Reservoir Gradient

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

28 September, 2015

Project looking at microbial composition and processes along a reservoir gradient.

Initial Setup

```
rm(list=ls())
getwd()
setwd("~/GitHub/ReservoirGradient/analyses")

# Import Required Packages
require("png")
```

```
## Loading required package: png
```

```
require("ggplot2"); require("ggmap")          # Mapping Tools
```

```
## Loading required package: ggplot2
## Loading required package: ggmap
```

```
require("reshape"); require("plyr")
```

```
## Loading required package: reshape
## Loading required package: plyr
##
## Attaching package: 'plyr'
##
## The following objects are masked from 'package:reshape':
##
##     rename, round_any
```

```
require("rgdal"); require("maptools")
```

```
## Loading required package: rgdal
## Loading required package: sp
## rgdal: version: 1.0-7, (SVN revision 559)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 1.11.2, released 2015/02/10
##  Path to GDAL shared files: C:/Program Files/R/R-3.1.3/library/rgdal/gdal
##  GDAL does not use iconv for recoding strings.
##  Loaded PROJ.4 runtime: Rel. 4.9.1, 04 March 2015, [PJ_VERSION: 491]
##  Path to PROJ.4 shared files: C:/Program Files/R/R-3.1.3/library/rgdal/proj
##  Linking to sp version: 1.2-0
```

```
## Loading required package: maptools
## Checking rgeos availability: FALSE
##      Note: when rgeos is not available, polygon geometry     computations in maptools depend on gpcl
##      which has a restricted licence. It is disabled by default;
##      to enable gpclib, type gpclibPermit()
```

```
require("raster"); require("akima")
```

```
## Loading required package: raster
## Loading required package: akima
```

FIGURE 1: NUTRIENT PATTERNS ACROSS DAM

```
# Load environmental data
env.dat <- read.csv("../data/ResGrad_EnvDat.csv", header = TRUE)
env.dat <- env.dat[-16,]  # Why are we removing this?

# Import University Lake Polygon
ul <- readOGR("../maps","UniversityLake")
summary(ul) # Check projection and and datum
#ul <- spTransform(ul, CRS("+proj=longlat +datum=WGS84")) # transform if necessary
ul <- fortify(ul) # raster image for plotting with ggplot2
```

```
## Regions defined for each Polygons
```

```
# Start Plotting File
png(filename="../figures/Figure1.png",
    width = 1200, height = 1200, res = 96*2)

# Download the map from GoogleMaps API - higher quality image than with RgoogleMaps
theMap <- get_map(location = c(lon = -86.503087, lat = 39.188686),
                  zoom = 17, maptype = "terrain",
                  source = "google", messaging = F, color = "bw")
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=39.188686,-86.503087&zoom=17&siz
```

```
base.map <- ggmap(theMap, extent = "device", legend = "topleft")
print(base.map)

# plot map with polygon and sample points
samples <- geom_point(
  aes(x = long, y = lat),
  size = 3, data = env.dat)

ul.pol <- geom_polygon(
  aes(x = long, y = lat, group = group),
  fill = 'grey', size = 0.5,
  color = 'black', data = ul, alpha=1)

base.map + ul.pol + samples
```

```
# Create the gradient contours for the plot
# Add color?
dev.off() # this writes plot to folder
graphics.off() # shuts down open devices
```

