

Resource Heterogeneity Structures Microbial Communities

Mario E. Muscarella

01 May, 2019

Introduction

Community diversity is strongly affected by the bottom-up effects of resource availability. However, because resource pools often exist as heterogeneous mixtures of individual resources, resource heterogeneity may also affect the diversity of local communities. To test this hypothesis, we surveyed bacterial communities in lakes that spanned a resource concentration gradient. In addition, we characterized resource heterogeneity in these lakes using high-resolution mass spectrometry of the dissolved organic matter (DOM) pool. Using these data, we will test for relationships between the available resources and the aquatic heterotrophic bacteria community, and we will use co-occurrence analysis to test for bacteria-resource interactions.

Initial Setup

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

# Import Tools and Standard Functions
source("../bin/MothurTools.R")
source("../bin/CommonFunctions.R")

# Save Standard Plot Settings
opar <- par(no.readonly = TRUE) # Saves plot defaults

# Load Required Packages
require("png"); require("grid"); require("vegan"); require("igraph")
require("picante"); require("gplots")# ;require("bioDist");
#require("xtable");require("phyloseq");require("car"); require("ade4");require("bioDist")
require("colorspace"); library("car")
source("../bin/box.cox.chord.R")
```

Load Data & Minor Processing

Lake Nutrient Concentrations and Physical Properties

```
nuts <- read.csv(file = "../data/HMWF_Nutrients.txt", header = T)
chl <- read.delim(file = "../data/ChlorophyllA.txt", header = T)
chl <- chl[order(chl$Year, chl$Lake), ]
phys <- read.csv(file = "../data/lake_data2.txt", header = T)
all.equal(nuts$Site, chl$Lake); all.equal(nuts$Year, chl$Year);
```

```
## [1] TRUE
```

```
## [1] TRUE
all.equal(nuts$Site[nuts$Year == 2012], phys$Lake)

## [1] TRUE
```

Figure S1: System Map

```
img <- readPNG("../figures/FigureS1.png")
grid.raster(img)
```



Load DOM Profiles

```
# Define Inputs
# Resource = raw site-by-resource matrix
resource.neg <- "../data/SpecAbundAveNeg.csv"
design.in <- "../data/design.txt"
annotations <- "../data/annotationSummary_MFconversion_output.csv"

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

# Import Resources
res.neg.in <- read.csv(resource.neg, header=T, row.names=1)
rownames(res.neg.in) <- c("Ann", "blank", "CanyonChemo", "Canyon", "CanyonHypo",
                          "CanyonI", "CanyonII", "CanyonIII", "CanyonIV", "Howe",
                          "Ives", "Jordan", "Lily", "Mountain", "Pony", "Rush",
                          "SecondPine", "UpperPine")

# Import Annotations
res.annot <- read.csv(annotations)
```

```

# Remove Blank and Extra Samples
blank.neg <- unlist(res.neg.in["blank", ])
res.hmwf.neg <- res.neg.in[-c(which(rownames(res.neg.in) %in%
                                c("blank", "CanyonChemo",
                                  "CanyonHypo", "CanyonI", "CanyonII",
                                  "CanyonIII", "CanyonIV", "Jordan"))), ]

# Remove Blank Peaks
for (i in 1:dim(res.hmwf.neg)[1]){
  res.hmwf.neg[i, ] <- res.hmwf.neg[i, ] - blank.neg * 1.1
}

# Remove Peaks Under Height of 50
res.hmwf.neg[res.hmwf.neg < 50] <- 0

# Remove Zero Sum Columns
res.hmwf.neg <- res.hmwf.neg[, colSums(res.hmwf.neg) > 0]

# Subset Annotations
missing.annot <- res.annot$Cmpd[which(res.annot$inferred.formula == 0)]
# res.hmwf.neg <- res.hmwf.neg[, -c(which(colnames(res.hmwf.neg) %in% missing.annot))]
res.annot <- res.annot[c(which(res.annot$Cmpd %in% colnames(res.hmwf.neg))), ]

# Data Transformations
# Reorder Sites
res.neg <- res.hmwf.neg[order(rownames(res.hmwf.neg)), ]

# Sequencing Coverage
coverage <- data.frame(Neg = rowSums(res.neg))
resources <- data.frame(Neg = dim(res.neg)[2])

# Make Relative Abundance Matrices
resREL.neg <- res.neg
for(i in 1:dim(res.neg)[1]){
  resREL.neg[i,] <- res.neg[i,]/sum(res.neg[i,])
}

# Log Transform Relative Resource Abundance
resREL.neg.log <- suppressWarnings(decostand(resREL.neg, method="log"))

# Box-Cox Chord Transformation
DOM.BCD <- box.cox.chord(res.neg) #Log Chord Transformation

```

Load DOM Metadata

```

# Read in the data
x <- scan("../data/data_20151106/neg/spectra/MarioAquaticNeg.msplib",
          what="", sep="\n")
ind.n <- grep("Name:", x)
neg.meta <- data.frame(matrix(NA, nrow = length(ind.n), ncol = 3))
colnames(neg.meta) <- c("Name", "Rt", "mz.max")

```

```

for (i in 1:length(ind.n)){
  neg.meta[i, 1] <- unlist(strsplit(x[ind.n[i]], ": "))[2]
  temp.comment <- unlist(strsplit(x[ind.n[i] + 12], " "))
  neg.meta[i, 2] <- as.numeric(gsub("Rt=", "",
                                temp.comment[grepl("Rt=", temp.comment)]))
  temp.peaks <- unlist(strsplit(x[ind.n[i] + 14], " "))
  temp.mz <- as.numeric(temp.peaks[seq(1, length(temp.peaks), by = 2)])
  temp.intensity <- as.numeric(temp.peaks[seq(2, length(temp.peaks), by = 2)])
  neg.meta[i, 3] <- max(temp.mz)
}

neg.meta <- neg.meta[match(colnames(resREL.neg), neg.meta$Name), ]

all.equal(neg.meta$Name, as.character(res.annot$Cmpd))

## [1] TRUE
all.equal(neg.meta$Rt, res.annot$Rt)

## [1] "Mean relative difference: 9.671294e-06"

```

Load Bacterial Community Data

```

# Define Inputs
# Design = general design file for experiment
# shared = OTU table from mothur with sequence similarity clustering
# Taxonomy = Taxonomic information for each OTU
design.in <- "../data/design.txt"
shared <- "../data/HMWF.final.opti.shared"
taxon <- "../data/HMWF.final.opti.taxonomy"

# Import Design
design <- read.delim(design.in, header=T, row.names=1)
design <- design[design$Molecule == "RNA" & design$Year == "2012", ]

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

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

# Remove Cyanobacteria
OTUs.in.2 <- OTUs.in[, -c(which(OTU.tax$Phylum == "Cyanobacteria/Chloroplast"))]
dim(OTUs.in.2)

## [1] 40 23946

OTU.tax.2 <- OTU.tax[which(OTU.tax$OTU %in% colnames(OTUs.in.2)), ]
#table(OTU.tax.2$Class)
#table(OTU.tax.2$Phylum)

# Data Transformations
# Reorder Site

```

```

OTUs.hmwf <- OTUs.in.2[rownames(design), ]

# Remove OTUs with less than two occurrences across all sites
# OTUs <- OTUs.hmwf[, which(colSums(OTUs.hmwf) >= 2)]
OTUs <- OTUs.hmwf[, colSums((OTUs.hmwf > 0) * 1) >= 2 | colSums(OTUs.hmwf) >= 3]
S.obs <- rowSums((OTUs > 0) * 1)

# Sequencing Coverage
coverage <- rowSums(OTUs)

bacteria <- dim(OTUs)[2]
dim(OTUs)

## [1] 10 5085

# Good's Coverage
goods.c <- goods(OTUs)

# Make Presence Absence Matrix
OTUsPA <- (OTUs > 0) * 1

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

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

# Box-Cox Chord Transformation
OTUs.BCD <- box.cox.chord(OTUs) #Log Chord Transformation

```

Resources

Figure S2: DOC, TN

```

# DOC TN correlation
cor.test(nuts$DOC, nuts$TN)

##
## Pearson's product-moment correlation
##
## data: nuts$DOC and nuts$TN
## t = 17.683, df = 18, p-value = 7.975e-13
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.9301241 0.9892442
## sample estimates:
## cor
## 0.9724041

```

```

cor.test(nuts$DOC, nuts$TP)

##
## Pearson's product-moment correlation
##
## data: nuts$DOC and nuts$TP
## t = 3.9123, df = 18, p-value = 0.001021
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.3362582 0.8618749
## sample estimates:
## cor
## 0.6779058

png(filename="../figures/FigureS2.png",
     width = 900, height = 900, res = 96*2, bg = "white")
par(opar)
par(mar = c(5, 5, 1, 1) + 0.1)
plot(x = log10(nuts$DOC), y = log10(nuts$TN),
     ylim = c(-0.6, 0.3), xlim = c(0.5, 1.6),
     xlab = "", ylab = "",
     pch = 21, bg = "gray", cex = 1.5, lwd=2, axes=F)

mod1 <- lm(log10(nuts$TN) ~ log10(nuts$DOC))
cor(log10(nuts$DOC), log10(nuts$TN))

## [1] 0.9422086

text(log10(4.6), log10(1.8), expression(paste(rho, " = 0.94", sep = "")),
     cex = 1.25)
clip(log10(4), log10(35.4), log10(0.24), log10(2))
abline(mod1, lwd = 2, col = "gray20", lty = 2)

axis(2, at = c(-0.6, -0.3, 0, 0.3), labels = c(0.25, 0.5, 1, 2),
     las = 1, lwd = 1.5, cex.axis = 1.25)
axis(1, at = c(0.7, 1, 1.47), labels = c(5, 10, 30), las = 1, lwd = 1.5, cex.axis = 1.25)
axis(4, at = c(-0.6, -0.3, 0, 0.3), tck = -0.02, labels = F, lwd = 1.5)
axis(3, at = c(0.7, 1, 1.47), tck = -0.02, labels = F, las = 1, lwd = 1.5)
axis(2, at = c(-0.6, -0.3, 0, 0.3), tck = 0.01, labels = F, lwd = 1.5)
axis(1, at = c(0.7, 1, 1.47), tck = 0.01, labels = F, las = 1, lwd = 1.5)
axis(4, at = c(-0.6, -0.3, 0, 0.3), tck = 0.01, labels = F, lwd = 1.5)
axis(3, at = c(0.7, 1, 1.47), tck = 0.01, labels = F, las = 1, lwd = 1.5)
mtext(expression(paste("TN (mg N L"-1, ")")), side = 2, line = 3, cex = 1.5)
mtext(expression(paste("DOC (mg C L"-1, ")")), side = 1, line = 3, cex = 1.5)
box(lwd = 2)

dev.off() # this writes plot to folder

## pdf
## 2

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

#```{r fig.width=4, fig.height=4,echo=FALSE,fig.cap="Diversity"}
img <- readPNG("../figures/FigureS2.png")

```

```
grid.raster(img)
```

DOM Alpha Diversity

```
# Observed Richness
S.res <- rowSums((res.neg > 0) * 1)

