Day 8: Beta Diversity (Guerrero Negro)

Back to Table of Contents

All of the code in this page is meant to be run in `R` unless otherwise specified.

Install biom package and vegan package if not installed.

```
install.packages(c('biom','vegan'),repo='http://cran.wustl.edu')
```

Load biom package, load data

```
library('biom')
library('vegan')

# load biom file
otus.biom <- read_biom('otu_table_json.biom')

# Extract data matrix (OTU counts) from biom table
otus <- as.matrix(biom_data(otus.biom))

# transpose so that rows are samples and columns are OTUs
otus <- t(otus)

# load mapping file
map <- read.table('map.txt', sep='\t', comment='', head=T, row.names=1)
```

It is extremely important to ensure that your OTU table and metadata table sample IDs are lined up correctly.

```
# see rownames of map and otus
rownames(map)
```

```
##  [1] "GN01P.484257"   "GN01P.o.484256" "GN02P.o.484250" "GN02P.484248"
##  [5] "GN03P.484253"   "GN03P.o.484249" "GN04P.484258"   "GN04P.o.484251"
##  [9] "GN05P.o.484260" "GN05P.484261"   "GN06P.o.484262" "GN06P.484247"
## [13] "GN07P.o.484246" "GN07P.484259"   "GN08P.484265"   "GN08P.o.484263"
## [17] "GN09P.484254"   "GN09P.o.484264" "GN10P.o.484252" "GN10P.484255"
```

```
rownames(otus)
```

```
##  [1] "GN01P.484257"   "GN07P.o.484246" "GN01P.o.484256" "GN06P.484247"
##  [5] "GN05P.484261"   "GN08P.484265"   "GN05P.o.484260" "GN06P.o.484262"
##  [9] "GN08P.o.484263" "GN07P.484259"   "GN04P.484258"   "GN04P.o.484251"
## [13] "GN09P.484254"   "GN09P.o.484264" "GN02P.484248"   "GN03P.484253"
## [17] "GN02P.o.484250" "GN03P.o.484249"
```

```
# find the overlap
common.ids <- intersect(rownames(map), rownames(otus))

# get just the overlapping samples
otus <- otus[common.ids,]
map <- map[common.ids,]
```

