Exploratory analysis of RNA-seq dataset

gg

2016-01-15

To run this file: Rscript -e "rmarkdown::render('biplots.Rmd')"

Setup

```
library(compositions)
library(zCompositions)
library(ALDEx2)
source("~/git/proprBayes/R/propr-functions.R")

library(igraph)

############
# these are functins that we need for phi.
sma.df <- function(df){
df.cor <- stats::cor(df, use="pairwise.complete.obs")
df.var <- stats::cov(df, use="pairwise.complete.obs")
df.sd <- sqrt(diag(df.var))
r.rf2 <-
    (outer(diag(df.var), diag(df.var), "-")^2 ) /
    (outer(diag(df.var), diag(df.var), "+")^2 - 4 * df.var^2 )
 diag(r.rf2) <- 0
  res.dof     <- nrow(df) - 2
  F           <- r.rf2/(1 - r.rf2) * res.dof
list(b=sign(df.cor) * outer(df.sd, df.sd, "/"),
p=1 - pf(F, 1, res.dof),
r2=df.cor^2)
}

lt.row.min <- function(X){

  result <- numeric(nrow(X) - 1)

  for(i in 2:nrow(X)){

    result[i-1] <- min(X[i, 1:i-1])

  }

  result

}


propr.phisym <- function (X)
{
 Cov    <- stats::var(X)
 tmp    <- 2 * Cov / outer(diag(Cov), diag(Cov), "+")
```

```
 return((1-tmp)/(1+tmp))
}
############ end functions
```

First let us explore the data. The idea here is to identify those samples that are closely related

