

# Supplemental 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("bioDist"); require("gplots")
#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
```

## Import Total Community

```
# 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$Molecule == "DNA" & design$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)

##
##          Acidobacteria_Gp1
##                   105
##          Acidobacteria_Gp10
##                   15
##          Acidobacteria_Gp11
##                   11
##          Acidobacteria_Gp13
##                   16
##          Acidobacteria_Gp15
##                   3
##          Acidobacteria_Gp16
##                   48
##          Acidobacteria_Gp17
##                   23
##          Acidobacteria_Gp18
##                   38
##          Acidobacteria_Gp19
##                   2
##          Acidobacteria_Gp2
```

```

##                11
##      Acidobacteria_Gp22
##                12
##      Acidobacteria_Gp23
##                13
##      Acidobacteria_Gp25
##                1
##      Acidobacteria_Gp3
##                210
##      Acidobacteria_Gp4
##                46
##      Acidobacteria_Gp5
##                5
##      Acidobacteria_Gp6
##                61
##      Acidobacteria_Gp7
##                21
##      Acidobacteria_Gp9
##                1
##      Acidobacteria_unclassified
##                20
##      Actinobacteria
##                1193
##      Actinobacteria_unclassified
##                1
##      Alphaproteobacteria
##                3480
##      Aminicenantes_unclassified
##                31
##      Anaerolineae
##                363
##      Ardenticatenia
##                5
##      Armatimonadetes_gp2
##                33
##      Armatimonadetes_gp4
##                6
##      Armatimonadetes_gp5
##                38
##      Armatimonadetes_unclassified
##                1
##      Armatimonadia
##                14
##      Bacilli
##                119
##      Bacteria_unclassified
##                1378
##      Bacteroidetes_unclassified
##                133
##      Bacteroidia
##                408
##      Betaproteobacteria
##                2456
##      BRC1_unclassified

```

```

##                                12
##                                Caldilineae
##                                60
## candidate_division_WPS-1_unclassified
##                                130
## candidate_division_WPS-2_unclassified
##                                10
##   candidate_division_ZB3_unclassified
##                                30
##           Candidatus_Cloacamonas
##                                15
##           Candidatus_Hydrogenedens
##                                41
##           Chitinivibrionia
##                                9
##           Chlamydiia
##                                339
##           Chlorobia
##                                11
##           Chloroflexi_unclassified
##                                101
##           Chloroflexia
##                                31
##           Chthonomonadetes
##                                41
##           Clostridia
##                                1227
##           Cytophagia
##                                318
##           Deferribacteres
##                                6
##           Dehalococcoidetes
##                                29
##           Dehalococcoidia
##                                6
##           Deinococci
##                                18
##           Deltaproteobacteria
##                                2380
##           Dictyoglomia
##                                2
##           Elusimicrobia
##                                12
##           Elusimicrobia_unclassified
##                                7
##           Endomicrobia
##                                35
##           Epsilonproteobacteria
##                                96
##           Erysipelotrichia
##                                28
##           Fibrobacteres_unclassified
##                                1
##           Fibrobacteria

```

##	6
##	Fimbriimonadia
##	11
##	Firmicutes_unclassified
##	370
##	Flavobacteriia
##	308
##	Fusobacteriia
##	27
##	Gammaproteobacteria
##	2467
##	Gemmatimonadetes
##	58
##	Holophagae
##	58
##	Ignavibacteria
##	62
##	Ktedonobacteria
##	13
##	Latescibacteria_unclassified
##	64
##	Lentisphaerae_unclassified
##	2
##	Lentisphaeria
##	8
##	Microgenomates_unclassified
##	6
##	Mollicutes
##	9
##	Negativicutes
##	89
##	Nitrospira
##	13
##	Oligoflexia
##	21
##	Oligosphaeria
##	20
##	Omnitrophica_unclassified
##	32
##	Opitutae
##	173
##	Parcubacteria_unclassified
##	386
##	Phycisphaerae
##	105
##	Planctomycetes_unclassified
##	70
##	Planctomycetia
##	1206
##	Proteobacteria_unclassified
##	679
##	Spartobacteria
##	323
##	Sphingobacteriia