# Simpson's Evenness
res.simpE <- round(apply(res.neg, 1, SimpE), 3)

# Shannon's Diversity
res.shan <- round(vegan::diversity(res.neg, index = "shannon"), 2)

# Combine Alpha Diversity
res.div <- data.frame("Lake" = row.names(res.neg), S.res, res.simpE, res.shan)

# Summary Stats
range(res.div$S.res);range(res.div$res.shan);range(res.div$res.simpE)

## [1] 529 569
## [1] 4.89 5.56
## [1] 0.053 0.152
CV(res.div$S.res);CV(res.div$res.shan);CV(res.div$res.simpE)

## [1] 2.118764
## [1] 3.629068
## [1] 32.91581
```

DOM Beta Diversity

```
# Calculate Bray-Curtis
hmf.bray.res <- vegdist(resREL.neg, method = "bray")
hmf.bray.res.log <- vegdist(resREL.neg.log, method = "bray")

dis.mean <- mean(hmf.bray.res)
dis.mean.l <- mean(hmf.bray.res.log)

hmf.bcd.res <- vegdist(DOM.BCD, method = "euclidean")
dis.mean.b <- mean(hmf.bcd.res)

# Principal Coordinates Analysis
pcoa.res <- cmdscale(hmf.bray.res, eig = TRUE, k = 3)
explainvar1.res <- round(pcoa.res$eig[1] / sum(pcoa.res$eig), 3) * 100
explainvar2.res <- round(pcoa.res$eig[2] / sum(pcoa.res$eig), 3) * 100
explainvar3.res <- round(pcoa.res$eig[3] / sum(pcoa.res$eig), 3) * 100
sum.eig.res <- sum(explainvar1.res, explainvar2.res, explainvar3.res)

# DOM Scores
```

```

dom.scores <- t(cor(pcoa.res$points, resREL.neg, method = "spearman"))
dom.scores <- as.matrix(dom.scores)
dom.scores.raw <- dom.scores
# dom.scores <- dom.scores[abs(dom.scores[,1]) > 0.7 |
#                               abs(dom.scores[,2]) > 0.7 , ]

dom.scores <- dom.scores[abs(dom.scores[,1]) > 0.7 |
                        abs(dom.scores[,2]) > 0.7 |
                        abs(dom.scores[,3]) > 0.7, ]

dim(dom.scores)

## [1] 144 3

dom.scores <- as.data.frame(dom.scores)
dom.scores[, 4:5] <- NA
colnames(dom.scores) <- c("Axis_1", "Axis_2", "Axis_3", "Rt", "mz")

for (i in 1:dim(dom.scores)[1]){
  temp <- which(neg.meta$Name == rownames(dom.scores)[i])
  dom.scores[i, 4] <- as.numeric(neg.meta$Rt[temp[1]])
  dom.scores[i, 5] <- as.numeric(neg.meta$mz.max[temp[1]])
}

write.table(round(dom.scores, 3), file = "../data/HMWF_DOM_NEG.txt",
            sep = "\t", quote = F, col.names = NA)

```

Figure 1: Organic Matter Ordination and Env Vectors

```

# Custom palette
palette(rainbow_hcl(10, c = 80, l = 60))
lake.col <- rep(NA, length(unique(design$Lake)))
names(lake.col) <- unique(design$Lake)
lake.col <- as.numeric(factor(design$Lake))

# Nutrients
all.equal(nuts$Site, chl$Lake); all.equal(nuts$Site[nuts$Year == 2012], phys$Lake)

## [1] TRUE
## [1] TRUE

lake.env <- cbind(nuts[nuts$Year == 2012, 3:5],
                 chl[chl$Year == 2012, 3], phys[, 4:6])
row.names(lake.env) <- phys$Lake

norm.tests <- apply(lake.env, 2, shapiro.test)
# Normal: TP
# Not Normal: DOC, TN, Chl, area, pH

lake.env.t <- lake.env * NA
for(i in 1:dim(lake.env.t)[2]){
  D.power <- powerTransform(lake.env[, i])
  lake.env.t[, i] <- as.numeric(scale(bcPower(lake.env[, i],

```



```

coef(D.power, round =F))))
}

dom.size <- log(lake.env$DOC)

all.equal(names(S.res), rownames(lake.env))

## [1] TRUE

S.res.mod <- lm(S.res ~ lake.env$DOC)
summary(S.res.mod)

##
## Call:
## lm(formula = S.res ~ lake.env$DOC)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.168  -2.068   1.675   4.577   7.470
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  567.0044     3.8064    149 4.61e-15 ***
## lake.env$DOC  -1.2972     0.3243     -4  0.00395 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.198 on 8 degrees of freedom
## Multiple R-squared:  0.6667, Adjusted R-squared:  0.6251
## F-statistic:    16 on 1 and 8 DF,  p-value: 0.003948

png(filename=" ../figures/Figure1.png",
     width = 900, height = 900, res = 96*2, bg = "white")
par(opar)

# Define Plot Parameters
layout(matrix(1))
par(mar = c(5, 5, 1, 1) + 0.5)

plot(pcoa.res$points[,1], pcoa.res$points[,2],
     ylim = c(-0.25, 0.3), xlim = c(-0.25, 0.3),
     xlab = paste("PCoA 1 (", explainvar1.res, "%)", sep = ""),
     ylab = paste("PCoA 2 (", explainvar2.res, "%)", sep = ""),
     #xlab = "", ylab = "",
     xaxt = "n", yaxt = "n",
     pch = 17, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1,
     axes = FALSE)

# Add Axes
axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1, las = 1)
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1, las = 1)
axis(side = 3, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=-0.02)
axis(side = 4, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=-0.02)
axis(side = 1, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=0.02)
axis(side = 2, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=0.02)

```

```

axis(side = 3, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=0.02)
axis(side = 4, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=0.02)
abline(h = 0, v = 0, lty = 3)
box(lwd = 2)

# Add DOM Scores
# arrows(0, 0, dom.scores[, 1] * 0.2, dom.scores[, 2] * 0.2,
#       col = "red", length = 0.1, lwd = 0.5)
# text(dom.scores[, 1] * 0.2, dom.scores[, 2] * 0.2, rownames(dom.scores),
#      col = "red", cex = 0.5)
# Add DOM Scores with mz/Rt info
neg.meta2 <- neg.meta[match(rownames(dom.scores), neg.meta$Name), ]

# Add Environmental Vectors
res.env <- envfit(pcoa.res, lake.env)
res.arrows <- res.env[[1]]$arrows * 0.2
arrows(0, 0, res.arrows[, 1], res.arrows[, 2],
      col = "gray10", length = 0.1, lwd = 2, lty = c(1, 1, 3, 1, 3, 1, 3, 3, 1))
arrows(res.arrows[, 1] * 0.99, res.arrows[, 2] * 0.99,
      res.arrows[, 1], res.arrows[, 2],
      col = "gray10", length = 0.1, lwd = 2)
text(res.arrows[, 1] +
     c(-0.03, 0.04, 0.03, 0.04, -0.06, -0.03, -0.03),
     res.arrows[, 2] +
     c(0.03, 0.01, 0.02, -0.01, 0.00, -0.03, -0.03),
     row.names(res.arrows), col = "gray40", cex = 1, font = 3)

# Add Points & Labels
points(pcoa.res$points[, 1], pcoa.res$points[, 2], pch = 22,
      cex = dom.size, bg = "gray", lwd = 2)

      #pch = 15, cex = 4, bg = "gray", col = lake.col)
text(pcoa.res$points[, 1] + c(0, 0, 0.04, 0, -0.04, 0, 0, 0, 0.04, 0),
     pcoa.res$points[, 2] + c(-0.035, 0.035, -0.035, 0.035, 0.035,
                              -0.035, 0.042, -0.035, -0.035, -0.04),
     labels = row.names(pcoa.res$points))

dev.off() # this writes plot to folder

## pdf
## 2

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

#```{r fig.width=4, fig.height=4,echo=FALSE,fig.cap="PCoA Plot Resources"}
img <- readPNG("../figures/Figure1.png")
grid.raster(img)

```

Figure S3: Organic Matter Ordination Figure

```

# Custom palette
palette(rainbow_hcl(10, c = 80, l = 60))

```

```

lake.col <- rep(NA, length(unique(design$Lake)))
names(lake.col) <- unique(design$Lake)
lake.col <- as.numeric(factor(design$Lake))

png(filename="../figures/FigureS3.png",
     width = 900, height = 900, res = 96*2, bg = "white")
par(opar)

# Define Plot Parameters
layout(matrix(1))
par(mar = c(5, 5, 1, 1) + 0.5)

plot(pcoa.res$points[,1], pcoa.res$points[,2],
     ylim = c(-0.25, 0.3), xlim = c(-0.25, 0.3),
     xlab = paste("PCoA 1 (", explainvar1.res, "%)", sep = ""),
     ylab = paste("PCoA 2 (", explainvar2.res, "%)", sep = ""),
     #xlab = "", ylab = "",
     xaxt = "n", yaxt = "n",
     pch = 17, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1,
     axes = FALSE)

# Add Axes
axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1, las = 1)
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1, las = 1)
axis(side = 3, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=-0.02)
axis(side = 4, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=-0.02)
axis(side = 1, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=0.02)
axis(side = 2, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=0.02)
axis(side = 3, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=0.02)
axis(side = 4, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=0.02)
abline(h = 0, v = 0, lty = 3)
box(lwd = 2)

# Add DOM Scores
arrows(0, 0, dom.scores[, 1] * 0.2, dom.scores[, 2] * 0.2, col = "red", length = 0.1)
text(dom.scores[, 1] * 0.2, dom.scores[, 2] * 0.2, rownames(dom.scores), col = "red", cex = 0.5)

# Add Points & Labels
points(pcoa.res$points[,1], pcoa.res$points[,2], pch = 22,
       cex = dom.size, bg = "gray", lwd = 2)

#pch = 15, cex = 4, bg = "gray", col = lake.col)
text(pcoa.res$points[,1] + c(0, 0, 0.04, 0, -0.04, 0, 0, 0, 0.04, 0),
     pcoa.res$points[,2] + c(-0.035, 0.035, -0.035, 0.035, 0.035,
                           -0.035, 0.035, -0.035, -0.035, -0.04),
     labels = row.names(pcoa.res$points))

dev.off() # this writes plot to folder

## pdf
## 2

graphics.off() # shuts down open devices

img <- readPNG("../figures/FigureS3.png")

```

```
grid.raster(img)
```

Alpha Diversity

Bacterial Alpha Diversity

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

# Simpson's Evenness
simpsE <- round(apply(OTUs, 1, SimpE), 3)

# Shannon's Diversity
shan <- vegan::diversity(OTUs, index = "shannon")

# Rarefied Richness
OTUs.rar <- rrarefy(OTUs, ceiling(min(rowSums(OTUs)) * 0.9))
S.rar <- round(rarefy(OTUs, ceiling(min(rowSums(OTUs)) * 0.9)), 0)

# Simpson's Evenness
simpsE.rar <- round(apply(OTUs.rar, 1, SimpE), 3)

# Shannon's Diversity
shan.rar <- vegan::diversity(OTUs.rar, index = "shannon")

alpha.div <- cbind(design, S.obs, simpsE, shan, S.rar, simpsE.rar, shan.rar)
alpha.div <- alpha.div[order(alpha.div$Lake, alpha.div$Year, alpha.div$Molecule), ]

# Summary Stats
range(alpha.div$S.rar);range(alpha.div$simpsE)

## [1] 706 2344
## [1] 0.010 0.053
CV(alpha.div$S.rar);CV(alpha.div$simpsE)

## [1] 50.77223
## [1] 43.62295
```

