(fdr3),]
dim(coeff3)
write.table(fdr3, "G3_DIFF_GENES.txt", sep="\t", quote=F)

##
fdr4<- topTable(ebays.dataset, n=nrow(nsfiltereddata), adjust="fdr",
coef=4,
                lfc=2, p.value=0.02, sort.by="logFC")
dim(fdr4)
coeff4<-nsfiltereddata[row.names(fdr4),]
dim(coeff4)
write.table(fdr4, "G4_DIFF_GENES.txt", sep="\t", quote=F)

##
```

```
fdr5<- topTable(ebays.dataset, n=nrow(nsfiltereddata), adjust="fdr",
coef=5,
                lfc=2, p.value=0.02, sort.by="logFC")
dim(fdr5)
coeff5<-nsfiltereddata[row.names(fdr5),]
dim(coeff5)
write.table(fdr5, "UNK_DIFF_GENES.txt", sep="\t", quote=F)

######
#15#draw heatmaps:
##
library(gplots)
my.dist <- function(x)dist(x, method="euclidean")
my.hclust <- function(d)hclust(d, method="complete")

#for the WNT:
windows(15,15)
png("heatmap wnt.png",width=1000, height=1000)

colnames(coeff1)<-schedule$Group
heatmap.2(as.matrix(coeff1), cex.main= 0.4,
          main= " Heatmap WNT genes",
            Rowv = TRUE,distfun=dist,hclustfun=hclust,scale="row",
            col=redgreen(75),key=TRUE, keysize=1,symkey=FALSE,
            density.info="none",trace="none", cexRow=0.1,cexCol=0.7)
dev.off()
#FOR THE shh:
windows(15,15)
png("heatmap SHH.png",width=1000, height=1000)

colnames(coeff2)<-schedule$Group
heatmap.2(as.matrix(coeff2), cex.main= 0.4,
          main= " Heatmap SHH genes",
          Rowv = TRUE,distfun=dist,hclustfun=hclust,scale="row",
          col=redgreen(75),key=TRUE, keysize=1,symkey=FALSE,
          density.info="none",trace="none", cexRow=0.1,cexCol=0.7)
dev.off()
#FOR THE G3:
windows(15,15)
png("heatmap G3 pvalue.png",width=1000, height=1000)

colnames(coeff3)<-schedule$Group
heatmap.2(as.matrix(coeff3), cex.main= 0.4,
          main= " Heatmap G3 genes",
          Rowv = TRUE,distfun=dist,hclustfun=hclust,scale="row",
          col=redgreen(75),key=TRUE, keysize=1,symkey=FALSE,
          density.info="none",trace="none", cexRow=0.1,cexCol=0.7)
dev.off()
#FOR THE G4:
windows(15,15)
png("heatmap G4.png",width=1000, height=1000)

colnames(coeff4)<-schedule$Group
heatmap.2(as.matrix(coeff4), cex.main= 0.4,
```

```
          main= " Heatmap for group 4 genes",
          Rowv = TRUE,distfun=dist,hclustfun=hclust,scale="row",
          col=redgreen(75),key=TRUE, keysize=1,symkey=FALSE,
          density.info="none",trace="none", cexRow=0.1,cexCol=0.7)
dev.off()
#FOR THE Unk:
windows(15,15)
png("heatmap Unk.png",width=1000, height=1000)

colnames(coeff5)<-schedule$Group
heatmap.2(as.matrix(coeff5), cex.main= 0.4,
          main= " Heatmap for Unspecified group genes",
          Rowv = TRUE,distfun=dist,hclustfun=hclust,scale="row",
          col=redgreen(75),key=TRUE, keysize=1,symkey=FALSE,
          density.info="none",trace="none", cexRow=0.1,cexCol=0.7)
dev.off()
##############
#collected heatmap for all:
collect<-rbind(coeff1,coeff2,coeff3, coeff4,coeff5)
windows(15,15)

colnames(collect)<-schedule$Group
heatmap.2(as.matrix(collect), cex.main= 0.4,
           main= " Heatmap for All DEGs across all subgroups",
           Rowv = TRUE,distfun=dist,hclustfun=hclust,scale="row",
           col=redgreen(75),key=TRUE, keysize=1,symkey=FALSE,
           density.info="none",trace="none", cexRow=0.1,cexCol=0.7)


