Reservoir Gradient

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

28 October, 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")
require("ggplot2")
require("maps")
require("rgdal")
```

```
## Warning: package 'rgdal' was built under R version 3.1.3
## Warning: package 'sp' was built under R version 3.1.3
```

```
require("raster")
require("OpenMx")
```

```
## Warning: package 'OpenMx' was built under R version 3.1.3
## Warning: package 'MASS' was built under R version 3.1.3
```

```
require("reshape")
require("ggmap")
```

```
## Warning: package 'ggmap' was built under R version 3.1.3
```

```
require("grid")
require("akima")
```

```
## Warning: package 'akima' was built under R version 3.1.3
```

```
require("plyr")
require("raster")
require("gridExtra")
```

```
## Warning: package 'gridExtra' was built under R version 3.1.3
```

FIGURE 1: NUTRIENT PATTERNS ACROSS DAM

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

```
# ggplot theme
theme_maps <- function(base_size = 12, base_family = "Arial"){
  theme_bw(base_size = base_size, base_family = base_family) %+replace%
    theme(panel.background = element_rect(fill = "white", color = "black", size = 1.5),
          #panel.border = element_rect(color = "black"),
        #panel.margin = unit(1,1,1,1),
        panel.grid.major = element_line(colour = "white"),
        panel.grid.minor = element_line(colour = "white"),
        axis.ticks = element_line(color = "black", size = 1),
        axis.text = element_text(size = 15),
        axis.title = element_text(size = 20, face = "bold"),
        legend.position = c(0.9,0.25)
        #,axis.title.x = element_blank(),axis.title.y = element_blank()
    )
}

# get shape files
## for map inset
usa <- map_data("usa")
IN <- map_data("state", region = "Indiana")

## for main plot
ul <- readOGR("../maps","UniversityLakePoly")
#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
```

```
# If using google map as baselayer
#ul.coords <- c(lon = -86.503087, lat = 39.188686)
#ul.map <- get_map(location = ul.coords,
#                  zoom = 17, #maptype = "terrain",
#                  source = "google", color = "bw")
#base.map <- ggmap(ul.map, extent = "device", legend = "topleft")



# Main Map
main.map <- ggplot(aes(long,lat), data = env.dat) +
  geom_polygon(fill = "grey", size = 1, color = 'black', data = ul, alpha = 1) +
  geom_point(size = 6, shape = 20) +
  theme_maps()  +
  labs(x = "\nLatitude", y = "Longitude\n")

print(main.map)

# Inset Map
```

```
inset <- ggplot() +
  theme_maps() +
  theme(axis.text = element_blank(),
        axis.ticks = element_blank(),
        panel.border = element_rect(color = NULL)) +
  labs(x = NULL, y = NULL)

inset <- inset + geom_map(data = usa, map = usa,
                    aes(x = long, y = lat, map_id = region),
                    color = "black", fill = "#e7e7e7", size = 0.15)

inset <- inset + geom_map(data = IN, map = IN,
                    aes(x = long, y=lat, map_id = region),
                    color = "black", fill = "black", size = 0.15)

inset <- inset + geom_point(aes(x = -86.503087, y = 39.188686), color = "red", size = 2)

print(inset)


# Start Plotting File
png(filename="../figures/Figure1.png",
    width = 1400, height = 1200, res = 96*2)

grid.newpage()
v1 <- viewport(width = 1, height = 1, x = 0.5, y = 0.5) #plot area for the main map
v2 <- viewport(width = 0.4, height = 0.3, x = 0.765, y = 0.31) #plot area for the inset map
print(main.map,vp = v1)
print(inset,vp = v2)

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

FIGURE S1: CHEMICAL PROPERTIES

```
# Start Plotting File
png(filename="../figures/FigureS1.png",
    width = 1200, height = 1200, res = 96*2)

par(mfrow = c(2,2), oma = c(0,2,0,2)+0.5)
    #, oma = c(5, 4, 0, 0) + 0.5)
#par(mfrow = c(1,1), mar = c(1, 1, 1, 7), oma = c(5, 4, 0, 0) + 0.5)

# Total Phosphorus
TP <- plot(env.dat$dist.dam, env.dat$TP,
     ylab = "", xlab = "", cex.lab = 2, las = 1,
     ylim = c(0,140), xlim = c(-15, 400),
     pch = 22, cex = 2, bg = "white", col = "black", lwd = 2,
     yaxt = "n", xaxt = "n")
box(lwd = 2)
```