Resource Heterogeneity and Community Diversity Relationships

```
# Organize Data
nuts2 <- nuts[nuts$Year == 2012, ]
nuts2 <- nuts2[order(nuts2$Site), ]

all.equal(nuts2$Site, alpha.div$Lake)

## [1] TRUE
```

```

all.equal(nuts2$Site, res.div$Lake)

## [1] TRUE
all.equal(nuts2$Site, phys$Lake)

## [1] TRUE
dat <- data.frame(alpha.div[, c(1, 4:8)], res.div[, 2:4],
                  nuts2[, 3:5], phys[, c(4,5,7,9)],
                  row.names = alpha.div[, 1])

shapiro.test(dat$DOC) # Not Normal

##
## Shapiro-Wilk normality test
##
## data: dat$DOC
## W = 0.63753, p-value = 0.0001555
shapiro.test(dat$S.rar) # Not Normal

##
## Shapiro-Wilk normality test
##
## data: dat$S.rar
## W = 0.69123, p-value = 0.0006898
shapiro.test(dat$simpsE.rar) # Normal

##
## Shapiro-Wilk normality test
##
## data: dat$simpsE.rar
## W = 0.90897, p-value = 0.274
shapiro.test(dat$S.res) # Normal

##
## Shapiro-Wilk normality test
##
## data: dat$S.res
## W = 0.91107, p-value = 0.2884
# Without Pony or Lily
shapiro.test(dat$DOC[dat$DOC < 10]) # Normal

##
## Shapiro-Wilk normality test
##
## data: dat$DOC[dat$DOC < 10]
## W = 0.95516, p-value = 0.7629
shapiro.test(dat$S.rar[dat$DOC < 10]) # Normal

##
## Shapiro-Wilk normality test
##
## data: dat$S.rar[dat$DOC < 10]

```

```
## W = 0.87643, p-value = 0.174
# Transform DOC and S.rar with Box-Cox
D.power <- powerTransform(dat$DOC)
S.power <- powerTransform(dat$S.rar)

## Warning in estimateTransform.default(X, Y, weights, family, ...):
## Convergence failure: return code = 52
dat$DOC.t <- as.numeric(scale(bcPower(dat$DOC, coef(D.power, round = F))))
dat$S.rar.t <- as.numeric(scale(bcPower(dat$S.rar, coef(S.power, round = F))))
shapiro.test(dat$DOC.t) # Normal

##
## Shapiro-Wilk normality test
##
## data: dat$DOC.t
## W = 0.95995, p-value = 0.7853
shapiro.test(dat$S.rar.t) # Normal

##
## Shapiro-Wilk normality test
##
## data: dat$S.rar.t
## W = 0.91531, p-value = 0.3195
# plot(dat$S.rar.t ~ dat$S.rar)

# Resource Concentration and Diversity
mod1 <- lm(S.rar.t ~ DOC.t, data = dat)
mod2 <- lm(simpsE.rar ~ DOC.t, data = dat)
summary(mod1);summary(mod2)

##
## Call:
## lm(formula = S.rar.t ~ DOC.t, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.1021 -0.5472  0.2161  0.4918  0.6657
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.710e-09  2.095e-01  0.000  1.00000
## DOC.t       7.810e-01  2.208e-01  3.537  0.00765 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6624 on 8 degrees of freedom
## Multiple R-squared:  0.61, Adjusted R-squared:  0.5612
## F-statistic: 12.51 on 1 and 8 DF, p-value: 0.007651
##
## Call:
## lm(formula = simpsE.rar ~ DOC.t, data = dat)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.015512 -0.007490 -0.004332  0.009511  0.015441
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.038800   0.003645   10.64 5.31e-06 ***
## DOC.t        -0.011258   0.003842   -2.93  0.019 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01153 on 8 degrees of freedom
## Multiple R-squared:  0.5177, Adjusted R-squared:  0.4574
## F-statistic: 8.586 on 1 and 8 DF, p-value: 0.01899
```

```
# Resource Heterogeneity and Divesity
mod3 <- lm(S.rar.t ~ S.res, data = dat)
mod4 <- lm(simpsE.rar ~ S.res, data = dat)
summary(mod3);summary(mod4)
```

```
##
## Call:
## lm(formula = S.rar.t ~ S.res, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.05162 -0.34668 -0.09042  0.55686  1.18499
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.29238   11.83034   2.814  0.0227 *
## S.res        -0.06001    0.02132  -2.815  0.0227 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7518 on 8 degrees of freedom
## Multiple R-squared:  0.4976, Adjusted R-squared:  0.4348
## F-statistic: 7.923 on 1 and 8 DF, p-value: 0.02268
```

```
##
## Call:
## lm(formula = simpsE.rar ~ S.res, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.016460 -0.005666  0.003103  0.006181  0.008315
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.5784574  0.1433841  -4.034  0.00377 **
## S.res        0.0011126  0.0002584   4.306  0.00260 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009112 on 8 degrees of freedom
```

```
## Multiple R-squared:  0.6986, Adjusted R-squared:  0.6609
## F-statistic: 18.54 on 1 and 8 DF,  p-value: 0.002595

# Stats
mod1.p <- round(summary(mod1)$coefficients[2,4], 3)
mod2.p <- round(summary(mod2)$coefficients[2,4], 3)
mod3.p <- round(summary(mod3)$coefficients[2,4], 3)
mod4.p <- round(summary(mod4)$coefficients[2,4], 3)

# Prediction Frames
pred.frame1 <- data.frame(DOC.t = seq(-2, 2, 0.1))
pred.frame2 <- data.frame(S.res = seq(525, 572, 1))

# Correlation Test
cor.test(~ S.res + DOC.t, data = dat)

##
## Pearson's product-moment correlation
##
## data:  S.res and DOC.t
## t = -2.7697, df = 8, p-value = 0.02431
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9227763 -0.1251564
## sample estimates:
##      cor
## -0.6996491

# plot(S.res ~ DOC.t, data = dat)
```

Figure 2: Structural Relationship Plots

```
png(filename="../figures/Figure2.png",
     width = 1400, height = 1400, res = 96*2, bg = "white")
par(opar)

layout(matrix(1:4, nrow = 2, byrow = F))
par(mar = c(0.5, 1, 1, 1) + 0.1, oma = c(6, 6, 0.5, 0.5) + 0.1)

# Resource Concentration vs Species Richness
plot(dat$S.rar.t ~ dat$DOC.t, type = "n",
     xlab = "", ylab = "", axes = F,
     xlim = c(-2, 2), ylim = c(-2, 2), las = 1,
     pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
add.hull(model = mod1, pred.frame = pred.frame1)
matlines(pred.frame1, predict(mod1, interval = "c", newdata=pred.frame1),
        lty=c(2,3,3), lwd=c(3,1.5,1.5), col="black")
points(dat$S.rar.t ~ dat$DOC.t,
       pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
legend("topleft", legend = bquote(italic(p) == .(mod1.p)),
      bty = "n", cex = 1.5, inset = c(-0.05, 0.01))
# mtext("Nutrients", side = 1, line = 3, cex = 1.5)
mtext("OTU Richness\n(Transformed)", side = 2, line = 3, cex = 1.5)
axis(1, lwd = 1.5, labels = F)
```



```

axis(1, lwd = 1.5, tck = -0.02, labels = F)
axis(2, lwd = 1.5, labels = T, las = 1, cex.axis = 1.5)
axis(3, lwd = 1.5, tck = -0.02, labels = F)
axis(4, lwd = 1.5, tck = -0.02, labels = F)
axis(1, lwd = 1.5, tck = 0.02, labels = F)
axis(2, lwd = 1.5, tck = 0.02, labels = F)
axis(3, lwd = 1.5, tck = 0.02, labels = F)
axis(4, lwd = 1.5, tck = 0.02, labels = F)
box(lwd = 1.5)
#text(1.7, 1245, "Pony")
#text(1.73, 1070, "Lily")

# Resource Concentration vs Species Evenness
plot(dat$simpsE.rar ~ dat$DOC.t, type = "n",
      xlab = "", ylab = "", axes = F,
      xlim = c(-2, 2), ylim = c(0, 0.07), las = 1,
      pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
add.hull(model = mod2, pred.frame = pred.frame1)
matlines(pred.frame1, predict(mod2, interval = "c", newdata=pred.frame1),
          lty=c(2,3,3), lwd=c(3,1.5,1.5), col="black")
points(dat$simpsE.rar ~ dat$DOC.t,
        pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
legend("topright", legend = bquote(italic(p) == .(mod2.p)),
       bty = "n", cex = 1.5, inset = 0.01)
mtext("[DOM] (Transformed)", side = 1, line = 3.5, cex = 1.5)
mtext("OTU Evenness", side = 2, line = 4, cex = 1.5)
axis(1, lwd = 1.5, labels = T, cex.axis = 1.5)
axis(1, lwd = 1.5, tck = -0.02, labels = F)
axis(2, lwd = 1.5, labels = T, las = 1, at = c(seq(0, 0.08, 0.02)), cex.axis = 1.5)
axis(3, lwd = 1.5, tck = -0.02, labels = F)
axis(4, lwd = 1.5, tck = -0.02, labels = F, at = c(seq(0, 0.08, 0.02)))
axis(1, lwd = 1.5, tck = 0.02, labels = F)
axis(2, lwd = 1.5, tck = 0.02, labels = F, at = c(seq(0, 0.08, 0.02)))
axis(3, lwd = 1.5, tck = 0.02, labels = F)
axis(4, lwd = 1.5, tck = 0.02, labels = F, at = c(seq(0, 0.08, 0.02)))
box(lwd = 1.5)

# Resource Richness vs Species Richness
plot(dat$S.rar.t ~ dat$S.res, type = "n",
      xlab = "", ylab = "", axes = F,
      xlim = c(525, 572), ylim = c(-2, 2), las = 1,
      pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
add.hull(model = mod3, pred.frame = pred.frame2)
matlines(pred.frame2, predict(mod3, interval = "c", newdata=pred.frame2),
          lty=c(2,3,3), lwd=c(3,1.5,1.5), col="black")
points(dat$S.rar.t ~ dat$S.res,
        pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
legend("topright", legend = bquote(italic(p) == .(mod3.p)),
       bty = "n", cex = 1.5, inset = 0.01)
# mtext("DOM Richness", side = 1, line = 3, cex = 1.5)
# mtext("Species Richness", side = 2, line = 3.5, cex = 1.5)
axis(1, lwd = 1.5, labels = F, at = c(seq(520, 570, 10)))
axis(2, lwd = 1.5, labels = F, las = 1)

```

```

axis(3, lwd = 1.5, tck = -0.02, labels = F, at = c(seq(520, 570, 10)))
axis(4, lwd = 1.5, tck = -0.02, labels = F)
axis(1, lwd = 1.5, tck = 0.02, labels = F, at = c(seq(520, 570, 10)))
axis(2, lwd = 1.5, tck = 0.02, labels = F)
axis(3, lwd = 1.5, tck = 0.02, labels = F, at = c(seq(520, 570, 10)))
axis(4, lwd = 1.5, tck = 0.02, labels = F)
box(lwd = 1.5)

# Resource Richness vs Species Evenness
plot(dat$simpsE.rar ~ dat$S.res, type = "n",
      xlab = "", ylab = "", axes = F,
      xlim = c(525, 572), ylim = c(0, 0.07), las = 1,
      pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
add.hull(model = mod4, pred.frame = pred.frame2)
matlines(pred.frame2, predict(mod4, interval = "c", newdata=pred.frame2),
         lty=c(2,3,3), lwd=c(3,1.5,1.5), col="black")
points(dat$simpsE.rar ~ dat$S.res,
        pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
legend("topleft", legend = bquote(italic(p) == .(mod4.p)),
      bty = "n", cex = 1.5, inset = c(-0.05, 0.01))
mtext("# DOM Components", side = 1, line = 3.5, cex = 1.5)
# mtext("Species Evenness", side = 2, line = 3.5, cex = 1.5)
axis(1, lwd = 1.5, labels = T, las = 1, at = c(seq(520, 570, 10)), cex.axis = 1.5)
axis(2, lwd = 1.5, labels = F, las = 1, at = c(seq(0, 0.08, 0.02)), cex.axis = 1.5)
axis(3, lwd = 1.5, tck = -0.02, labels = F, at = c(seq(520, 570, 10)))
axis(4, lwd = 1.5, tck = -0.02, labels = F, at = c(seq(0, 0.08, 0.02)))
axis(1, lwd = 1.5, tck = 0.02, labels = F, at = c(seq(520, 570, 10)))
axis(2, lwd = 1.5, tck = 0.02, labels = F, at = c(seq(0, 0.08, 0.02)))
axis(3, lwd = 1.5, tck = 0.02, labels = F, at = c(seq(520, 570, 10)))
axis(4, lwd = 1.5, tck = 0.02, labels = F, at = c(seq(0, 0.08, 0.02)))
box(lwd = 1.5)

dev.off() # this writes plot to folder

## pdf
## 2

graphics.off() # shuts down open devices

img <- readPNG("../figures/Figure2.png")
grid.raster(img)

