HJA.Rmd

Initial Setup

```
rm(list=ls())
setwd("~/GitHub/HJA-streams/")
source("./analysis/MothurTools.R")
```

```
## Loading required package: reshape
```

```
require("vegan")
```

```
## Loading required package: vegan
## Loading required package: permute
## Loading required package: lattice
## This is vegan 2.3-1
```

```
se <- function(x, ...){sd(x, na.rm = TRUE)/sqrt(length(na.omit(x)))}
```

Import Shared and Design Files

```
# Define Inputs
# Design = general design file for experiment
# shared = OTU table from mothur with sequence similarity clustering
# Taxonomy = Taxonomic information for each OTU
design <- "./data/design.txt"
shared <- "./data/hja_streams.final.shared"
taxon  <- "./data/hja_streams.final.0.03.taxonomy"
env    <- "./data/hja_env.csv"

# Import Design
design <- read.delim(design, header=T, row.names=1)

# Import Shared Files
OTUs <- read.otu(shared = shared, cutoff = "0.03")         # 97% Similarity

# Import Taxonomy
OTU.tax <- read.tax(taxonomy = taxon, format = "rdp")

# Import Env
env <- read.csv(env, header=T)
```

Data Transormations

```r
# Remove OTUs with less than two occurances across all sites
OTUs <- OTUs[, which(colSums(OTUs) >= 2)]

# Sequencing an Good's Coverage
# Sequencing Coverage
coverage <- rowSums(OTUs)

# Good's Coverage
goods <- function(x = ""){
  1 - (sum(x == 1) / rowSums(x))
}
goods.c <- goods(OTUs)

# Remove Low Coverage Samples (This code removes two sites: Site 5DNA, Site 6cDNA)
lows <- which(coverage < 7000)
lows
```

```
## LC_01_W LC_03_S LC_04_S LC_06_S LC_06_W LC_07_S LC_12_W LC_14_S LC_15_S
##       2       5       7      11      12      13      23      26      28
## LC_17_W LC_19_W LC_20_S LC_21_S LC_22_S W1_01_W W1_03_S W1_04_S W1_07_S
##      33      37      38      40      42      45      48      50      56
## W1_07_W W1_09_S W1_09_W W1_10_W W1_11_S W1_13_S W1_14_S W1_15_S W1_16_S
##      57      60      61      63      64      68      70      72      74
## W1_16_W W1_17_S W1_18_W W1_19_W
##      75      76      79      81
```

```r
OTUs <- OTUs[-which(coverage < 7000), ]
design <- design[-which(coverage < 7000), ]
env <- env[-which(coverage < 7000), ]

# Make Relative Abundence Matrices
OTUsREL <- OTUs
for(i in 1:dim(OTUs)[1]){
  OTUsREL[i,]<- OTUs[i,]/sum(OTUs[i,])
}

# Log Transform Relative Abundances
OTUsREL.log <- decostand(OTUs, method="log")
```

Calculate Alpha Diversity

```r
# Observed Richness
S.obs <- rowSums((OTUs > 0) * 1)

# Simpson's Evenness
SimpE <- function(x = ""){
  x <- as.data.frame(x)
  D <- diversity(x, "inv")
  S <- sum((x > 0) * 1)
  E <- (D)/S
  return(E)
```

```
}
simpsE <- round(apply(OTUs, 1, SimpE), 3)

# Shannon's Diversity
H <- function(x = ""){
  x <- x[x>0]
  H = 0
  for (n_i in x){
    p = n_i / sum(x)
    H = H - p*log(p)
  }
  return(H)
}

shan <- round(apply(OTUs, 1, H), 2)
shan2 <- diversity(OTUs, index = "shannon")

# Rarefaction
hja.S <- rowSums((OTUs > 0) * 1)
min.N <- min(rowSums(OTUs))
#S.rarefy <- rarefy(x = OTUs, sample = min.N, se = TRUE)
#rarecurve(x = OTUs, step = 20, col = "blue", cex = 0.6, las = 1)
#rared <- t(S.rarefy)

alpha.div <- cbind(design, S.obs, simpsE, shan)
```