See dimensions of OTU table

```
dim(otus)
```

```
## [1]   18 1750
```

See dimensions of mapping file

```
dim(map)
```

```
## [1] 18 60
```

Get three different distances metrics

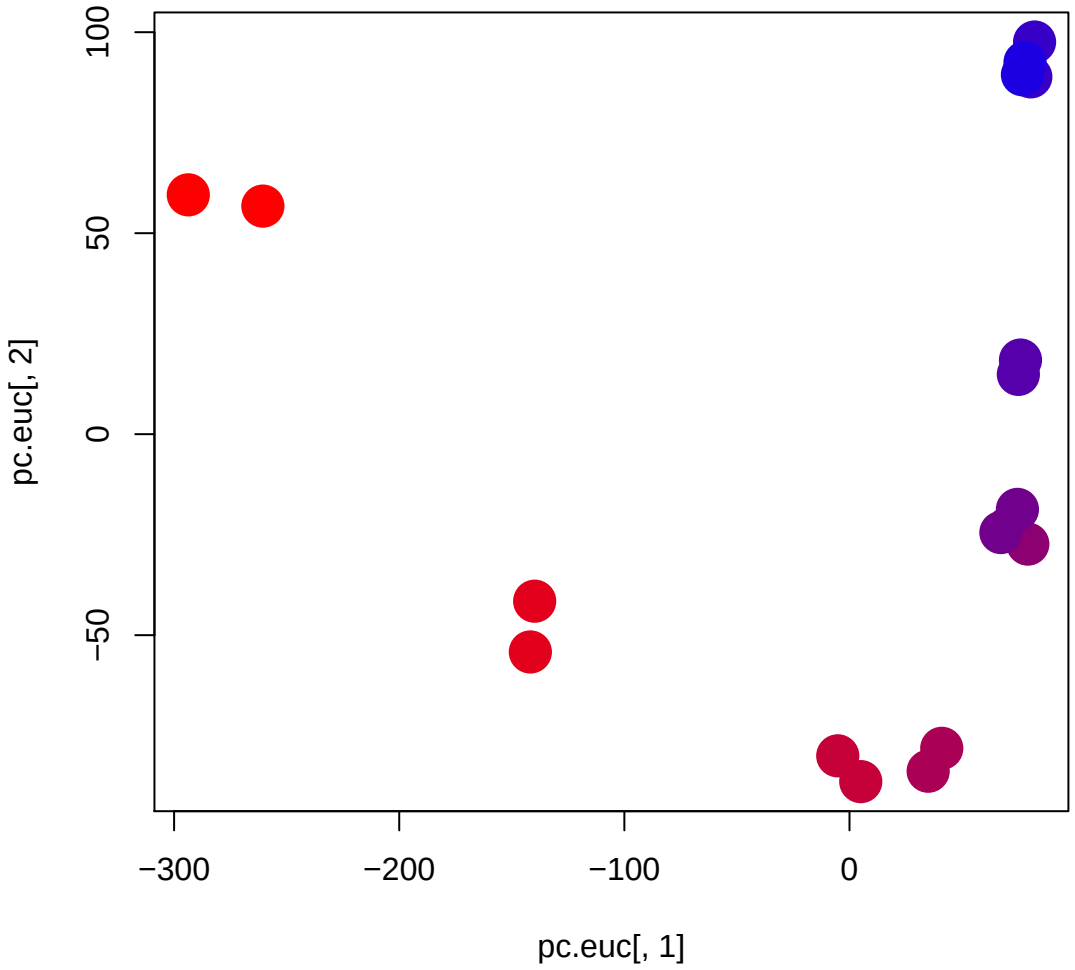
```
# get Euclidean distance
d.euc <- dist(otus)

# get Bray-Curtis distances (default for Vegan)
d.bray <- vegdist(otus)

# get Chi-square distances using vegan command
# we will extract chi-square distances from correspondence analysis
my.ca <- cca(otus)
d.chisq <- as.matrix(dist(my.ca$CA$u[,1:2]))
```

Now run principal coordinates embedding on the distance metrics

```
# Run PCoA (not PCA)
pc.euc <- cmdscale(d.euc, k=2)

# Bray-Curtis principal coords
pc.bray <- cmdscale(d.bray,k=2)

# get first two dimensions of chi-square coordinates:
pc.chisq <- my.ca$CA$u[,1:2]
```

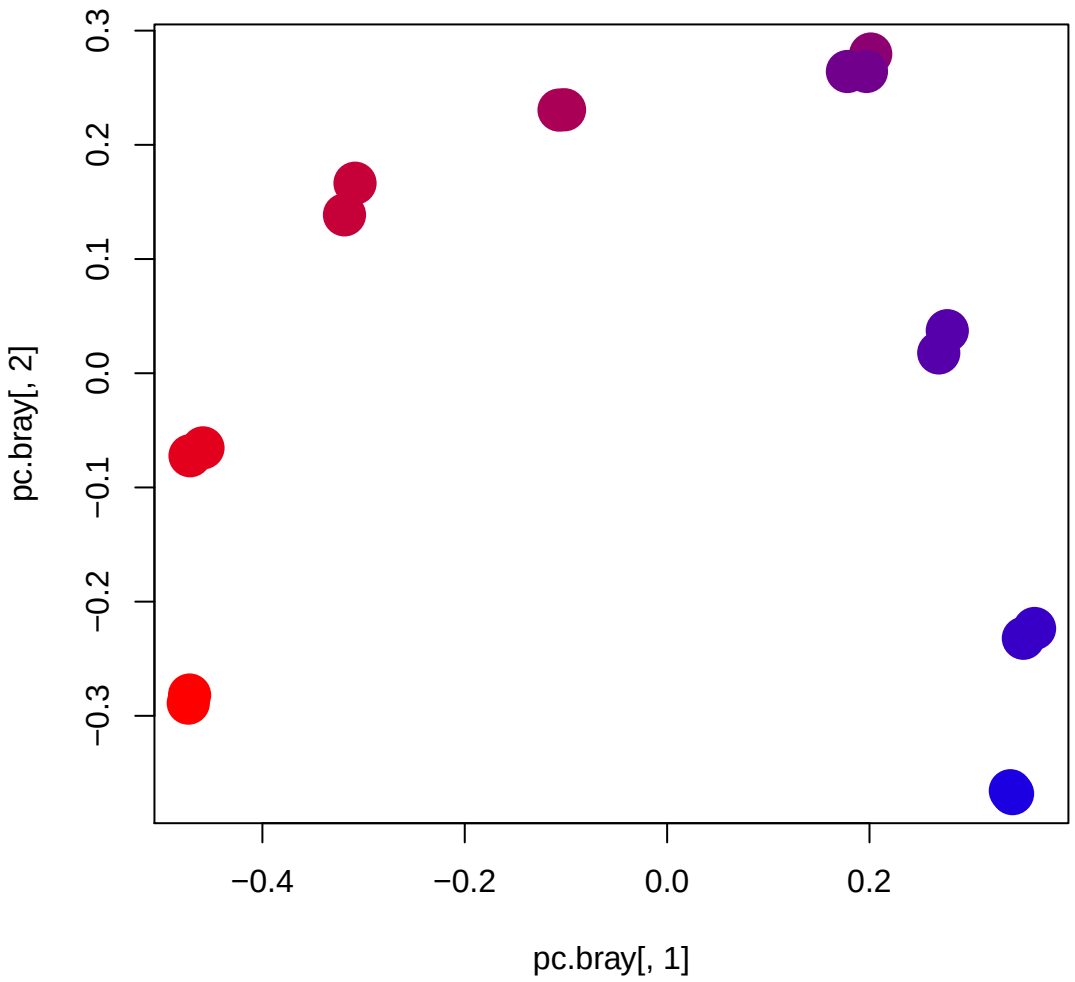
Plot Euclidean distances with gradient colors

```
# makes a gradient from red to blue
my.colors <- colorRampPalette(c('red','blue'))(10)

# plot Euclidean PCoA coords using color gradient
# based on layer (1...10)
layer <- map[,'LAYER']
plot(pc.euc[,1], pc.euc[,2], col=my.colors[layer], cex=3, pch=16)
```



