

# R Tutorial: Geometric Interpretation of Gene Co-Expression Network Analysis, Applied to Yeast Microarray Data

Jun Dong, Steve Horvath

Correspondence: [shorvath@mednet.ucla.edu](mailto:shorvath@mednet.ucla.edu), <http://www.ph.ucla.edu/biostat/people/horvath.htm>

## General instructions:

There are two network function files. One (NetworkFunctions.txt) is used in Zhang and Horvath (2005). The other (NetworkFunctions1.txt) contains additional functions used only in this analysis, which are also maintained separately.

```
# =====  
# Read in the libraries and check the size of memory.  
# =====  
  
# Libraries  
library(sna) # this is needed for closeness, DegreeInOut  
library(MASS)  
library(cluster)  
library(sma) # different from sna! this is needed for plot.mat below  
library(impute) # needed for imputing missing value before PCA  
library(splines) # for the spline predictor to estimate the number of clusters  
library(Hmisc) # install it for the C-index calculations  
  
# Memory  
# memory.size(TRUE) # check the maximum memory that can be allocated  
memory.limit(size=3448) # increase the available memory to 3GB  
  
# =====  
# Read in all the functions for network analysis and load the data.  
# =====  
  
# put the data file and the two function files in this directory. You should change the  
# path to your own where you put your files.  
setwd("E:/Documents and Settings/jundong/My Documents/Steve/Network  
Char/tutorial");  
source("NetworkFunctions.txt"); source("NetworkFunctions1.txt");  
  
# =====  
# Prepare the data.  
# =====  
  
dat0 <- read.csv("YEASTCellCycle4000.csv",header=T, row.names=1)  
  
# the following gene summary file contains information on the yeast genes  
datSummary=dat0[,1:7]  
# the column essentiality indicates which gene is essential for yeast survival
```

```

table(datSummary$essentiality)
# message: there are 645 essential genes

# the following data frame contains the gene expression data: columns are genes, rows
are arrays (samples)
datExpr <- dat0[,8:51]

# Now we define the adjacency matrix for a weighted network, i.e. soft thresholding.
# Calculate the Pearson Correlation Matrix
corhelp=cor(t(datExpr),use="pairwise.complete.obs")
# use the power adjacency function
power1=7
AdjMat1 = abs(corhelp)^ power1
diag(AdjMat1)=0
# This is the definition of the connectivity, a.k.a degree
Degree <- apply(AdjMat1,2,sum)

# =====
# Construct modules.
# =====

# This code allows one to restrict the analysis to the most connected genes, which
may speed up calculations when it comes to module detection.
DegCut = 2001 # number of most connected genes that will be considered
DegreeRank <- rank(-Degree)
restDegree <- DegreeRank <= DegCut
# thus our module detection uses the following number of genes
sum(restDegree)

# Now we restrict the adjacency matrix to the most connected genes

AdjMat1rest <- AdjMat1[restDegree,restDegree]

# The following code computes the topological overlap matrix based on the adjacency
matrix.

dissGTOM1=TOMdist1(AdjMat1rest)
collect_garbage()

# Now we carry out hierarchical clustering with the TOM matrix. Branches of the
resulting clustering tree will be used to define gene modules.

hierGTOM1 <- hclust(as.dist(dissGTOM1),method="average");
colorh1=as.character(modulecolor2(hierGTOM1,h1=.97))
table(colorh1)
colorlevel1=levels(factor(colorh1))
grey=5 # To flag which module is the grey module

Alldegrees1=DegreeInOut(as.matrix(AdjMat1rest),colorh1)

```

```
datExpr2=datExpr[restDegree,]
datalab="Yeast"
nrow=3; ncol=3;
```

```
GS= datSummary$essentiality [restDegree]
```

```
# =====
# Analysis based on Singular Value Decomposition (SVD) of Gene Expression
# Profiles.
# =====
# The following computes the proportion of variance explained by the eigengenes in
the gene expression profiles, in each module and for the whole network. Note that the
first eigengene is also called Module Eigengene, and will be studied further.
```

```
library(impute) # needed for imputing missing value before PCA
```

```
m1=ModulePrinComps2(t(datExpr2), colorh1)
names(m1); list(dim(m1[[1]]), dim(m1[[2]]))
```

```
no.MEs=5
# in the following, rows are genes and columns are samples
datModule= datExpr2
# impute missing data in expression profiles
datModule=impute.knn(as.matrix(datModule))
datModule=t(scale(t(datModule)))
svd1=svd(datModule)
varexplained= (svd1$d[1:no.MEs])^2/sum(svd1$d^2)
```

```
# construct table 2: variance explained by the first 5 eigengenes
varexplained1=matrix(0,nrow=no.MEs, ncol=length(colorlevel1)+1)
varexplained1[,1:length(colorlevel1)]=as.matrix(m1[[2]])
varexplained1[(length(colorlevel1)+1)]=varexplained
colnames(varexplained1)=c(colorlevel1, "Network")
rownames(varexplained1)=c(paste("EigenGene",1:no.MEs, sep=""))
signif(varexplained1,2)
```

```
# This table shows that in tight modules (tightly connected genes), the proportion of
variance explained by the module (first) eigengenes is very high (0.57-0.71). While
for the grey module or the whole network, it is only around 0.3.
```

```
# =====
# Relating the Module Eigengenes to each other and to external traits
# =====
```

```
tmp=data.frame(m1[[1]])
names(tmp)= paste("ME.",colorlevel1,sep="")
```

```
setwd("E:/Documents and Settings/jundong/My
Documents/Steve/January2007PNAS/plotsPNAS")
```

```

# windows()

postscript(paste(datalab, "-ME-pairs.ps", sep=""), width=8.5, height=8.5,
horizontal=F, paper="special")
pairs(tmp, upper.panel = panel.lm, lower.panel = panel.cor1 , diag.panel=panel.hist,
cex.labels=1)
dev.off()
#### <plot>: GBM-ME-pairs

#### Figure 1 ####
ClusterSamples=hclust(dist(scale(t(datExpr2))),method="average")
# plot(ClusterSamples)

for(i in 1:length(colorlevel1) ){
which.module=colorlevel1[i]
#windows(height=5,width=7)
postscript(paste(datalab, "-ME-", colorlevel1[i], ".ps", sep=""), width=8.5, height=6,
horizontal=F, paper="special")
par(mfrow=c(2,1), mar=c(0.3, 4, 3, 2))
plot.mat(t(scale(t(datExpr2)[ClusterSamples$order,][,colorh1==which.module ] ),nrg
cols=30,rlabels=F,rcols=which.module, main=which.module, cex.main=3)
#windows(height=5,width=7)
#par(mfrow=c(1,1), mar=c(1, 3, 3, 1))
par(mar=c(1, 2.6, 0, 0.6))
barplot(m1[[1]][ClusterSamples$order,i], col=which.module, main="", cex.main=3)
dev.off()
}
#### <plot>: GBM-ME-brown

#===== For dendrograms =====
#windows()
postscript(paste(datalab, "-hierGTOM1-soft.ps", sep=""), width=8.5, height=6,
horizontal=F, paper="special")
par(mfrow=c(2,1), mar=c(0,2,2,0))
plot(hierGTOM1, main="Yeast Co-expression Network: Network Dendrogram",
labels=F, xlab="", sub="");
hclustplot1(hierGTOM1,colorh1, title1="Colored by module membership")
dev.off()
#### <plot>: GBM-hierGTOM1-soft

AdjMat1resthard= abs(corhelp) >= 0.65
AdjMat1resthard = AdjMat1resthard [restDegree,restDegree]
dissGTOM1hard=TOMdist1(AdjMat1resthard)
hierGTOM1hard <- hclust(as.dist(dissGTOM1hard),method="average");
#windows()
postscript(paste(datalab, "-hierGTOM1-hard.ps", sep=""), width=8.5, height=6,
horizontal=F, paper="special")
par(mfrow=c(2,1), mar=c(0,2,2,0))
plot(hierGTOM1hard, main="Yeast Co-expression Network: Hard Thresholding",
labels=F, xlab="", sub="");