Alpha Diversity Across Gradient

```
# Seperate data based on water and sediment samples
water <- alpha.div[alpha.div$habitat == "water",]
sed <- alpha.div[alpha.div$habitat == "sediment", ]

lc   <- alpha.div[alpha.div$watershed == "LC",]
ws01 <- alpha.div[alpha.div$watershed == "WS01",]

# Calculate Linear Model
model.rich <- lm(alpha.div$S.obs ~ alpha.div$habitat)
summary(model.rich)
```

```
##
## Call:
## lm(formula = alpha.div$S.obs ~ alpha.div$habitat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5004.1 -1726.5  -129.7  1092.9  7327.9
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3876.3      491.9   7.880 1.54e-10 ***
## alpha.div$habitatwater     2001.9      650.7   3.076  0.00329 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2410 on 54 degrees of freedom
## Multiple R-squared:  0.1491, Adjusted R-squared:  0.1334
## F-statistic: 9.464 on 1 and 54 DF,  p-value: 0.003287
```

```
model.div <- lm(alpha.div$shan ~ alpha.div$habitat)
summary(model.div)
```

```
##
## Call:
## lm(formula = alpha.div$shan ~ alpha.div$habitat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0634 -0.1500  0.1125  0.4066  0.8966
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               6.5175     0.1282  50.840   <2e-16 ***
## alpha.div$habitatwater    0.2659     0.1696   1.568    0.123
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.628 on 54 degrees of freedom
## Multiple R-squared:  0.04356,    Adjusted R-squared:  0.02584
## F-statistic: 2.459 on 1 and 54 DF,  p-value: 0.1227
```

```
model.even <- lm(alpha.div$simpsE ~ alpha.div$habitat)
summary(model.even)
```

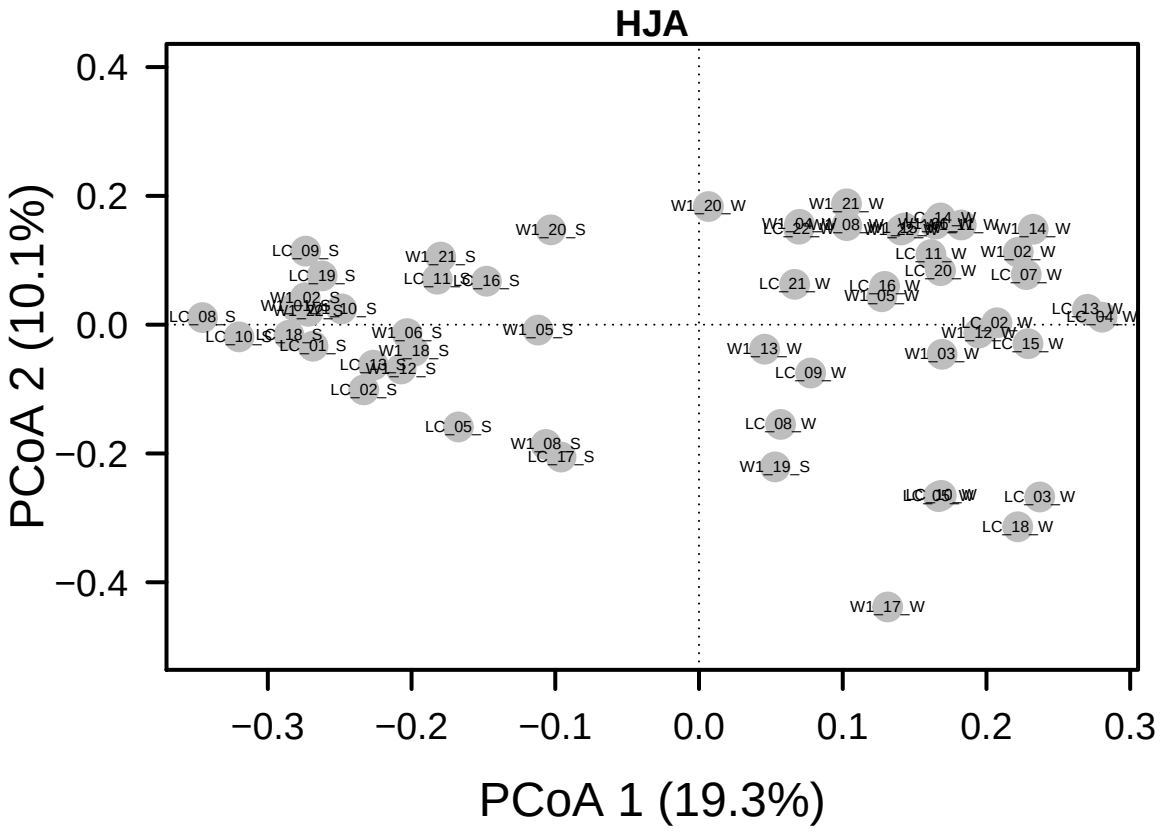
```
##
## Call:
## lm(formula = alpha.div$simpsE ~ alpha.div$habitat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.03646 -0.02019 -0.01119  0.01081  0.11254
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              0.050458   0.006283   8.031 8.79e-11 ***
## alpha.div$habitatwater -0.021271   0.008311  -2.559   0.0133 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03078 on 54 degrees of freedom
## Multiple R-squared:  0.1082, Adjusted R-squared:  0.09165
## F-statistic:  6.55 on 1 and 54 DF,  p-value: 0.01333
```

```r
# Calculate Confidance Intervals of Model
newdata.rich <- data.frame(alpha.div$habitat)
conf95.rich <- predict(model.rich, newdata.rich, interval="confidence")
```