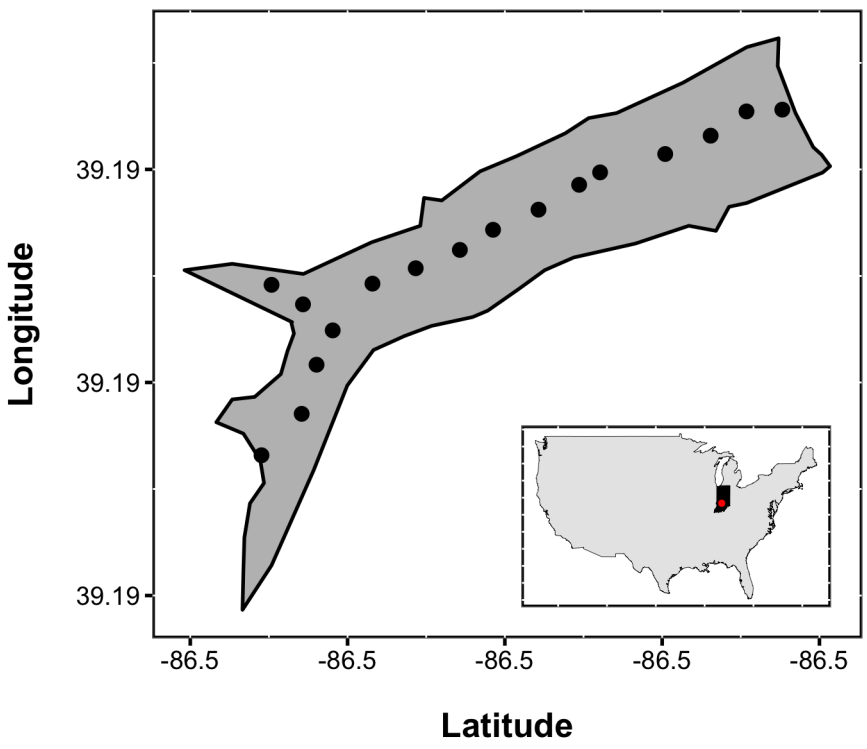

Figure 1: University Lake Map

```
axis(side = 2, lwd.ticks = 2, cex.axis = 1.5, las = 1,
    labels = c("0", "40", "80", "120"), at = c(0, 40, 80, 120))

axis(side = 1, lwd.ticks = 2,  labels = F, 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))


mtext(expression(paste('Total Phosphorus (mg P L'^-1*')')), side = 2, line = 4, cex = 1)

# Chlorophyll
chla <- plot(env.dat$dist.dam, env.dat$chla,
     ylab = "", xlab = "", cex.lab = 2, las = 1,
     ylim = c(0,30), xlim = c(-15, 400),
     pch = 22, cex = 2, bg = "white", col = "black", lwd = 2,
     yaxt = "n", xaxt = "n")
box(lwd = 2)

axis(side = 2, lwd.ticks = 2, cex.axis = 1.5, las = 1,
    labels = c("0", "10", "20", "30"), at = c(0, 10, 20, 30))

axis(side = 1, lwd.ticks = 2, labels = F, 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))

mtext(expression(paste('Chlorophyll A (mg L'^-1*')')), side = 2, line = 4, cex = 1)
```

```
#Dissolved Oxygen
plot(env.dat$dist.dam, env.dat$chla,
     ylab = "", xlab = "", cex.lab = 2, las = 1,
     ylim = c(0,30), xlim = c(-15, 400),
     pch = 22, cex = 2, bg = "white", col = "black", lwd = 2,
     yaxt = "n", xaxt = "n")
box(lwd = 2)

axis(side = 2, lwd.ticks = 2, cex.axis = 1.5, las = 1,
    labels = c("0", "10", "20", "30"), at = c(0, 10, 20, 30))

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

mtext(expression(paste('Dissolved Oxygen (mg L'^-1*')')), side = 2, line = 4, cex = 1)

#pH
plot(env.dat$dist.dam, env.dat$chla,
     ylab = "", xlab = "", cex.lab = 2, las = 1,
     ylim = c(0,30), xlim = c(-15, 400),
     pch = 22, cex = 2, bg = "white", col = "black", lwd = 2,
     yaxt = "n", xaxt = "n")
box(lwd = 2)

axis(side = 2, lwd.ticks = 2, cex.axis = 1.5, las = 1,
    labels = c("0", "10", "20", "30"), at = c(0, 10, 20, 30))

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

mtext("pH", side = 2, line = 4, cex = 1)

dev.off() # this writes plot to folder
```