```

```
hclustplot1(hierGTOM1hard,colorh1, title1="Colored by module membership of Soft
Thresholding")
dev.off()
### <plot>: GBM-hierGTOM1-hard
```

```
rm(DegreeRank, AdjMat1, AdjMat1rest, AdjMat1resthard, corhelp, dissGTOM1,
dissGTOM1hard, datExpr, Alldegrees1); collect_garbage();
```

```
save.image(paste(datalab, "-all.RData", sep=""))
```

```
#===== End For dendrograms =====
# Start with other beta value (power1)
```

```
load("E:/Documents and Settings/jundong/My
Documents/Steve/January2007PNAS/plotsHubGeneRelevance/Yeast-all.RData")
setwd("E:/Documents and Settings/jundong/My
Documents/Steve/January2007PNAS/plotsHubGeneRelevance")
```

```
power1=1
```

```
# =====
# Analysis based on Factorizability Decomposition (FD).
# =====
# FDADJ() is a function that calculates various module specific quantities. The name
is the abbreviation for "Factorizability Decomposition of ADJacency matrices". In
default, this function uses the function NPC.iterate() to calculate the conformity,
which is based on the iterative algorithm described in the Horvath, Dong, Yip (2006).
In very rare cases, it takes a long time for the algorithm to converge.
```

```
# =====
# Weighted network: Soft Thresholding Method
# =====
```

```
# First, we apply FDADJ() to a weighted network that uses power adjacency function.
```

```
time.start=Sys.time()
for(i in 1:length(colorlevel1) ){
adjtemp=cor(t(datExpr2[colorh1==colorlevel1[i], ]), use="pairwise.complete.obs" )
adjtemp= abs(adjtemp)^power1
assign(paste("FDsoft", colorlevel1[i],sep=""), FDADJ(adjtemp))
rm(adjtemp);collect_garbage()
}
time.end=Sys.time()
time.end-time.start # total time for running this part
```

```

# Construct the table 1.
module.soft=matrix(NA, ncol=length(colorlevel1), nrow=
length(FDsoftblue$summary.names))
colnames(module.soft)=colorlevel1
rownames(module.soft)= FDsoftblue$summary.names
for(i in 1:length(colorlevel1)){
temp=eval(as.name(paste("FDsoft",colorlevel1[i],sep="")))
module.soft[,i]=temp$summary
}
signif(module.soft[-1,],3)

# =====
# Calculation based on approximation: CFApprox=|cor(x, ME)|^power1
# =====

## Yeast, only GS, ModuleSignificance and HubGeneSignificance are available
## GSExp, ModuleSignificanceExp, HubGeneSignificanceExp and
## EigengeneSignificance are NA.
#EigengeneSignificance= abs(cor(Trait, m1[[1]]) )^power1; EigengeneSignificance
#GS= abs(cor(t(datExpr2), Trait, use="pairwise.complete.obs") )^power1
#windows(height=5)
if(power1==1){
postscript(paste(datalab, "-Enrichment1-p", power1, ".ps", sep=""), width=8.5,
height=6, horizontal=F, paper="special")
par(mar=c(2,5,5,2)+0.1))
ModuleEnrichment2(GS,colorh1,cex.lab1=2.3, cex.main1=1.8)
dev.off()
### <plot>: GBM-Enrichment1-p1
}

#GSExp=rep(NA, length(GS))

# based on power1
for(i in 1:length(colorlevel1)){
y1=cor(t(datExpr2[colorh1==colorlevel1[i],]),m1[[1]][,i],use="pairwise.complete.obs
"); y1=abs(y1)^power1; #Expression Conformity
Size=length(y1);
y2=sum(y1)*y1; #Expression Connectivity
y3=sum(y2)/(Size*(Size-1)); #Expression Density
y4=Size*(max(y2)-mean(y2))/((Size-1)*(Size-2)); #Expression Centralization
y5=sqrt(Size*sum(y2^2)/sum(y2)^2-1); #Expression Heterogeneity
y6=(sum(y1^2)/sum(y1))^2; ##Expression ClusterCoef
y7=y1* sum(y1^2)/sum(y1)
assign(paste("Expsoft",colorlevel1[i],sep=""), list(CFExp=y1, DensityExp = y3,
CentralizationExp =y4, HeterogeneityExp=y5, ClusterCoefExp=y6, MARExp=y7))
#GSExp[colorh1==colorlevel1[i]]=y1* EigengeneSignificance[i]
rm(y1,y2,y3,y4,y5,y6,y7)
}

```

```

# Construct the table 1.
Exp.soft=matrix(-666, nc=length(colorlevel1), nrow=4)
colnames(Exp.soft)=colorlevel1
rownames(Exp.soft)=c("DensityExp", "CentralizationExp", "HeterogeneityExp",
"ClusterCoefExp")
for(i in 1:length(colorlevel1)){
temp=eval(as.name(paste("Expsoft", colorlevel1[i], sep="")))
Exp.soft[,i]=c(temp$DensityExp, temp$CentralizationExp, temp$HeterogeneityExp,
temp$ClusterCoefExp)
}
signif(Exp.soft,3)

```

```

ModuleSignificance=tapply(GS, colorh1, mean); ModuleSignificance;
#ModuleSignificanceExp=tapply(GSExp, colorh1, mean); ModuleSignificanceExp;

```

```

HubGeneSignificance=rep(NA, length(ModuleSignificance) )
#HubGeneSignificanceExp=rep(NA, length(ModuleSignificance) )
for(i in 1:length(colorlevel1)){
temp=eval(as.name(paste("FDsoft", colorlevel1[i], sep="")))
GS.temp=GS[colorh1==colorlevel1[i]]
K=temp$kWithin/max(temp$kWithin)
HubGeneSignificance[i]=sum(GS.temp*K)/sum(K^2)
#temp=eval(as.name(paste("Expsoft", colorlevel1[i], sep="")))
#GS.temp=GSExp[colorh1==colorlevel1[i]]
#K=temp$CFExp/max(temp$CFExp)
#HubGeneSignificanceExp[i]= sum(GS.temp*K)/sum(K^2)
}

```

```

HubGeneSignificance; #HubGeneSignificanceExp;

```

```

if(0){
#windows(width=7, height=5)
postscript(paste(datalab, "-ModuleSignificance-p", power1, ".ps", sep=""), width=8.5,
height=6, horizontal=F, paper="special")
par(mar=c(6,5,5,2)+0.1))
plot(ModuleSignificanceExp, ModuleSignificance, col=colorlevel1, pch=19, cex=2,
xlab="Eigengene-based Module Significance", ylab="Module Significance",
cex.main=3, cex.lab=2.3, cex.axis=1.6, cex.axis=1.6)
abline(0,1, col=2)
title(substitute( {beta == exp1} * " " * {R^2 == exp2}, list(exp1 = power1, exp2 =
signif(cor(ModuleSignificanceExp[-grey], ModuleSignificance[-grey])
^2,2))),cex.main=3)
#title(paste("beta=", power1, ", R^2=",signif(cor(ModuleSignificanceExp[-grey],
ModuleSignificance[-grey]) ^2,2), sep=""),cex.main=3)
dev.off()
### <plot>: GBM-ModuleSignificance-p1

```

```

#windows(width=7, height=5)

```

```

postscript(paste(datalab, "-HubGeneSignificance-p", power1, ".ps", sep=""),
width=8.5, height=6, horizontal=F, paper="special")
par(mar=(c(6,5,5,2)+0.1))
plot(HubGeneSignificanceExp, HubGeneSignificance, col=colorlevel1, pch=19,
cex=2, xlab="Eigengene-based Hub Gene Significance", ylab="Hub Gene
Significance", cex.main=3, cex.lab=2.3, cex.axis=1.6)
abline(0,1, col=2)
title(substitute( {beta == exp1} * ", " * {R^2 == exp2}, list(exp1 = power1, exp2 =
signif(cor(HubGeneSignificanceExp[-grey], HubGeneSignificance[-grey])
^2,2))),cex.main=3)
#title(paste("beta=", power1, ", R^2=",signif(cor(HubGeneSignificanceExp[-grey],
HubGeneSignificance[-grey]) ^2,2), sep=""),cex.main=3)
dev.off()
### <plot>: GBM-HubGeneSignificance-p1
}

```

```

#windows(width=7, height=5)
postscript(paste(datalab, "-MS-HGS-Density-p", power1, ".ps", sep=""), width=8.5,
height=6, horizontal=F, paper="special")
par(mar=(c(6,5,5,2)+0.1))
tmp1= HubGeneSignificance*sqrt(module.soft[3,])
tmp3= ModuleSignificance
plot(tmp1, tmp3, col= colorlevel1, pch=19, ylab="Module Significance", xlab= "",
cex=2, cex.main=3, cex.lab=2.3, cex.axis=1.6)
#"HubGeneSignificance*sqrt(Density)"
abline(0,1, col=2)
abline(lm(tmp3 [-grey]~ tmp1 [-grey]), col=4)
title(substitute( {beta == exp1} * ", " * {R^2 == exp2}, list(exp1 = power1, exp2 =
signif(cor(tmp1[-grey], tmp3[-grey])^2,2))),cex.main=3)
title(xlab= expression(paste("HubGeneSignificance * ", sqrt(Density) ) ), line=4,
cex.lab=2.3)
#title(paste("beta=", power1, ", R^2=",signif(cor(tmp1[-grey], tmp3[-grey])^2,2),
sep=""),cex.main=3)
dev.off()
### <plot>: GBM-MS-HGS-Density-p1