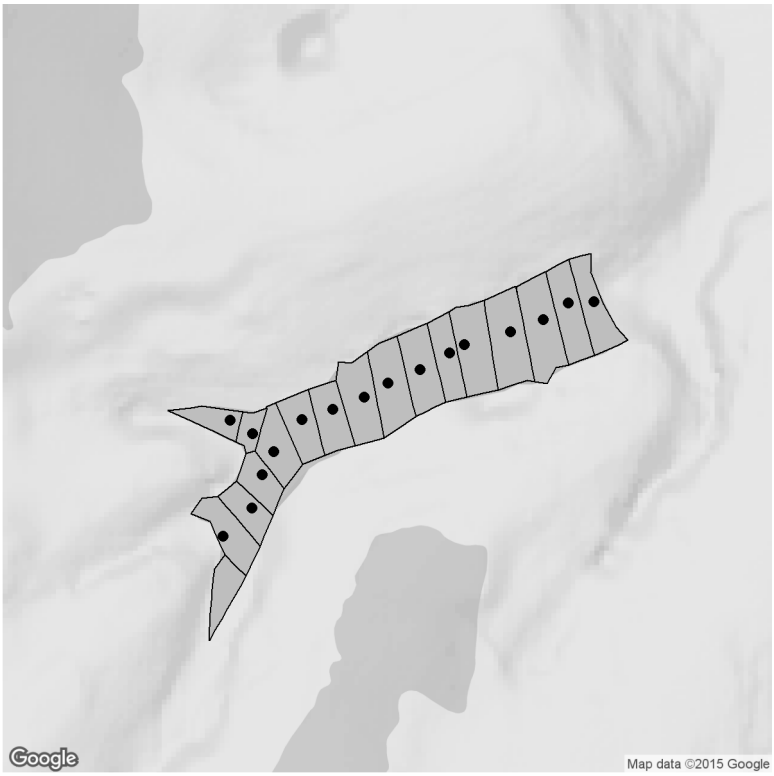

Figure 1: Water Chemistry

FIGURE 2: METABOLISM ALONG GRADIENT

MICROBIAL METABOLISM: BP, BR, BGE

Microbial Functional Groups: Phototroph:Heterotroph

Read in data

```
metab <- read.table("../data/res.grad.metab.txt", sep="\t", header=TRUE)
colnames(metab)[1] <- "dist"
colnames(metab)[2] <- "BP"
colnames(metab)[3] <- "BR"
BGE <- round((metab$BP/(metab$BP + metab$BR)),3)
metab <- cbind(metab, BGE)
```

```
png(filename="../figures/Figure2.png",
    width = 1200, height = 1200, res = 96*2)
```

```r
par(mfrow = c(1,1), mar = c(1, 7, 1, 7), oma = c(5, 4, 0, 0) + 0.5)
bar.layout <- layout(rbind(1, 2, 3), height = c(4, 4, 4))
#layout.show(bar.layout)

# Baterial Producivity (BP)

plot(metab$dist, metab$BP, ylab = "", xlab = "", pch = 22, ylim = c(0, 2), xlim = c(-15, 400),
     cex = 2, bg = "white", col = "black", cex.lab = 2, las = 1, lwd = 2,
     yaxt = "n", xaxt = "n")
box(lwd = 2)

axis(side = 2, lwd.ticks = 2, cex.axis = 1.5, las = 1,
    labels = c("0.0", "1.0", "2.0"), at = c(0, 1.0, 2.0))

axis(side = 4, lwd.ticks = 2, labels = F,  cex.axis = 2, las = 1,
   at = c(0, 1, 2))

# axis(side = 1, lwd.ticks = 2, cex.axis = 1.5, las = 1,
#     labels = c("0", "100", "200", "300", "400"), at = c(0, 100, 200, 300, 400))

axis(side = 1, lwd.ticks = 2, labels = F,  cex.axis = 2, las = 1,
   at = c(0, 100, 200, 300, 400))

axis(side = 3, lwd.ticks = 2, labels = F,  cex.axis = 2, las = 1,
   at = c(0, 100, 200, 300, 400))

mtext(expression(paste('BP (', mu ,'M C h'^-1* ')')), side = 2, line = 4, cex = 1.5)

# Quadratic regression for BP
dist <- metab$dist
dist2 <- metab$dist^2
BP.fit <- lm(metab$BP ~ dist + dist2)
dist.vals <- seq(0, 375, 25)
BP.pred <- predict(BP.fit,list(dist = dist.vals, dist2 = dist.vals^2))
lines(dist.vals, BP.pred, col = "black", lwd = 2.5, lty = 6)
text(40, 1.8, labels = expression(r^2 == "0.40"), cex = 1.5)

# Bacterial Respiration (BR)

plot(metab$dist, metab$BR, ylab = "", xlab = "", pch = 22, ylim = c(0.75, 3.75), xlim = c(-15, 400),
     cex = 2, bg = "white", col = "black", cex.lab = 2, las = 1, lwd = 2,
     yaxt = "n", xaxt = "n")
box(lwd = 2)

axis(side = 2, lwd.ticks = 2, cex.axis = 1.5, las = 1,
    labels = c("1.0", "2.0", "3.0"), at = c(1, 2, 3))

axis(side = 4, lwd.ticks = 2, labels = F,  cex.axis = 2, las = 1,
   at = c(1, 2, 3))

axis(side = 1, lwd.ticks = 2, labels = F,  cex.axis = 2, las = 1,
   at = c(0, 100, 200, 300, 400))
```

```r
axis(side = 3, lwd.ticks = 2, labels = F,  cex.axis = 2, las = 1,
   at = c(0, 100, 200, 300, 400))

mtext(expression(paste('BR (', mu ,'M C h'^-1* ')')), side = 2, line = 4, cex = 1.5)

# Simple linear regression for BR
BR.fit <- lm(metab$BR ~ metab$dist)
BR.int <- BR.fit$coefficients[1]
BR.slp <- BR.fit$coefficients[2]
clip(0, 375, 0, 3.75)
abline(a = BR.int, b = BR.slp, col = "black", lwd = 2.5, lty = 6)
text(40, 3.5, labels = expression(r^2 == 0.75), cex = 1.5)

# Bacterial Growth Efficiency

plot(metab$dist, metab$BGE, ylab = "", xlab = "", pch = 22, ylim = c(0, 0.6), xlim = c(-15, 400),
     cex = 2, bg = "white", col = "black", cex.lab = 2, las = 1, lwd = 2,
     yaxt = "n", xaxt = "n")
box(lwd = 2)

axis(side = 2, lwd.ticks = 2, cex.axis = 1.5, las = 1,
    labels = c("0.0", "0.3", "0.6"), at = c(0, 0.3, 0.6))

axis(side = 4, lwd.ticks = 2, labels = F,  cex.axis = 2, las = 1,
   at = c(0, 0.3, 0.6))

axis(side = 1, lwd.ticks = 2,  cex.axis = 2, las = 1, mgp = c(3, 1.5, 0),
   labels = c("0", "100", "200", "300", "400"), at = c(0, 100, 200, 300, 400))

axis(side = 3, lwd.ticks = 2, labels = F, cex.axis = 2, las = 1,
   at = c(0, 100, 200, 300, 400))

mtext("BGE", side = 2, line = 4, cex = 1.5)
mtext("Distance (m)", side = 1, line = 4, cex = 1.5)

# Simple linear regression for BGE
BGE.fit <- lm(metab$BGE ~ metab$dist)
BGE.int <- BGE.fit$coefficients[1]
BGE.slp <- BGE.fit$coefficients[2]
clip(0, 375, 0, 0.58)
abline(a = BGE.int, b = BGE.slp, col = "black", lwd = 2.5, lty = 6)
text(40, 0.535, labels = expression(r^2 == 0.23), cex = 1.5)

# Phototroph to Heterotroph Ratio



dev.off() # this writes plot to folder
graphics.off() # shuts down open devices
```