################
#16#Venn_diagram:
install.packages("VennDiagram")
library(VennDiagram)
trial<-decideTests(ebays.dataset,method="separate",adjust.method="fdr",
                p.value=0.02,lfc=2)

trial_gps<-trial[,1:5]
colnames(trial_gps)<-c("WNT","SHH","G3", "G4", "unknown")
vennDiagram(trial_gps, main="Venn diagram showing differentially
expressed genes across 5 groups",

cex.main=0.7,include=c("up","down"),counts.col=c("red","green"),
            cex=1, lwd=3, cex.lab=0.6,
            circle.col="yellow")
#####
#DEGs in WNT:
diff_genes_WNT<-which(trial_gps[,1]!=0)
diff_genes_WNT2<-trial_gps[diff_genes_WNT,]
write.table(diff_genes_WNT2[,0], "WNT_diff_exp_final.txt", sep="\t",
quote=FALSE)
#overexpressed WNT:
over_WNT<-which(trial_gps[,1]==1)
over_WNT<-trial_gps[over_WNT,]
write.table(over_WNT[,0], "overexp.txt", sep="\t", quote=FALSE)
```

```
#Under expressed WNT:
under_WNT<-which(trial_gps[,1]==-1)
under_WNT<-trial_gps[under_WNT,]
write.table(under_WNT[,0], "underexp.txt", sep="\t", quote=FALSE)
############################
#DEGs SHH:
diff_genes_SHH<-which(trial_gps[,2]!=0)
diff_genes_SHH2<-trial_gps[diff_genes_SHH,]
write.table(diff_genes_SHH2[,0], "SHH_diff_exp_final.txt", sep="\t",
quote=FALSE)
#overexpressed SHH:
over_SHH<-which(trial_gps[,2]==1)
over_SHH<-trial_gps[over_SHH,]
write.table(over_SHH[,0], "overexp_SHH.txt", sep="\t", quote=FALSE)
#Under expressed SHH:
under_SHH<-which(trial_gps[,2]==-1)
under_SHH<-trial_gps[under_SHH,]
write.table(under_SHH[,0], "underexp_SHH.txt", sep="\t", quote=FALSE)
#######
#DEGs G3:
diff_genes_G3<-which(trial_gps[,3]!=0)
diff_genes_G3_2<-trial_gps[diff_genes_G3,]
write.table(diff_genes_G3_2[,0], "G3_diff_exp_final.txt", sep="\t",
quote=FALSE)
#overexpressed G3:
over_G3<-which(trial_gps[,3]==1)
over_G3<-trial_gps[over_G3,]
write.table(over_G3[,0], "overexp_G3.txt", sep="\t", quote=FALSE)
#Under expressed G3:
under_G3<-which(trial_gps[,3]==-1)
under_G3<-trial_gps[under_G3,]
write.table(under_G3[,0], "underexp_G3.txt", sep="\t", quote=FALSE)

#######
#DEGs G4:
diff_genes_G4<-which(trial_gps[,4]!=0)
diff_genes_G4_2<-trial_gps[diff_genes_G4,]
write.table(diff_genes_G4_2[,0], "G4_diff_exp_final.txt", sep="\t",
quote=FALSE)
#overexpressed G4:
over_G4<-which(trial_gps[,4]==1)
over_G4<-trial_gps[over_G4,]
write.table(over_G4[,0], "overexp_G4.txt", sep="\t", quote=FALSE)
#Under expressed G4:
under_G4<-which(trial_gps[,4]==-1)
under_G4<-trial_gps[under_G4,]
write.table(under_G4[,0], "underexp_G4.txt", sep="\t", quote=FALSE)

#######
#DEGs UNS:
diff_genes_UNS<-which(trial_gps[,5]!=0)
diff_genes_UNS_2<-trial_gps[diff_genes_UNS,]
write.table(diff_genes_UNS_2[,0], "UNS_diff_exp_final.txt", sep="\t",
quote=FALSE)
```

```
#overexpressed UNS:
over_UNS<-which(trial_gps[,5]==1)
over_UNS<-trial_gps[over_UNS,]
write.table(over_UNS[,0], "overexp_UNS.txt", sep="\t", quote=FALSE)
#Under expressed UNS:
under_UNS<-which(trial_gps[,5]==-1)
under_UNS<-trial_gps[under_UNS,]
write.table(under_UNS[,0], "underexp_UNS.txt", sep="\t", quote=FALSE)