```

```

if(0){
#windows(width=7, height=5)
postscript(paste(datalab, "-HubGene-Eigengene-Significance-p", power1, ".ps",
sep=""), width=8.5, height=6, horizontal=F, paper="special")
par(mar=(c(6,5,5,2)+0.1))
tmp1= HubGeneSignificance
tmp3= as.vector(EigengeneSignificance)
plot(tmp1, tmp3, col= colorlevel1, pch=19, ylab="Eigengene Significance",
xlab="Hub Gene Significance", cex=2, cex.main=3, cex.lab=2.3, cex.axis=1.6)
abline(0,1, col=2)
title(substitute( {beta == exp1} * ", " * {R^2 == exp2}, list(exp1 = power1, exp2 =
signif(cor(tmp1[-grey], tmp3[-grey])^2,2))),cex.main=3)

```

```
#title(paste("beta=", power1, ", R^2=", signif(cor(tmp1[-grey], tmp3[-grey])^2,2),
sep=""),cex.main=3)
dev.off()
#### <plot>: GBM-HubGene-Eigengene-Significance-p1
}
```

```
### check max a_{e,i}
```

```
for(i in 1:length(colorlevel1)){
temp=eval(as.name(paste("Expsoft",colorlevel1[i],sep="")))
print(max(temp$CFExp))
}
```

```
for(i in 1:length(colorlevel1) ){
#i=1 # 2 for brown and 1 for blue
temp=eval(as.name(paste("FDsoft",colorlevel1[i],sep="")))
tmp3=GS[colorh1== colorlevel1[i]];
tmp1=temp$kWithin/ max(temp$kWithin) #K
# tmp=lm(tmp3~tmp1)
tmp=lm(tmp3~tmp1-1)
#print(cor.test(tmp1, tmp3), method="s")
#windows(width=8, height=5)
postscript(paste(datalab, "-GS-K-", colorlevel1[i], "-0Intercept-p", power1, ".ps",
sep=""), width=8.5, height=6, horizontal=F, paper="special")
par(mar=(c(6,5,5,2)+0.1))
plot(tmp1, tmp3, col=colorlevel1[i], pch="o", xlab="K", ylab="Gene Significance",
cex.lab=2.3, cex.axis=1.6, xlim=c(0,1), ylim=c(0,max(tmp3)))
title(substitute( {exp0} * " , " * {beta == exp1} * " , slope = " * {exp2} ,
list(exp0=colorlevel1[i], exp1 = power1, exp2 = signif(tmp$coef,2) )),cex.main=3)
# points(0,0, pch=19, col= colorlevel1[i], cex=1.5)
# title(paste(colorlevel1[i], " , beta=",power1, " , slope=",signif(tmp$coef,2), sep=""),
cex.main=3)
# abline(tmp$coef, col=2)
## for no intercept (Hub Gene Significance)
abline(c(0,tmp$coef), col=2)
dev.off()
}
#### <plot>: GBM-GS-K-brown-0Intercept-p1
```

```
#windows(width=9, height=4)
postscript(paste(datalab, "-K-EigenCF-p", power1, ".ps", sep=""), width=(ncol*4),
height=(nrow*3), horizontal=F, paper="special")
par(mar=(c(6,5,5,2)+0.1))
par(mfrow=c(nrow,ncol))
for(i in 1:length(colorlevel1) ){
temp=eval(as.name(paste("FDsoft",colorlevel1[i],sep="")))
temp1=eval(as.name(paste("Expsoft",colorlevel1[i],sep="")))
cor=cor(temp1$CFExp, temp$kWithin)
plot(temp1$CFExp,temp$kWithin/max(temp$kWithin),xlab="Eigengene
Conformity", ylab="K",col=colorlevel1[i], cex.lab=2, cex.axis=1.6, cex.main=2.5)
```

```

abline(0,1,col=2)
abline(lm((temp$kWithin/max(temp$kWithin)) [-grey]~ temp1$CFExp [-grey]),
col=4)
title(substitute( {exp0} * " , " * {beta == exp1} * " , " * {R^2 == exp2},
list(exp0=colorlevel1[i], exp1 = power1, exp2 = signif(cor^2,2) )),cex.main=2.6)
#title(paste(colorlevel1[i], " , beta=", power1, " , R^2=",signif(cor^2,2), sep=""))
}
dev.off()
### <plot>: GBM-K-EigenCF-p1

#windows(width=7, height=5)
postscript(paste(datalab, "-Centralization-sqrtDensity-p", power1, ".ps", sep=""),
width=8.5, height=6, horizontal=F, paper="special")
par(mar=(c(6,6,5,2)+0.1))
plot(module.soft[6,], sqrt(module.soft[3,])*(1- sqrt(module.soft[3,])), col=colorlevel1,
xlab="Centralization", ylab= expression(paste(sqrt(Density), " ", (1-
sqrt(Density)))) ,cex.lab=2.3, cex.axis=1.6, pch=19, cex=2)
#"sqrt(Density)*(1-sqrt(Density))"
abline(0,1, col=2)
abline(lm( (sqrt(module.soft[3,])*(1- sqrt(module.soft[3,])))[-grey]~ (module.soft[6,])
[-grey]), col=4)
title(substitute({beta == exp1} * " , " * {R^2 == exp2}, list(exp0=colorlevel1[i], exp1
= power1, exp2 = signif(cor(module.soft[6,-grey], sqrt(module.soft[3,-grey]) *(1-
sqrt(module.soft[3,-grey])))^2,2) )),cex.main=3)
#title(paste("beta=",power1, " , R^2=",signif(cor(module.soft[6,], sqrt(module.soft[3,])
*(1- sqrt(module.soft[3,])))^2,2), sep=""), cex.main=3)
dev.off()
### <plot>: GBM-Centralization-sqrtDensity-p1

maxconnectivity=rep(NA, length(colorlevel1))
for(i in 1:length(colorlevel1) ){
temp=eval(as.name(paste("FDsoft",colorlevel1[i],sep="")))
maxconnectivity[i]=max(temp$kWithin)
}
maxconnectivity
#windows(width=7, height=5)
postscript(paste(datalab, "-maxk-sqrtDensity-p", power1, ".ps", sep=""), width=8.5,
height=6, horizontal=F, paper="special")
par(mar=(c(6,6,5,2)+0.1))
plot(maxconnectivity/(module.soft[1,]-1), sqrt(module.soft[3,]), col=colorlevel1,
xlab="max(k)/(n-1)", ylab= expression(sqrt(Density) ),cex.lab=2.3, cex.axis=1.6,
pch=19, cex=2)
abline(0,1, col=2)
abline(lm( (sqrt(module.soft[3,]))[-grey]~ (maxconnectivity/(module.soft[1,]-1)) [-
grey]), col=4)
title(substitute({beta == exp1} * " , " * {R^2 == exp2}, list(exp0=colorlevel1[i], exp1
= power1, exp2 = signif(cor(maxconnectivity[-grey]/(module.soft[1,-grey]-1),
sqrt(module.soft[3,-grey]))^2,2))), cex.main=3)
# title(paste("beta=",power1, " , R^2=", signif(cor(maxconnectivity/(module.soft[1,]-
1), sqrt(module.soft[3,]))^2,2), sep=""), cex.main=3)

```

```

dev.off()
### <plot>: GBM-maxk-sqrtDensity-p1

#####
#### Factorizability of microarray data
#####
library(impute) # needed for imputing missing value before PCA

# Calculate the Factorizability of X (microarray matrix)
Factorizability.X=rep(-666, length=length(colorlevel1)) # defined for 0-diag matrix
names(Factorizability.X)=colorlevel1
Factorizability.X.approx= Factorizability.X # Approximation
Factorizability.X.approx1= Factorizability.X # defined for 1-diag matrix

for(i in 1:length(colorlevel1)){
  datModule=datExpr2[colorh1==colorlevel1[i],]
  datModule=impute.knn(as.matrix(datModule))
  y1=as.vector(cor(t(datModule), m1[[1]][,i],use="pairwise.complete.obs"))
  mat1=cor(t(datModule),use="pairwise.complete.obs")
  mat2=as.matrix(outer(y1,y1))
  Factorizability.X.approx1[i]=1-sum( (mat1-mat2)^2 )/sum(mat1^2)
  Factorizability.X.approx[i]=sum( mat2^2 )/sum(mat1^2)
  diag(mat1)=0; diag(mat2)=0;
  Factorizability.X[i]=1-sum( (mat1-mat2)^2 )/sum(mat1^2)
}
rm(y1, mat1, mat2)

signif(Factorizability.X, 3)
signif(Factorizability.X.approx, 3)
signif(Factorizability.X.approx1, 3)
signif(module.soft[2,], 3)
signif(varexplained1[1,],3)

### Figure 2 ###

### Weighted ###
CC.soft=NULL
for(i in 1:length(colorlevel1) ){
  temp=eval(as.name(paste("FDsoft",colorlevel1[i],sep="")))
  CC.soft=c(CC.soft, temp$ClusterCoef)
}
CC.soft.new=(module.soft[9,]^2+1)^2*module.soft[3,]

greys=( sum(module.soft[1,1:(grey-1)])+1 ) : (sum(module.soft[1,1:grey]) )

#windows(width=7, height=5)
# par(mar=(c(6,5,5,2)+0.1))