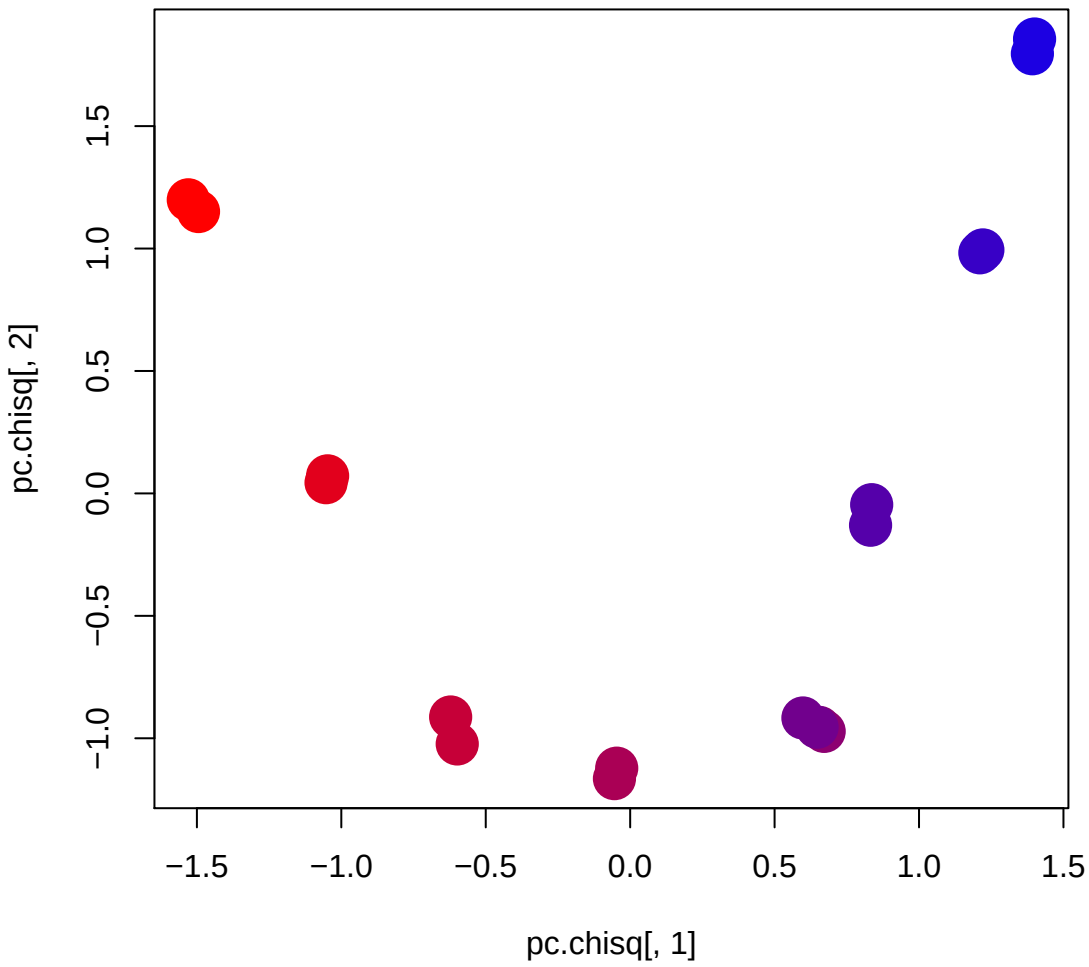
Plot Bray-Curtis distances with gradient colors

```
# Plot Bray-Curtis PCoA
plot(pc.bray[,1], pc.bray[,2], col=my.colors[layer], cex=3, pch=16)
```