WNT=nsfiltereddata_01[,c(3,15,25,47,59,62:64)]
SHH=nsfiltereddata_01[,c(5,29:32, 42:44,70,72)]
G3=nsfiltereddata_01[,c(9:12,17,23:24,34:35,37,47, 51, 58:59, 61:62)]
G4=nsfiltereddata_01[,c(1:2, 4, 6:8, 13:15, 18:22, 25, 27:28, 33:36,
38:40,45:46, 49:50, 52:57, 67:70, 72, 74:75)]
G5=nsfiltereddata_01[,c(41,66)]
##########################################
#Pvalue t test for WNT:
##########################################
#WNT vs SHH pvalue:
pvalue_WNT_SHH=NULL
for(i in 1:nrow(WNT)){pvalue_WNT_SHH [i]=t.test(WNT[i,],SHH[i,])$p.value}
write.table(pvalue_WNT_SHH, "pvalue_WNT_SHH.txt", sep="\t", quote=F)

#WNT vs G3:
pvalue_WNT_G3_adj=NULL
for(i in 1:nrow(WNT)){pvalue_WNT_G3 [i]=t.test(WNT[i,],G3[i,])$p.value}

pvalue_WNT_G3_adj<- p.adjust(pvalue_WNT_G3, method="fdr",
n=length(pvalue_WNT_G3))
write.table(pvalue_WNT_G3_adj, "pvalue_WNT_G3_adj.txt", sep="\t",
quote=F)


write.table(pvalue_WNT_G3, "pvalue_WNT_G3.txt", sep="\t", quote=F)

#WNT vs G4:
pvalue_WNT_G4=NULL
for(i in 1:nrow(WNT)){pvalue_WNT_G4 [i]=t.test(WNT[i,],G3[i,])$p.value}

pvalue_WNT_G4_adj<- p.adjust(pvalue_WNT_G4, method="fdr",
n=length(pvalue_WNT_G4))
write.table(pvalue_WNT_G4_adj, "pvalue_WNT_G4_adj.txt", sep="\t",
quote=F)

write.table(pvalue_WNT_G4, "pvalue_WNT_G4.txt", sep="\t", quote=F)

#WNT vs G5:
pvalue_WNT_G5=NULL
for(i in 1:nrow(WNT)){pvalue_WNT_G5 [i]=t.test(G5[i,],WNT[i,])$p.value}

pvalue_WNT_G5_adj<- p.adjust(pvalue_WNT_G5, method="fdr",
n=length(pvalue_WNT_G5))
write.table(pvalue_WNT_G5_adj, "pvalue_WNT_G5_adj.txt", sep="\t",
quote=F)
```

```
write.table(pvalue_WNT_G5, "pvalue_WNT_G5.txt", sep="\t", quote=F)

########################################
#Pvalue t test for SHH:
########################################
#SHH vs WNT:
pvalue_SHH_WNT=NULL
for(i in 1:nrow(SHH)){pvalue_SHH_WNT [i]=t.test(SHH[i,],WNT[i,])$p.value}

pvalue_SHH_WNT_adj<- p.adjust(pvalue_SHH_WNT, method="fdr",
n=length(pvalue_SHH_WNT))
write.table(pvalue_SHH_WNT_adj, "pvalue_SHH_WNT_adj.txt", sep="\t",
quote=F)

write.table(pvalue_SHH_WNT, "pvalue_SHH_WNT.txt", sep="\t", quote=F)

#SHH vs G3:
pvalue_SHH_G3=NULL
for(i in 1:nrow(SHH)){pvalue_SHH_G3 [i]=t.test(SHH[i,],G3[i,])$p.value}

pvalue_SHH_G3_adj<- p.adjust(pvalue_SHH_G3, method="fdr",
n=length(pvalue_SHH_G3))
write.table(pvalue_SHH_G3_adj, "pvalue_SHH_G3_adj.txt", sep="\t",
quote=F)

write.table(pvalue_SHH_G3, "pvalue_SHH_G3.txt", sep="\t", quote=F)

#SHH vs G4:
pvalue_SHH_G4=NULL
for(i in 1:nrow(SHH)){pvalue_SHH_G4 [i]=t.test(SHH[i,],G4[i,])$p.value}

pvalue_SHH_G4_adj<- p.adjust(pvalue_SHH_G4, method="fdr",
n=length(pvalue_SHH_G4))
write.table(pvalue_SHH_G4_adj, "pvalue_SHH_G4_adj.txt", sep="\t",
quote=F)

write.table(pvalue_SHH_G4, "pvalue_SHH_G4.txt", sep="\t", quote=F)