```

Pony and Lily Test

```

# Remove Lily and Pony
dat.b <- dat[-c(5,7), ]

# Resource Concentration and Diversity
mod1b <- lm(S.rar.t ~ DOC.t, data = dat.b)
mod2b <- lm(simpsE.rar ~ DOC.t, data = dat.b)
summary(mod1b);summary(mod2b)

##

```

```

## Call:
## lm(formula = S.rar.t ~ DOC.t, data = dat.b)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7225 -0.4067 -0.2127  0.4548  0.9189
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.2583     0.2684  -0.963   0.373
## DOC.t         0.3472     0.3666   0.947   0.380
##
## Residual standard error: 0.65 on 6 degrees of freedom
## Multiple R-squared:  0.1301, Adjusted R-squared:  -0.01489
## F-statistic: 0.8973 on 1 and 6 DF,  p-value: 0.3801
##
## Call:
## lm(formula = simpsE.rar ~ DOC.t, data = dat.b)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.01335 -0.010531  0.003401  0.007638  0.011503
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.043513   0.004604   9.452 7.98e-05 ***
## DOC.t       -0.003271   0.006288  -0.520   0.622
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01115 on 6 degrees of freedom
## Multiple R-squared:  0.04315, Adjusted R-squared:  -0.1163
## F-statistic: 0.2706 on 1 and 6 DF,  p-value: 0.6216
##
## Resource Heterogeneity and Divesity
mod3b <- lm(S.rar.t ~ S.res, data = dat.b)
mod4b <- lm(simpsE.rar ~ S.res, data = dat.b)
summary(mod3b);summary(mod4b)
##
## Call:
## lm(formula = S.rar.t ~ S.res, data = dat.b)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.6941 -0.3495 -0.2172  0.3338  0.9714
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  16.48023   17.28337   0.954   0.377
## S.res       -0.03020    0.03094  -0.976   0.367
##
## Residual standard error: 0.6474 on 6 degrees of freedom
## Multiple R-squared:  0.137, Adjusted R-squared:  -0.006776

```

```
## F-statistic: 0.9529 on 1 and 6 DF, p-value: 0.3667
##
## Call:
## lm(formula = simpE.rar ~ S.res, data = dat.b)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.012089 -0.001771  0.002282  0.005150  0.005956
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.5054231  0.2052977  -2.462   0.0490 *
## S.res         0.0009849  0.0003675   2.680   0.0365 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.00769 on 6 degrees of freedom
## Multiple R-squared:  0.5449, Adjusted R-squared:  0.469
## F-statistic: 7.183 on 1 and 6 DF, p-value: 0.03653
# Only E.rar (mod4) is still significant
```

Figure S4: Structural Relationship Plots - linear

```
png(filename="../figures/FigureS4.png",
      width = 1600, height = 1600, res = 96*2, bg = "white")
par(opar)

layout(matrix(1:4, nrow = 2, byrow = F))
par(mar = c(0.5, 1, 1, 1) + 0.1, oma = c(6, 5.5, 0, 0) + 0.1)

# Resource Concentration vs Species Richness
plot(dat$S.rar ~ dat$DOC, type = "n",
      xlab = "", ylab = "", axes = F,
      xlim = c(0, 30), ylim = c(600, 2600), las = 1,
      pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
#add.hull(model = mod1, pred.frame = pred.frame1)
#matlines(pred.frame1, predict(mod1, interval = "c", newdata=pred.frame1),
#         lty=c(2,3,3), lwd=c(4,2,2), col="black")
points(dat$S.rar ~ dat$DOC,
        pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
# legend("topleft", legend = bquote(italic(P) == .(mod1.p)),
#       bty = "n", cex = 1.5, inset = c(-0.05, 0.01))
# mtext("Nutrients", side = 1, line = 3, cex = 1.5)
mtext("OTU Richness", side = 2, line = 4, cex = 1.5)
axis(1, lwd = 2, labels = F, at = c(seq(0, 30, 10)))
axis(1, lwd = 2, tck = -0.02, labels = F, at = c(seq(0, 30, 10)))
axis(2, lwd = 2, labels = T, las = 1, cex.axis = 1.5,
      at = seq(600, 20000, by = 400))
axis(3, lwd = 2, tck = -0.02, labels = F, at = c(seq(0, 30, 10)))
axis(4, lwd = 2, tck = -0.02, labels = F,
      at = seq(600, 20000, by = 400))
```

```

axis(1, lwd = 2, tck = 0.02, labels = F, at = c(seq(0, 30, 10)))
axis(2, lwd = 2, tck = 0.02, labels = F,
     at = seq(600, 20000, by = 400))
axis(3, lwd = 2, tck = 0.02, labels = F, at = c(seq(0, 30, 10)))
axis(4, lwd = 2, tck = 0.02, labels = F,
     at = seq(600, 20000, by = 400))
box(lwd = 2)
#text(1.7, 1245, "Pony")
#text(1.73, 1070, "Lily")

# Resource Concentration vs Species Evenness
plot(dat$simpsE.rar ~ dat$DOC, type = "n",
     xlab = "", ylab = "", axes = F,
     xlim = c(0, 30), ylim = c(0, 0.07), las = 1,
     pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
# add.hull(model = mod2, pred.frame = pred.frame1)
# matlines(pred.frame1, predict(mod2, interval = "c", newdata=pred.frame1),
#          lty=c(2,3,3), lwd=c(4,2,2), col="black")
points(dat$simpsE.rar ~ dat$DOC,
       pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
# legend("topright", legend = bquote(italic(P) == .(mod2.p)),
#       bty = "n", cex = 1.5, inset = 0.01)
mtext(expression(paste("DOC (mg C L"-1, ")")), side = 1, line = 3.5, cex = 1.5)
mtext("OTU Evenness", side = 2, line = 4, cex = 1.5)
axis(1, lwd = 2, labels = T, cex.axis = 1.5, at = c(seq(0, 30, 10)))
axis(1, lwd = 2, tck = -0.02, labels = F, at = c(seq(0, 30, 10)))
axis(2, lwd = 2, labels = T, las = 1, at = c(seq(0, 0.06, 0.02)), cex.axis = 1.5)
axis(3, lwd = 2, tck = -0.02, labels = F, at = c(seq(0, 30, 10)))
axis(4, lwd = 2, tck = -0.02, labels = F, at = c(seq(0, 0.06, 0.02)))
axis(1, lwd = 2, tck = 0.02, labels = F, at = c(seq(0, 30, 10)))
axis(2, lwd = 2, tck = 0.02, labels = F, at = c(seq(0, 0.06, 0.02)))
axis(3, lwd = 2, tck = 0.02, labels = F, at = c(seq(0, 30, 10)))
axis(4, lwd = 2, tck = 0.02, labels = F, at = c(seq(0, 0.06, 0.02)))
box(lwd = 2)

# Resource Richness vs Species Richness
plot(dat$S.rar ~ dat$S.res, type = "n",
     xlab = "", ylab = "", axes = F,
     xlim = c(525, 572), ylim = c(600, 2600), las = 1,
     pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
# add.hull(model = mod3, pred.frame = pred.frame2)
# matlines(pred.frame2, predict(mod3, interval = "c", newdata=pred.frame2),
#          lty=c(2,3,3), lwd=c(4,2,2), col="black")
points(dat$S.rar ~ dat$S.res,
       pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
# legend("topright", legend = bquote(italic(P) == .(mod3.p)),
#       bty = "n", cex = 1.5, inset = 0.01)
# mtext("DOM Richness", side = 1, line = 3, cex = 1.5)
# mtext("Species Richness", side = 2, line = 3.5, cex = 1.5)
axis(1, lwd = 2, labels = F, at = c(seq(520, 570, 10)))
axis(2, lwd = 2, labels = F, las = 1,
     at = seq(600, 20000, by = 400))
axis(3, lwd = 2, tck = -0.02, labels = F, at = c(seq(520, 570, 10)))

```

```

axis(4, lwd = 2, tck = -0.02, labels = F,
     at = seq(600, 20000, by = 400))
axis(1, lwd = 2, tck = 0.02, labels = F, at = c(seq(520, 570, 10)))
axis(2, lwd = 2, tck = 0.02, labels = F,
     at = seq(600, 20000, by = 400))
axis(3, lwd = 2, tck = 0.02, labels = F, at = c(seq(520, 570, 10)))
axis(4, lwd = 2, tck = 0.02, labels = F,
     at = seq(600, 20000, by = 400))
box(lwd = 2)

# Resource Richness vs Species Evenness
plot(dat$simpsE.rar ~ dat$S.res, type = "n",
     xlab = "", ylab = "", axes = F,
     xlim = c(525, 572), ylim = c(0, 0.07), las = 1,
     pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
# add.hull(model = mod4, pred.frame = pred.frame2)
# matlines(pred.frame2, predict(mod4, interval = "c", newdata=pred.frame2),
#          lty=c(2,3,3), lwd=c(4,2,2), col="black")
points(dat$simpsE.rar ~ dat$S.res,
       pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
# legend("topleft", legend = bquote(italic(P) == .(mod4.p)),
#       bty = "n", cex = 1.5, inset = c(-0.05, 0.01))
mtext("# DOM Components", side = 1, line = 3.5, cex = 1.5)
# mtext("Species Evenness", side = 2, line = 3.5, cex = 1.5)
axis(1, lwd = 2, labels = T, las = 1, at = c(seq(520, 570, 10)), cex.axis = 1.5)
axis(2, lwd = 2, labels = F, las = 1, at = c(seq(0, 0.06, 0.02)), cex.axis = 1.5)
axis(3, lwd = 2, tck = -0.02, labels = F, at = c(seq(520, 570, 10)))
axis(4, lwd = 2, tck = -0.02, labels = F, at = c(seq(0, 0.06, 0.02)))
axis(1, lwd = 2, tck = 0.02, labels = F, at = c(seq(520, 570, 10)))
axis(2, lwd = 2, tck = 0.02, labels = F, at = c(seq(0, 0.06, 0.02)))
axis(3, lwd = 2, tck = 0.02, labels = F, at = c(seq(520, 570, 10)))
axis(4, lwd = 2, tck = 0.02, labels = F, at = c(seq(0, 0.06, 0.02)))
box(lwd = 2)

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

img <- readPNG("../figures/FigureS4.png")
grid.raster(img)