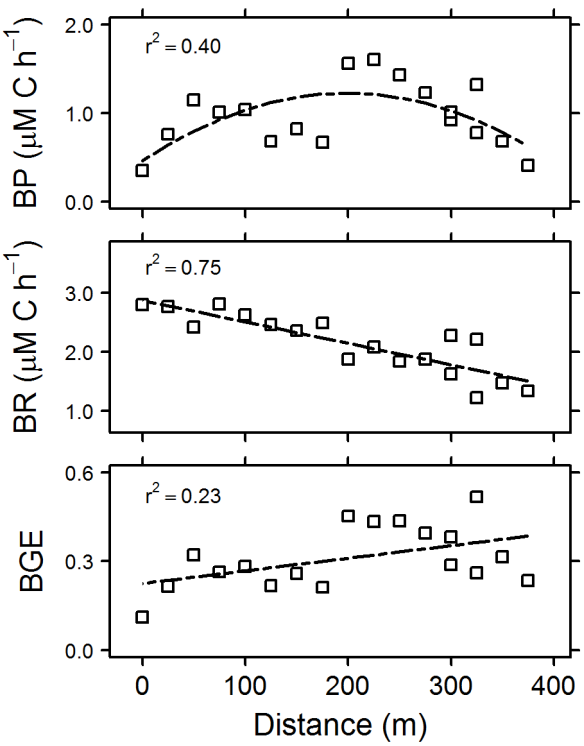

Figure 2: Microbial Processes

FIGURE 3: Shifts in Microbial Metabolism

Crump Model: Mass Effects vs. Species Sorting

Load required R packages and tools

```
source("../bin/MothurTools.R")
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")
```

```
# cyanos
# phytos
# cyan <- "../data/UL."
# cyanos <- read.otu(shared = cyan, cutoff = "0.03")
# photos <- read.otu(shared = photo, cutoff = "0.03")
```

