Reservoir Gradient: Microbial Communities

Jay T. Lennon, Megan L. Larsen, Mario E. Muscarella

10 September, 2015

Project looking at microbial composition and processes along a reservoir gradient

Initial Setup

```
rm(list=ls())
setwd("~/GitHub/ReservoirGradient")
source("./bin/MothurTools.R")
```

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

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

```
## Loading required package: vegan
## Loading required package: permute
## Loading required package: lattice
## This is vegan 2.2-1
```

```
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/UL.design.txt"
shared <- "../data/UL.bac.final.shared"
taxon  <- "../data/UL.bac.final.0.03.taxonomy"

# 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")
```

Data Transormations

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

# 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")

# Remove Low Coverage Samples
coverage <- rowSums(OTUs)
```

Calculate Alpha Diversity

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

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

```
##  ULSoil_01  ULSoil_02  ULSoil_03  UL_01_DNA UL_01_cDNA  UL_02_DNA
##  0.9541560  0.9537302  0.9389795  0.7123476  0.9521973  0.7763052
## UL_02_cDNA  UL_03_DNA UL_03_cDNA  UL_04_DNA UL_04_cDNA  UL_05_DNA
##  0.9500715  0.7293705  0.9464016  0.7562412  0.9409259 -0.9708085
## UL_05_cDNA  UL_06_DNA UL_06_cDNA  UL_07_DNA UL_07_cDNA  UL_08_DNA
##  0.8688629  0.9381043 -0.3017887  0.7216556  0.9220278  0.9156897
## UL_08_cDNA  UL_09_DNA UL_09_cDNA  UL_10_DNA UL_10_cDNA  UL_11_DNA
##  0.9475004  0.7773052  0.9121607  0.9363162  0.8771606  0.9381341
## UL_11_cDNA  UL_12_DNA UL_12_cDNA  UL_13_DNA UL_13_cDNA  UL_14_DNA
##  0.9493440  0.9452848  0.9369904  0.9456289  0.9431292  0.9236055
## UL_14_cDNA  UL_15_DNA UL_15_cDNA  UL_16_DNA UL_16_cDNA  UL_17_DNA
##  0.9564206  0.9089748  0.9232507  0.9075687  0.9518381  0.8627176
## UL_17_cDNA  UL_18_DNA UL_18_cDNA
##  0.9315844  0.9318416  0.9592666
```

```
# 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)