Plot Chi-square distances with gradient colors

```
# Plot Chi-square PCoA
plot(pc.chisq[,1], pc.chisq[,2], col=my.colors[layer], cex=3, pch=16)
```



Visualizing UniFrac distances

Calculate UniFrac distances in QIIME

```
# Note: This command is on the command line, not in R
# (load macqiime if necessary)
beta_diversity.py -i otu_table.biom -o beta -t ../ref/greengenes/97_otus.tree
```

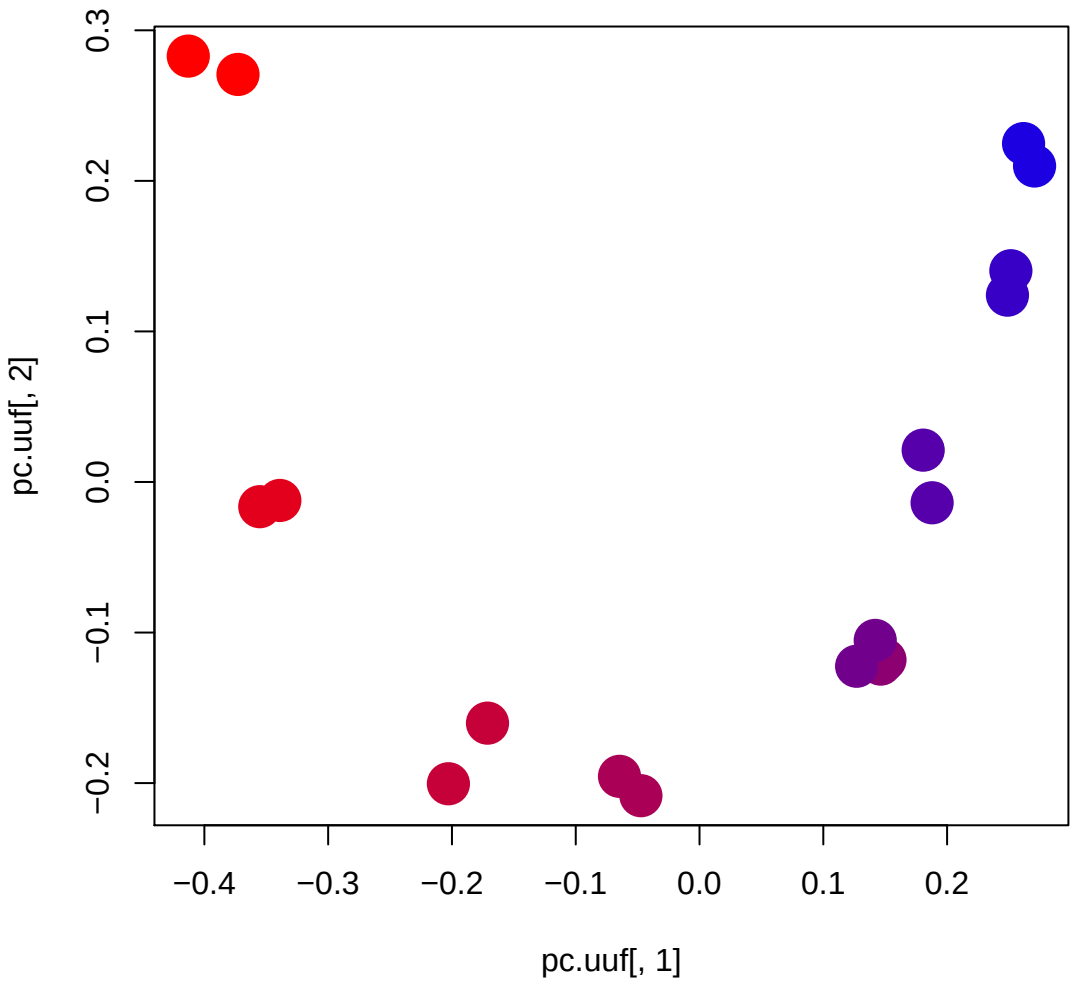
Load UniFrac distances, calculate PCoA

```
# load unweighted and weighted unifrac
d.uuf <- read.table('beta/unweighted_unifrac_otu_table.txt', sep='\t',head=T,row=1)
d.wuf <- read.table('beta/weighted_unifrac_otu_table.txt', sep='\t',head=T,row=1)

# ensure that these last two matrices have the same samples in the
```

```
# same order as the metadata table
d.uuf <- d.uuf[common.ids, common.ids]
d.wuf <- d.wuf[common.ids, common.ids]

# get first two dimensions of unifrac PCoA:
pc.uuf <- cmdscale(d.uuf, k=2)
pc.wuf <- cmdscale(d.wuf, k=2)
```