```

Figure X1: DOM Richness and Species Evenness

```

png(filename="../figures/FigureX1.png",
     width = 900, height = 900, res = 96*2, bg = "white")
par(opar)

par(mar = c(0.5, 1, 1, 1) + 0.1, oma = c(5, 5.5, 0, 0) + 0.1)

# Resource Richness vs Species Evenness
plot(dat$simpsE.rar ~ dat$S.res,
     xlab = "", ylab = "", type = "n", axes = F,
     xlim = c(525, 572), ylim = c(0, 0.06), las = 1,

```

```

    pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
add.hull(model = mod4, pred.frame = pred.frame2)
matlines(pred.frame2, predict(mod4, interval = "c", newdata=pred.frame2),
        lty=c(2,3,3), lwd=c(4,2,2), col="black")
points(dat$simpsE.rar ~ dat$S.res,
       pch = 22, col = "black", bg = "gray", cex = 1.5, lwd = 2)
legend("topleft", legend = bquote(italic(p) == .(mod4.p)),
      bty = "n", cex = 1.25, inset = c(-0.05, 0.01))
mtext("# DOM Components", side = 1, line = 3.5, cex = 1.5)
mtext("OTU Evenness", side = 2, line = 3.5, cex = 1.5)
axis(1, lwd = 2, labels = T, las = 1, at = c(seq(520, 570, 10)), cex.axis = 1.25)
axis(2, lwd = 2, labels = T, las = 1, at = c(seq(0, 0.06, 0.02)), cex.axis = 1.25)
axis(3, lwd = 2, tck = -0.02, labels = F, at = c(seq(520, 570, 10)))
axis(4, lwd = 2, tck = -0.02, labels = F, at = c(seq(0, 0.06, 0.02)))
axis(1, lwd = 2, tck = 0.02, labels = F, at = c(seq(520, 570, 10)))
axis(2, lwd = 2, tck = 0.02, labels = F, at = c(seq(0, 0.06, 0.02)))
axis(3, lwd = 2, tck = 0.02, labels = F, at = c(seq(520, 570, 10)))
axis(4, lwd = 2, tck = 0.02, labels = F, at = c(seq(0, 0.06, 0.02)))
box(lwd = 2)

dev.off() # this writes plot to folder

## pdf
## 2

graphics.off() # shuts down open devices

img <- readPNG("../figures/FigureX1.png")
grid.raster(img)

```

Beta Diversity

Bacterial Compositional Diversity

```

# Calculate Bray-Curtis
hmf.bray.REL <- vegdist(OTUs.REL.log, method = "bray")
dis.mean <- mean(hmf.bray.REL)

# Log-Chord Distance
hmf.bcd.euc <- vegdist(OTUs.BCD, method = "euclidean")
mean(hmf.bcd.euc)

## [1] 0.9891855

# Principal Coordinates Analysis w/ BC
pcoa.rel <- cmdscale(hmf.bray.REL, eig = TRUE, k = 3)
explainvar1.rel <- round(pcoa.rel$eig[1] / sum(pcoa.rel$eig), 3) * 100
explainvar2.rel <- round(pcoa.rel$eig[2] / sum(pcoa.rel$eig), 3) * 100
explainvar3.rel <- round(pcoa.rel$eig[3] / sum(pcoa.rel$eig), 3) * 100
sum.eig.rel <- sum(explainvar1.rel, explainvar2.rel, explainvar3.rel)

# Principal Coordinates Analysis w/ Log-Chord

```

```

pcoa.bcd <- cmdscale(hmwf.bcd.euc, eig = TRUE, k = 3)
explainvar1.bcd <- round(pcoa.bcd $eig[1] / sum(pcoa.bcd $eig), 3) * 100
explainvar2.bcd <- round(pcoa.bcd $eig[2] / sum(pcoa.bcd $eig), 3) * 100
explainvar3.bcd <- round(pcoa.bcd $eig[3] / sum(pcoa.bcd $eig), 3) * 100
sum.eig.bcd <- sum(explainvar1.bcd, explainvar2.bcd, explainvar3.bcd)

# OTU Scores
otu.scores <- t(cor(pcoa.rel$points, OTUsREL))
otu.scores <- as.matrix(otu.scores)[,1:2]
otu.scores <- otu.scores[abs(otu.scores[,1]) > 0.7 |
                        abs(otu.scores[,2]) > 0.7,]
dim(otu.scores)

## [1] 1125    2

```

Compositional Relationships (dbRDA)

```

# Resource Concentration dbRDA
hmwf.bray.REL <- vegdist(OTUsREL.log, method = "bray")
pcoa.rel <- cmdscale(hmwf.bray.REL, eig = TRUE, k = 3)
dbRDA <- capscale(hmwf.bray.REL ~ dat$DOC.t, comm = OTUsREL.log, add = T)
anova(dbRDA, permutations = how(nperm=9999))

## Permutation test for capscale under reduced model
## Permutation: free
## Number of permutations: 9999
##
## Model: capscale(formula = hmwf.bray.REL ~ dat$DOC.t, comm = OTUsREL.log, add = T)
##           Df SumOfSqs      F Pr(>F)
## Model      1  0.54264 3.1035 9e-04 ***
## Residual    8  1.39880
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

RsquareAdj(dbRDA)

```

```

## $r.squared
## [1] 0.2795044
##
## $adj.r.squared
## [1] 0.1894425

```

```

# DOM Diversity dbRDA; using: hmwf.bray.res; pcoa.res
# Calculate Bray-Curtis
hmwf.bray.res <- vegdist(resREL.neg, method = "bray")
hmwf.bray.res.log <- vegdist(resREL.neg.log, method = "bray")
hmwf.bcd.res <- vegdist(box.cox.chord(res.neg), method = "euclidean")
pcoa.res <- cmdscale(hmwf.bray.res.log, eig = TRUE, k = 3)
dbRDA.dom <- capscale(hmwf.bray.REL ~ pcoa.res$points[, 1:3], add = T)
anova(dbRDA.dom)

## Permutation test for capscale under reduced model
## Permutation: free
## Number of permutations: 999

```



```
##
## Model: capscale(formula = hmwf.bray.REL ~ pcoa.res$points[, 1:3], add = T)
##           Df SumOfSqs      F Pr(>F)
## Model      3  0.87948 1.6563  0.056 .
## Residual    6  1.06197
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
RsquareAdj(dbrDA.dom)
```

```
## $r.squared
## [1] 0.4530018
##
## $adj.r.squared
## [1] 0.1795026
```

```
anova(dbrDA.dom, by = 'axis')
```

```
## Permutation test for capscale under reduced model
## Forward tests for axes
## Permutation: free
## Number of permutations: 999
##
## Model: capscale(formula = hmwf.bray.REL ~ pcoa.res$points[, 1:3], add = T)
##           Df SumOfSqs      F Pr(>F)
## CAP1        1  0.57114 3.2269  0.058 .
## CAP2        1  0.17128 0.9677  0.810
## CAP3        1  0.13706 0.7744  0.722
## Residual     6  1.06197
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
bray.BAC <- vegdist(decostand(OTUsREL, "log"), "bray")
pcoa.BAC <- cmdscale(bray.BAC, k = 3, eig = T)
bray.RES <- vegdist(decostand(resREL.neg, "log"), "bray")
pcoa.RES <- cmdscale(bray.RES, k = 3, eig = T)
dbrDA.dom <- capscale(bray.BAC ~ pcoa.RES$points[, 1:3], add = T)
anova(dbrDA.dom)
```

```
## Permutation test for capscale under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: capscale(formula = bray.BAC ~ pcoa.RES$points[, 1:3], add = T)
##           Df SumOfSqs      F Pr(>F)
## Model      3  0.87016 1.6541  0.04 *
## Residual    6  1.05212
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
RsquareAdj(dbrDA.dom)
```

```
## $r.squared
## [1] 0.4526707
##
## $adj.r.squared
## [1] 0.1790061
```

```

anova(dbrDA.dom, by = 'axis')

## Permutation test for capscale under reduced model
## Forward tests for axes
## Permutation: free
## Number of permutations: 999
##
## Model: capscale(formula = bray.BAC ~ pcoa.RES$points[, 1:3], add = T)
##           Df SumOfSqs      F Pr(>F)
## CAP1      1  0.56169 3.2032  0.040 *
## CAP2      1  0.17121 0.9764  0.825
## CAP3      1  0.13726 0.7828  0.738
## Residual   6   1.05212
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

res.com <- envfit(pcoa.BAC, pcoa.RES$points)
res.com

##
## ***VECTORS
##
##           Dim1      Dim2      r2 Pr(>r)
## [1,]  0.20782  0.97817 0.1909  0.470
## [2,] -0.99992 -0.01258 0.7025  0.009 **
## [3,]  0.65630 -0.75450 0.0518  0.841
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999

cor.test(~ pcoa.RES$points[, 1] + dat$DOC.t)

##
## Pearson's product-moment correlation
##
## data:  pcoa.RES$points[, 1] and dat$DOC.t
## t = 0.94717, df = 8, p-value = 0.3713
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.3900706  0.7893520
## sample estimates:
##           cor
## 0.3175442

cor.test(~ pcoa.RES$points[, 2] + dat$DOC.t)

##
## Pearson's product-moment correlation
##
## data:  pcoa.RES$points[, 2] and dat$DOC.t
## t = -2.7488, df = 8, p-value = 0.0251
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.9219884 -0.1199549
## sample estimates:

```

```
##          cor
## -0.6969433

cor.test(~ pcoa.RES$points[, 3] + dat$DOC.t)

##
## Pearson's product-moment correlation
##
## data:  pcoa.RES$points[, 3] and dat$DOC.t
## t = 0.93311, df = 8, p-value = 0.3781
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3940634  0.7875671
## sample estimates:
##          cor
## 0.3132955
```

