Dormancy and dispersal structure bacterial communities across ecosystem boundaries

Nathan I. Wisnoski, Mario E. Muscarella, Megan L. Larsen, and Jay T. Lennon

24 January, 2019

Initial Setup

First, we'll load the packages we'll need for the analysis, as well as some other functions.

```
# Import Required Packages
library("png")
library("grid")
library("tidyverse")
library("vegan")
library("xtable")
library("viridis")
library("cowplot")
library("adespatial")
library("ggrepel")
library("gganimate")
library("maps")
library("rgdal")
library("iNEXT")
library("officer")
library("flextable") #must have gdtools installed also
library("broom")
library("ggpmisc")
library("pander")

source("bin/mothur_tools.R")
se <- function(x, ...){sd(x, na.rm = TRUE)/sqrt(length(na.omit(x)))}
```

Next, we'll set the aesthetics of the figures we will produce.

```
my.cols <- RColorBrewer::brewer.pal(n = 4, name = "Greys")[3:4]

# Set theme for figures in the paper
theme_set(theme_classic() +
  theme(axis.title = element_text(size = 20),
        axis.title.x = element_text(margin = margin(t = 15, b = 15)),
        axis.title.y = element_text(margin = margin(l = 15, r = 15)),
        axis.text = element_text(size = 14),
        axis.text.x = element_text(margin = margin(t = 5)),
        axis.text.y = element_text(margin = margin(r = 5)),
        #axis.line.x = element_line(size = 1),
        #axis.line.y = element_line(size = 1),
        axis.line.x = element_blank(),
        axis.line.y = element_blank(),
        axis.ticks.x = element_line(size = 1),
        axis.ticks.y = element_line(size = 1),
```

```
        axis.ticks.length = unit(.1, "in"),
        panel.border = element_rect(color = "black", fill = NA, size = 1.5),
        legend.title = element_blank(),
        legend.text = element_text(size = 16)
        ))
```

Import Data

Here, we read in the processed sequence files from mothur (shared and taxonomy) and a design of the sampling. We also load in the environmental data. We then remove the mock community from the dataset and ensure the the design and OTU table are aligned by row.

```
# Define Inputs
# Design = general design file for experiment
# shared = OTU table from mothur with sequence similarity clustering
# Taxonomy = Taxonomic information for each OTU
design <- "data/UL.design.txt"
shared <- "data/ul_resgrad.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.pick.opti_m
taxon  <- "data/ul_resgrad.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.pick.opti_m

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

# Import Shared Files
OTUs <- read.otu(shared = shared, cutoff = "0.03")    # 97% Similarity

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

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

# Subset to just the reservoir gradient sites
OTUs <- OTUs[str_which(rownames(OTUs), "RG"),]
OTUs <- OTUs[-which(rownames(OTUs) == "RGMockComm"),]

# make sure OTU table matches up with design order
OTUs <- OTUs[match(rownames(design), rownames(OTUs)),]
```

Clean and transform OTU table

Here, we remove OTUs with low incidence across sites, we remove any samples with low coverage, and we standardize the OTU table by log-transforming the abundances and relativizing by site.