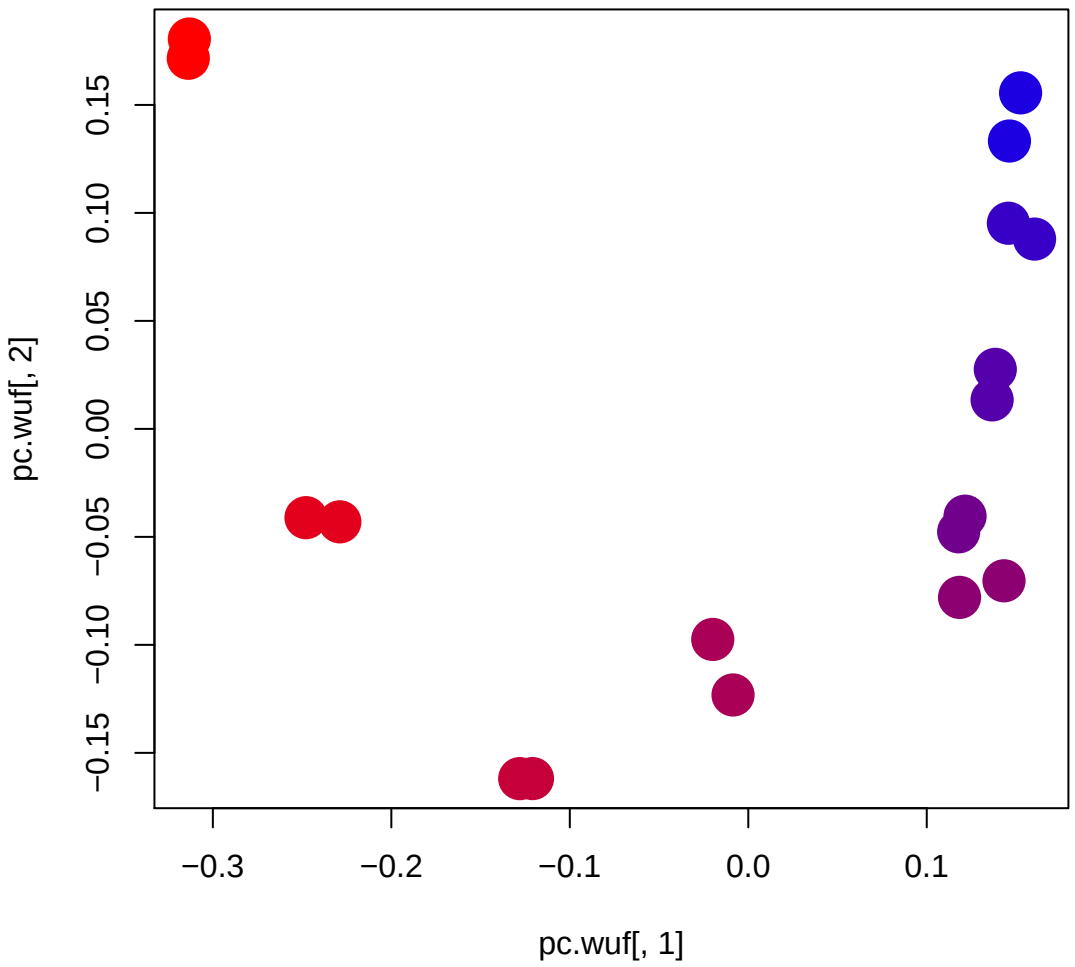
Plot unweighted UniFrac distances with gradient colors

```
plot(pc.uuf[,1], pc.uuf[,2], col=my.colors[layer], cex=3, pch=16)
```