Filter the refseq to a mean count across samples of greater than 5

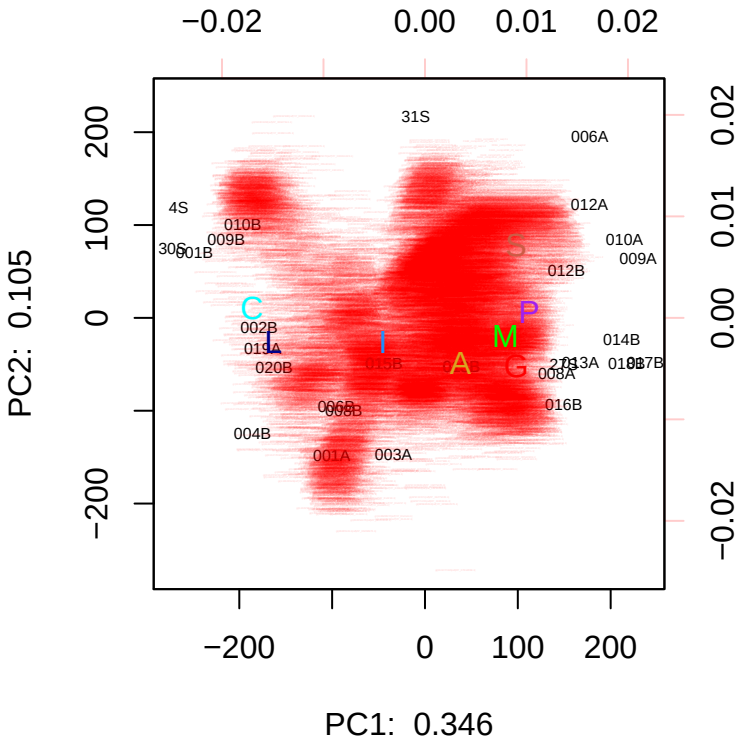
This is a form biplot that represents distances between samples with variances of components mapped onto the plot.

```
# this is the taxon table
o <- read.table(
   "~/git/twntyfr/data/merged_readcounts_taxonomy_qiime_L6_16Sorder_color_sum.txt", header=T, row.names=
)


# this is the refseq set
d <- read.table("data/merged_readcounts_subsys4.txt", header=T,
   row.names=1, check.names=F, sep="\t", comment.char="", quote="")

sub4 <- d$subsys4
d$subsys4 <- NULL

refseq <- d$refseqID
d$refseqID <- NULL

len <- d$length
d$length <- NULL

d.min <- d[which(apply(d,1,mean)> 5),]

# we need to replace 0 values with a best estimate
# use zCompositions CZM by default
# but samples must be by row, so use t()

d.n0.CZM <- cmultRepl(t(d.min), label=0, method="CZM")
```

```
## No. corrected values:  20044
```

```
# turn this into a centered log-ratio transform
# samples are by row
# remember that apply by row rotates the data
# R is terrible, so we need to use t() again

d.clr <- t( apply(d.n0.CZM, 1, function(x){log(x) - mean(log(x))}) )

d.mvar.clr <- mvar(d.clr)

d.pcx <- prcomp(d.clr)

# TODO color samples by most abundant organism
biplot(d.pcx, cex=c(0.5,0.05), col=c("black",rgb(1,0,0,0.2)), var.axes=F,
    xlab=paste("PC1: ", round(sum(d.pcx$sdev[1]^2)/d.mvar.clr, 3)),
```

```
    ylab=paste("PC2: ", round(sum(d.pcx$sdev[2]^2)/d.mvar.clr, 3)),
    scale=0
)

# filter the taxon table by what is in e.min
o.d <- o[,colnames(d)]
o.d.CZM <- cmultRepl(t(o.d), label=0, method="CZM")
o.d.clr <- t( apply(o.d.CZM, 1, function(x){log(x) - mean(log(x))}) )

# overplot taxon correlations with location of samples in the first two components

points(cor(d.pcx$x[,1], o.d.clr[,1], method="kendall") /50,
    cor(d.pcx$x[,2], o.d.clr[,1], method="kendall")/50, pch="I", col="DodgerBlue")
points(cor(d.pcx$x[,1], o.d.clr[,6], method="kendall") /50,
    cor(d.pcx$x[,2], o.d.clr[,6], method="kendall")/50, pch="L", col="DarkBlue")

points(cor(d.pcx$x[,1], o.d.clr[,2], method="kendall") /50,
    cor(d.pcx$x[,2], o.d.clr[,2], method="kendall")/50, pch="C", col="cyan")

points(cor(d.pcx$x[,1], o.d.clr[,10], method="kendall") /50,
    cor(d.pcx$x[,2], o.d.clr[,10], method="kendall")/50, pch="M", col="green")

points(cor(d.pcx$x[,1], o.d.clr[,8], method="kendall") /50,
    cor(d.pcx$x[,2], o.d.clr[,8], method="kendall")/50, pch="P", col="purple")

points(cor(d.pcx$x[,1], o.d.clr[,7], method="kendall") /50,
    cor(d.pcx$x[,2], o.d.clr[,7], method="kendall")/50, pch="G", col="red")

points(cor(d.pcx$x[,1], o.d.clr[,9], method="kendall") /50, cor(d.pcx$x[,2], o.d.clr[,9],
    method="kendall")/50, pch="A", col="Goldenrod")

points(cor(d.pcx$x[,1], o.d.clr[,12], method="kendall") /50, cor(d.pcx$x[,2],
    o.d.clr[,12], method="kendall")/50, pch="S", col="coral3")
```



This is a good representation of the data explaining 0.346 proportion of the variance in the data on the first component, and a much smaller amount on the second and later components. This reduced dataset contains 19496 genes. We can see that the samples cluster into a few fairly discrete groups. These groups contain fairly discrete sets of genes, and we can infer that the sample set distributions are driven by gene occurrence in different genomes.

