```
slist<-search()
cname<-"h:\\Cluster"

if(length(slist[slist == cname]) != 1){
   attach(cname,pos=1)
}
else{
   detach(cname)
   attach(cname,pos=1)
}

#Need to set the path for the data, and select one of the experiments from A to H
#I have assumed that you have selected experiment F

root<-"h:\\Cluster\\CWRU-F\\"

#
#Now read in the data
#
cluster.files<-dos(paste("dir ",root,"*.gpr /b /on",sep=""))
time.courses.F<-numeric()

for(i in 1:length(cluster.files)){

   print(i)

   fname1<-paste(root,cluster.files[i],sep="")
   f1.norm<-importData(fname1,type="ASCII",startRow=13,endRow=17)
   f1.dat<-importData(fname1,type="ASCII",startRow=27,colNameRow=27)

   n.factors<-unpaste(f1.norm[1,],sep="=")
   n.factor<-as.numeric(unlist(n.factors[2]))

   nrow(f1.dat)

   f1.colnames<-names(f1.dat)
   row.ids<-c(1:ncol(f1.dat))
   cy5.colf<-row.ids[f1.colnames == "F635.Median"]
   cy5.colb<-row.ids[f1.colnames == "B635.Median"]
   cy3.colf<-row.ids[f1.colnames == "F532.Median"]
   cy3.colb<-row.ids[f1.colnames == "B532.Median"]

   red.f<-f1.dat[,cy5.colf]
   red.b<-f1.dat[,cy5.colb]
   gre.f<-f1.dat[,cy3.colf]
   gre.b<-f1.dat[,cy3.colb]

   obs.id<-c(1:length(red.f))

   #red.y1<-red.f[red.f > red.b & gre.f > gre.b]
   #red.y0<-red.b[red.f > red.b & gre.f > gre.b]

   red.y1<-red.f
   red.y0<-red.b
   red.y<-red.y1-red.y0

   #gre.y1<-gre.f[red.f > red.b & gre.f > gre.b]
   #gre.y0<-gre.b[red.f > red.b & gre.f > gre.b]

   gre.y1<-gre.f
   gre.y0<-gre.b
   gre.y<-gre.y1-gre.y0

   de.y<-log(red.y)-log(gre.y)+log(n.factor)

   time.courses.F<-cbind(time.courses.F,de.y)

}

#Remove missing and "Inf" data

index<-c(1:nrow(time.courses.F))
time.maxes<-apply(time.courses.F,1,max)
```

```
length(time.maxes[time.maxes == "NA"])
index[time.maxes == "NA"]

gene.courses.F<-time.courses.F[time.maxes != "NA",]
gene.courses.F[gene.courses.F == Inf]<-0.0
gene.courses.F[gene.courses.F == -Inf]<-0.0


#The object gene.courses.F now contains the treated microarray data

print(dim(gene.courses.F))


#Histogram of the data
hist(gene.courses.F)

#####################################################################

#
#PCA
#

gene.experiment.F<-as.data.frame(gene.courses.F)

#Carry out the pca
ge.F.pca<-princomp(gene.experiment.F)

#Look at the "loadings" or weights

loadings(ge.F.pca)

plot.loadings(loadings(ge.F.pca))


#Look for patterns in the plot of the first against second principal components
biplot(ge.F.pca)

#Does this identify any gene subsets.

#Examine how much each of the seven principal components explains in terms of
#the overall variability
screeplot(ge.F.pca)



#####################################################################
#Cluster Analysis

#You may wish to cut down the number of genes that you are looking at
#do this by
#gene.courses.F.full<-gene.courses.F
#gene.courses.F<-gene.courses.F[1:1000,]
#to take the first 1000 genes.



gene.clusters.F<-hclust(dist(gene.courses.F))
plclust(gene.clusters.F)

###################

plclust(hclust(dist(gene.courses.F, metric="euclidean"), method="ave"))

plclust(hclust(dist(gene.courses.F, metric="maximum"), method="single"))



#Some post-processing for the clustering data
#ncuts defines the number of clusters

ncuts<-3
bic.cuts<-c(1:ncuts)
tvec<-c(1:7)
```

```
tmat<-matrix(rep(tvec,nrow(gene.courses.F)),nrow=nrow(gene.courses.F),byrow=T)
index<-c(1:nrow(gene.courses.F))
gene.cuts<-matrix(0,nrow=nrow(gene.courses.F),ncol=ncuts)

gene.y<-as.vector(t(gene.courses.F))
gene.id<-rep(c(1:nrow(gene.courses.F)),each=5)
gene.t<-rep(tvec,nrow(gene.courses.F))

for(i in 1:ncuts){
    gene.cut<-cutree(gene.clusters.F,k=i+1)
    gene.cuts[,i]<-gene.cut
    nin.clusters<-table(gene.cut)

    for(j in 1:(i+1)){
        plot(tvec,gene.courses.F[1,],ylim=range(gene.courses.F),type="n")
        id<-index[gene.cut == j]
        for(k in 1:nin.clusters[j]){
            lines(tvec,gene.courses.F[id[k],])
        }
    }
}

#########################################################################
#Trying to select the number of clusters using BIC
#This section of code computes the BIC value for the
#optimal clusters found under hierarchical clustering

ncuts<-500
aic.cuts<-c(1:ncuts)
bic.cuts<-c(1:ncuts)
nval<-nrow(gene.courses.F)

tvec<-c(6,20,40,96,192,336,480)
tmat<-matrix(rep(tvec,nval),nrow=nval,byrow=T)
index<-c(1:nval)
gene.cuts<-matrix(0,nrow=nval,ncol=ncuts)

gene.y<-as.vector(t(gene.courses.F))
gene.id<-rep(c(1:nval),each=length(tvec))
gene.t<-rep(tvec,nval)

plot(c(1:1000),ylim=range(-180000,-50000),type="n")

for(i in 1:ncuts){

    gene.cut<-cutree(gene.clusters.F,k=i)
    nin.clusters<-table(gene.cut)
    nc<-length(nin.clusters)


    gene.clusterdata<-cbind(as.numeric(gene.cut),gene.courses.F)

    cluster.means<-matrix(0,nrow=nc,ncol=length(tvec))
    cluster.vars<-matrix(0,nrow=nc,ncol=length(tvec))

    for(ic in 1:nc){
           if(nin.clusters[ic] == 1){
              cluster.means[ic,]<-gene.clusterdata[gene.clusterdata[,1] == ic,2:(length(tvec)+
    1)]
              }
           else{
              cluster.means[ic,]<-apply(gene.clusterdata[gene.clusterdata[,1] == ic,2:(length
    (tvec)+1)],2,mean)
           }
    }

    mean.mat<-gene.courses.F*0
    index<-c(1:nval)
    for(ic in 1:nc){
        id<-index[as.numeric(gene.cut) == ic]
        mean.mat[id,]<-matrix(rep(cluster.means[ic,],length(id)),nrow=length(id),byrow=T)
    }
```

```
    residual.ssq<-sum((gene.courses.F-mean.mat)^2)
    residual.var<-sum((gene.courses.F-mean.mat)^2)/(nval*length(tvec)-nc)

    bic.cuts[i]<-(nval*length(tvec))*log(residual.ssq/(nval*length(tvec)))+2*nc*length(tvec)
    *log(nval*length(tvec))

    print(c(i,aic.cuts[i],bic.cuts[i],round(residual.var,6)))

    points(i,bic.cuts[i])

}

plot(c(1:500),bic.cuts,type="l")
```