```
##          735
##          Spirochaetia
##          151
##          SR1_unclassified
##          39
##          Subdivision3
##          594
##          Subdivision5
##          138
##          Synergistia
##          3
##          Thermoflexia
##          1
##          Thermoleophilia
##          8
##          Thermomicrobia
##          33
##          Thermotogae
##          1
##          Verrucomicrobia_unclassified
##          82
##          Verrucomicrobiae
##          259
```

```
table(OTU.tax.2$Phylum)
```

```
##
##          Acidobacteria          Actinobacteria          Aminicenantes
##          720          1202          31
##          Armatimonadetes    Bacteria_unclassified    Bacteroidetes
##          144          1378          1902
##          BRC1 candidate_division_WPS-1 candidate_division_WPS-2
##          12          130          10
##          candidate_division_ZB3          Chlamydiae          Chlorobi
##          30          339          11
##          Chloroflexi          Cloacimonetes          Deferribacteres
##          642          15          6
##          Deinococcus-Thermus          Dictyoglomi          Elusimicrobia
##          18          2          54
##          Fibrobacteres          Firmicutes          Fusobacteria
##          16          1833          27
##          Gemmatimonadetes          Hydrogenedentes          Ignavibacteriae
##          58          41          62
##          Latescibacteria          Lentisphaerae          Microgenomates
##          64          30          6
##          Nitrospirae          Omnitrophica          Parcubacteria
##          13          32          386
##          Planctomycetes          Proteobacteria          Spirochaetes
##          1381          11579          151
##          SR1          Synergistetes          Tenericutes
##          39          3          9
##          Thermotogae          Verrucomicrobia
##          1          1569
```

```

# 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 5684

# 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

```

## Load DOM Profiles

```

# Define Inputs
resource.neg <- "../data/SpecAbundAveNeg.csv"
annotations <- "../data/annotationSummary_MFconversion_output.csv"

# 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

```

## 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.simpsE, res.shan)

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

## [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.simpsE)

## [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)

```

## Community Alpha Diversity

```

# Total Community Alpha# 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), ]

# 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$S.rar
## W = 0.68855, p-value = 0.0006402
shapiro.test(dat$simpsE.rar) # Normal

##
## Shapiro-Wilk normality test
##
## data: dat$simpsE.rar
## W = 0.9373, p-value = 0.5233
shapiro.test(dat$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 = 0.91346, p-value = 0.3791
# Transform DOC and S.rar with Box-Cox
D.power <- powerTransform(dat$DOC)
S.power <- powerTransform(dat$S.rar)
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 = 0.95708, p-value = 0.7521
# 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
## -0.8903 -0.6265  0.1937  0.3904  0.9846
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.04650   10.77388   3.346  0.0101 *
## S.res        -0.06497    0.01942  -3.346  0.0101 *

```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6847 on 8 degrees of freedom
## Multiple R-squared:  0.5833, Adjusted R-squared:  0.5312
## F-statistic: 11.2 on 1 and 8 DF,  p-value: 0.01013

##
## Call:
## lm(formula = simpsE.rar ~ S.res, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.0164030 -0.0079378 -0.0007338  0.0055267  0.0204006
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.5013339   0.1864667  -2.689   0.0276 *
## S.res         0.0009577   0.0003360    2.850   0.0215 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01185 on 8 degrees of freedom
## Multiple R-squared:  0.5038, Adjusted R-squared:  0.4418
## F-statistic: 8.123 on 1 and 8 DF,  p-value: 0.02148

mod4.p <- round(summary(mod4)$coefficients[2,4], 3)

pred.frame2 <- data.frame(S.res = seq(525, 572, 1))

png(filename="../figures/FigureS7.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, 6, 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, 0.07), 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 = 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\n(Total Community)", side = 2, line = 4, 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/FigureS7.png")
grid.raster(img)

```

## Community Beta Diversity