PCoA Plots

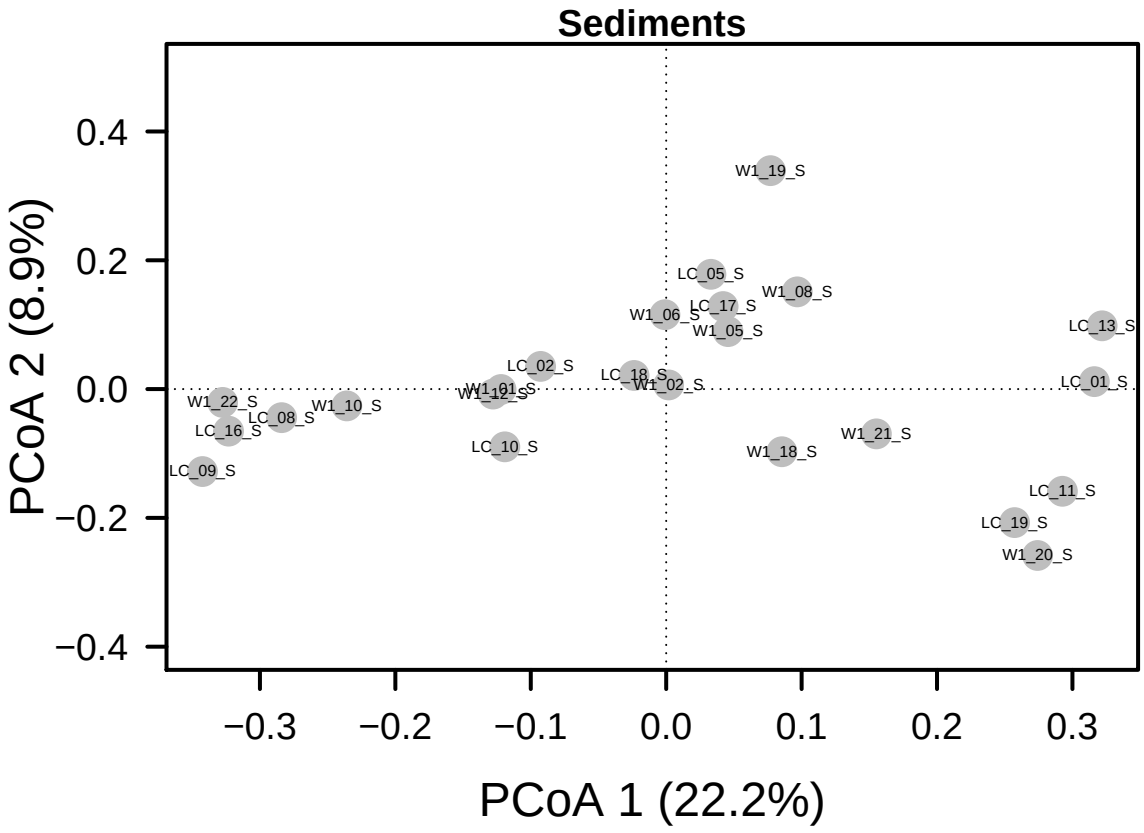
HJA

```r
hja.db <- vegdist(OTUsREL, method = "bray", upper = TRUE, diag = TRUE)
hja.pcoa <- cmdscale(hja.db, eig=TRUE, k=3)
var1 <- round(hja.pcoa$eig[1] / sum(hja.pcoa$eig),3) * 100
var2 <- round(hja.pcoa$eig[2] / sum(hja.pcoa$eig),3) * 100
var3 <- round(hja.pcoa$eig[3] / sum(hja.pcoa$eig),3) * 100

par(mar = c(5, 5, 1, 2) + 0.1)
plot(hja.pcoa$points[ ,1], hja.pcoa$points[ ,2], ylim = c(-0.5, 0.4),
     xlab = paste("PCoA 1 (", var1, "%)", sep = ""),
     ylab = paste("PCoA 2 (", var2, "%)", sep = ""),
     pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = F,
     main="HJA")
axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
abline(h = 0, v = 0, lty = 3)
box(lwd = 2)
points(hja.pcoa$points[ ,1], hja.pcoa$points[ ,2],
       pch=19, cex=2, bg="gray", col="gray")
text(hja.pcoa$points[ ,1], hja.pcoa$points[ ,2],
     cex = 0.5, labels = row.names(hja.pcoa$points))
```