Plot weighted UniFrac distances with gradient colors

```
plot(pc.wuf[,1], pc.wuf[,2], col=my.colors[layer], cex=3, pch=16)
```

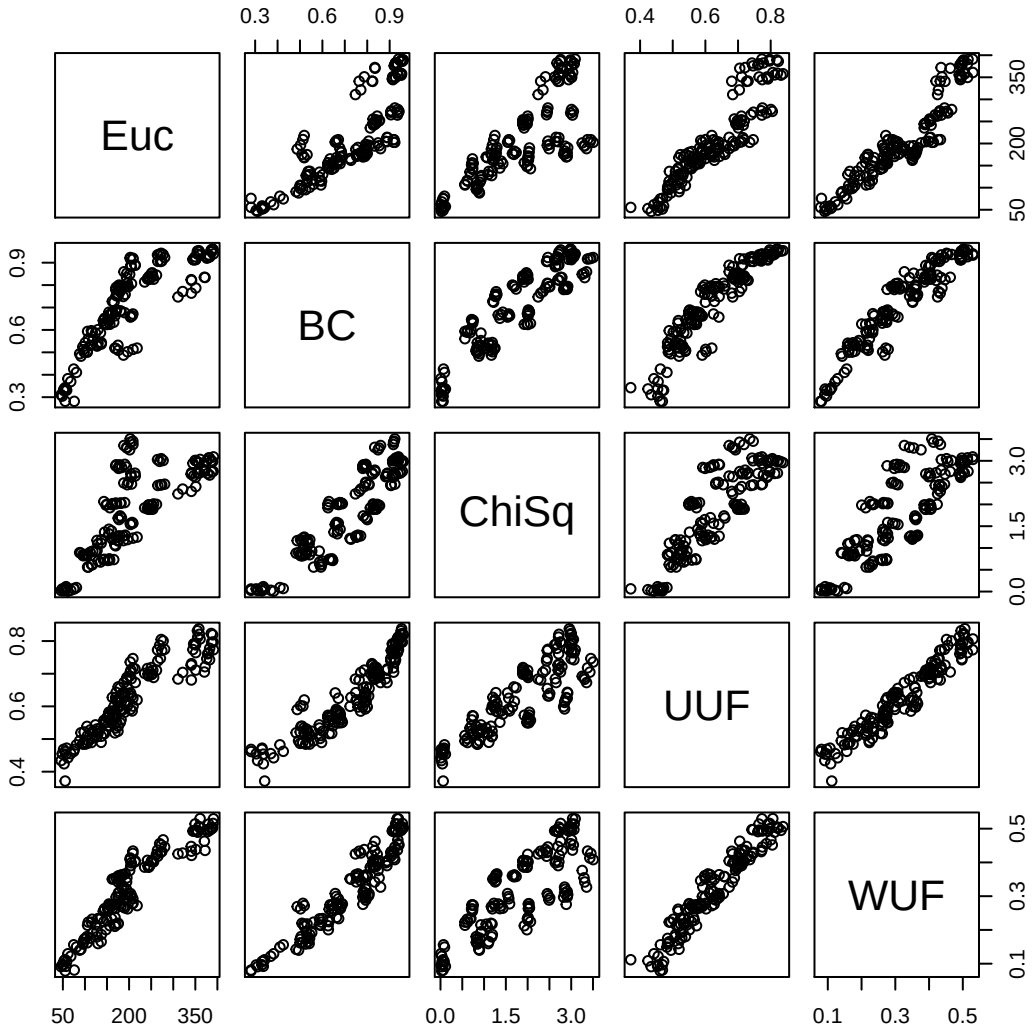


Note: to make a PDF:

```
pdf("chisq.pdf",width=5,height=5)
plot(pc.chisq[,1], pc.chisq[,2], col=my.colors[map[,'LAYER']], cex=3, pch=16)
dev.off()
```