```

```

postscript(paste(datalab, "-CCHeteroDensity-p", power1, ".ps", sep=""), width=8.5,
height=6, horizontal=F, paper="special")
par(mar=c(6,6,5,2)+0.1))
plot(CC.soft.new, module.soft[13,], col=colorlevel1, pch="-", ylab="Clustering
Coefficient", xlab="", cex=2, cex.main=3, cex.lab=2.3, cex.axis=1.6,
ylim=range(CC.soft))
#(1+Heterogeneity^2)^2*Density
abline(0,1, col=2)
points(rep(CC.soft.new, module.soft[1,]), CC.soft, col=rep(colorlevel1,
module.soft[1,]), pch="o", cex=0.6)
rm.grey=rep(T, length(CC.soft))
rm.grey[(sum(module.soft[1,1:(grey-1)]+1): sum(module.soft[1,1:grey]) ] =F
abline(lm( (CC.soft)[-greys]~ (rep(CC.soft.new, module.soft[1,]) [-greys]), col=4)
title(substitute( {beta == exp1} * ", " * {R^2 == exp2}, list(exp1 = power1, exp2 =
signif(cor(rep(CC.soft.new[-grey], module.soft[1,-grey]),
CC.soft[rm.grey]^2,2) )),cex.main=3)
title(xlab= expression(paste( "(", 1+ Heterogeneity ^2, ")"), ""^2, " * Density",
sep="" ) ), line=4, cex.lab=2.3)
#title(paste("beta=", power1, ", R^2=",signif(cor(rep(CC.soft.new, module.soft[1,]),
CC.soft)^2,2), sep=""),cex.main=3)
dev.off()
### <plot>: GBM-CCHeteroDensity-p1

```

### ### Weighted ###

```

datExpr.hub=matrix(NA, ncol=length(colorlevel1), nrow=dim(datExpr2)[2])
for(i in 1:length(colorlevel1) ){
temp=eval(as.name(paste("FDsoft",colorlevel1[i],sep="")))
temp=temp$kWithin
select= match(min(rank(-temp)), rank(-temp))
datExpr.hub[,i]= as.numeric( datExpr2[colorh1==colorlevel1[i],][select,] )
}
for(i in 1:length(colorlevel1) ){
#windows(width=8, height=5); par(mar=c(5,5,4,2)+0.1)
postscript(paste(datalab, "-ME-Hub-", colorlevel1[i], "-p", power1, ".ps", sep=""),
width=8.5, height=6, horizontal=F, paper="special")
par(mar=c(6,6,5,2)+0.1))
plot(datExpr.hub[,i], m1[[1]][,i], col=colorlevel1[i], xlab="Hub Gene Expression",
ylab="Module Eigengene Expression", pch=19, cex.lab=2.3, main="", cex=1,
cex.axis=1.6)
title(substitute( {exp0} * ", " * {beta == exp1} * ", " * {R^2 == exp2}, list(exp0=
colorlevel1[i], exp1 = power1, exp2 = signif(cor(datExpr.hub[,i], m1[[1]][,i],
use="pairwise.complete.obs")^2,2) )),cex.main=3)
abline(lm(m1[[1]][,i] ~ datExpr.hub[,i]),col=2)
dev.off()
}
### <plot>: GBM-ME-Hub-brown-p1

```

### ### Figure 7 ####

### ### Weighted ###

```
#windows(width=7, height=5)
#par(mar=(c(5,5,4,1)+0.1))
postscript(paste(datalab, "-Density-p", power1, ".ps", sep=""), width=8.5, height=6,
horizontal=F, paper="special")
par(mar=(c(6,6,5,2)+0.1))
plot(Exp.soft[1,], module.soft[3,], col=colorlevel1, xlab="Eigengene-based Density",
ylab="Density",cex.lab=2.3, cex.axis=1.6, pch=19, cex=2)
abline(0,1, col=2)
abline(lm( (module.soft[3,])[-grey]~ (Exp.soft[1,]) [-grey]), col=4)
title(substitute( {beta == exp1} * " , " * {R^2 == exp2}, list(exp1 = power1, exp2 =
signif(cor(Exp.soft[1,-grey],module.soft[3,-grey])^2,2))),cex.main=3)
dev.off()
### <plot>: GBM-Density-p1
```

```
#windows(width=7, height=5)
#par(mar=(c(5,5,4,1)+0.1))
postscript(paste(datalab, "-Centralization-p", power1, ".ps", sep=""), width=8.5,
height=6, horizontal=F, paper="special")
par(mar=(c(6,6,5,2)+0.1))
plot(Exp.soft[2,], module.soft[6,], col=colorlevel1, xlab="Eigengene-based
Centralization", ylab="Centralization",cex.lab=2.3, cex.axis=1.6, pch=19, cex=2)
abline(0,1, col=2)
abline(lm( (module.soft[6,])[-grey]~ (Exp.soft[2,]) [-grey]), col=4)
#title(main=paste("beta=",power1, ",
R^2=",signif(cor(Exp.soft[2,],module.soft[6,])^2,2), sep=""), cex.main=3)
title(substitute( {beta == exp1} * " , " * {R^2 == exp2}, list(exp1 = power1, exp2 =
signif(cor(Exp.soft[2,-grey],module.soft[6,-grey])^2,2))),cex.main=3)
dev.off()
### <plot>: GBM-Centralization-p1
### Figure 9a
```

```
#windows(width=7, height=5)
#par(mar=(c(5,5,4,1)+0.1))
postscript(paste(datalab, "-Heterogeneity-p", power1, ".ps", sep=""), width=8.5,
height=6, horizontal=F, paper="special")
par(mar=(c(6,6,5,2)+0.1))
plot(Exp.soft[3,], module.soft[9,], col=colorlevel1, xlab="Eigengene-based
Heterogeneity", ylab="Heterogeneity", cex.lab=2.3, cex.axis=1.6, pch=19, cex=2)
# abline(test2$coef)
abline(0,1, col=2)
abline(lm( (module.soft[9,])[-grey]~ (Exp.soft[3,]) [-grey]), col=4)
#title( main=paste("beta=",power1, sep=""),cex.main=3)
title(substitute( {beta == exp1} * " , " * {R^2 == exp2}, list(exp1 = power1, exp2 =
signif(cor(Exp.soft[3,-grey],module.soft[9,-grey])^2,2))),cex.main=3)
dev.off()
### <plot>: GBM-Heterogeneity-p1
### Figure 9c
```

```

CC.soft=NULL
for(i in 1:length(colorlevel1) ){
temp=eval(as.name(paste("FDsoft",colorlevel1[i],sep="")))
CC.soft=c(CC.soft, temp$ClusterCoef)
}
CC.soft.new=rep(Exp.soft[4,],module.soft[1,])

#windows(width=7, height=5)
#par(mar=c(5,5,4,1)+0.1))
postscript(paste(datalab, "-ClusterCoef-p", power1, ".ps", sep=""), width=8.5,
height=6, horizontal=F, paper="special")
par(mar=c(6,6,5,2)+0.1))
plot(Exp.soft[4,], module.soft[13,], col=colorlevel1, pch="-", ylab="Clustering
Coefficients", xlab="Eigengene-based Clustering Coefficients", cex=2, ylim=
range(CC.soft),cex.main=3,cex.lab=2.3, cex.axis=1.6)
abline(0,1, col=2)
points(CC.soft.new, CC.soft, col=rep(colorlevel1, module.soft[1,]), pch="o", cex=0.6)
#title(paste("beta=", power1, ", R^2=",signif(cor(CC.soft.new, CC.soft)^2,2),
sep=""),cex.main=3)
rm.grey=rep(T, length(CC.soft))
rm.grey[(sum(module.soft[1,1:(grey-1)]+1): sum(module.soft[1,1:grey]) ] =F
abline(lm( (CC.soft)[-greys]~ (CC.soft.new) [-greys]), col=4)
title(substitute( {beta == exp1} * ", " * {R^2 == exp2}, list(exp1 = power1, exp2 =
signif(cor(CC.soft.new[rm.grey], CC.soft[rm.grey])^2,2) )),cex.main=3)
dev.off()
### <plot>: GBM-ClusterCoef-p1
### Figure 9e