Data Transormations

```
# 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 < 10000)
lows
```

```
##  UL_05_DNA UL_06_cDNA
##         12         15
```

```
OTUs <- OTUs[-which(coverage < 10000), ]
design <- design[-which(coverage < 10000), ]

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

```r
# 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")

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

Alpha Diversity Across Gradient

```r
# Seperate data based on lake and soil samples
lake <- alpha.div[alpha.div$type == "water",]
soil <- alpha.div[alpha.div$type == "soil", ]

# Calculate Linear Model
model.rich <- lm(lake$S.obs ~ lake$distance * lake$molecule)
summary(model.rich)
```

```
##
## Call:
## lm(formula = lake$S.obs ~ lake$distance * lake$molecule)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -394.78 -164.65  -18.63  122.24  722.23
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   368.9165   125.7785   2.933  0.00637 **
## lake$distance                   4.4396     0.5291   8.390 2.30e-09 ***
## lake$moleculeRNA              113.2278   176.7261   0.641  0.52658
## lake$distance:lake$moleculeRNA  -4.4788     0.7445  -6.016 1.33e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 253.6 on 30 degrees of freedom
## Multiple R-squared:  0.8407, Adjusted R-squared:  0.8247
## F-statistic: 52.76 on 3 and 30 DF,  p-value: 4.472e-12
```

```
# Calculate Confidance Intervals of Model
newdata.rich <- data.frame(cbind(lake$molecule, lake$distance))
conf95.rich <- predict(model.rich, newdata.rich, interval="confidence")

# Average Richess in Terrestrial Habitat
mean(soil$S.obs)
```

```
## [1] 7158.667
```

Similarity To Terrestrial Habitat Across Gradient

```
# Similarity to Soil Sample
UL.bray      <- 1 - as.matrix(vegdist(OTUsREL.log, method="bray"))
UL.bray.lake <- UL.bray[-c(1:3), 1:3]
bray.mean    <- round(apply(UL.bray.lake, 1, mean), 3)
bray.se      <- round(apply(UL.bray.lake, 1, se), 3)
UL.sim       <- cbind(design[-c(1:3), ], bray.mean, bray.se)

# Calculate Linear Model
model.terr <- lm(UL.sim$bray.mean ~ UL.sim$distance * UL.sim$molecule)
summary(model.terr)
```

```
##
## Call:
## lm(formula = UL.sim$bray.mean ~ UL.sim$distance * UL.sim$molecule)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.051051 -0.012638 -0.002573  0.008963  0.091666
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       1.567e-02  1.461e-02   1.073 0.291795
## UL.sim$distance                   4.143e-04  6.144e-05   6.743 1.78e-07
```

```
## UL.sim$moleculeRNA                   1.127e-02  2.052e-02   0.549 0.586965
## UL.sim$distance:UL.sim$moleculeRNA -3.855e-04  8.646e-05  -4.459 0.000107
##
## (Intercept)
## UL.sim$distance                     ***
## UL.sim$moleculeRNA
## UL.sim$distance:UL.sim$moleculeRNA ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02945 on 30 degrees of freedom
## Multiple R-squared:  0.754,  Adjusted R-squared:  0.7294
## F-statistic: 30.65 on 3 and 30 DF,  p-value: 2.868e-09
```

```
# Calculate Confidance Intervals of Model
newdata.terr <- data.frame(cbind(UL.sim$molecule, UL.sim$distance))
conf95.terr <- predict(model.terr, newdata.terr, interval="confidence")
```