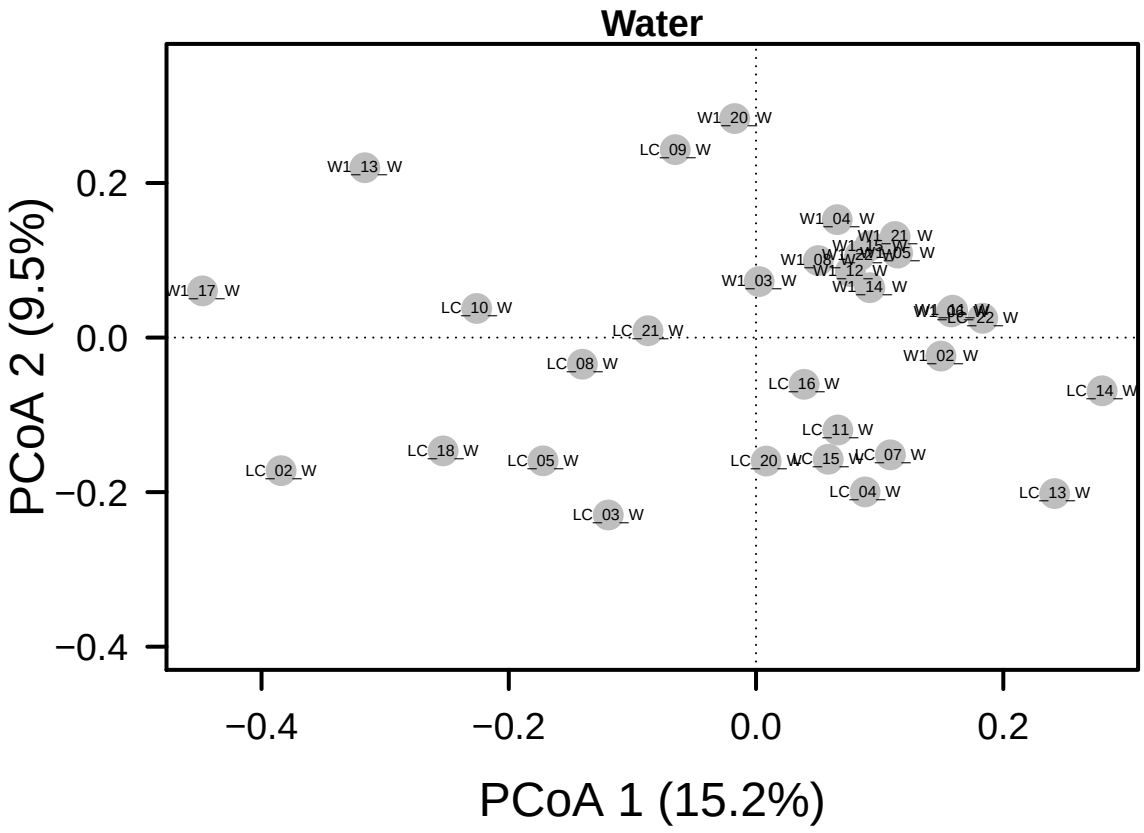
Sediments

```
seds <- OTUsREL.log[which(design$habitat == "sediment"),]
sediment.db <- vegdist(seds, method = "bray", diag = T)
sediment.pcoa <- cmdscale(sediment.db, eig=TRUE, k=3)
s.var1 <- round(sediment.pcoa$eig[1] / sum(sediment.pcoa$eig),3) * 100
s.var2 <- round(sediment.pcoa$eig[2] / sum(sediment.pcoa$eig),3) * 100
s.var3 <- round(sediment.pcoa$eig[3] / sum(sediment.pcoa$eig),3) * 100

par(mar = c(5, 5, 1, 2) + 0.1)
plot(sediment.pcoa$points[ ,1], sediment.pcoa$points[ ,2], ylim = c(-0.4, 0.5),
     xlab = paste("PCoA 1 (", s.var1, "%)", sep = ""),
     ylab = paste("PCoA 2 (", s.var2, "%)", sep = ""),
     pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = F,
     main="Sediments")
axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
abline(h = 0, v = 0, lty = 3)
box(lwd = 2)
points(sediment.pcoa$points[ ,1], sediment.pcoa$points[ ,2],
       pch=19, cex=2, bg="gray", col="gray")
text(sediment.pcoa$points[ ,1], sediment.pcoa$points[ ,2],
     cex=0.5, labels = row.names(sediment.pcoa$points))
```