```

```

#===== kME =====

```

```

# Fuzzy

```

```

kME =matrix(NA, nrow=length(colorh1), ncol=length(colorlevel1) )
colnames(kME)=colorlevel1

```

```

for(i in 1:length(colorlevel1)){
kME[,i] = abs(cor(t(datExpr2), m1[[1]][,i], use="pairwise.complete.obs"))
}

```

```

#windows(width=8, height=5); par(mar=c(5,5,4,2)+0.1)
fuzzy.list="turquoise"
postscript(paste(datalab, "-Fuzzy.", fuzzy.list[1], ".vs.GS-p", power1, ".ps", sep=""),
width=8.5, height=6, horizontal=F, paper="special")
par(mar=c(6,6,5,2)+0.1))
plot(kME[,match(fuzzy.list, colorlevel1)], GS, col=colorh1, cex=0.5, pch=19,
xlab=paste("Membership in", fuzzy.list,"module"), ylab="Gene Significance",
cex.lab=2, cex.axis=1.6)
title(main="Gene Significance", cex.main=2.5)
dev.off()
### <plot>: GBM-Fuzzy.brown.vs.GS-p1

```

```

if(power1==1){
#indows(width=8, height=5); par(mar=c(5,5,4,2)+0.1)
fuzzy.list=c("blue", "brown")
postscript(paste(datalab, "-Fuzzy.", fuzzy.list[1], ".vs.", fuzzy.list[2], ".ps", sep=""),
width=8.5, height=6, horizontal=F, paper="special")
par(mar=c(6,6,5,2)+0.1))
plot(kME[,match(fuzzy.list[1], colorlevel1)], kME[,match(fuzzy.list[2], colorlevel1)],
col=colorh1, cex=0.5, pch=19, xlab=paste("Membership in ", fuzzy.list[1], " module",
sep=""), ylab= paste("Membership in ", fuzzy.list[2], " module", sep=""),
main="Fuzzy Module Assignment", cex.main=2.5, cex.lab=2, cex.axis=1.6)
dev.off()
### <plot>: GBM-Fuzzy.blue.vs.brown

#indows(width=8, height=5); par(mar=c(5,5,4,2)+0.1)
fuzzy.list=c("blue", "turquoise")
postscript(paste(datalab, "-Fuzzy.", fuzzy.list[1], ".vs.", fuzzy.list[2], ".ps", sep=""),
width=8.5, height=6, horizontal=F, paper="special")
par(mar=c(6,6,5,2)+0.1))
plot(kME[,match(fuzzy.list[1], colorlevel1)], kME[,match(fuzzy.list[2], colorlevel1)],
col=colorh1, cex=0.5, pch=19, xlab=paste("Membership in ", fuzzy.list[1], " module",
sep=""), ylab= paste("Membership in ", fuzzy.list[2], " module", sep=""),
main="Fuzzy Module Assignment", cex.main=2.5, cex.lab=2, cex.axis=1.6)
dev.off()
### <plot>: GBM-Fuzzy.blue.vs.brown

#indows(width=8, height=5); par(mar=c(5,5,4,2)+0.1)
fuzzy.list=c("brown", "turquoise")
postscript(paste(datalab, "-Fuzzy.", fuzzy.list[1], ".vs.", fuzzy.list[2], ".ps", sep=""),
width=8.5, height=6, horizontal=F, paper="special")
par(mar=c(6,6,5,2)+0.1))
plot(kME[,match(fuzzy.list[1], colorlevel1)], kME[,match(fuzzy.list[2], colorlevel1)],
col=colorh1, cex=0.5, pch=19, xlab=paste("Membership in ", fuzzy.list[1], " module",
sep=""), ylab= paste("Membership in ", fuzzy.list[2], " module", sep=""),
main="Fuzzy Module Assignment", cex.main=2.5, cex.lab=2, cex.axis=1.6)
dev.off()
### <plot>: GBM-Fuzzy.blue.vs.brown
}### Only when power1=1

save.image(paste(datalab, "-p", power1, ".RData", sep=""))

```

**#The Saved R Session Ends Here!**

```
##### Hard Thresholding #####  
### To draw plots for unweighted network, load R session "p1.RData"
```

```
tau1=0.65 ### 0.7 for GBM and 0.65 for Yeast and Mouseweight
```

```
# =====  
# Unweighted network: Hard Thresholding Method  
# =====
```

```
time.start=Sys.time()  
for(i in 1:length(colorlevel1) ){  
  adjtemp=cor(t(datExpr2[colorh1==colorlevel1[i], ] ), use="pairwise.complete.obs" )  
  adjtemp= abs(adjtemp) >= tau1  
  assign(paste("FDhard", colorlevel1[i],sep=""), FDADJ(adjtemp))  
  rm(adjtemp);collect_garbage()  
}  
time.end=Sys.time()  
time.end-time.start # total time for running this part
```

```
# Construct the table 1.  
module.hard=matrix(NA, ncol=length(colorlevel1), nrow=  
length(FDhardblue$summary.names))  
colnames(module.hard)=colorlevel1  
rownames(module.hard)= FDhardblue$summary.names  
for(i in 1:length(colorlevel1)){  
  temp=eval(as.name(paste("FDhard",colorlevel1[i],sep=""))) )  
  module.hard[,i]=temp$summary  
}  
signif(module.hard[-1,],3)
```

```
HubGeneSignificance.hard=rep(NA, length(ModuleSignificance) )  
for(i in 1:length(colorlevel1)){  
  temp=eval(as.name(paste("FDhard",colorlevel1[i],sep=""))) )  
  GS.temp=GS[colorh1==colorlevel1[i]]  
  K=temp$kWithin/max(temp$kWithin)  
  HubGeneSignificance.hard[i]=sum(GS.temp*K)/sum(K^2)  
}  
HubGeneSignificance.hard; # HubGeneSignificanceExp;
```

```
if(0){  
  #windows(width=7, height=5)  
  postscript(paste(datalab, "-HubGeneSignificance-tau", tau1, ".ps", sep=""), width=8.5,  
  height=6, horizontal=F, paper="special")  
  par(mar=(c(6,5,5,2)+0.1))  
  plot(HubGeneSignificanceExp, HubGeneSignificance.hard, col=colorlevel1, pch=19,  
  cex=2, xlab="Eigengene-based Hub Gene Significance", ylab="Hub Gene  
  Significance", cex.main=3, cex.lab=2.3, cex.axis=1.6)
```

```

abline(0,1, col=2)
title(substitute( {tau == exp1} * ", " * {R^2 == exp2}, list(exp1 = tau1, exp2 =
signif(cor(HubGeneSignificanceExp[-grey], HubGeneSignificance.hard[-grey])
^2,2))),cex.main=3)
dev.off()
### <plot>: GBM-HubGeneSignificance-p1
}

#windows(width=7, height=5)
postscript(paste(datalab, "-MS-HGS-Density-tau", tau1, ".ps", sep=""), width=8.5,
height=6, horizontal=F, paper="special")
par(mar=(c(6,5,5,2)+0.1))
tmp1= HubGeneSignificance.hard*sqrt(module.hard[3,])
tmp3= ModuleSignificance
plot(tmp1, tmp3, col= colorlevel1, pch=19, ylab="Module Significance", xlab= "",
cex=2, cex.main=3, cex.lab=2.3, cex.axis=1.6)
#"HubGeneSignificance*sqrt(Density)"
abline(0,1, col=2)
abline(lm(tmp3 [-grey]~ tmp1 [-grey]), col=4)
title(substitute( {tau == exp1} * ", " * {R^2 == exp2}, list(exp1 = tau1, exp2 =
signif(cor(tmp1[-grey], tmp3[-grey])^2,2))),cex.main=3)
title(xlab= expression(paste("HubGeneSignificance * ", sqrt(Density) ) ), line=4,
cex.lab=2.3)
dev.off()
### <plot>: GBM-MS-HGS-Density-p1

if(0){
#windows(width=7, height=5)
postscript(paste(datalab, "-HubGene-Eigengene-Significance-tau", tau1, ".ps",
sep=""), width=8.5, height=6, horizontal=F, paper="special")
par(mar=(c(6,5,5,2)+0.1))
tmp1= HubGeneSignificance.hard
tmp3= as.vector(EigengeneSignificance)
plot(tmp1, tmp3, col= colorlevel1, pch=19, ylab="Eigengene Significance",
xlab="Hub Gene Significance", cex=2, cex.main=3, cex.lab=2.3, cex.axis=1.6)
abline(0,1, col=2)
title(substitute( {tau == exp1} * ", " * {R^2 == exp2}, list(exp1 = tau1, exp2 =
signif(cor(tmp1[-grey], tmp3[-grey])^2,2))),cex.main=3)
#title(paste("beta=", tau1, ", R^2=",signif(cor(tmp1[-grey], tmp3[-grey])^2,2),
sep=""),cex.main=3)
dev.off()
### <plot>: GBM-HubGene-Eigengene-Significance-p1
}

for(i in 1:length(colorlevel1) ){
#i=1 # 2 for brown and 1 for blue
temp=eval(as.name(paste("FDhard",colorlevel1[i],sep="")))
tmp3=GS[colorh1== colorlevel1[i]];
tmp1=temp$kWithin/ max(temp$kWithin) #K
# tmp=lm(tmp3~tmp1)