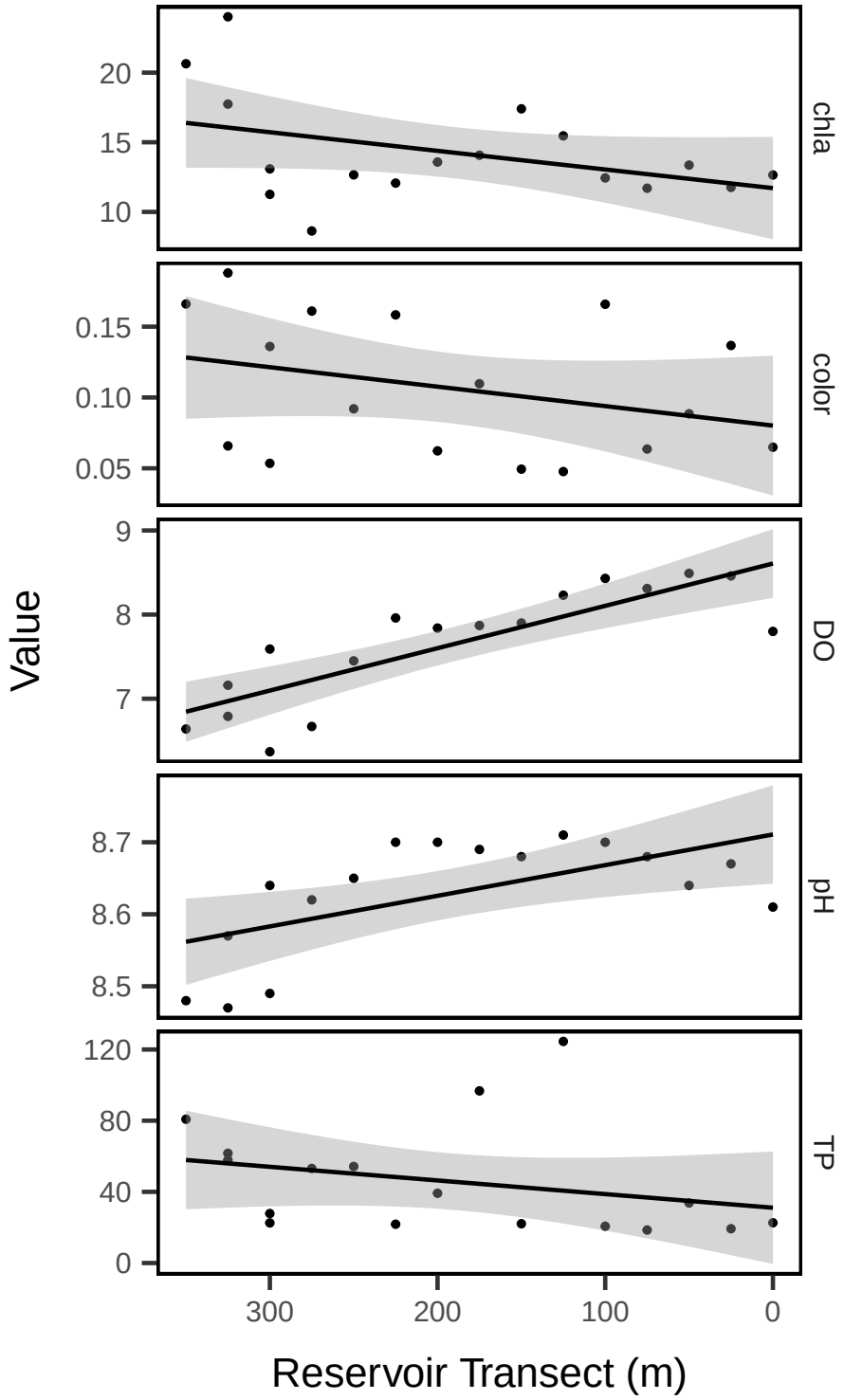
```
# Remove OTUs with less than two occurences across all sites
OTUs <- OTUs[, which(colSums(OTUs) >= 2)]

# Sequencing Coverage
coverage <- rowSums(OTUs)

# Remove Low Coverage Samples (This code removes two sites: Site 5DNA, Site 6cDNA)
```

```
lows <- which(coverage < 10000)
OTUs <- OTUs[-which(coverage < 10000), ]
design <- design[-which(coverage < 10000), ]
# Remove OTUs with less than two occurences across all sites
OTUs <- OTUs[, which(colSums(OTUs) >= 2)]
# OTUs <- rrarefy(OTUs, min(coverage))

# Make Relative Abundance Matrices
OTUsREL <- decostand(OTUs, method = "total")

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

Reservoir environmental gradients

Just to see if there are any strong underlying resource or nutrient gradients in the reservoir, we'll plot them along the distance of the reservoir.