#SHH vs G5:
pvalue_SHH_G5=NULL
for(i in 1:nrow(SHH)){pvalue_SHH_G5 [i]=t.test(SHH[i,],G5[i,])$p.value}

pvalue_SHH_G5_adj<- p.adjust(pvalue_SHH_G5, method="fdr",
n=length(pvalue_SHH_G5))
write.table(pvalue_SHH_G5_adj, "pvalue_SHH_G5_adj.txt", sep="\t",
quote=F)

write.table(pvalue_SHH_G5, "pvalue_SHH_G5.txt", sep="\t", quote=F)

########################################
#Pvalue for G3 marker:
########################################
```

```
#G3 vs WNT:
pvalue_G3_WNT=NULL
for(i in 1:nrow(G3)){pvalue_G3_WNT [i]=t.test(G3[i,],WNT[i,])$p.value}

pvalue_G3_WNT_adj<- p.adjust(pvalue_G3_WNT, method="fdr",
n=length(pvalue_G3_WNT))
write.table(pvalue_G3_WNT_adj, "pvalue_G3_WNT_adj.txt", sep="\t",
quote=F)

write.table(pvalue_G3_WNT, "pvalue_G3_WNT.txt", sep="\t", quote=F)

#G3 vs SHH:
pvalue_G3_SHH=NULL
for(i in 1:nrow(G3)){pvalue_G3_SHH [i]=t.test(G3[i,],SHH[i,])$p.value}

pvalue_G3_SHH_adj<- p.adjust(pvalue_G3_SHH, method="fdr",
n=length(pvalue_G3_SHH))
write.table(pvalue_G3_SHH_adj, "pvalue_G3_SHH_adj.txt", sep="\t",
quote=F)

write.table(pvalue_G3_SHH, "pvalue_G3_SHH.txt", sep="\t", quote=F)

#G3 vs G4:
pvalue_G3_G4=NULL
for(i in 1:nrow(G3)){pvalue_G3_G4 [i]=t.test(G3[i,],G4[i,])$p.value}

pvalue_G3_G4_adj<- p.adjust(pvalue_G3_G4, method="fdr",
n=length(pvalue_G3_G4))
write.table(pvalue_G3_G4_adj, "pvalue_G3_G4_adj.txt", sep="\t", quote=F)

write.table(pvalue_G3_G4, "pvalue_G3_G4.txt", sep="\t", quote=F)

#G3 vs G5:
pvalue_G3_G5=NULL
for(i in 1:nrow(G3)){pvalue_G3_G5 [i]=t.test(G3[i,],G5[i,])$p.value}

pvalue_G3_G5_adj<- p.adjust(pvalue_G3_G5, method="fdr",
n=length(pvalue_G3_G5))
write.table(pvalue_G3_G5_adj, "pvalue_G3_G5_adj.txt", sep="\t", quote=F)

write.table(pvalue_G3_G5, "pvalue_G3_G5.txt", sep="\t", quote=F)

##################################
#Pvalue t test for G4:
##################################

#G4 vs WNT:
pvalue_G4_WNT=NULL
for(i in 1:nrow(G4)){pvalue_G4_WNT [i]=t.test(G4[i,],WNT[i,])$p.value}

pvalue_G4_WNT_adj<- p.adjust(pvalue_G4_WNT, method="fdr",
n=length(pvalue_G4_WNT))
```

```
write.table(pvalue_G4_WNT_adj, "pvalue_G4_WNT_adj.txt", sep="\t",
quote=F)

write.table(pvalue_G4_WNT, "pvalue_G4_WNT.txt", sep="\t", quote=F)

#G4 vs SHH:
pvalue_G4_SHH=NULL
for(i in 1:nrow(G4)){pvalue_G4_SHH [i]=t.test(G4[i,],SHH[i,])$p.value}

pvalue_G4_SHH_adj<- p.adjust(pvalue_G4_SHH, method="fdr",
n=length(pvalue_G4_SHH))
write.table(pvalue_G4_SHH_adj, "pvalue_G4_SHH_adj.txt", sep="\t",
quote=F)

write.table(pvalue_G4_SHH, "pvalue_G4_SHH.txt", sep="\t", quote=F)

#G4 vs G3:
pvalue_G4_G3=NULL
for(i in 1:nrow(G4)){pvalue_G4_G3 [i]=t.test(G4[i,],G3[i,])$p.value}

pvalue_G4_G3_adj<- p.adjust(pvalue_G4_G3, method="fdr",
n=length(pvalue_G4_G3))
write.table(pvalue_G4_G3_adj, "pvalue_G4_G3_adj.txt", sep="\t", quote=F)

write.table(pvalue_G4_G3, "pvalue_G4_G3.txt", sep="\t", quote=F)