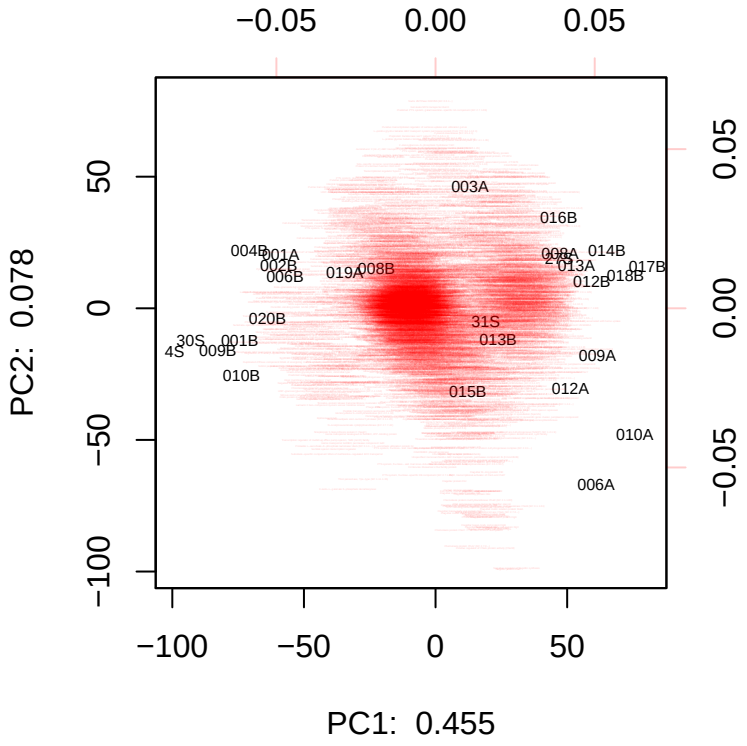
To get around the gene occurrence problem, we aggregate the data to the SEED subsystem 4 level.

```
e <- read.table("data/AitchisonTransformedDataForALDExInput.txt", header=T,
   row.names=1, check.names=F, sep="\t", comment.char="", quote="")

e.n0.CZM <- cmultRepl(t(e), label=0, method="CZM")
```

```
## No. corrected values:  2869
```

```
# turn this into a centered log-ratio transform
# samples are by row
# remember that apply by row rotates the data
# R is terrible, so we need to use t() again

e.clr <- t( apply(e.n0.CZM, 1, function(x){log(x) - mean(log(x))}) )

e.mvar.clr <- mvar(e.clr)

e.pcx <- prcomp(e.clr)

# TODO color samples by most abundant organism
biplot(e.pcx, cex=c(0.5,0.05), col=c("black",rgb(1,0,0,0.2)), var.axes=F,
    xlab=paste("PC1: ", round(sum(e.pcx$sdev[1]^2)/e.mvar.clr, 3)),
    ylab=paste("PC2: ", round(sum(e.pcx$sdev[2]^2)/e.mvar.clr, 3)),
```

