The following program codes focus on overlapped grouping on acute leukemia gene expression and they can be conducted on R. The red font is a description of the codes behind it.

# Download packages that we need

library(MASS)

library(class)

library(cluster)

library(impute)

library(WGCNA)

library(PCIT)

options(stringsAsFactors=F)

# Import input matrix for each class of acute leukemia gene expression

T1golub=read.csv("golub1.csv",header=T)

# Now we investigate soft thesholding with the power adjacency function

powers1=c(seq(1,10,by=1),seq(12,20,by=2)) RpowerTable=pickSoftThreshold(T1golub, powerVector=powers1)[[2]]

gc()

cex1=0.7

par(mfrow=c(1,2))

plot(RpowerTable[,1], -sign(RpowerTable[,3])\*RpowerTable[,2],xlab="

Soft Threshold (power)",ylab="Scale Free Topology Model Fit,signed R^2",type="n")

text(RpowerTable[,1], -sign(RpowerTable[,3])\*RpowerTable[,2],

labels=powers1,cex=cex1,col="red")

# this line corresponds to using an R^2 cut-off of h

abline(h=0.8,col="red")

plot(RpowerTable[,1], RpowerTable[,5],xlab="Soft Threshold (power)",ylab="Mean

Connectivity", type="n")

text(RpowerTable[,1], RpowerTable[,5], labels=powers1, cex=cex1,col="red")

# Note that at power=6, the curve has an elbow or kink, i.e. for this power the scale free topology , fit does not improve after increasing the power. This is why we choose beta1=6

beta1=6

ConnectivityT1golub=softConnectivity(T1golub,power=beta1)-1

par(mfrow=c(1,1))

scaleFreePlot(ConnectivityT1golub, main=paste("soft threshold, power=",beta1), truncated=T);

ConnectivityCut = 3571 # number of most connected genes that will be considered

ConnectivityRankT1golub = rank(-ConnectivityT1golub)

restConnectivityT1golub = ConnectivityRankT1golub<= ConnectivityCut

sum(restConnectivityT1golub)

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

ADJrest = adjacency(T1golub[,restConnectivityT1golub], power=beta1)

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

dissTOM=TOMdist(ADJrest)

gc()

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

hierTOM = hclust(as.dist(dissTOM),method="average");

par(mfrow=c(1,1))

plot(hierTOM,labels=F,sub="",xlab="")

bwT1golub=blockwiseModules(T1golub[,restConnectivityT1golub],blocks=rep(1,sum(restConnectivityT1golub)),power=beta1,TOMType="unsigned",minModuleSize=30,verbose=3)

##### Count the number of selected modules and plot

table(bwT1golub$colors)

plotDendroAndColors(bwT1golub$dendrograms[[1]],bwT1golub$colors,"Module color",main="Cluster dendrogram and module colors in BALLNet", dendroLabels=FALSE)

TOMplot(dissTOM , hierTOM, bwT1golub$colors, terrainColors=TRUE)

# To get a sense of how related the modules are one can summarize each module by its first eigengene (referred to as principal components), and then correlate these module eigengenes with each other.

datT1=moduleEigengenes(T1golub[,restConnectivityT1golub],bwT1golub$colors)[[1]]

# We define a dissimilarity measure between the module eigengenes that keeps track of the sign of the correlation between the module eigengenes.

dissimT1golub=1-(t(cor(datT1, method="p")))/2

hclustdatT1=hclust(as.dist(dissimT1golub), method="average" )

par(mfrow=c(1,1))

plot(hclustdatT1, main="Clustering tree based on the module eigengenes of modules")