#G4 vs G5:
pvalue_G4_G5=NULL
for(i in 1:nrow(G4)){pvalue_G4_G5 [i]=t.test(G4[i,],G5[i,])$p.value}

pvalue_G4_G5_adj<- p.adjust(pvalue_G4_G5, method="fdr",
n=length(pvalue_G4_G5))
write.table(pvalue_G4_G5_adj, "pvalue_G4_G5_adj.txt", sep="\t", quote=F)

write.table(pvalue_G4_G5, "pvalue_G4_G5.txt", sep="\t", quote=F)

###################################
#Pvalue t test for G5:
###################################

#G5 vs WNT:
pvalue_G5_WNT=NULL
for(i in 1:nrow(G5)){pvalue_G5_WNT [i]=t.test(G5[i,],WNT[i,])$p.value}

pvalue_G5_WNT_adj<- p.adjust(pvalue_G5_WNT, method="fdr",
n=length(pvalue_G5_WNT))
write.table(pvalue_G5_WNT_adj, "pvalue_G5_WNT_adj.txt", sep="\t",
quote=F)

write.table(pvalue_G5_WNT, "pvalue_G5_WNT.txt", sep="\t", quote=F)

#G5 vs SHH:
pvalue_G5_SHH=NULL
for(i in 1:nrow(G5)){pvalue_G5_SHH [i]=t.test(G5[i,],SHH[i,])$p.value}
```

```
pvalue_G5_SHH_adj<- p.adjust(pvalue_G5_SHH, method="fdr",
n=length(pvalue_G5_SHH))
write.table(pvalue_G5_SHH_adj, "pvalue_G5_SHH_adj.txt", sep="\t",
quote=F)

write.table(pvalue_G5_SHH, "pvalue_G5_SHH.txt", sep="\t", quote=F)

#G5 vs G3:
pvalue_G5_G3=NULL
for(i in 1:nrow(G5)){pvalue_G5_G3 [i]=t.test(G5[i,],G3[i,])$p.value}

pvalue_G5_G3_adj<- p.adjust(pvalue_G5_G3, method="fdr",
n=length(pvalue_G5_G3))
write.table(pvalue_G5_G3_adj, "pvalue_G5_G3_adj.txt", sep="\t", quote=F)

write.table(pvalue_G5_G3, "pvalue_G5_G3.txt", sep="\t", quote=F)

#G5 vs G4:
pvalue_G5_G4=NULL
for(i in 1:nrow(G5)){pvalue_G5_G4 [i]=t.test(G5[i,],G4[i,])$p.value}

pvalue_G5_G4_adj<- p.adjust(pvalue_G5_G4, method="fdr",
n=length(pvalue_G5_G4))
write.table(pvalue_G5_G4_adj, "pvalue_G5_G4_adj.txt", sep="\t", quote=F)

write.table(pvalue_G5_G4, "pvalue_G5_G4.txt", sep="\t", quote=F)

############################################################
############################################################
#Now calculate average expression of each subgroup:

WNT_mean=apply(WNT, 1, mean)
write.table(WNT_mean, "WNT_mean.txt", sep="\t", quote=F)

SHH_mean=apply(SHH, 1, mean)
write.table(SHH_mean, "SHH_mean.txt", sep="\t", quote=F)

G3_mean=apply(G3, 1, mean)
write.table(G3_mean, "G3_mean.txt", sep="\t", quote=F)

G4_mean=apply(G4, 1, mean)
write.table(G4_mean, "G4_mean.txt", sep="\t", quote=F)

G5_mean=apply(G5, 1, mean)
write.table(G5_mean, "G5_mean.txt", sep="\t", quote=F)

#Attach them together in one file:
av_exp_all_groups= cbind(WNT_mean, SHH_mean, G3_mean, G4_mean, G5_mean)
write.table(av_exp_all_groups, "avg_exp_all_gps.txt", sep="\t", quote=F)
############################################################
############################################################
#after uploading the marker genes on R for each subgroup, search for
their expression profiles:
```

```
##############################

#for wnt:
wnt_marker_exp<-merge(avg_exp_all_gps, wnt_specific_list,
by.x="Probe_ids", by.y="x")
write.table(wnt_marker_exp, "wnt_marker_exp.txt", sep="\t", quote=F)

#for shh:
shh_marker_exp<-merge(avg_exp_all_gps, shh_specific_list,
by.x="Probe_ids", by.y="x")
write.table(shh_marker_exp, "shh_marker_exp.txt", sep="\t", quote=F)