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

```
graphics.off() # shuts down open devices
```

FIGURE 2: METABOLISM ALONG GRADIENT

MICROBIAL METABOLISM: BP, BR, BGE

Microbial Functional Groups: Phototroph:Heterotroph

Read in data

```r
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)
```

```r
png(filename="../figures/Figure2.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))
#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))

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
```

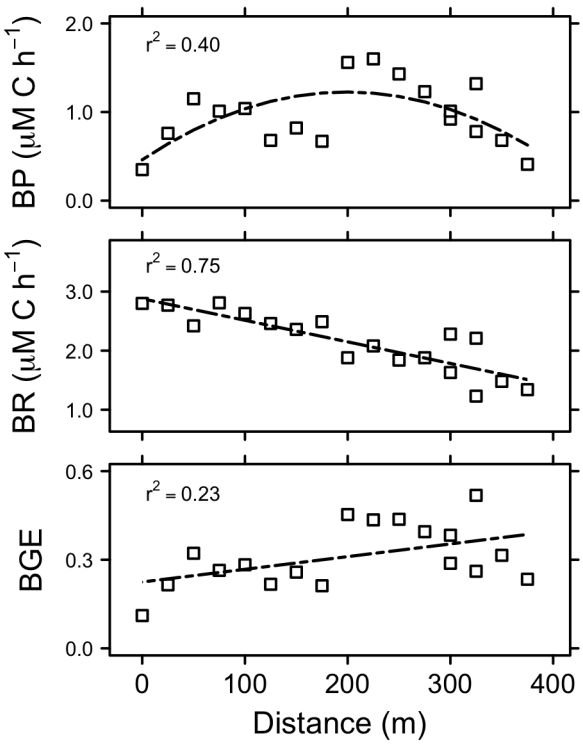

Figure 2: Microbial Processes

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 occurences 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,])
}

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

Calculate Alpha Diversity

```
# 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

```
# 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$molecule)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -394.8 -164.6  -18.6  122.2  722.2
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     368.916    125.779    2.93   0.0064 **
## lake$distance                     4.440      0.529    8.39  2.3e-09 ***
## lake$moleculeRNA                113.228    176.726    0.64   0.5266
## lake$distance:lake$moleculeRNA   -4.479      0.745   -6.02  1.3e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 254 on 30 degrees of freedom
## Multiple R-squared:  0.841,  Adjusted R-squared:  0.825
## F-statistic: 52.8 on 3 and 30 DF,  p-value: 4.47e-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] 7159
```

```
# Dummy Variables Regression Model ("Species Richness"")
D1 <- (lake$molecule == "RNA")*1
fit.Fig.3a <- lm(lake$S.obs ~ lake$distance + D1 + lake$distance*D1)
summary(fit.Fig.3a)
```

```
##
## Call:
## lm(formula = lake$S.obs ~ lake$distance + D1 + lake$distance *
##     D1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -394.8 -164.6  -18.6  122.2  722.2
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       368.916    125.779    2.93   0.0064 **
## lake$distance       4.440      0.529    8.39  2.3e-09 ***
## D1                113.228    176.726    0.64   0.5266
## lake$distance:D1   -4.479      0.745   -6.02  1.3e-06 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 254 on 30 degrees of freedom
## Multiple R-squared:  0.841,  Adjusted R-squared:  0.825
## F-statistic: 52.8 on 3 and 30 DF,  p-value: 4.47e-12
```