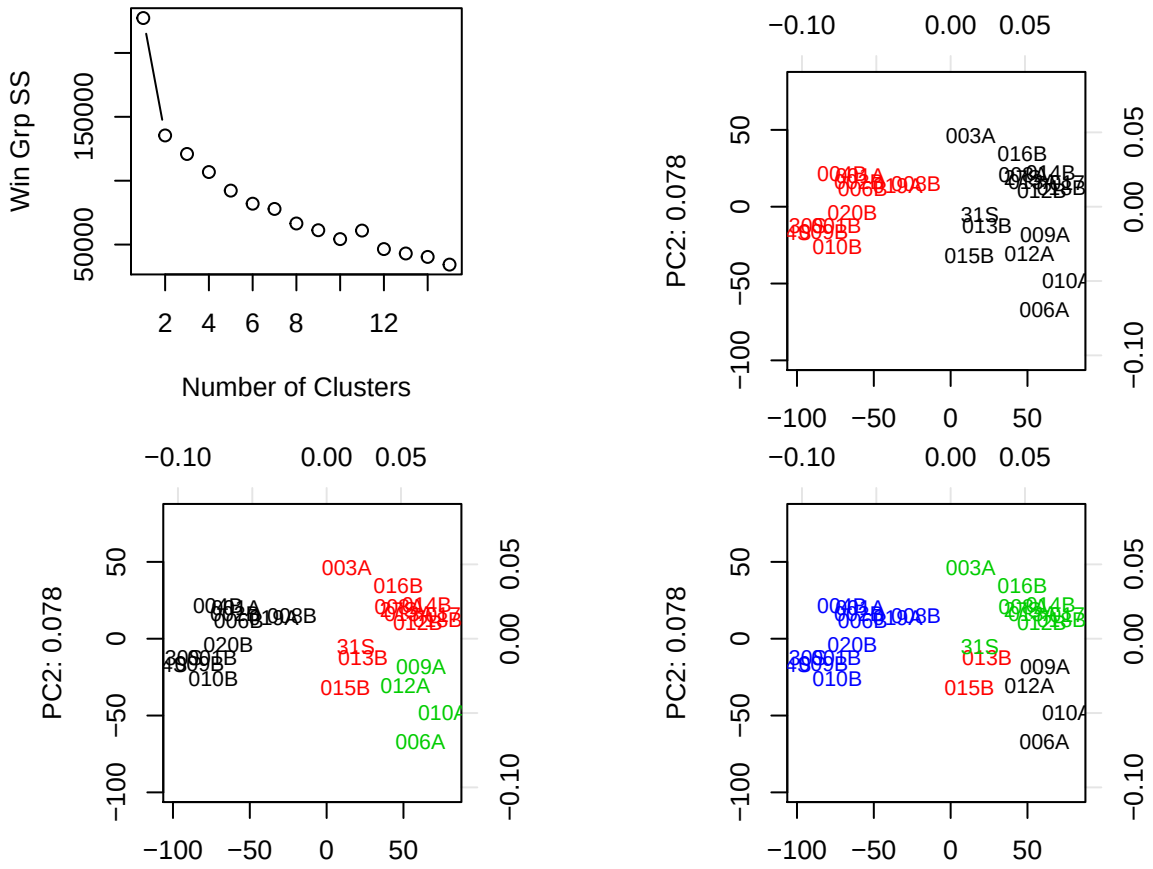
```
    scale=0
)
```



This is a much better representation of the data. We did not bother filtering by abundance since there is a much smaller number of features (3874). Now we can see a fairly clear split in the data that is less driven by gene occurrence and more by functions within the microbiome (cite Beiko review regarding ecosystems of genes rather than organisms here!).

```
mydata <- as.data.frame(e.clr)

# calculate the within group sum of squares
wss <- (nrow(mydata) -1) * sum(apply(mydata,2,var))

# summarize the fit for the first 15 groups
for (i in 2:15) wss[i] <- sum(kmeans(mydata,
        centers=i)$withinss)

# the plot suggests a reasonable fit can be had with about 2-3 clusters
# and after that there are only small improvements with additional clusters
layout( matrix(c(1,2,3,4),2,2, byrow=T), widths=c(10,10), height=c(10,10))

par(mar=c(5,4,0,5)+0.1)

# make the sum of squares distance plot
plot(1:15, wss[1:15], type="b", xlab="Number of Clusters",
    ylab="Win Grp SS")

par(mar=c(2,0,2,0)+0.1)

# plot the data for cluster sizes 2-4
```

```
for(i in 2:4){
    fit <- kmeans(mydata,i)
    mydata$fit.cluster <- NULL # clear any previous data
    mydata <- data.frame(mydata, fit$cluster) # add the cluster data

    coloredBiplot(e.pcx,col=rgb(0,0,0,0.1),cex=c(0.8,0.05),
     xlab=paste("PC1: ", round(sum(e.pcx$sdev[1]^2)/e.mvar.clr, 3), sep=""),
     ylab=paste("PC2: ", round(sum(e.pcx$sdev[2]^2)/e.mvar.clr, 3), sep=""),
     xlabs.col=mydata$fit.cluster, arrow.len=0.05, var.axes=F, expand=0.8,  scale=0)
}
```



K-means clustering supports only 2 groups. In fact, only the first plot needs to be shown to support this: maybe include the barplot?