#for g3:
g3_marker_exp<-merge(avg_exp_all_gps, g3_specific_list, by.x="Probe_ids",
by.y="x")
write.table(g3_marker_exp, "g3_marker_exp.txt", sep="\t", quote=F)

#for G4:
g4_marker_exp<-merge(avg_exp_all_gps, g4_specific_list, by.x="Probe_ids",
by.y="x")
write.table(g4_marker_exp, "g4_marker_exp.txt", sep="\t", quote=F)

#for g5:
g5_marker_exp<-merge(avg_exp_all_gps, g5_specific_list, by.x="Probe_ids",
by.y="x")
write.table(g5_marker_exp, "g5_marker_exp.txt", sep="\t", quote=F)

####################################
#final step: extract your marker geens from differentially expressed list
of each group:
####################################
#for wnt:
wnt_diff_markers<-merge( WNT_GENES_0.02_list,wnt_marker_exp,
by.x="row.names", by.y="Probe_ids", all=T)
row.has.na_wnt<-apply(wnt_diff_markers[,3:7], 1,
function(x){any(is.na(x))})
sum(row.has.na_wnt)
wnt_diff_markers<-wnt_diff_markers[!row.has.na_wnt,]
xx<-WNT_GENES_0.02_list[!row.has.na_wnt,]
wnt_diff_markers<-cbind(wnt_diff_markers, xx)
wnt_diff_markers<-wnt_diff_markers[,-c(0,2)]
write.table(wnt_diff_markers, "wnt_diff_markers.txt", sep="\t", quote=F)

#for SHH:

shh_diff_markers<-merge( SHH_GENES_0.02_list,shh_marker_exp,
by.x="row.names", by.y="Probe_ids", all=T)
row.has.na_shh<-apply(shh_diff_markers[,3:7], 1,
function(x){any(is.na(x))})
sum(row.has.na_shh)
shh_diff_markers<-shh_diff_markers[!row.has.na_shh,]
logfc<-SHH_GENES_0.02_list[!row.has.na_shh,]
shh_diff_markers<-cbind(shh_diff_markers, logfc)
shh_diff_markers<-shh_diff_markers[,-c(0,2)]
```

```
write.table(shh_diff_markers, "shh_diff_markers.txt", sep="\t", quote=F)

#for G3:

g3_diff_markers<-merge( G3_GENES_0.02_list,g3_marker_exp,
by.x="row.names", by.y="Probe_ids", all=T)
row.has.na_g3<-apply(g3_diff_markers[,3:7], 1,
function(x){any(is.na(x))})
sum(row.has.na_g3)
g3_diff_markers<-g3_diff_markers[!row.has.na_g3,]
logfc<-G3_GENES_0.02_list[!row.has.na_g3,]
g3_diff_markers<-cbind(g3_diff_markers, logfc)
g3_diff_markers<-g3_diff_markers[,-c(0,2)]
write.table(g3_diff_markers, "g3_diff_markers.txt", sep="\t", quote=F)

#for G4:

g4_diff_markers<-merge( G4_GENES_0.02_list,g4_marker_exp,
by.x="row.names", by.y="Probe_ids", all=T)
row.has.na_g4<-apply(g4_diff_markers[,3:7], 1,
function(x){any(is.na(x))})
sum(row.has.na_g4)
g4_diff_markers<-g4_diff_markers[!row.has.na_g4,]
logfc<-G4_GENES_0.02_list[!row.has.na_g4,]
g4_diff_markers<-cbind(g4_diff_markers, logfc)
g4_diff_markers<-g4_diff_markers[,-c(0,2)]
write.table(g4_diff_markers, "g4_diff_markers.txt", sep="\t", quote=F)

#for G5:

g5_diff_markers<-merge( G5_GENES_0.02_list,g5_marker_exp,
by.x="row.names", by.y="Probe_ids", all=T)
row.has.na_g5<-apply(g5_diff_markers[,3:7], 1,
function(x){any(is.na(x))})
sum(row.has.na_g5)
g5_diff_markers<-g5_diff_markers[!row.has.na_g5,]
logfc<-G5_GENES_0.02_list[!row.has.na_g5,]
g5_diff_markers<-cbind(g5_diff_markers, logfc)
g5_diff_markers<-g5_diff_markers[,-c(0,2)]
write.table(g5_diff_markers, "g5_diff_markers.txt", sep="\t", quote=F)

##########################
```