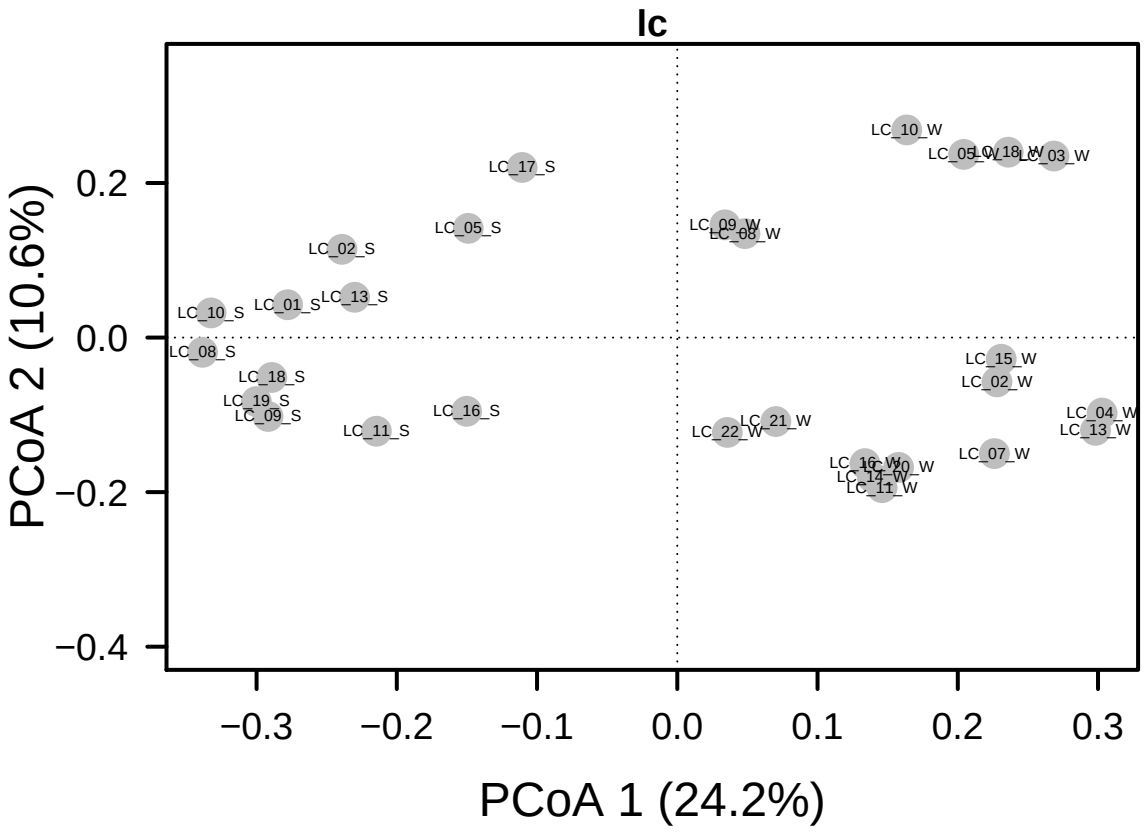
Water

```
water <- OTUsREL.log[which(design$habitat == "water"),]
water.db <- vegdist(water, method = "bray", diag = T)
water.pcoa <- cmdscale(water.db, eig=TRUE, k=3)
w.var1 <- round(water.pcoa$eig[1] / sum(water.pcoa$eig),3) * 100
w.var2 <- round(water.pcoa$eig[2] / sum(water.pcoa$eig),3) * 100
w.var3 <- round(water.pcoa$eig[3] / sum(water.pcoa$eig),3) * 100

par(mar = c(5, 5, 1, 2) + 0.1)
plot(water.pcoa$points[ ,1], water.pcoa$points[ ,2], ylim = c(-0.4, 0.35),
     xlab = paste("PCoA 1 (", w.var1, "%)", sep = ""),
     ylab = paste("PCoA 2 (", w.var2, "%)", sep = ""),
     pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = F,
     main="Water")
axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
abline(h = 0, v = 0, lty = 3)
box(lwd = 2)
points(water.pcoa$points[ ,1], water.pcoa$points[ ,2],
       pch=19, cex=2, bg="gray", col="gray")
text(water.pcoa$points[ ,1], water.pcoa$points[ ,2],
     cex=0.5, labels = row.names(water.pcoa$points))
```