```

# Total Community Beta
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)
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)
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  0.85723 1.6431 0.032 *
## Residual   6  1.04340
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

RsquareAdj(dbRDA.dom)

## $r.squared
## [1] 0.4510238
##
## $adj.r.squared
## [1] 0.1765357

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  0.54701 3.1455 0.029 *
## CAP2      1  0.17208 0.9895 0.847
## CAP3      1  0.13814 0.7944 0.733
## Residual   6  1.04340
## ---
## 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,]  0.19142  0.98151 0.2467 0.376
## [2,] -0.99878 -0.04930 0.7276 0.009 **
## [3,]  0.52501 -0.85110 0.0399 0.859
## ---
## 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

# Resource Concentration dbRDA
hmf.bray.REL <- vegdist(OTUsREL.log, method = "bray")
pcoa.rel <- cmdscale(hmf.bray.REL, eig = TRUE, k = 3)
dbRDA <- capscale(hmf.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 = hmf.bray.REL ~ dat$DOC.t, comm = OTUsREL.log, add = T)
##      Df SumOfSqs      F Pr(>F)
## Model    1  0.50647 2.8894 0.0031 **
## Residual  8  1.40227
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

RsquareAdj(dbRDA)

## $r.squared
## [1] 0.2653416
##
## $adj.r.squared
## [1] 0.1735093

# DOM Diversity dbRDA; using: hmf.bray.res; pcoa.res
# Calculate Bray-Curtis
hmf.bray.res <- vegdist(resREL.neg, method = "bray")
hmf.bray.res.log <- vegdist(resREL.neg.log, method = "bray")
hmf.bcd.res <- vegdist(box.cox.chord(res.neg), method = "euclidean")
pcoa.res <- cmdscale(hmf.bray.res.log, eig = TRUE, k = 3)
dbRDA.dom <- capscale(hmf.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 = hmf.bray.REL ~ pcoa.res$points[, 1:3], add = T)
##      Df SumOfSqs      F Pr(>F)
## Model    3  0.86375 1.6531 0.033 *
## Residual  6  1.04498
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

RsquareAdj(dbrDA.dom)

## $r.squared
## [1] 0.452526
##
## $adj.r.squared
## [1] 0.178789

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  0.55667 3.1962 0.038 *
## CAP2       1  0.17125 0.9833 0.830
## CAP3       1  0.13583 0.7799 0.742
## Residual    6  1.04498
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# PCoA of Total 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/FigureS8.png",
     width = 1300, height = 900, 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 = 1.5, 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$points[,1] +
     c(0.04, 0, 0, 0, 0, 0, -0.04, -0.02, 0.02),
     pcoa.BAC$points[,2] +
     c(-0.04, 0.04, 0.04, -0.04, 0.04, 0.04, 0.04, -0.04, 0.04, -0.04),
     labels = dat$Lake, , col = "black", cex = 0.8)

# DOM Composition Vectors
cor.test(~ pcoa.RES$points[, 2] + pcoa.BAC$points[, 1])

##
## Pearson's product-moment correlation
##
## data: pcoa.RES$points[, 2] and pcoa.BAC$points[, 1]
## t = -4.6147, df = 8, p-value = 0.001722
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.9644759 -0.4813889
## sample estimates:
## cor
## -0.8525972

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 = 1.25, font = 3)
text(-com.arrows[2, 1] * 1.2, com.arrows[2, 2] + 0.08, "DOM 2",

```

```
col = "gray40", cex = 1.25, font = 3)

dev.off() # this writes plot to folder

## pdf
## 2
graphics.off() # shuts down open devices

img <- readPNG("../figures/FigureS8.png")
grid.raster(img)
```