```
DNA.int.3a <- fit.Fig.3a$coefficients[1]
DNA.slp.3a <- fit.Fig.3a$coefficients[2]
RNA.int.3a <- DNA.int.3a + fit.Fig.3a$coefficients[3]
RNA.slp.3a <- DNA.slp.3a + fit.Fig.3a$coefficients[4]
```

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$molecule)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.05105 -0.01264 -0.00257  0.00896  0.09167
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        1.57e-02   1.46e-02    1.07  0.29180
## UL.sim$distance                    4.14e-04   6.14e-05    6.74  1.8e-07
## UL.sim$moleculeRNA                 1.13e-02   2.05e-02    0.55  0.58696
## UL.sim$distance:UL.sim$moleculeRNA -3.85e-04   8.65e-05   -4.46  0.00011
##
## (Intercept)
## UL.sim$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: 0.0294 on 30 degrees of freedom
## Multiple R-squared:  0.754,  Adjusted R-squared:  0.729
## F-statistic: 30.7 on 3 and 30 DF,  p-value: 2.87e-09
```

```r
# 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")

# Dummy Variables Regression Model ("Terrestrial Influence")
D2 <- (UL.sim$molecule == "RNA")*1
fit.Fig.3b <- lm(UL.sim$bray.mean ~ UL.sim$distance + D2 + UL.sim$distance*D2)
summary(fit.Fig.3b)
```

```
##
## Call:
## lm(formula = UL.sim$bray.mean ~ UL.sim$distance + D2 + UL.sim$distance *
##     D2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.05105 -0.01264 -0.00257  0.00896  0.09167
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.57e-02   1.46e-02    1.07  0.29180
## UL.sim$distance     4.14e-04   6.14e-05    6.74  1.8e-07 ***
## D2                  1.13e-02   2.05e-02    0.55  0.58696
## UL.sim$distance:D2 -3.85e-04   8.65e-05   -4.46  0.00011 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0294 on 30 degrees of freedom
## Multiple R-squared:  0.754,  Adjusted R-squared:  0.729
## F-statistic: 30.7 on 3 and 30 DF,  p-value: 2.87e-09
```

```r
DNA.int.3b <- fit.Fig.3b$coefficients[1]
DNA.slp.3b <- fit.Fig.3b$coefficients[2]
RNA.int.3b <- DNA.int.3b + fit.Fig.3b$coefficients[3]
RNA.slp.3b <- DNA.slp.3b + fit.Fig.3b$coefficients[4]
```

Similarity To Lake Habitat Across Gradient

```r
# 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$molecule)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.2128 -0.0760 -0.0062  0.0545  0.2546
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         0.780483   0.049355   15.81  4.3e-16
## UL.sim2$distance                   -0.001590   0.000208   -7.66  1.5e-08
## UL.sim2$moleculeRNA                -0.463977   0.069346   -6.69  2.1e-07
## UL.sim2$distance:UL.sim2$moleculeRNA  0.001409   0.000292    4.82  3.8e-05
##
## (Intercept)                        ***
## UL.sim2$distance                   ***
## UL.sim2$moleculeRNA                ***
## UL.sim2$distance:UL.sim2$moleculeRNA ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0995 on 30 degrees of freedom
## Multiple R-squared:  0.739,  Adjusted R-squared:  0.712
## F-statistic: 28.2 on 3 and 30 DF,  p-value: 7.11e-09
```