# 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)
diversity(OTUs, index = "shannon")
```

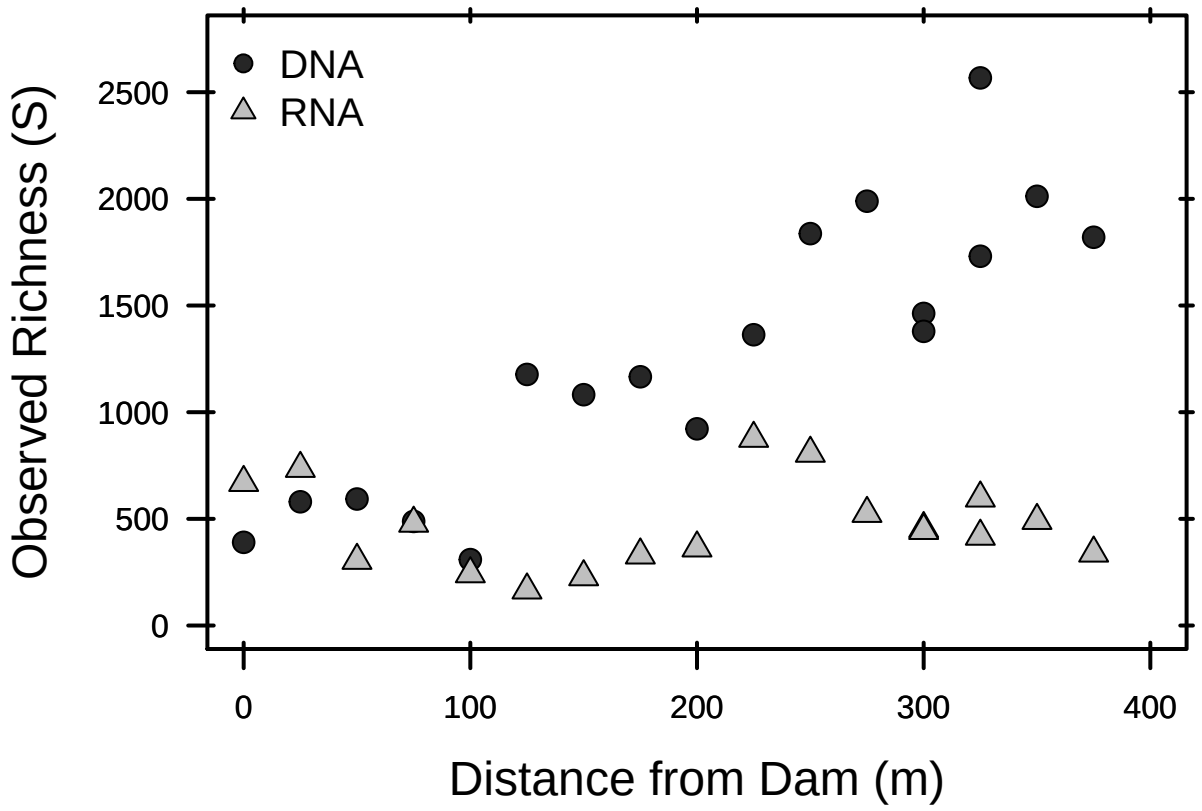
```
##  ULSoil_01  ULSoil_02  ULSoil_03  UL_01_DNA UL_01_cDNA  UL_02_DNA
##  6.5548875  6.4528725  6.5691267  4.0795977  3.1684072  4.2357046
## UL_02_cDNA  UL_03_DNA UL_03_cDNA  UL_04_DNA UL_04_cDNA  UL_05_DNA
##  3.4726269  4.1754273  1.9109298  4.2176218  0.8091232  4.2789561
## UL_05_cDNA  UL_06_DNA UL_06_cDNA  UL_07_DNA UL_07_cDNA  UL_08_DNA
##  1.5800722  4.1328886  1.3139208  4.4590932  1.4832583  4.2888036
## UL_08_cDNA  UL_09_DNA UL_09_cDNA  UL_10_DNA UL_10_cDNA  UL_11_DNA
##  2.8562167  4.6499951  1.6225924  4.7479989  4.3440328  5.0381606
## UL_11_cDNA  UL_12_DNA UL_12_cDNA  UL_13_DNA UL_13_cDNA  UL_14_DNA
##  3.6131264  5.4000309  3.0954874  4.0250790  2.6736182  4.2285382
## UL_14_cDNA  UL_15_DNA UL_15_cDNA  UL_16_DNA UL_16_cDNA  UL_17_DNA
##  3.1389321  5.2472604  3.0578964  3.5985282  0.8868672  4.9131101
## UL_17_cDNA  UL_18_DNA UL_18_cDNA
##  2.3490868  5.5273013  1.5167291
```

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

Alpha Diversity Plots

```
# Seperate data based on lake and soil samples
lake <- alpha.div[-c(1:3),]
soil <- alpha.div[c(1:3), ]

# Richness across Reservoir Gradient
opar <- par()
par(mar = c(5,6, 1, 1))
mol <- rep(NA, length(lake$molecule))
  for (i in 1:length(mol)){
    if (lake$molecule[i] == "DNA"){
      mol[i] <- 21
    } else {
      mol[i] <- 24
```

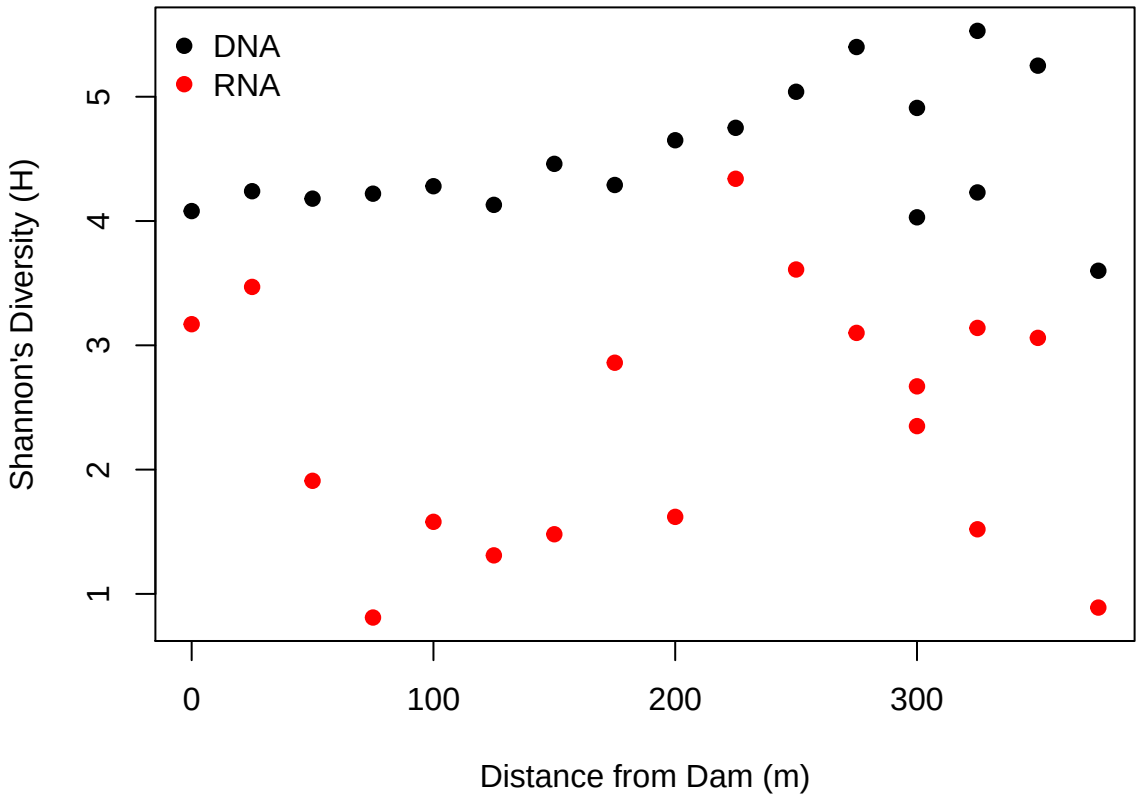
```
    }
  }
cols <- rep(NA, length(lake$molecule))
  for (i in 1:length(cols)){
    if (lake$molecule[i] == "DNA"){
      cols[i] <- "gray15"
    } else {
      cols[i] <- "gray75"
    }
  }