Similarity To Lake Habitat Across Gradient

```
# Similarity to Lake Sample 1
UL.bray2      <- 1 - as.matrix(vegdist(OTUsREL.log, method="bray"))
UL.bray.lake2 <- UL.bray[-c(1:3), 4:7]
UL.sim2       <- cbind(design[-c(1:3), ], "DNA" = apply(UL.bray.lake2[,c(1,3)], 1, mean),
                                          "RNA" = apply(UL.bray.lake2[,c(2,4)], 1, mean))

# Calculate Linear Model
model.lake1 <- lm(UL.sim2$DNA ~ UL.sim2$distance * UL.sim2$molecule)
model.lake2 <- lm(UL.sim2$RNA ~ UL.sim2$distance * UL.sim2$molecule)
summary(model.lake1)
```

```
##
## Call:
## lm(formula = UL.sim2$DNA ~ UL.sim2$distance * UL.sim2$molecule)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.212825 -0.075949 -0.006199  0.054511  0.254650
##
## Coefficients:
##                                     Estimate Std. Error t value
## (Intercept)                         0.7804831  0.0493547  15.814
## UL.sim2$distance                   -0.0015905  0.0002076  -7.660
## UL.sim2$moleculeRNA                -0.4639770  0.0693462  -6.691
## UL.sim2$distance:UL.sim2$moleculeRNA  0.0014089  0.0002921   4.823
##                                     Pr(>|t|)
## (Intercept)                         4.27e-16 ***
## UL.sim2$distance                    1.52e-08 ***
## UL.sim2$moleculeRNA                 2.06e-07 ***
## UL.sim2$distance:UL.sim2$moleculeRNA 3.84e-05 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09951 on 30 degrees of freedom
## Multiple R-squared:  0.7385, Adjusted R-squared:  0.7124
## F-statistic: 28.24 on 3 and 30 DF,  p-value: 7.107e-09
```

```
summary(model.lake2)
```

```
##
## Call:
## lm(formula = UL.sim2$RNA ~ UL.sim2$distance * UL.sim2$molecule)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.278785 -0.037188  0.002748  0.040844  0.290619
##
## Coefficients:
##                                       Estimate Std. Error t value
## (Intercept)                           4.249e-01  5.839e-02   7.276
## UL.sim2$distance                     -7.120e-04  2.456e-04  -2.898
## UL.sim2$moleculeRNA                   1.850e-02  8.205e-02   0.226
## UL.sim2$distance:UL.sim2$moleculeRNA -3.571e-05  3.457e-04  -0.103
##                                      Pr(>|t|)
## (Intercept)                          4.22e-08 ***
## UL.sim2$distance                      0.00695 **
## UL.sim2$moleculeRNA                   0.82311
## UL.sim2$distance:UL.sim2$moleculeRNA  0.91840
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1177 on 30 degrees of freedom
## Multiple R-squared:  0.3743, Adjusted R-squared:  0.3117
## F-statistic: 5.982 on 3 and 30 DF,  p-value: 0.002539
```

```
# Calculate Confidance Intervals of Model
newdata.lake <- data.frame(cbind(UL.sim2$molecule, UL.sim2$distance))
conf95.lake <- predict(model.lake1, newdata.lake, interval="confidence")
```