Samples 3A, 31S, 13B and 15B are near the midpoint on the right, and 19A and 8B are near the midpoint on the left. Five of these six microbiomes are atypical, and the other 19A is exclusively L. iners, but phenotypically is BV. We next exclude these six samples and re-examine the partitioning.

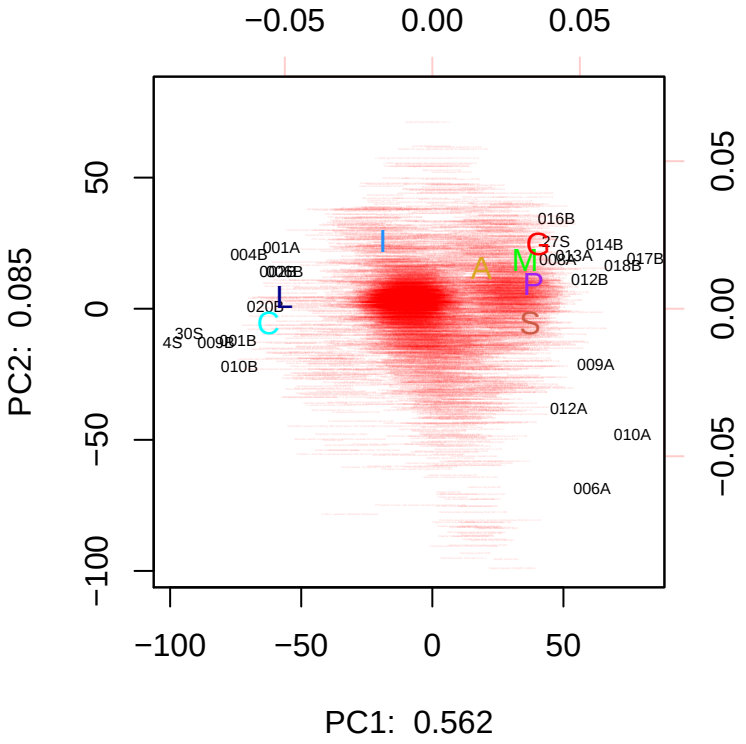
```
e.min <- e

e.min[,"003A"] <- NULL
e.min[,"31S"] <- NULL
e.min[,"013B"] <- NULL
e.min[,"015B"] <- NULL
e.min[,"019A"] <- NULL
e.min[,"008B"] <- NULL
```

```
e.min.n0.CZM <- cmultRepl(t(e.min), label=0, method="CZM")
```

```
## No. corrected values:  545
```