plot(lake$S.obs ~ lake$distance, col= "black", bg = cols, pch=mol, las = 1,
     xlim = c(0,400), ylim = c(0, 2750), cex = 1.5,
     xlab="", ylab="")

axis(side = 1, lwd.ticks = 2, cex.axis = 1, las = 1)
axis(side = 2, lwd.ticks = 2, cex.axis = 1, las = 1)
axis(side = 3, lwd.ticks = 2, tck=-0.01, labels = F,  cex.axis = 2, las = 1)
axis(side = 4, lwd.ticks = 2, tck=-0.01, labels = F,  cex.axis = 2, las = 1)
axis(side = 1, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)
axis(side = 2, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)
axis(side = 3, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)
axis(side = 4, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)

mtext("Distance from Dam (m)" , side = 1, line = 3, cex=1.5)
mtext("Observed Richness (S)" , side = 2, line = 4, cex=1.5)

box(lwd=2)

legend("topleft", legend = levels(lake$molecule), pch=c(21, 24),
       pt.bg = c("gray15", "gray75"), bty='n', cex = 1.25)

abline(h = mean(soil$S.obs), lty=2, col="blue")
```




```
mean(soil$S.obs)
```

```
## [1] 7168.667
```

```
# S
plot(lake$shan ~ lake$distance, col=lake$molecule, pch=19,
     xlab="Distance from Dam (m)", ylab="Shannon's Diversity (H)")
legend("topleft", legend = levels(lake$molecule), col=c(1,2), pch=19, bty='n')
```



Beta Diversity

```
# Calculate Bray-Curtis
UL.db <- vegdist(OTUsREL, method = "bray")

UL.pcoa <- cmdscale(UL.db, eig = TRUE, k = 3)

explainvar1 <- round(UL.pcoa$eig[1] / sum(UL.pcoa$eig), 3) * 100
explainvar2 <- round(UL.pcoa$eig[2] / sum(UL.pcoa$eig), 3) * 100
explainvar3 <- round(UL.pcoa$eig[3] / sum(UL.pcoa$eig), 3) * 100
sum.eig <- sum(explainvar1, explainvar2, explainvar3)

# Define Plot Parameters
par(mar = c(5, 5, 1, 2) + 0.1)

# Initiate Plot
plot(UL.pcoa$points[ ,1], UL.pcoa$points[ ,2], ylim = c(-0.8, 0.8),
     xlab = paste("PCoA 1 (", explainvar1, "%)", sep = ""),
     ylab = paste("PCoA 2 (", explainvar2, "%)", sep = ""),
     pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = FALSE)

# Add Axes
axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
```

```r
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
abline(h = 0, v = 0, lty = 3)
box(lwd = 2)

# Add Points & Labels
points(UL.pcoa$points[ ,1], UL.pcoa$points[ ,2],
       pch = 19, cex = 3, bg = "gray", col = "gray")
text(UL.pcoa$points[ ,1], UL.pcoa$points[ ,2],
     labels = row.names(UL.pcoa$points))

# Add Points & Labels
points(UL.pcoa$points[ ,1], UL.pcoa$points[ ,2],
       pch = 19, cex = 3, bg = "gray", col = design$molecule)
text(UL.pcoa$points[ ,1], UL.pcoa$points[ ,2],
     labels = row.names(UL.pcoa$points))
```