```
env.dat %>% select(dist.dam, DO, pH, TP, color, chla) %>%
  gather(variable, value, -dist.dam) %>%
  ggplot(aes(x = dist.dam, y = value)) +
  geom_point() +
  geom_smooth(method = "lm", color = "black") +
  facet_grid(variable ~., scales = "free") +
  theme(strip.background = element_blank(), strip.text = element_text(size = 14)) +
  labs(x = "Reservoir Transect (m)",
       y = "Value") +
  scale_x_reverse()
```



So, there are some weak gradients, but nothing too prevailing.

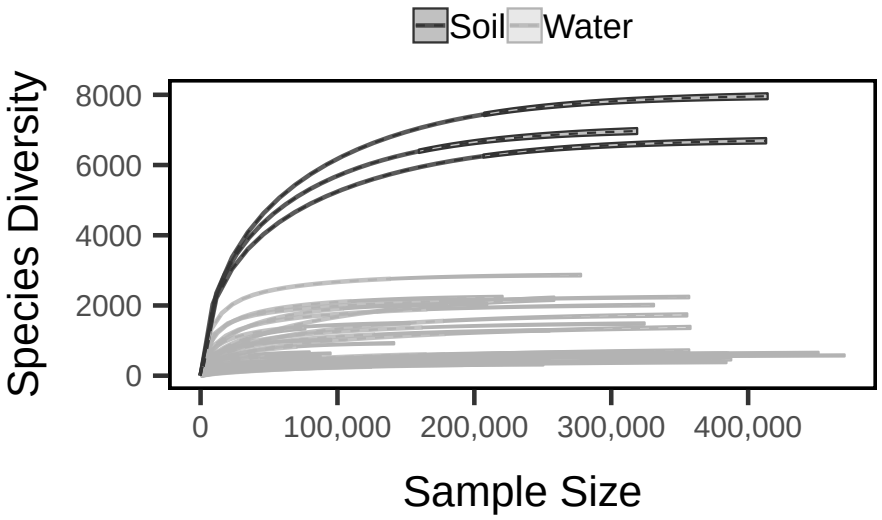
Analyze Diversity

Now, we will analyze the bacterial diversity in the reservoir and nearby soils to figure out how well they support different mechanisms of community assembly.

How does α-diversity vary along the reservoir?

First, we use the method of rarefaction and extrapolation developed by Chao et al. in the iNEXT package.

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

# Simpson's Evenness
SimpE <- function(x = ""){
  x <- as.data.frame(x)
  D <- diversity(x, "inv")
  S <- sum((x > 0) * 1)
  E <- (D)/S
  return(E)
}
simpsE <- round(apply(OTUs, 1, SimpE), 3)
shan <- diversity(OTUs, index = "shannon")
exp.shan <- exp(shan)
alpha.div <- cbind(design, S.obs, simpsE, shan, exp.shan)


# # estimate asymptotic richness
# divestim <- iNEXT(t(OTUs), datatype = "abundance", nboot = 999)
# saveRDS(divestim, file = "intermediate-data/inext-output-999boots.rda")
divestim <- read_rds("intermediate-data/inext-output-999boots.rda")
divestim.df <- fortify(divestim) %>%
  mutate(habitat = str_to_title(design[as.character(site),"type"]))

divestim.df %>%
  ggplot(aes(x = x, y = y,
             ymin = y.lwr, ymax = y.upr,
             color = habitat, fill = habitat, group = site)) +
  geom_ribbon(data=subset(divestim.df, method == "extrapolated"), alpha = 0.3) +
  geom_line(data=subset(divestim.df, method == "interpolated"), size = 1, alpha = .8) +
  geom_line(alpha = 1, linetype = "dashed") +
  scale_x_continuous(labels = scales::comma) +
  labs(x = "Sample Size", y = "Species Diversity") +
  theme(legend.position = "top") +
  scale_color_grey(end = .7) +
  scale_fill_grey(end = .7)
```