Figure 3: Bacterial PCoA Plot with Env Vectors

```
# PCoA of Active Community
bray.BAC <- vegdist(decostand(OTUsREL, "log"), "bray")

## Warning: non-integer data: divided by smallest positive value

pcoa.BAC <- cmdscale(bray.BAC, k = 3, eig = T)
explainvar1 <- round(pcoa.BAC$eig[1] / sum(pcoa.BAC$eig), 3) * 100
explainvar2 <- round(pcoa.BAC$eig[2] / sum(pcoa.BAC$eig), 3) * 100

# PCoA of Resources
bray.RES <- vegdist(decostand(resREL.neg, "log"), "bray")

## Warning: non-integer data: divided by smallest positive value

pcoa.RES <- cmdscale(bray.RES, k = 3, eig = T)

# Resource Concentrations
cons.RES <- dat$DOC.t

# Initial Plot as PNG
png(filename="../figures/Figure3.png",
     width = 1400, height = 1000, res = 96*2, bg = "white")

# Define Plot Parameters
par(opar)
par(mar = c(4.75, 5, 1, 1) + 0.5)

# Initiate Plot 1
plot(pcoa.BAC$points[, 1], pcoa.BAC$points[, 2],
     ylim = c(-0.3, 0.4), xlim = c(-0.4, 0.6),
     xlab = paste("PCoA 1 (", explainvar1, "%)", sep = ""),
     ylab = paste("PCoA 2 (", explainvar2, "%)", sep = ""),
     #xlab = "", ylab = "", xaxt = "n", yaxt = "n",
     pch = 22, cex = 2.0, type = "n", cex.lab = 1.75, cex.axis = 1,
     axes = FALSE)
```

```

# Add Axes
axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1.25, las = 1)
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1.25, las = 1,
     at = c(-0.2, 0, 0.2, 0.4))
axis(side = 3, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=-0.02)
axis(side = 4, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=-0.02,
     at = c(-0.2, 0, 0.2, 0.4))
axis(side = 1, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=0.01)
axis(side = 2, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=0.01,
     at = c(-0.2, 0, 0.2, 0.4))
axis(side = 3, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=0.01)
axis(side = 4, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=0.01,
     at = c(-0.2, 0, 0.2, 0.4))
abline(h = 0, v = 0, lty = 3)
box(lwd = 2)

# Add Points & Labels
points(pcoa.BAC$points[,1], pcoa.BAC$points[,2], pch = 22,
       cex = 3.2, bg = "gray", lwd = 2)
#text(pcoa.BAC$points[,1], pcoa.BAC$points[,2],
#      unlist(lapply(strsplit(row.names(pcoa.BAC$points),
#      split = "_"), "[[", 1)))

# Resource Concentration Vector
# res.con <- envfit(pcoa.BAC, cons.RES)
# con.arrows <- res.con[[1]]$arrows * 0.3
# arrows(0, 0, con.arrows[, 1], con.arrows[, 2], col = "gray10", length = 0.1, lwd = 2)
# text(con.arrows[, 1] * 1.2, con.arrows[, 2] * 1.2, "Conc.", col = "black", cex = 1)

# DOM Composition Vectors
cor.test(~ pcoa.RES$points[, 2] + pcoa.BAC$points[, 1])
res.com <- envfit(pcoa.BAC, pcoa.RES$points[,1:2])
com.arrows <- res.com[[1]]$arrows * 0.3
arrows(0, 0, -com.arrows[1, 1], com.arrows[1, 2],
       col = "gray30", length = 0.1, lwd = 4)
arrows(0, 0, -com.arrows[2, 1], -com.arrows[2, 2],
       col = "gray30", length = 0.1, lwd = 4)
text(-com.arrows[1, 1] - 0.02, com.arrows[1, 2] * 1.2, "DOM 1",
     col = "gray40", cex = 1.5, font = 3)
text(-com.arrows[2, 1] * 1.2, com.arrows[2, 2] + 0.08, "DOM 2",
     col = "gray40", cex = 1.5, font = 3)

text(0.48, -0.05, "Lily", col = "black", cex = 1.2)
text(0.56, 0.1, "Pony", col = "black", cex = 1.2)
text(-0.08, -0.23, "Ann", col = "black", cex = 1.2)
text(0.05, -0.145, "Canyon", col = "black", cex = 1.2)
text(-0.14, -0.08, "Howe", col = "black", cex = 1.2)
text(-0.11, -0.025, "Ives", col = "black", cex = 1.2)
text(-0.255, 0.08, "Mountain", col = "black", cex = 1.2)
text(-0.24, -0.18, "Rush", col = "black", cex = 1.2)
text(-0.235, 0.23, "Second", col = "black", cex = 1.2)
text(-0.21, 0.32, "Upper", col = "black", cex = 1.2)

```

```

# legend("bottomright", c("Resource Concentration Model",
#                          "Resource Composition Model"),
#        lwd = 2, col = c("gray10", "gray60", "gray60"), bty = "n")

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

# ```{r fig.width=4, fig.height=4,echo=FALSE,fig.cap="Diversity"}
img <- readPNG("../figures/Figure3.png")
grid.raster(img)

```

Figure X2: Example

```

# Initial Plot as PNG
png(filename="../figures/FigureX2.png",
     width = 1300, height = 900, res = 96*2, bg = "white")

# Define Plot Parameters
par(opar)
par(mar = c(4, 5, 1, 1) + 0.5)

plot(pcoa.BAC$points[,1], pcoa.BAC$points[,2],
     ylim = c(-0.3, 0.4), xlim = c(-0.4, 0.6),
     xlab = paste("PCoA 1 (", explainvar1, "%)", sep = ""),
     ylab = paste("PCoA 2 (", explainvar2, "%)", sep = ""),
     #xlab = "", ylab = "", xaxt = "n", yaxt = "n",
     pch = 22, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1,
     axes = FALSE)

# Add Axes
axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1.25, las = 1)
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1.25, las = 1,
     at = c(-0.2, 0, 0.2, 0.4))
axis(side = 3, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=-0.02)
axis(side = 4, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=-0.02,
     at = c(-0.2, 0, 0.2, 0.4))
axis(side = 1, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=0.01)
axis(side = 2, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=0.01,
     at = c(-0.2, 0, 0.2, 0.4))
axis(side = 3, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=0.01)
axis(side = 4, labels = F, lwd.ticks = 2, cex.axis = 1, las = 1, tck=0.01,
     at = c(-0.2, 0, 0.2, 0.4))
abline(h = 0, v = 0, lty = 3)
box(lwd = 2)

# Add Points & Labels
points(pcoa.BAC$points[,1], pcoa.BAC$points[,2], pch = 22,
       cex = 2.5, bg = "gray", lwd = 2)
#text(pcoa.BAC$points[,1], pcoa.BAC$points[,2],
#     unlist(lapply(strsplit(row.names(pcoa.BAC$points),
#     split = "_"), "[[", 1)))

```

```

# Resource Concentration Vector
# res.con <- envufit(pcoa.BAC, cons.RES)
# con.arrows <- res.con[[1]]$arrows * 0.3
# arrows(0, 0, con.arrows[, 1], con.arrows[, 2], col = "gray10", length = 0.1, lwd = 2)
# text(con.arrows[, 1] * 1.2, con.arrows[, 2] * 1.2, "Conc.", col = "black", cex = 1)

text(0.48, -0.045, "Lily", col = "black", cex = 0.8)
text(0.56, 0.1, "Pony", col = "black", cex = 0.8)
text(-0.085, -0.24, "Ann", col = "black", cex = 0.8)
text(0.05, -0.145, "Canyon", col = "black", cex = 0.8)
text(-0.14, -0.08, "Howe", col = "black", cex = 0.8)
text(-0.12, -0.02, "Ives", col = "black", cex = 0.8)
text(-0.255, 0.08, "Mountain", col = "black", cex = 0.8)
text(-0.23, -0.18, "Rush", col = "black", cex = 0.8)
text(-0.235, 0.25, "Second", col = "black", cex = 0.8)
text(-0.19, 0.32, "Upper", col = "black", cex = 0.8)

# legend("bottomright", c("Resource Concentration Model",
#                          "Resource Composition Model"),
#       lwd = 2, col = c("gray10", "gray60", "gray60"), bty = "n")

dev.off() # this writes plot to folder

## pdf
## 2

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

# ```{r fig.width=4, fig.height=4,echo=FALSE,fig.cap="Diversity"}
img <- readPNG("../figures/FigureX2.png")
grid.raster(img)

```

Generalist and Specialists Patterns

Spatial Generalists

```

# Define generalists as those found at all sites
sp.gens <- which(colSums(OTUsPA) >= 8)
length(sp.gens); length(sp.gens)/dim(OTUsPA)[2]

## [1] 233
## [1] 0.04582104

sp.gen.tax <- OTU.tax[which(OTU.tax$OTU %in% names(sp.gens)), ]
#table(sp.gen.tax$Phylum)
#table(sp.gen.tax$Class)
#table(sp.gen.tax$Order)
#table(sp.gen.tax$Family)
#table(sp.gen.tax$Genus)

```

```

# Save Spatial Generalists
write.table(sp.gen.tax, "../data/SpatialGeneralists.txt", quote = F,
            row.names = F, sep = "\t")

# Define specialists as those found at only 2 or fewer sites
sp.spec <- which(colSums(OTUsPA) <= 5)
sp.spe.tax <- OTU.tax[which(OTU.tax$OTU %in% names(sp.spec)), ]
#table(sp.spe.tax$Order)

```

Consumer-Resource Interaction

```

# Subset OTUs for most dominant
all.equal(colnames(OTUsREL), colnames(OTUsPA))

## [1] TRUE

# OTUsREL.dom <- OTUsREL[,which(colSums(as.matrix(OTUsREL)) > 0.005 )]

OTUsREL.dom <- OTUsREL[,which(colSums(as.matrix(OTUsREL)) > 0.01 &
                               colSums(OTUsPA) > 1)]
dim(OTUsREL.dom)

## [1] 10 158

row.names(OTUsREL.dom) <- gsub("2012_RNA", "", row.names(OTUsREL.dom))
write.table(OTUsREL.dom, "../data/OTUsREL.subset.txt", quote = F)

# which((log10(apply(OTUsREL.dom, 2, max)) -
#          log10(apply(OTUsREL.dom, 2, min)))) > 2)

# Subset Resources for 100 most dominant
resREL.dom <- resREL.neg[,which(colSums(as.matrix(resREL.neg)) > 0.01)]
resREL.dom <- resREL.dom[,order(colSums(as.matrix(resREL.dom)), decreasing = T)[1:100]]

# Subset Resource for most influential
resREL.sc <- t(cor(pcoa.RES$points, resREL.neg))
resREL.sc <- as.matrix(resREL.sc)[,1:3]
resREL.sc <- resREL.sc[abs(resREL.sc[,1]) >= 0.7 | abs(resREL.sc[,2]) >= 0.7 |
                      abs(resREL.sc[,3]) >= 0.7, ]
resREL.inf <- resREL.neg[, which(colnames(resREL.neg) %in% rownames(resREL.sc))]
dim(resREL.inf)

## [1] 10 172

# Save Data
write.table(resREL.inf, "../data/ResREL.subset.txt", quote = F)

# Annotations for Influential Resources
DOM.inf.annot <- res.annot[which(res.annot$Cmpd %in% (names(resREL.inf))), ]

# Add DOM Scores
dom.scores.raw2 <- dom.scores.raw[which(row.names(dom.scores.raw) %in% DOM.inf.annot$Cmpd), ]
all.equal(as.character(DOM.inf.annot$Cmpd[1:5]), rownames(dom.scores.raw2)[1:5])

## [1] TRUE

```

```

DOM.inf.annot$DIM1 <- dom.scores.raw2[, 1]
DOM.inf.annot$DIM2 <- dom.scores.raw2[, 2]

write.table(DOM.inf.annot, "../data/ResRel.inf.annot.txt", quote = F,
            sep = "\t", row.names = F)

# Define Function
spear.matrix <- function(mat1 = "", mat2 = "", dim1 = "", dim2 = ""){
  mat.temp <- matrix(NA, dim1, dim2)
  row.names(mat.temp) <- colnames(mat1)
  colnames(mat.temp) <- colnames(mat2)
  mat.temp <- cor(mat1, mat2, method = "spearman")
  return(mat.temp)
}

dim1 <- dim(OTUsREL.dom)[2]
dim2 <- dim(resREL.inf)[2]

# Calculate Consumer Resource Correlations
spear.ConRes <- spear.matrix(mat1 = OTUsREL.dom, mat2 = resREL.inf, dim1, dim2)

# Isolate Relevant Interactions
spear.ConRes2 <- spear.ConRes
spear.ConRes2[which(spear.ConRes < 0.6 & spear.ConRes > -0.6)] <- 0
spear.ConRes3 <- spear.ConRes2

# Summary
str(spear.ConRes3)

## num [1:158, 1:172] 0 0 0 0 0 ...
## - attr(*, "dimnames")=List of 2
## ..$ : chr [1:158] "Otu00002" "Otu00003" "Otu00004" "Otu00005" ...
## ..$ : chr [1:172] "C1" "C3" "C8" "C10" ...

sum(spear.ConRes3 <= -0.7)

## [1] 779

sum(spear.ConRes3 >= 0.7)

## [1] 499

pos.cor.ConRes <- which(spear.ConRes3 >= 0.7)
neg.cor.ConRes <- which(spear.ConRes3 <= -0.7)

# Significance Test
sig.test.neg <- vector("list", 1000)
sig.test.pos <- vector("list", 1000)
for (i in 1:1000) {
  # Two Independent Runs
  testA <- randomizeMatrix(as.matrix(OTUsREL.dom), null.model = "independentswap")
  testB <- randomizeMatrix(as.matrix(resREL.inf), null.model = "independentswap")
  # Co-Occurrence Analysis
  spear.ConRes.test <- spear.matrix(testA, testB, dim1, dim2)
  spear.ConRes2.test <- spear.ConRes.test