```

```

tmp=lm(tmp3~tmp1-1)
#print(cor.test(tmp1, tmp3), method="s")
#windows(width=8, height=5)
postscript(paste(datalab, "-GS-K-", colorlevel1[i], "-0Intercept-tau", tau1, ".ps",
sep=""), width=8.5, height=6, horizontal=F, paper="special")
par(mar=(c(6,5,5,2)+0.1))
plot(tmp1, tmp3, col=colorlevel1[i], pch="o", xlab="K", ylab="Gene Significance",
cex.lab=2.3, cex.axis=1.6, xlim=c(0,1), ylim=c(0,max(tmp3)))
title(substitute( {exp0} * " , " * {tau == exp1} * " , slope = " * {exp2},
list(exp0=colorlevel1[i], exp1 = tau1, exp2 = signif(tmp$coef,2) )),cex.main=3)
# points(0,0, pch=19, col= colorlevel1[i], cex=1.5)
# title(paste(colorlevel1[i]," , beta=",power1," , slope=",signif(tmp$coef,2), sep=""),
cex.main=3)
# abline(tmp$coef, col=2)
## for no intercept (Hub Gene Significance)
abline(c(0,tmp$coef), col=2)
dev.off()
}
### <plot>: GBM-GS-K-brown-0Intercept-p1

#windows(width=9, height=4)
postscript(paste(datalab, "-K-EigenCF-tau", tau1, ".ps", sep=""), width=(ncol*4),
height=(nrow*3), horizontal=F, paper="special")
par(mar=(c(6,5,5,2)+0.1))
par(mfrow=c(nrow,ncol))
for(i in 1:length(colorlevel1) ){
temp=eval(as.name(paste("FDhard",colorlevel1[i],sep="")))
temp1=eval(as.name(paste("Expsoft",colorlevel1[i],sep="")))
cor=cor(temp1$CFExp, temp$kWithin)
plot(temp1$CFExp,temp$kWithin/max(temp$kWithin),xlab="Eigengene
Conformity", ylab="K",col=colorlevel1[i], cex.lab=2, cex.axis=1.6, cex.main=2.5)
abline(0,1,col=2)
abline(lm((temp$kWithin/max(temp$kWithin)) [-grey]~ temp1$CFExp [-grey]),
col=4)
title(substitute( {exp0} * " , " * {tau == exp1} * " , " * {R^2 == exp2},
list(exp0=colorlevel1[i], exp1 = tau1, exp2 = signif(cor^2,2) )),cex.main=2.6)
#title(paste(colorlevel1[i], " , beta=", tau1," , R^2=",signif(cor^2,2), sep=""))
}
dev.off()
### <plot>: GBM-K-EigenCF-p1

#windows(width=7, height=5)
postscript(paste(datalab, "-Centralization-sqrtDensity-tau", tau1, ".ps", sep=""),
width=8.5, height=6, horizontal=F, paper="special")
par(mar=(c(6,6,5,2)+0.1))
plot(module.hard[6,], sqrt(module.hard[3,])*(1- sqrt(module.hard[3,])),
col=colorlevel1, xlab="Centralization", ylab= expression(paste(sqrt(Density), " , (1-
sqrt(Density)))) ,cex.lab=2.3, cex.axis=1.6, pch=19, cex=2)
#"sqrt(Density)*(1-sqrt(Density))"
abline(0,1, col=2)

```

```

abline(lm( sqrt(module.hard[3,])*(1- sqrt(module.hard[3,]))[-grey]~
(module.hard[6,]) [-grey]), col=4)
title(substitute({tau == exp1} * " , " * {R^2 == exp2}, list(exp0=colorlevel1[i], exp1 =
tau1, exp2 = signif(cor(module.hard[6,-grey], sqrt(module.hard[3,-grey]) *(1-
sqrt(module.hard[3,-grey]))^2,2) )),cex.main=3)
#title(paste("beta=",power1, " , R^2=",signif(cor(module.hard[6,],
sqrt(module.hard[3,]) *(1- sqrt(module.hard[3,]))^2,2), sep=""), cex.main=3)
dev.off()
### <plot>: GBM-Centralization-sqrtDensity-p1

maxconnectivity=rep(NA, length(colorlevel1))
for(i in 1:length(colorlevel1) ){
temp=eval(as.name(paste("FDhard",colorlevel1[i],sep="")))
maxconnectivity[i]=max(temp$kWithin)
}
maxconnectivity
#windows(width=7, height=5)
postscript(paste(datalab, "-maxk-sqrtDensity-tau", tau1, ".ps", sep=""), width=8.5,
height=6, horizontal=F, paper="special")
par(mar=c(6,6,5,2)+0.1))
plot(maxconnectivity/(module.hard[1,]-1), sqrt(module.hard[3,]), col=colorlevel1,
xlab="max(k)/(n-1)", ylab= expression(sqrt(Density) ),cex.lab=2.3, cex.axis=1.6,
pch=19, cex=2)
abline(0,1, col=2)
abline(lm( sqrt(module.hard[3,])[-grey]~ (maxconnectivity/(module.hard[1,]-1)) [-
grey]), col=4)
title(substitute({tau == exp1} * " , " * {R^2 == exp2}, list(exp0=colorlevel1[i], exp1 =
tau1, exp2 = signif(cor(maxconnectivity[-grey]/(module.hard[1,-grey]-1),
sqrt(module.hard[3,-grey]))^2,2))), cex.main=3)
# title(paste("beta=",power1, " , R^2=", signif(cor(maxconnectivity/(module.hard[1,]-
1), sqrt(module.hard[3,]))^2,2), sep=""), cex.main=3)
dev.off()
### <plot>: GBM-maxk-sqrtDensity-p1

```

### Figure 2 ####

### Weighted ###

```

CC.soft=NULL
for(i in 1:length(colorlevel1) ){
temp=eval(as.name(paste("FDhard",colorlevel1[i],sep="")))
CC.soft=c(CC.soft, temp$ClusterCoef)
}
CC.soft.new=(module.hard[9,]^2+1)^2*module.hard[3,]

#windows(width=7, height=5)
# par(mar=c(6,5,5,2)+0.1))
postscript(paste(datalab, "-CCHeteroDensity-tau", tau1, ".ps", sep=""), width=8.5,
height=6, horizontal=F, paper="special")
par(mar=c(6,6,5,2)+0.1))

```

```

plot(CC.soft.new, module.hard[13,], col=colorlevel1, pch="-", ylab="Clustering
Coefficient", xlab="", cex=2, cex.main=3, cex.lab=2.3, cex.axis=1.6,
ylim=range(CC.soft))
#(1+Heterogeneity^2)^2*Density
abline(0,1, col=2)
points(rep(CC.soft.new, module.hard[1,]), CC.soft, col=rep(colorlevel1,
module.hard[1,]), pch="o", cex=0.6)
rm.grey=rep(T, length(CC.soft))
rm.grey[(sum(module.hard[1,1:(grey-1)]+1): sum(module.hard[1,1:grey]) ] =F
abline(lm( (CC.soft)[-greys]~ (rep(CC.soft.new, module.hard[1,]) [-greys]), col=4)
title(substitute( {tau == exp1} * ", " * {R^2 == exp2}, list(exp1 = tau1, exp2 =
signif(cor(rep(CC.soft.new[-grey], module.hard[1,-grey]),
CC.soft[rm.grey])^2,2) )),cex.main=3)
title(xlab= expression(paste( "(", 1+ Heterogeneity ^2, ")", ""^2, " * Density",
sep="" ) ), line=4, cex.lab=2.3)
#title(paste("beta=", tau1, ", R^2=",signif(cor(rep(CC.soft.new, module.hard[1,]),
CC.soft)^2,2), sep=""),cex.main=3)
dev.off()
### <plot>: GBM-CCHeteroDensity-p1