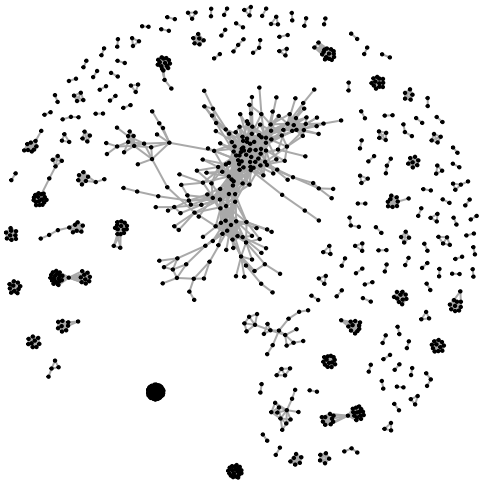
```
# turn this into a centered log-ratio transform
# samples are by row
# remember that apply by row rotates the data
# R is terrible, so we need to use t() again

e.min.clr <- t( apply(e.min.n0.CZM, 1, function(x){log(x) - mean(log(x))}) )

e.min.mvar.clr <- mvar(e.min.clr)

e.min.pcx <- prcomp(e.min.clr)

# TODO color samples by most abundant organism
biplot(e.min.pcx, cex=c(0.5,0.05), col=c("black",rgb(1,0,0,0.2)), var.axes=F,
    xlab=paste("PC1: ", round(sum(e.min.pcx$sdev[1]^2)/e.min.mvar.clr, 3)),
    ylab=paste("PC2: ", round(sum(e.min.pcx$sdev[2]^2)/e.min.mvar.clr, 3)),
    scale=0
)

# filter the taxon table by what is in e.min
o.min <- o[,colnames(e.min)]
o.CZM <- cmultRepl(t(o.min), label=0, method="CZM")
o.clr <- t( apply(o.CZM, 1, function(x){log(x) - mean(log(x))}) )

# overplot taxon correlations with first two components
cor.scale.0 <-  max(abs(1/range(e.min.pcx$rotation[,1:2])))

points(cor(e.min.pcx$x[,1], o.clr[,1], method="kendall") /cor.scale.0,
    cor(e.min.pcx$x[,2], o.clr[,1], method="kendall")/cor.scale.0, pch="I",
    col="DodgerBlue")
points(cor(e.min.pcx$x[,1], o.clr[,6], method="kendall") /cor.scale.0,
    cor(e.min.pcx$x[,2], o.clr[,6], method="kendall")/cor.scale.0, pch="L",
    col="DarkBlue")

points(cor(e.min.pcx$x[,1], o.clr[,2], method="kendall") /cor.scale.0,
    cor(e.min.pcx$x[,2], o.clr[,2], method="kendall")/cor.scale.0, pch="C", col="cyan")

points(cor(e.min.pcx$x[,1], o.clr[,10], method="kendall") /cor.scale.0,
    cor(e.min.pcx$x[,2], o.clr[,10], method="kendall")/cor.scale.0, pch="M", col="green")

points(cor(e.min.pcx$x[,1], o.clr[,8], method="kendall") /cor.scale.0,
    cor(e.min.pcx$x[,2], o.clr[,8], method="kendall")/cor.scale.0, pch="P", col="purple")

points(cor(e.min.pcx$x[,1], o.clr[,7], method="kendall") /cor.scale.0,
    cor(e.min.pcx$x[,2], o.clr[,7], method="kendall")/cor.scale.0, pch="G", col="red")

points(cor(e.min.pcx$x[,1], o.clr[,9], method="kendall") /cor.scale.0,
    cor(e.min.pcx$x[,2], o.clr[,9], method="kendall")/cor.scale.0, pch="A",
    col="Goldenrod")
```

```
points(cor(e.min.pcx$x[,1], o.clr[,12], method="kendall") /cor.scale.0,
    cor(e.min.pcx$x[,2], o.clr[,12], method="kendall")/cor.scale.0, pch="S", col="coral3")
```




```
# We start with the log-ratio transformed data

# generate the phi statistic
# this is a measure of the concordance of ratios across samples
# low values are better, a good place to start is 0.05, but this has only been
# examined in one dataset
# this is from Lovell's R package in progress
e.min.sym.phi <- propr.phisym(e.min.clr)

# generate a symmetrical regression line of best fit
e.min.sma <- sma.df(e.min.clr)

# make an empty dataset of the correct size
lt <- which(col(e.min.sym.phi)<row(e.min.sym.phi), arr.ind=FALSE)
lt.ind <- which(col(e.min.sym.phi)<row(e.min.sym.phi), arr.ind=TRUE)

# find the minimum value of each row
e.min.phi.min <- lt.row.min(e.min.sym.phi)

# dataframe to hold the info,
# data is a set of vectors where only the lower triangle is kept, because the matrix is symmetrical
# this is needed so the subset function works properly
e.min.sma.df <- data.frame(row=factor(rownames(e.min.sym.phi)[lt.ind[,"row"]]),
    col=factor(colnames(e.min.sym.phi)[lt.ind[,"col"]]))
e.min.sma.df$phi <- e.min.sym.phi[lt]
# find the set of connections with phi less than some value
```