```



```

spear.ConRes2.test[which(spear.ConRes.test <= 0.7 &
                        spear.ConRes.test >= -0.7)] <- 0
spear.ConRes3.test <- as.matrix(spear.ConRes2.test)
sig.test.neg[[i]] <- which(spear.ConRes3.test <= -0.7)
sig.test.pos[[i]] <- which(spear.ConRes3.test >= 0.7)
}

# False Negatives
false.neg <- which((table(unlist(sig.test.neg)) / 1000) > 0.05)
false.pos <- which((table(unlist(sig.test.pos)) / 1000) > 0.05)
weak.neg <- which((table(unlist(sig.test.neg)) / 1000) > 0.05 &
                  (table(unlist(sig.test.neg)) / 1000) < 0.1)
weak.pos <- which((table(unlist(sig.test.pos)) / 1000) > 0.05 &
                  (table(unlist(sig.test.pos)) / 1000) < 0.1)

length(setdiff(neg.cor.ConRes, c(false.neg, weak.neg)))

## [1] 765

length(setdiff(pos.cor.ConRes, c(false.pos, weak.neg)))

## [1] 492

sum(length(setdiff(neg.cor.ConRes, c(false.neg, weak.neg))),
     length(setdiff(pos.cor.ConRes, c(false.pos, weak.neg)))) / 10000

## [1] 0.1257

# Remove Non Significant Interactions and Define Weak Interactions
spear.ConRes4 <- spear.ConRes3
bad.ints.neg <- sort(false.neg)
bad.ints.pos <- sort(false.pos)
for (i in 1:length(bad.ints.neg)){
  if (spear.ConRes4[bad.ints.neg[i]] < 0){
    spear.ConRes4[bad.ints.neg[i]] <- 0
  } else {}
}
for (i in 1:length(bad.ints.pos)){
  if (spear.ConRes4[bad.ints.pos[i]] > 0){
    spear.ConRes4[bad.ints.pos[i]] <- 0
  } else {}
}
weak.ints.neg <- sort(weak.neg)
weak.ints.pos <- sort(weak.pos)
for (i in 1:length(weak.ints.neg)){
  if (spear.ConRes4[weak.ints.neg[i]] < 0){
    spear.ConRes4[weak.ints.neg[i]] <- 0
  } else {}
}
for (i in 1:length(weak.ints.pos)){
  if (spear.ConRes4[weak.ints.pos[i]] > 0){
    spear.ConRes4[weak.ints.pos[i]] <- 0
  } else {}
}

# Remove Zero Sum Col and Rows

```

```
spear.ConRes5 <- spear.ConRes4[which(apply(spear.ConRes4, 1, min) < 0),
                                which(apply(spear.ConRes4, 2, min) < 0)]
```

```
# Summary
str(spear.ConRes5)
```

```
## num [1:149, 1:165] 0 0 0 0 0 ...
## - attr(*, "dimnames")=List of 2
## ..$ : chr [1:149] "Otu00002" "Otu00003" "Otu00004" "Otu00005" ...
## ..$ : chr [1:165] "C1" "C3" "C8" "C10" ...
```

```
dim(spear.ConRes5)
```

```
## [1] 149 165
```

```
sum(spear.ConRes5 < -0.7)
```

```
## [1] 765
```

```
sum(spear.ConRes5 > 0.7)
```

```
## [1] 473
```

```
percent.total <- (sum(spear.ConRes5 <= -0.7) + sum(spear.ConRes5 >= 0.7)) /
  (dim(spear.ConRes5)[1] * dim(spear.ConRes5)[2])
pos.cor.ConRes <- which(spear.ConRes5 >= 0.7)
neg.cor.ConRes <- which(spear.ConRes5 <= -0.7)
sum(colSums(spear.ConRes5) != 0) / dim(spear.ConRes)[2]
```

```
## [1] 0.9593023
```

```
sum(rowSums(spear.ConRes5) != 0) / dim(spear.ConRes)[1]
```

```
## [1] 0.943038
```

```
sum(pos.cor.ConRes != 0)
```

```
## [1] 473
```

```
sum(neg.cor.ConRes != 0)
```

```
## [1] 765
```

Resource Generalists

```
ConRes.matrix <- as.matrix(spear.ConRes5)
re.gens <- rownames(ConRes.matrix)[rowSums(ConRes.matrix <= -0.7) >= 4]
# re.gens <- rownames(ConRes.matrix)[rowSums(abs(ConRes.matrix) >= 0.7) >= 6]
length(re.gens); length(re.gens)/dim(OTUsPA)[2] # (67; 88)
```

```
## [1] 67
```

```
## [1] 0.01317601
```

```
re.gen.tax <- OTU.tax[which(OTU.tax$OTU %in% re.gens), ]
#table(re.gen.tax$Phylum)
#table(re.gen.tax$Class)
#table(re.gen.tax$Order)
#table(re.gen.tax$Family)
```

```

# Save Resource Generalists
write.table(re.gen.tax, "../data/ResourceGeneralists.txt", quote = F,
           row.names = F, sep = "\t")

re.spec <- rownames(ConRes.matrix)[rowSums(ConRes.matrix <= -0.7) <= 2]
re.spe.tax <- OTU.tax[which(OTU.tax$OTU %in% re.spec), ]
#table(re.spe.tax$Order)

length(re.gens); length(re.gens) / dim(OTUsPA) [2]

## [1] 67
## [1] 0.01317601

sp.gens.n <- names(sp.gens)
length(sp.gens); length(sp.gens) / dim(OTUsPA) [2]

## [1] 233
## [1] 0.04582104

overlap.gens <- intersect(re.gens, sp.gens.n)
overlap.gens

## [1] "Otu00004" "Otu00006" "Otu00010" "Otu00012" "Otu00013" "Otu00014"
## [7] "Otu00015" "Otu00016" "Otu00023" "Otu00024" "Otu00032" "Otu00034"
## [13] "Otu00036" "Otu00037" "Otu00038" "Otu00041" "Otu00042" "Otu00043"
## [19] "Otu00049" "Otu00058" "Otu00060" "Otu00062" "Otu00063" "Otu00070"
## [25] "Otu00074" "Otu00075" "Otu00078" "Otu00083" "Otu00085" "Otu00090"
## [31] "Otu00091" "Otu00094" "Otu00097" "Otu00108" "Otu00110" "Otu00117"
## [37] "Otu00120" "Otu00122" "Otu00124" "Otu00156" "Otu00161" "Otu00167"
## [43] "Otu00171" "Otu00181" "Otu00195" "Otu00198" "Otu00207" "Otu00219"
## [49] "Otu00290"

length(overlap.gens); length(overlap.gens) / length(re.gens)

## [1] 49
## [1] 0.7313433

ov.gen.tax <- re.gen.tax[which(re.gen.tax$OTU %in% overlap.gens), ]
write.table(ov.gen.tax, "../data/OverlapGeneralists.txt", quote = F,
           row.names = F, sep = "\t")

```

Figure 4: Generalists Plot

```

png(filename="../figures/Figure4.png",
     width = 1000, height = 1600, res = 96*2, bg = "white")
par(opar)

layout(matrix(1:2, nrow = 2, byrow = F))
par(mar = c(0.5, 1, 1, 1) + 0.1, oma = c(5, 6, 0, 0) + 0.1)

# Resource Generalists
#Organize Data

```

```

per.re.gens <- rowSums(OTUsREL[, c(re.gens)])
all.equal(row.names(dat), gsub("2012_RNA", "", names(per.re.gens)))

## [1] TRUE

# Stats & Prediction Frames
mod2 <- lm(per.re.gens ~ DOC.t, data = dat)
mod2.p <- round(summary(mod2)$coefficients[2,4], 3)
pred.frame1 <- data.frame(DOC.t = seq(-2, 2, 0.1))

# Initiate Plot
plot(per.re.gens ~ dat$DOC.t, # type = "n",
      xlab = "", ylab = "", xlim = c(-2, 2), ylim = c(0, 0.6), axes = F,
      pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)

# Add Regression Model
add.hull(model = mod2, pred.frame = pred.frame1)
matlines(pred.frame1, predict(mod2, interval = "c", newdata=pred.frame1),
         lty=c(2,3,3), lwd=c(4,2,2), col="black")
points(per.re.gens ~ dat$DOC.t,
       pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
legend("bottomleft", legend = bquote(italic(p) == .(mod2.p)),
      bty = "n", cex = 1.25, inset = c(-0.05, 0.01))

# Axes and Labels
#mtext("[DOC] (Transformed)", side = 1, line = 3, cex = 1.5)
mtext("Resource Generalists\n(Proportion)", side = 2, line = 3.5, cex = 1.5)
axis(1, lwd = 2, labels = F, cex.axis = 1.5)
axis(1, lwd = 2, tck = -0.02, labels = F)
axis(2, lwd = 2, labels = T, las = 1, at = c(seq(0.1, 0.9, 0.2)), cex.axis = 1.5)
axis(3, lwd = 2, tck = -0.02, labels = F)
axis(4, lwd = 2, tck = -0.02, labels = F, at = c(seq(0.1, 0.9, 0.2)))
axis(1, lwd = 2, tck = 0.02, labels = F)
axis(2, lwd = 2, tck = 0.02, labels = F, at = c(seq(0.1, 0.9, 0.2)))
axis(3, lwd = 2, tck = 0.02, labels = F)
axis(4, lwd = 2, tck = 0.02, labels = F, at = c(seq(0.1, 0.9, 0.2)))
box(lwd = 2)
abline(h = 0.5, lty = 3, lwd = 1, col = "gray50")

# Spatial Generalists
#Organize Data
per.sp.gens <- rowSums(OTUsREL[, c(sp.gens)])
all.equal(row.names(dat), gsub("2012_RNA", "", names(per.sp.gens)))

## [1] TRUE

# Stats & Prediction Frames
mod1 <- lm(per.sp.gens ~ DOC.t, data = dat)
mod1.p <- round(summary(mod1)$coefficients[2,4], 3)
pred.frame1 <- data.frame(DOC.t = seq(-2, 2, 0.1))

cor.test(~ pcoa.RES$points[, 2] + per.sp.gens)

##
## Pearson's product-moment correlation