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

```
##
## Call:
## lm(formula = UL.sim2$RNA ~ UL.sim2$distance * UL.sim2$molecule)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27878 -0.03719  0.00275  0.04084  0.29062
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         4.25e-01   5.84e-02    7.28  4.2e-08
## UL.sim2$distance                   -7.12e-04   2.46e-04   -2.90   0.0069
## UL.sim2$moleculeRNA                 1.85e-02   8.20e-02    0.23   0.8231
## UL.sim2$distance:UL.sim2$moleculeRNA -3.57e-05   3.46e-04   -0.10   0.9184
##
## (Intercept)                        ***
## UL.sim2$distance                   **
## UL.sim2$moleculeRNA
## UL.sim2$distance:UL.sim2$moleculeRNA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.118 on 30 degrees of freedom
## Multiple R-squared:  0.374,  Adjusted R-squared:  0.312
## F-statistic: 5.98 on 3 and 30 DF,  p-value: 0.00254
```

```r
# 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")

# Dummy Variables Regression Model ("Lake Influence")
D3 <- (UL.sim2$molecule == "RNA")*1
fit.Fig.3c <- lm(UL.sim2$DNA ~ UL.sim2$distance + D3 + UL.sim2$distance*D3)
summary(fit.Fig.3c)
```

```
##
## Call:
## lm(formula = UL.sim2$DNA ~ UL.sim2$distance + D3 + UL.sim2$distance *
##     D3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.2128 -0.0760 -0.0062  0.0545  0.2546
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.780483   0.049355   15.81  4.3e-16 ***
## UL.sim2$distance     -0.001590   0.000208   -7.66  1.5e-08 ***
## D3                   -0.463977   0.069346   -6.69  2.1e-07 ***
## UL.sim2$distance:D3   0.001409   0.000292    4.82  3.8e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0995 on 30 degrees of freedom
## Multiple R-squared:  0.739,  Adjusted R-squared:  0.712
## F-statistic: 28.2 on 3 and 30 DF,  p-value: 7.11e-09
```

```r
DNA.int.3c <- fit.Fig.3c$coefficients[1]
DNA.slp.3c <- fit.Fig.3c$coefficients[2]
RNA.int.3c <- DNA.int.3c + fit.Fig.3c$coefficients[3]
RNA.slp.3c <- DNA.slp.3c + fit.Fig.3c$coefficients[4]
```

Figure 3 Plot

```r
# 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))

# Add multiple regression lines
clip(400, 0, 0, 2750)
abline(a = DNA.int.3a, b = DNA.slp.3a, col = "black", lwd = 2.5, lty = 6)

clip(400, 0, 0, 2750)
abline(a = RNA.int.3a, b = RNA.slp.3a, col = "black", lwd = 2.5, lty = 4)

  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))

# Add multiple regression lines
clip(400, 0, 0, 0.27)
abline(a = DNA.int.3b, b = DNA.slp.3b, col = "black", lwd = 2.5, lty = 6)

clip(400, 0, 0, 0.27)
abline(a = RNA.int.3b, b = RNA.slp.3b, col = "black", lwd = 2.5, lty = 4)

  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$DNA ~ UL.sim2$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))

# Add multiple regression lines
clip(400, 0, 0, 1)
abline(a = DNA.int.3c, b = DNA.slp.3c, col = "black", lwd = 2.5, lty = 6)

clip(400, 0, 0, 1)
abline(a = RNA.int.3c, b = RNA.slp.3c, col = "black", lwd = 2.5, lty = 4)

  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()
```

Identifying the Soil Bacteria

```
soil.only <- OTUs[, which(colSums(OTUs[-c(1:3),]) == 0)]
lake.n.soil <- OTUs[, setdiff(colnames(OTUs),colnames(soil.only))]

w.dna <- OTUs[which(design$molecule == "DNA" & design$type == "water"), ]
w.rna <- OTUs[which(design$molecule == "RNA" & design$type == "water"), ]

nvr.act <- which(colSums(w.rna) == 0)

terr.lake <- w.dna[ , c(names(nvr.act))]
```