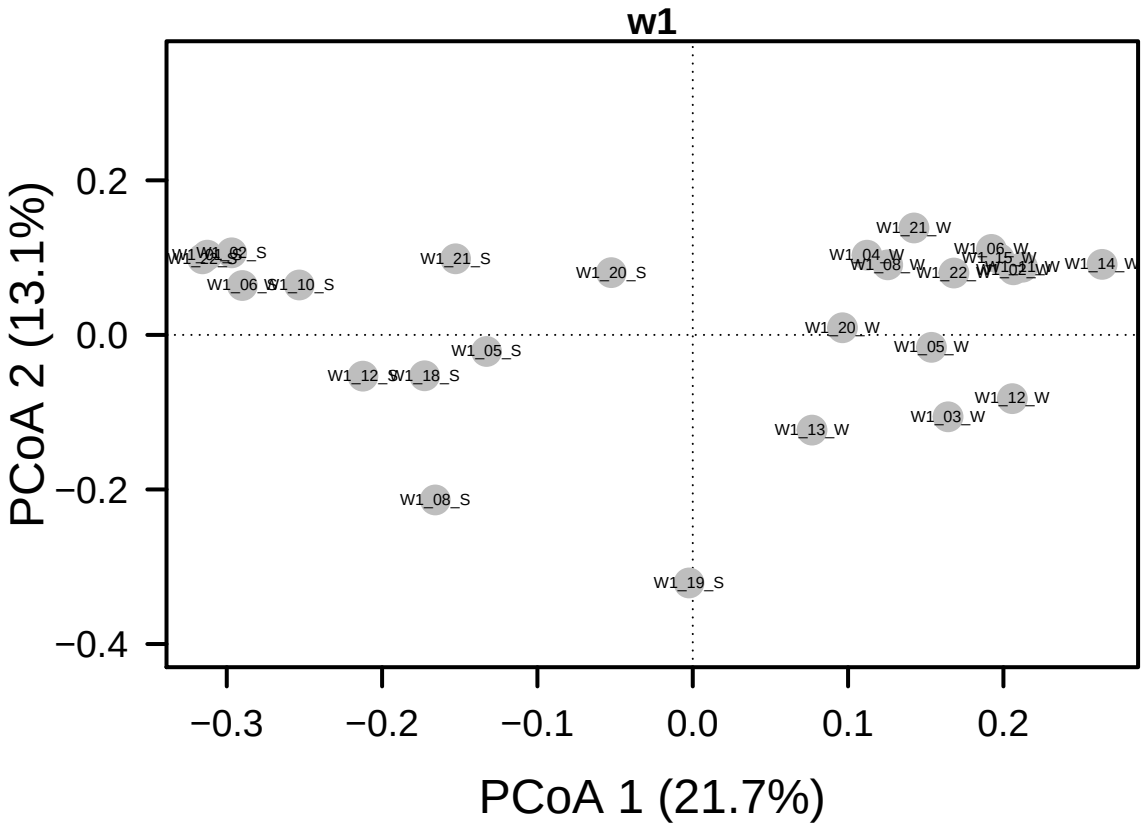
LC

```
lc <- OTUsREL[which(design$watershed == "LC"),]
lc.db <- vegdist(lc, method = "bray", diag = T)
lc.pcoa <- cmdscale(lc.db, eig=TRUE, k=3)
lc.var1 <- round(lc.pcoa$eig[1] / sum(lc.pcoa$eig),3) * 100
lc.var2 <- round(lc.pcoa$eig[2] / sum(lc.pcoa$eig),3) * 100
lc.var3 <- round(lc.pcoa$eig[3] / sum(lc.pcoa$eig),3) * 100

par(mar = c(5, 5, 1, 2) + 0.1)
plot(lc.pcoa$points[ ,1], lc.pcoa$points[ ,2], ylim = c(-0.4, 0.35),
     xlab = paste("PCoA 1 (", lc.var1, "%)", sep = ""),
     ylab = paste("PCoA 2 (", lc.var2, "%)", sep = ""),
     pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = F,
     main="lc")
axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
abline(h = 0, v = 0, lty = 3)
box(lwd = 2)
points(lc.pcoa$points[ ,1], lc.pcoa$points[ ,2],
       pch=19, cex=2, bg="gray", col="gray")
text(lc.pcoa$points[ ,1], lc.pcoa$points[ ,2],
     cex=0.5, labels = row.names(lc.pcoa$points))
```