```

```
##
## data:  pcoa.RES$points[, 2] and per.sp.gens
## t = 3.9327, df = 8, p-value = 0.00434
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.372741 0.953883
## sample estimates:
##      cor
## 0.8118389

cor.test(~ pcoa.RES$points[, 1] + per.sp.gens)

##
## Pearson's product-moment correlation
##
## data:  pcoa.RES$points[, 1] and per.sp.gens
## t = 0.22491, df = 8, p-value = 0.8277
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.5792705 0.6751948
## sample estimates:
##      cor
## 0.07926608

# Initiate Plot
plot(per.sp.gens ~ dat$DOC.t, type = "n",
     xlab = "", ylab = "", xlim = c(-2, 2), ylim = c(0.45, 0.9), axes = F,
     pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)

# Add Regression Model
add.hull(model = mod1, pred.frame = pred.frame1)
matlines(pred.frame1, predict(mod1, interval = "c", newdata=pred.frame1),
         lty=c(2,3,3), lwd=c(4,2,2), col="black")
points(per.sp.gens ~ dat$DOC.t,
       pch = 22, col = "black", bg = "gray", cex = 2, lwd = 2)
legend("bottomleft", legend = bquote(italic(p) == .(mod1.p)),
      bty = "n", cex = 1.25, inset = c(-0.05, 0.01))

# Axes and Labels
mtext("[DOM] (Transformed)", side = 1, line = 3, cex = 1.5)
mtext("Cosmopolitan Taxa\n(Proportion)", side = 2, line = 3.5, cex = 1.5)
axis(1, lwd = 2, labels = T, cex.axis = 1.5)
axis(1, lwd = 2, tck = -0.02, labels = F)
axis(2, lwd = 2, labels = T, las = 1, at = c(seq(0.1, 0.9, 0.2)), cex.axis = 1.5)
axis(3, lwd = 2, tck = -0.02, labels = F)
axis(4, lwd = 2, tck = -0.02, labels = F, at = c(seq(0.1, 0.9, 0.2)))
axis(1, lwd = 2, tck = 0.02, labels = F)
axis(2, lwd = 2, tck = 0.02, labels = F, at = c(seq(0.1, 0.9, 0.2)))
axis(3, lwd = 2, tck = 0.02, labels = F)
axis(4, lwd = 2, tck = 0.02, labels = F, at = c(seq(0.1, 0.9, 0.2)))
box(lwd = 2)
abline(h = 0.5, lty = 3, lwd = 1, col = "gray50")

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

```
## pdf
## 2

graphics.off() # shuts down open devices

img <- readPNG("../figures/Figure4.png")
grid.raster(img)
```

Figure S5: Interaction Plot

```
png(filename="../figures/FigureS5.png",
     width = 2400, height = 2400, res = 96*2, bg = "white")
par(opar)
# Custome Color Palette
jet.colors <- colorRampPalette(c("#00007F", "blue", "#007FFF", "cyan",
                                "#7FFF7F", "yellow", "#FF7F00", "red",
                                "#7F0000"))
jet.colors.W <- colorRampPalette(c("#00007F", "blue", "#007FFF", "cyan",
                                "white", "white", "white", "white",
                                "yellow", "#FF7F00", "red", "#7F0000"))

ConRes.matrix <- as.matrix(spear.ConRes5)
rownames(ConRes.matrix) <- gsub("Otu.0", "OTU", rownames(ConRes.matrix))
res.order <- order(colSums(abs(ConRes.matrix) >= 0.5), decreasing = T)
cons.order <- order(rowSums(abs(ConRes.matrix) >= 0.5), decreasing = T)
heatmap.2(ConRes.matrix[cons.order, res.order],
          col = jet.colors.W, Rowv = F, Colv = F,
          dendrogram = "none", na.rm = F, na.color = "white", trace = "none",
          density.info = "none", key.xlab = "Interaction", key.title = "",
          key.par=list(cex = 1, cex.lab = 1.5, mar = c(4, 4, 4, 4)),
          cexCol = 0.5, cexRow = 0.5, main = "",
          lhei = c(1, 5), lwid = c(1.5, 4))
mtext("Resource Availability", side = 3, line = 7, cex = 2, adj = 0.60)
#mtext("Available", side = 3, line = 0.8, cex = 1, adj = 0.2, padj = 1)
#mtext("Restrictive", side = 3, line = 0.8, cex = 1, adj = 1, padj = 1)
mtext("Consumer Strategy", side = 2, line = -3, cex = 2, adj = 0.325)
#mtext("Generalist", side = 2, line = -5, cex = 1, adj = 1, las = 1, at = 0.90)
#mtext("Specialist", side = 2, line = -5, cex = 1, adj = 1, las = 1, at = -0.12, xpd = T)
#arrows(x0 = 0.33, y0 = 1.015, x1 = 0.83, y1 = 1.015, length = 0.1, angle = 45, lwd = 3, xpd = T)
#arrows(x0 = 0.1, y0 = 0.85, x1 = 0.1, y1 = -0.08, length = 0.1, angle = 45, lwd = 3, xpd = T)

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

# ``{r fig.width=4, fig.height=4,echo=FALSE,fig.cap="Interaction Heatmap"}
img <- readPNG("../figures/FigureS5.png")
grid.raster(img)
```

Figure S6: Example Responders

```

png(filename="../figures/FigureS6.png",
     width = 1600, height = 900, res = 96*2)
par(opar)

# Example Taxa
abundancesR <- OTUsREL.dom
colnames(abundancesR) <- gsub("Otu.0", "OTU", colnames(abundancesR))
examples <- ConRes.matrix[cons.order, res.order]
otu.spread <- which((log10(apply(abundancesR, 2, max)) -
                      log10(apply(abundancesR, 2, min))) > 2)
#examples[which(row.names(examples) %in% names(otu.spread)), ]

# Set Plot Parameters
layout(matrix(c(1:2), 1, 2))
par(mar = c(1, 2, 0, 0) + 0.5, oma = c(4, 3, 0.5, 0.5))

# Negative Example
OTU.neg <- abundancesR[, which(colnames(abundancesR) == "OTU024")]
DOM.neg <- resREL.inf[, which(colnames(resREL.inf) == "C52")]
plot(OTU.neg * 100, DOM.neg * 1000, pch = 22, bg = "gray", cex = 2, lwd = 2,
     xlim = c(0, 3.5), ylim = c(0.8, 3.5), las = 1, cex.lab = 1.5,
     xlab = "", ylab = "", axes = F)
cor(OTU.neg, DOM.neg, method = "spearman")
lines(stats::lowess(DOM.neg * 1000 ~ OTU.neg * 100), lwd = 4, lty = 2)
axis(side=1, lwd.ticks = 2, tck=-0.04, labels = T, cex.axis = 0.8, las = 1)
axis(side=1, lwd.ticks = 2, tck=0.01, labels = F, cex.axis = 1)
axis(side=2, lwd.ticks = 2, tck=0.01, labels = T, cex.axis = 0.8, las = 1)
axis(side=2, lwd.ticks = 2, tck=-0.04, labels = F, cex.axis = 1)
axis(side=3, lwd.ticks = 2, tck=-0.02, labels = F, cex.axis = 1)
axis(side=3, lwd.ticks = 2, tck=0.01, labels = F, cex.axis = 1)
axis(side=4, lwd.ticks = 2, tck=-0.02, labels = F, cex.axis = 1)
axis(side=4, lwd.ticks = 2, tck=0.01, labels = F, cex.axis = 1)
box(lwd = 2)

# Positive Example
# examples[which(row.names(examples) %in% names(otu.spread)), which(colnames(examples) == "C789")]
OTU.pos <- abundancesR[, which(colnames(abundancesR) == "OTU070")]
DOM.pos <- resREL.inf[, which(colnames(resREL.inf) == "C789")]
plot(OTU.pos * 100, DOM.pos * 1000, pch = 22, bg = "gray", cex = 2, lwd = 2,
     xlim = c(0, 1.8), ylim = c(0, 1.15), las = 1, cex.lab = 1.5,
     xlab = "", ylab = "", axes = F)
cor(OTU.pos, DOM.pos, method = "spearman")
lines(stats::lowess(DOM.pos * 1000 ~ OTU.pos * 100), lwd = 4, lty = 2)
axis(side=1, lwd.ticks = 2, tck=-0.04, labels = T, cex.axis = 0.8, las = 1)
axis(side=1, lwd.ticks = 2, tck=0.01, labels = F, cex.axis = 1)
axis(side=2, lwd.ticks = 2, tck=0.01, labels = T, cex.axis = 0.8, las = 1)
axis(side=2, lwd.ticks = 2, tck=-0.04, labels = F, cex.axis = 1)
axis(side=3, lwd.ticks = 2, tck=-0.02, labels = F, cex.axis = 1)
axis(side=3, lwd.ticks = 2, tck=0.01, labels = F, cex.axis = 1)
axis(side=4, lwd.ticks = 2, tck=-0.02, labels = F, cex.axis = 1)
axis(side=4, lwd.ticks = 2, tck=0.01, labels = F, cex.axis = 1)
box(lwd = 2)

```

```

mtext("Species Abundance (%)", 1, line = 2, cex = 1.25, outer = T)
mtext("Resource Abundance (\u2030)", 2, line = 1, cex = 1.25, outer = T)

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

img <- readPNG("../figures/FigureS6.png")
grid.raster(img)

```

Who are the Generalists and Specialists

```

# length(specialists)
# length(generalists)
#
# generalists
# OTU.tax[,1] <- gsub("Otu000", "OTU", OTU.tax[,1])
# Gen.Tax <- OTU.tax[which(OTU.tax[,1] %in% generalists),]
# Spe.Tax <- OTU.tax[which(OTU.tax[,1] %in% specialists),]
#
# Gen.Tax.Class <- Gen.Tax$Class
# Spe.Tax.Class <- Spe.Tax$Class
# Gen.Tax.Gen <- Gen.Tax$Genus
# Spe.Tax.Gen <- Spe.Tax$Genus
#
# unique(Gen.Tax.Gen)
# unique(Spe.Tax.Gen)
#
# table(Gen.Tax.Gen)
# table(Spe.Tax.Gen)
#
# setdiff(unique(Gen.Tax.Gen), unique(Spe.Tax.Gen))
# setdiff(unique(Spe.Tax.Gen), unique(Gen.Tax.Gen))
# intersect(unique(Gen.Tax.Gen), unique(Spe.Tax.Gen))

```

Resource Generalists

```

# per.gens <- rowSums(OTUsREL[, c(generalists)])
# all.equal(row.names(dat), gsub("2012_RNA", "", names(per.gens)))
# plot(per.gens ~ dat$DOC.t)
# abline(h = 0.5)
#
# per.spes <- rowSums(OTUsREL[, c(specialists)])
# all.equal(row.names(dat), gsub("2012_RNA", "", names(per.spes)))
# plot(per.spes ~ dat$DOC.t)
#
# length(generalists)
#
# abundancesR <- OTUsREL
# colnames(abundancesR) <- gsub("Otu.0", "OTU", colnames(abundancesR))
# rownames(abundancesR) <- gsub("2012_RNA", "", rownames(abundancesR))

```



```

#
# spec.matrix <- matrix(NA, nrow = 10, ncol = 6)
# colnames(spec.matrix) <- c("Total.Rel.Gen", "Active.Rel.Gen", "Total.Num.Gen",
#                             "Total.Rel.Spe", "Active.Rel.Spe", "Total.Num.Spe")
# rownames(spec.matrix) <- rownames(abundancesD)
#
# for (i in 1:(dim(spec.matrix)[1])){
#   temp.gensT <- abundancesD[i, colnames(abundancesD) %in% generalists]
#   temp.gensA <- abundancesR[i, colnames(abundancesR) %in% generalists]
#   temp.specT <- abundancesD[i, colnames(abundancesD) %in% specialists]
#   temp.specA <- abundancesR[i, colnames(abundancesR) %in% specialists]
#   spec.matrix[i,1] <- round(sum(temp.gensT), 3)
#   spec.matrix[i,2] <- round(sum(temp.gensA), 3)
#   spec.matrix[i,3] <- sum(temp.gensT > 0)
#   spec.matrix[i,4] <- round(sum(temp.specT), 3)
#   spec.matrix[i,5] <- round(sum(temp.specA), 3)
#   spec.matrix[i,6] <- sum(temp.specA > 0)
# }
#
#
# t.test(spec.matrix[, 2], spec.matrix[,4], var.equal = T)
#

```