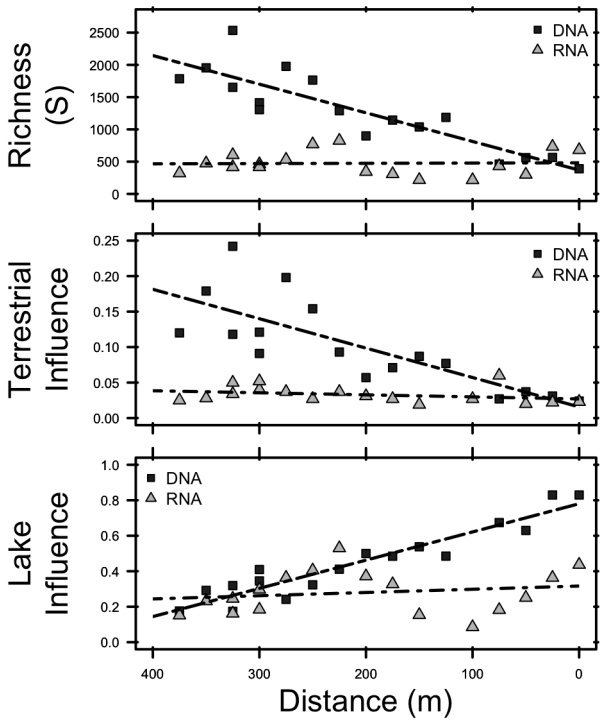

Figure 3: Microbial Commuinity Shifts

```
terr.rich <- rowSums((terr.lake > 0) * 1)

terr.REL <- rowSums(terr.lake) / rowSums(w.dna)

design.dna <- design[which(design$molecule == "DNA" & design$type == "water"), ]
```

Soil Organisms Plot

```
# Initial Plot
png(filename="../figures/Figure4.png",
    width = 1200, height =1200, res = 96*2)

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

# Soil OTU Richness Across Gradient Plot
plot(terr.rich ~ design.dna$distance, col= "black", pch=22, las = 1,
     xlim = c(400, 0), ylim = c(0, 1800), cex = 1.5,
     xlab="", ylab="", xaxt="n")

  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("Transient\nRichness\n(S)" , side = 2, line = 4, cex=1.5)

  box(lwd=2)


# Soil OTU Relative Abundance Across Gradient Plot
plot(terr.REL ~ design.dna$distance, col= "black", pch=22, las = 1,
     xlim = c(400, 0), ylim = c(0, 0.18), 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=-0.05, 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("Transient\nRelative\nAbundance", side = 2, line = 4, cex=1.5)

  box(lwd=2)


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

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

```
graphics.off()
```

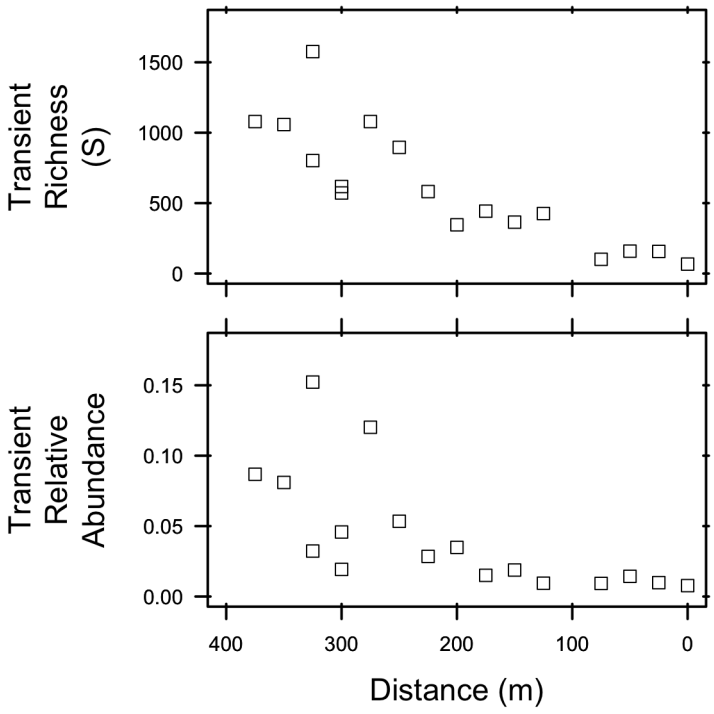

Figure 4: Transient Species Distributions