Figure 3 Plot

```
# Define Plot Parameters
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] <- 22
    } 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"
    }
  }

# Initial Plot
png(filename="../figures/Figure3.png",
    width = 1200, height =1200, res = 96*2)

par(mfrow = c(1,1), mar = c(1, 7, 1, 7), oma = c(5, 4, 0, 0) + 0.5)
bar.layout <- layout(rbind(1, 2, 3), height = c(4, 4, 4))

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

  matlines(lake$distance[lake$molecule == "DNA"], conf95.rich[lake$molecule == "DNA", ],
         lty = c(1, 0, 0), col=c("black", "gray50", "gray50"), lwd=c(2, 1, 1))
  matlines(lake$distance[lake$molecule == "RNA"], conf95.rich[lake$molecule == "RNA", ],
         lty = c(1, 0, 0), col=c("black", "gray50", "gray50"), lwd=c(2, 1, 1))

  axis(side = 1, lwd.ticks = 2, labels = F, 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 (m)" , side = 1, line = 3, cex=1.5)
  mtext("Richness\n(S)" , side = 2, line = 4, cex=1.5)

  legend("topright", legend = levels(lake$molecule), pch=c(22, 24),
       pt.bg = c("gray15", "gray75"), bty='n', cex = 1.25)

  box(lwd=2)

# Terrestrial Influence Plot
#   mol <- rep(NA, length(UL.sim$molecule))
#     for (i in 1:length(mol)){
#       if (UL.sim$molecule[i] == "DNA"){
#         mol[i] <- 21
#       } else {
#         mol[i] <- 24
#       }
#     }
#   cols <- rep(NA, length(UL.sim$molecule))
```

```
#     for (i in 1:length(cols)){
#       if (UL.sim$molecule[i] == "DNA"){
#         cols[i] <- "gray15"
#       } else {
#         cols[i] <- "gray75"
#       }
#     }

plot(UL.sim$bray.mean ~ UL.sim$distance, col= "black", bg = cols, pch=mol, las = 1,
     xlim = c(400, 0), ylim = c(0, 0.25), cex = 1.5,
     xlab="", ylab="", xaxt="n")

  matlines(lake$distance[lake$molecule == "DNA"], conf95.terr[lake$molecule == "DNA", ],
         lty = c(1, 0, 0), col=c("black", "gray50", "gray50"), lwd=c(2, 1, 1))
  matlines(lake$distance[lake$molecule == "RNA"], conf95.terr[lake$molecule == "RNA", ],
         lty = c(1, 0, 0), col=c("black", "gray50", "gray50"), lwd=c(2, 1, 1))

  axis(side = 1, lwd.ticks = 2, labels = F, cex.axis = 1, las = 1)
  axis(side = 2, lwd.ticks = 2, cex.axis = 1, las = 1)
  axis(side = 3, lwd.ticks = 2, tck=-0.05, 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 (m)" , side = 1, line = 3, cex=1.5)
  mtext("Terrestrial\nInfluence" , side = 2, line = 4, cex=1.5)

  legend("topright", legend = levels(UL.sim$molecule), pch=c(22, 24),
       pt.bg = c("gray15", "gray75"), bty='n', cex = 1.25)

  box(lwd=2)

# Lake Influence Plot
plot(UL.sim2$DNA ~ UL.sim$distance, col= "black", bg = cols, pch=mol, las = 1,
     xlim = c(400, 0), ylim = c(0, 1), cex = 1.5,
     xlab="", ylab="")

  matlines(lake$distance[lake$molecule == "DNA"], conf95.lake[lake$molecule == "DNA", ],
         lty = c(1, 0, 0), col=c("black", "gray50", "gray50"), lwd=c(2, 1, 1))
  matlines(lake$distance[lake$molecule == "RNA"], conf95.lake[lake$molecule == "RNA", ],
         lty = c(1, 0, 0), col=c("black", "gray50", "gray50"), lwd=c(2, 1, 1))

  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=-0.05, 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 (m)" , side = 1, line = 3, cex=1.5)
  mtext("Lake\nInfluence", side = 2, line = 4, cex=1.5)

  legend("topleft", legend = levels(UL.sim$molecule), pch=c(22, 24),
       pt.bg = c("gray15", "gray75"), bty='n', cex = 1.25)

  box(lwd=2)



# Close Plot Defice
dev.off()
```

```
## pdf
##   2
```

```
graphics.off()
```

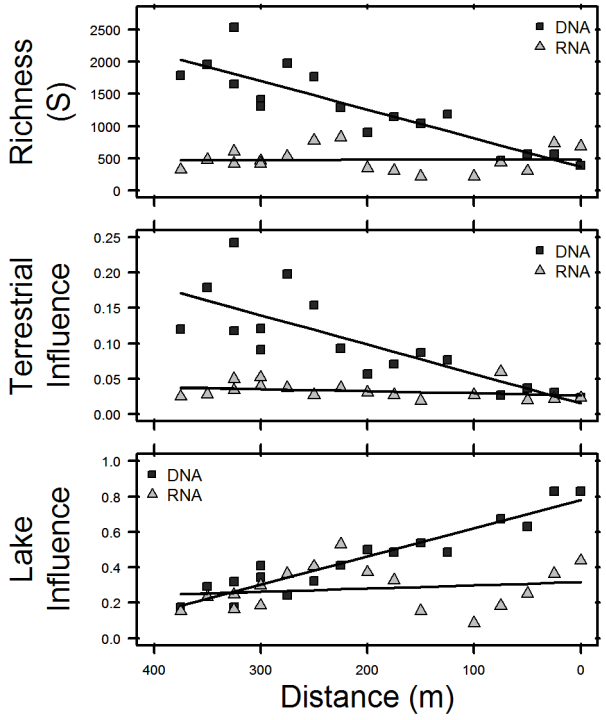

Figure 3: Microbial Commuinity Shifts