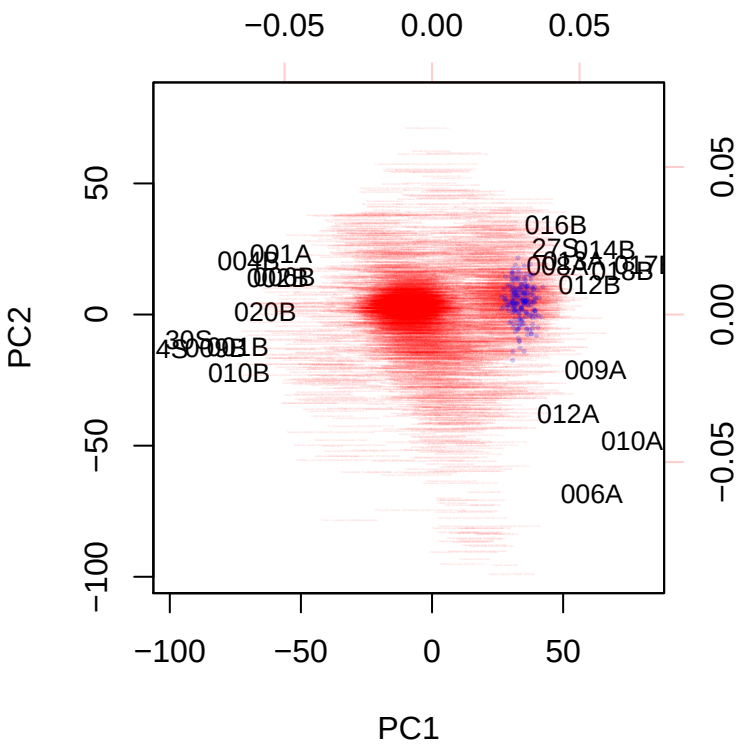
```
 phi.cutoff <- 0.01

e.min.sma.lo.phi <- subset(e.min.sma.df, phi < phi.cutoff)

# generate a graphical object
g <- graph.data.frame(e.min.sma.lo.phi, directed=FALSE)

# overview of all the proportional relationships
# this can take a long time!!!
plot(g, layout=layout.fruchterman.reingold.grid(g, weight=0.05/E(g)$phi), vertex.size=1,
    vertex.color="black", vertex.label=NA)
```



```
# # get the clusters from the graph object
 g.clust <- clusters(g)
```

```
# write them to a file

# # data frame containing the names and group memberships of each cluster
g.df <- data.frame(Systematic.name=V(g)$name, cluster=g.clust$membership,
    cluster.size=g.clust$csize[g.clust$membership])
write.table(g.df, file="g.df.txt", sep="/t", quote=F, col.names=NA)

max.clust <- induced.subgraph(g, which(g.clust$membership %in% which(g.clust$csize ==
    max(g.clust$csize))))
max.clust.names <- V(max.clust)$name
```

```
par(mfrow=c(1,1))

biplot(e.min.pcx,col=c("black", rgb(1,0,0,0.2)),cex=c(0.8,0.05),
var.axes=F, scale=0)

#nms <- rownames(g.df)[g.df$cluster==max.clust.names]

points(e.min.pcx$rotation[max.clust.names,][,1],e.min.pcx$rotation[max.clust.names,][,2],
    col=rgb(0,0,1,0.2),pch=19,cex=0.2)
```


−0.05
0.00
0.05
50
0
−50
−100
PC2
0.05
0.00
−0.05
016B
012B
020B
010B
009A
012A
010A
006A
−100
−50
0
50
PC1