```

### ### Weighted ###

```

datExpr.hub=matrix(NA, ncol=length(colorlevel1), nrow=dim(datExpr2)[2])
for(i in 1:length(colorlevel1) ){
temp=eval(as.name(paste("FDhard",colorlevel1[i],sep="")))
temp=temp$kWithin
select= match(min(rank(-temp)), rank(-temp))
datExpr.hub[,i]= as.numeric( datExpr2[colorh1==colorlevel1[i],[select,] )
}
for(i in 1:length(colorlevel1) ){
#windows(width=8, height=5); par(mar=c(5,5,4,2)+0.1)
postscript(paste(datalab, "-ME-Hub-", colorlevel1[i], "-tau", tau1, ".ps", sep=""),
width=8.5, height=6, horizontal=F, paper="special")
par(mar=c(6,6,5,2)+0.1))
plot(datExpr.hub[,i], m1[[1]][,i], col=colorlevel1[i], xlab="Hub Gene Expression",
ylab="Module Eigengene Expression", pch=19, cex.lab=2.3, main="", cex=1,
cex.axis=1.6)
title(substitute( {exp0} * ", " * {tau == exp1} * ", " * {R^2 == exp2}, list(exp0=
colorlevel1[i], exp1 = tau1, exp2 = signif(cor(datExpr.hub[,i], m1[[1]][,i],
use="pairwise.complete.obs")^2,2) )),cex.main=3)
abline(lm(m1[[1]][,i] ~ datExpr.hub[,i]),col=2)
dev.off()
}
### <plot>: GBM-ME-Hub-brown-p1

```

### ### Figure 7 #####

### ### Weighted ###

```

#windows(width=7, height=5)
#par(mar=c(5,5,4,1)+0.1))

```

```

postscript(paste(datalab, "-Density-tau", tau1, ".ps", sep=""), width=8.5, height=6,
horizontal=F, paper="special")
par(mar=(c(6,6,5,2)+0.1))
plot(Exp.soft[1,], module.hard[3,], col=colorlevel1, xlab="Eigengene-based Density",
ylab="Density",cex.lab=2.3, cex.axis=1.6, pch=19, cex=2)
abline(0,1, col=2)
abline(lm( (module.hard[3,])[-grey]~ (Exp.soft[1,]) [-grey]), col=4)
title(substitute( {tau == exp1} * " , " * {R^2 == exp2}, list(exp1 = tau1, exp2 =
signif(cor(Exp.soft[1,-grey],module.hard[3,-grey])^2,2))),cex.main=3)
dev.off()
### <plot>: GBM-Density-p1

```

```

#windows(width=7, height=5)
#par(mar=(c(5,5,4,1)+0.1))
postscript(paste(datalab, "-Centralization-tau", tau1, ".ps", sep=""), width=8.5,
height=6, horizontal=F, paper="special")
par(mar=(c(6,6,5,2)+0.1))
plot(Exp.soft[2,], module.hard[6,], col=colorlevel1, xlab="Eigengene-based
Centralization", ylab="Centralization",cex.lab=2.3, cex.axis=1.6, pch=19, cex=2)
abline(0,1, col=2)
abline(lm( (module.hard[6,])[-grey]~ (Exp.soft[2,]) [-grey]), col=4)
#title(main=paste("beta=",power1, ",
R^2=",signif(cor(Exp.soft[2,],module.hard[6,])^2,2), sep=""), cex.main=3)
title(substitute( {tau == exp1} * " , " * {R^2 == exp2}, list(exp1 = tau1, exp2 =
signif(cor(Exp.soft[2,-grey],module.hard[6,-grey])^2,2))),cex.main=3)
dev.off()
### <plot>: GBM-Centralization-p1
### Figure 9a

```

```

#windows(width=7, height=5)
#par(mar=(c(5,5,4,1)+0.1))
postscript(paste(datalab, "-Heterogeneity-tau", tau1, ".ps", sep=""), width=8.5,
height=6, horizontal=F, paper="special")
par(mar=(c(6,6,5,2)+0.1))
plot(Exp.soft[3,], module.hard[9,], col=colorlevel1, xlab="Eigengene-based
Heterogeneity", ylab="Heterogeneity", cex.lab=2.3, cex.axis=1.6, pch=19, cex=2)
# abline(test2$coef)
abline(0,1, col=2)
abline(lm( (module.hard[9,])[-grey]~ (Exp.soft[3,]) [-grey]), col=4)
#title( main=paste("beta=",power1, sep=""),cex.main=3)
title(substitute( {tau == exp1} * " , " * {R^2 == exp2}, list(exp1 = tau1, exp2 =
signif(cor(Exp.soft[3,-grey],module.hard[9,-grey])^2,2))),cex.main=3)
dev.off()
### <plot>: GBM-Heterogeneity-p1
### Figure 9c

```

```

CC.soft=NULL
for(i in 1:length(colorlevel1) ){
temp=eval(as.name(paste("FDhard",colorlevel1[i],sep="")))

```

```

CC.soft=c(CC.soft, temp$ClusterCoef)
}
CC.soft.new=rep(Exp.soft[4,],module.hard[1,])

#windows(width=7, height=5)
#par(mar=(c(5,5,4,1)+0.1))
postscript(paste(datalab, "-ClusterCoef-tau", tau1, ".ps", sep=""), width=8.5, height=6,
horizontal=F, paper="special")
par(mar=(c(6,6,5,2)+0.1))
plot(Exp.soft[4,], module.hard[13,], col=colorlevel1, pch="-", ylab="Clustering
Coefficients", xlab="Eigengene-based Clustering Coefficients", cex=2, ylim=
range(CC.soft),cex.main=3,cex.lab=2.3, cex.axis=1.6)
abline(0,1, col=2)
points(CC.soft.new, CC.soft, col=rep(colorlevel1, module.hard[1,]), pch="o", cex=0.6)
#title(paste("beta=", tau1, ", R^2=", signif(cor(CC.soft.new, CC.soft)^2,2),
sep=""),cex.main=3)
rm.grey=rep(T, length(CC.soft))
rm.grey[(sum(module.hard[1,1:(grey-1)]+1): sum(module.hard[1,1:grey]) ] =F
abline(lm( (CC.soft)[-greys]~ (CC.soft.new) [-greys]), col=4)
title(substitute( {tau == exp1} * ", " * {R^2 == exp2}, list(exp1 = tau1, exp2 =
signif(cor(CC.soft.new[rm.grey], CC.soft[rm.grey])^2,2) )),cex.main=3)
dev.off()
### <plot>: GBM-ClusterCoef-p1
### Figure 9e

```

## ### Generate the dendrograms for all beta and tau values

### Run in Unix @ /home/jdong/Steve/Dendrograms

```
source("NetworkFunctions.txt"); source("NetworkFunctions1.txt");
datalab="Yeast"

# =====
# Prepare the data.
# =====

dat0 <- read.csv("YEASTCellCycle4000.csv",header=T, row.names=1)

# the following gene summary file contains information on the yeast genes
datSummary=dat0[,1:7]
# the column essentiality indicates which gene is essential for yeast survival
table(datSummary$essentiality)
# message: there are 645 essential genes

# the following data frame contains the gene expression data: columns are genes, rows
are arrays (samples)
datExpr <- dat0[,8:51]

# Now we define the adjacency matrix for a weighted network, i.e. soft thresholding.
# Calculate the Pearson Correlation Matrix
corhelp=cor(t(datExpr),use="pairwise.complete.obs")
# use the power adjacency function
AdjMat1 = abs(corhelp)^ 7
diag(AdjMat1)=0
# This is the definition of the connectivity, a.k.a degree
Degree <- apply(AdjMat1,2,sum)

# =====
# Construct modules.
# =====

# This code allows one to restrict the analysis to the most connected genes, which
may speed up calculations when it comes to module detection.
DegCut = 2001 # number of most connected genes that will be considered
DegreeRank <- rank(-Degree)
restDegree <- DegreeRank <= DegCut
# thus our module detection uses the following number of genes
sum(restDegree)

# Now we restrict the adjacency matrix to the most connected genes

corr.soft = corhelp[restDegree,restDegree]
```

```

AdjMat1rest <- AdjMat1[restDegree,restDegree]

# The following code computes the topological overlap matrix based on the adjacency
matrix.

dissGTOM1=TOMdist1(AdjMat1rest)
collect_garbage()

# Now we carry out hierarchical clustering with the TOM matrix. Branches of the
resulting clustering tree will be used to define gene modules.

hierGTOM1 <- hclust(as.dist(dissGTOM1),method="average");
colorh1=as.character(modulecolor2(hierGTOM1,h1=.97))
table(colorh1)
colorlevel1=levels(factor(colorh1))

rm(dat0, datExpr, corhelp, AdjMat1, Degree, AdjMat1rest, dissGTOM1, hierGTOM1)

save.image("Yeast.Rdata")