WS01

```
w1 <- OTUsREL[which(design$watershed == "WS01"),]
w1.db <- vegdist(w1, method = "bray", diag = T)
w1.pcoa <- cmdscale(w1.db, eig=TRUE, k=3)
w1.var1 <- round(w1.pcoa$eig[1] / sum(w1.pcoa$eig),3) * 100
w1.var2 <- round(w1.pcoa$eig[2] / sum(w1.pcoa$eig),3) * 100
w1.var3 <- round(w1.pcoa$eig[3] / sum(w1.pcoa$eig),3) * 100

par(mar = c(5, 5, 1, 2) + 0.1)
plot(w1.pcoa$points[ ,1], w1.pcoa$points[ ,2], ylim = c(-0.4, 0.35),
     xlab = paste("PCoA 1 (", w1.var1, "%)", sep = ""),
     ylab = paste("PCoA 2 (", w1.var2, "%)", sep = ""),
     pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = F,
     main="w1")
axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
abline(h = 0, v = 0, lty = 3)
box(lwd = 2)
points(w1.pcoa$points[ ,1], w1.pcoa$points[ ,2],
       pch=19, cex=2, bg="gray", col="gray")
text(w1.pcoa$points[ ,1], w1.pcoa$points[ ,2],
     cex=0.5, labels = row.names(w1.pcoa$points))
```



CCA

```
#env.mat <- as.matrix(env[which(env$habitat == "hja"),10:17])
env.mat <- as.matrix(env[,10:17])
#hja.cca <- vegan::cca(OTUs ~ env.mat)
#anova(hja.cca, by = "axis")
cca.fit <- envfit(hja.cca, env.mat, perm = 999)
cca.explainvar1 <- round(hja.cca$CCA$eig[1] /
                         sum(c(hja.cca$CCA$eig, hja.cca$CA$eig)), 3) * 100
cca.explainvar2 <- round(hja.cca$CCA$eig[2] /
                         sum(c(hja.cca$CCA$eig, hja.cca$CA$eig)), 3) * 100

par(mar = c(5, 5, 4, 4) + 0.1)
plot(scores(hja.cca, display = "wa"), xlim = c(-2, 2.5), ylim = c(-4, 2),
     xlab = paste("CCA 1 (", cca.explainvar1, "%)", sep = ""),
     ylab = paste("CCA 2 (", cca.explainvar2, "%)", sep = ""),
     pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = FALSE)
axis(side=1, labels=T, lwd.ticks=2, cex.axis=1.2, las=1)
axis(side=2, labels=T, lwd.ticks=2, cex.axis=1.2, las=1)
abline(h=0, v=0, lty=3)
box(lwd=2)
points(scores(hja.cca, display = "wa"),
       pch=19, cex=3, bg="gray", col="gray")
text(scores(hja.cca, display = "wa"),
     labels = row.names(scores(hja.cca, display = "wa")))
```