Let's plot pairwise comparisons of the different distance metrics

```
d.vector.matrix <- cbind(as.numeric(d.euc), as.numeric(d.bray), as.numeric(as.dist(d.chisq)), as.numeri
colnames(d.vector.matrix) <- c('Euc','BC','ChiSq','UUF','WUF')
pairs(d.vector.matrix)
```



And display the pairwise pearson correlations

```
cor(d.vector.matrix)
```

```
##             Euc        BC     ChiSq       UUF       WUF
## Euc   1.0000000 0.8324774 0.7664577 0.9159583 0.9344522
## BC    0.8324774 1.0000000 0.8976043 0.9298203 0.9454640
## ChiSq 0.7664577 0.8976043 1.0000000 0.8535358 0.8192654
## UUF   0.9159583 0.9298203 0.8535358 1.0000000 0.9631557
## WUF   0.9344522 0.9454640 0.8192654 0.9631557 1.0000000
```

Which distance metric best recovered physical sample distances based on END_DEPTH?

```
# y is the euclidean distance matrix based on ending depth of each layer
y <- as.vector(dist(map$END_DEPTH))
```

```r
# Test the correlation of END_DEPTH distance and ecological distance
# for each metric
metrics <- list(d.euc, d.bray, d.chisq, d.uuf, d.wuf)
names(metrics) <- colnames(d.vector.matrix) # reuse pairwise column names

for(i in 1:length(metrics)){
    d.name <- names(metrics)[i]

    # convert distance matrix to vector form
    d <- as.vector(as.dist(metrics[[i]]))

    cat('Correlation of ',d.name,':','\n',sep='')
    print(cor.test(d, y, method='spear', exact=FALSE))
}
```

```
## Correlation of Euc:
##
##  Spearman's rank correlation rho
##
## data:  d and y
## S = 367040, p-value = 8.875e-07
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.3850855
##
## Correlation of BC:
##
##  Spearman's rank correlation rho
##
## data:  d and y
## S = 222770, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.6267841
##
## Correlation of ChiSq:
##
##  Spearman's rank correlation rho
##
## data:  d and y
## S = 149150, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.7501234
##
## Correlation of UUF:
##
##  Spearman's rank correlation rho
##
## data:  d and y
```

```
## S = 282320, p-value = 2.59e-12
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.527028
##
## Correlation of WUF:
##
##  Spearman's rank correlation rho
##
## data:  d and y
## S = 340610, p-value = 3.055e-08
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.4293781
```

Here is the one-liner version if you just want to test one distance metric vs. your continuous variable of interest. There are two ways to do this:

1. ask whether the **overall** distance metric is correlated with your gradient:

```
# y is the euclidean distance matrix based on your variable of interest (here depth)
# note: euclidean diatnace makes sense in the Guerrero Negro sampling depth because
# it is measuring physical distance (in millimeters)
# Note: the "as.vector" stretches it out to a single numeric vector (no longer a matrix)
y <- as.vector(dist(map$END_DEPTH))

# Stretch out the d
d <- as.vector(as.dist(d.chisq))
cor.test(d, y, method='spear', exact=FALSE)
```

```
##
##  Spearman's rank correlation rho
##
## data:  d and y
## S = 149150, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.7501234
```

2. You could ask whether PC1, the **first principal axis of variation** (not overall distance), is significantly correlated with your variable of interest. This is an even stronger result if significant.

```
cor.test(map$END_DEPTH, pc.chisq[,1], method='spear', exact=FALSE)
```

```
##
##  Spearman's rank correlation rho
##
## data:  map$END_DEPTH and pc.chisq[, 1]
## S = 20.585, p-value = 1.952e-12
```

```
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.9787561
```

Note that Chi-square has the highest correlation with END_DEPTH.