```

## load("Yeast.Rdata")

```

power1=6 # 1:7
AdjMat.soft=abs(corr.soft)^power1
collect_garbage()
dissGTOM1=TOMdist1(as.matrix(AdjMat.soft))
collect_garbage()
hierGTOM1 <- hclust(as.dist(dissGTOM1),method="average");
rm(AdjMat.soft, dissGTOM1); collect_garbage()
collect_garbage()
postscript(paste(datalab, "-hierGTOM1-p", power1, ".ps", sep=""), width=8.5,
height=6, horizontal=F, paper="special")
par(mfrow=c(2,1), mar=c(1,2,3,0))
plot(hierGTOM1, main="", labels=F, xlab="", sub="");
title(substitute( "Network Dendrogram, " * {beta == exp1 }, list(exp1 =
power1)),cex.main=2)
hclustplot1(hierGTOM1,colorh1, title="")
title(substitute("Colored by module membership"), cex.main=1.5)
dev.off()

```

```

tau1 = 0.65 #0.65 & 0.5
AdjMat.soft=abs(corr.soft) >= tau1
collect_garbage()
dissGTOM1=TOMdist1(as.matrix(AdjMat.soft))
collect_garbage()
hierGTOM1 <- hclust(as.dist(dissGTOM1),method="average");

```

```

rm(AdjMat.soft, dissGTOM1); collect_garbage()
collect_garbage()
postscript(paste(datalab, "-hierGTOM1-tau", tau1, ".ps", sep=""), width=8.5, height=6,
horizontal=F, paper="special")
par(mfrow=c(2,1), mar=c(1,2,3,0))
plot(hierGTOM1, main="", labels=F, xlab="", sub="");
title(substitute( "Network Dendrogram, " * {tau == exp1}, list(exp1 =
tau1)),cex.main=2)
hclustplot1(hierGTOM1,colorh1, title="")
title(substitute("Colored by module membership"), cex.main=1.5)
dev.off()

```

## ### Generate table 2 – Soft Thresholding

### Load R session locally

power1

```

maxaei=rep(NA, length(colorlevel1))
for(i in 1:length(colorlevel1)){
temp=eval(as.name(paste("Expsoft",colorlevel1[i],sep="")))
maxaei[i]=max(temp$CFExp)
}

if(1){
cat("Module", colnames(module.soft), sep=" & "); cat("\\\\[3pt] \\hline
&&&&&&&&& \\\\[-5pt]n");
cat("Size ($n^{(q)})$", module.soft[1,], sep=" & "); cat("\\\\[3pt] \\hline
&&&&&&&&& \\\\[-5pt]n");
cat("Eigengene Factorizability ($EF(X^{(q)})$)", signif(Factorizability.X.approx, 3) ,
sep=" & "); cat("\\\\[3pt] &&&&&&&&& \\\\[-5pt]n");
cat("$VarExplained(\\textbf{\\emph{E}}^{(q)}) $",
signif(varexplained1[1,1:length(colorlevel1)],3), sep=" & "); cat("\\\\[3pt]
&&&&&&&&& \\\\[-5pt]n");
cat("$max(a_{e,i})$", signif(maxaei,3), sep=" & "); cat("\\\\\\hline &&&&&&&&&
\\\\[-5pt]n");
cat("$Density$", signif(module.soft[3,],3), sep=" & "); cat("\\\\[3pt] &&&&&&&&&
\\\\[-5pt]n");
cat("$Density_{E}$", signif(Exp.soft[1,],3), sep=" & "); cat("\\\\[3pt] \\hline
&&&&&&&&& \\\\[-5pt]n");
cat("$Centralization$", signif(module.soft[6,],3), sep=" & "); cat("\\\\[3pt]
&&&&&&&&& \\\\[-5pt]n");
cat("$Centralization_{E}$", signif(Exp.soft[2,],3), sep=" & "); cat("\\\\[3pt] \\hline
&&&&&&&&& \\\\[-5pt]n");
cat("$Heterogeneity$", signif(module.soft[9,],3), sep=" & "); cat("\\\\[3pt]
&&&&&&&&& \\\\[-5pt]n");
cat("$Heterogeneity_{E}$", signif(Exp.soft[3,],3), sep=" & "); cat("\\\\[3pt] \\hline
&&&&&&&&& \\\\[-5pt]n");
cat("$Mean(ClusterCoef)$", signif(module.soft[12,],3), sep=" & "); cat("\\\\[3pt]
&&&&&&&&& \\\\[-5pt]n");

```

```

cat("$ClusterCoef_{E}$", signif(Exp.soft[4,],3), sep=" & "); cat("\\\\[3pt] \\hline
&&&&&&&&& \\\\[-5pt]n");
cat("$ModuleSignif$", signif(ModuleSignificance,3), sep=" & "); cat("\\\\[3pt]
&&&&&&&&& \\\\[-5pt]n");
cat("$HubGeneSignif$", signif(HubGeneSignificance,3), sep=" & "); cat("\\\\[3pt]
\\hline \\hline\n");
}

```

## ### Generate table 2 – Hard Thresholding

### Load R session locally

tau1

```

maxaei=rep(NA, length(colorlevel1))
for(i in 1:length(colorlevel1)){
temp=eval(as.name(paste("Expsoft",colorlevel1[i],sep="")))
maxaei[i]=max(temp$CFExp)
}
if(1){
cat("Module", colnames(module.hard), sep=" & "); cat("\\\\[3pt] \\hline
&&&&&&&&& \\\\[-5pt]n");
cat("Size (N^{(q)})", module.hard[1,], sep=" & "); cat("\\\\[3pt] \\hline
&&&&&&&&& \\\\[-5pt]n");
cat("Eigengene Factorizability (EF(X^{(q)}))", signif(Factorizability.X.approx, 3) ,
sep=" & "); cat("\\\\[3pt] &&&&&&&&& \\\\[-5pt]n");
cat("$VarExplained(\\textbf{\\emph{E}}^{(q)})$",
signif(varexplained1[1,1:length(colorlevel1)],3), sep=" & "); cat("\\\\[3pt]
&&&&&&&&& \\\\[-5pt]n");
cat("$max(a_{e,i})$", signif(maxaei,3), sep=" & "); cat("\\\\\\hline &&&&&&&&&
\\\\[-5pt]n");
cat("$Density$", signif(module.hard[3,],3), sep=" & "); cat("\\\\[3pt]
&&&&&&&&& \\\\[-5pt]n");
cat("$Density_{E}$", signif(Exp.soft[1,],3), sep=" & "); cat("\\\\[3pt] \\hline
&&&&&&&&& \\\\[-5pt]n");
cat("$Centralization$", signif(module.hard[6,],3), sep=" & "); cat("\\\\[3pt]
&&&&&&&&& \\\\[-5pt]n");
cat("$Centralization_{E}$", signif(Exp.soft[2,],3), sep=" & "); cat("\\\\[3pt] \\hline
&&&&&&&&& \\\\[-5pt]n");
cat("$Heterogeneity$", signif(module.hard[9,],3), sep=" & "); cat("\\\\[3pt]
&&&&&&&&& \\\\[-5pt]n");
cat("$Heterogeneity_{E}$", signif(Exp.soft[3,],3), sep=" & "); cat("\\\\[3pt] \\hline
&&&&&&&&& \\\\[-5pt]n");
cat("$Mean(ClusterCoef)$", signif(module.hard[12,],3), sep=" & "); cat("\\\\[3pt]
&&&&&&&&& \\\\[-5pt]n");
cat("$ClusterCoef_{E}$", signif(Exp.soft[4,],3), sep=" & "); cat("\\\\[3pt] \\hline
&&&&&&&&& \\\\[-5pt]n");

```

```
cat("$ModuleSignif$", signif(ModuleSignificance,3), sep=" & "); cat("\\\\[3pt]
&&&&&&&& \\[-5pt]\n");
cat("$HubGeneSignif$", signif(HubGeneSignificance.hard,3), sep=" & ");
cat("\\\\[3pt] \\hline \\hline\n");
}
```