Next, we'll extract the estimates for the Hill numbers at different levels of q, which differentially weight common versus rare species.

```
hill.estim <- divestim$AsyEst %>% filter(Diversity == "Species richness") %>%
  left_join(rownames_to_column(alpha.div), by = c("Observed" = "S.obs")) %>%
  select(Site, rowname, station, molecule, type, distance) %>%
  left_join(divestim$AsyEst, by = "Site")

hill.water <- as_tibble(hill.estim) %>% filter(type == "water")
hill.water.rich <- subset(hill.water, Diversity == "Species richness")
hill.water.shan <- subset(hill.water, Diversity == "Shannon diversity")
hill.water.simp <- subset(hill.water, Diversity == "Simpson diversity")

hill.water.mod.rich <- lm(Estimator ~ distance * molecule, data = hill.water.rich)
hill.water.mod.shan <- lm(Estimator ~ distance * molecule, data = hill.water.shan)
hill.water.mod.simp <- lm(Estimator ~ distance * molecule, data = hill.water.simp)

summary(hill.water.mod.rich)
```

```
##
## Call:
## lm(formula = Estimator ~ distance * molecule, data = hill.water.rich)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -518.38 -137.60    0.71   98.61  718.25
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          468.6274   128.2862   3.653 0.000918 ***
## distance               5.2143     0.5397   9.662 5.22e-11 ***
## moleculeRNA          104.9450   166.1694   0.632 0.532164
## distance:moleculeRNA  -5.1222     0.7163  -7.151 4.07e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 258.7 on 32 degrees of freedom
## Multiple R-squared:  0.8714, Adjusted R-squared:  0.8593
```

```
## F-statistic: 72.26 on 3 and 32 DF,  p-value: 2.431e-14
```

```
summary(hill.water.mod.shan)
```

```
##
## Call:
## lm(formula = Estimator ~ distance * molecule, data = hill.water.shan)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -123.915  -14.841   -3.500    6.902  157.964
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           55.5031    24.9703   2.223  0.03342 *
## distance               0.2892     0.1050   2.753  0.00965 **
## moleculeRNA          -35.5144    32.3441  -1.098  0.28039
## distance:moleculeRNA  -0.2905     0.1394  -2.084  0.04525 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 50.35 on 32 degrees of freedom
## Multiple R-squared:  0.5556, Adjusted R-squared:  0.514
## F-statistic: 13.34 on 3 and 32 DF,  p-value: 8.138e-06
```

```
summary(hill.water.mod.simp)
```

```
##
## Call:
## lm(formula = Estimator ~ distance * molecule, data = hill.water.simp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.589  -7.222  -0.977   6.321  39.440
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           31.61377    7.45207   4.242 0.000176 ***
## distance               0.05218    0.03135   1.664 0.105804
## moleculeRNA          -22.18812    9.65268  -2.299 0.028205 *
## distance:moleculeRNA  -0.04408    0.04161  -1.059 0.297369
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.02 on 32 degrees of freedom
## Multiple R-squared:  0.5693, Adjusted R-squared:  0.529
## F-statistic:  14.1 on 3 and 32 DF,  p-value: 4.985e-06
```

```
hill.water.mods <- as_tibble(rbind.data.frame(
  tidy(hill.water.mod.rich) %>% add_column(Diversity = "Species richness"),
  tidy(hill.water.mod.shan) %>% add_column(Diversity = "Shannon diversity"),
  tidy(hill.water.mod.simp) %>% add_column(Diversity = "Simpson diversity")
))

# Summary table of the model results.
hill.water.mods %>%
```

```
  group_by(Diversity) %>%
  rename("Term" = term,
         "Estimate" = estimate,
         "Std. Error" = std.error,
         "Statistic" = statistic,
         "p-value" = p.value) %>%
  filter(Term != "(Intercept)") %>%
  select(Diversity, everything()) %>%
  mutate_if(is.double, signif, digits = 3) %>%
  pander()
```

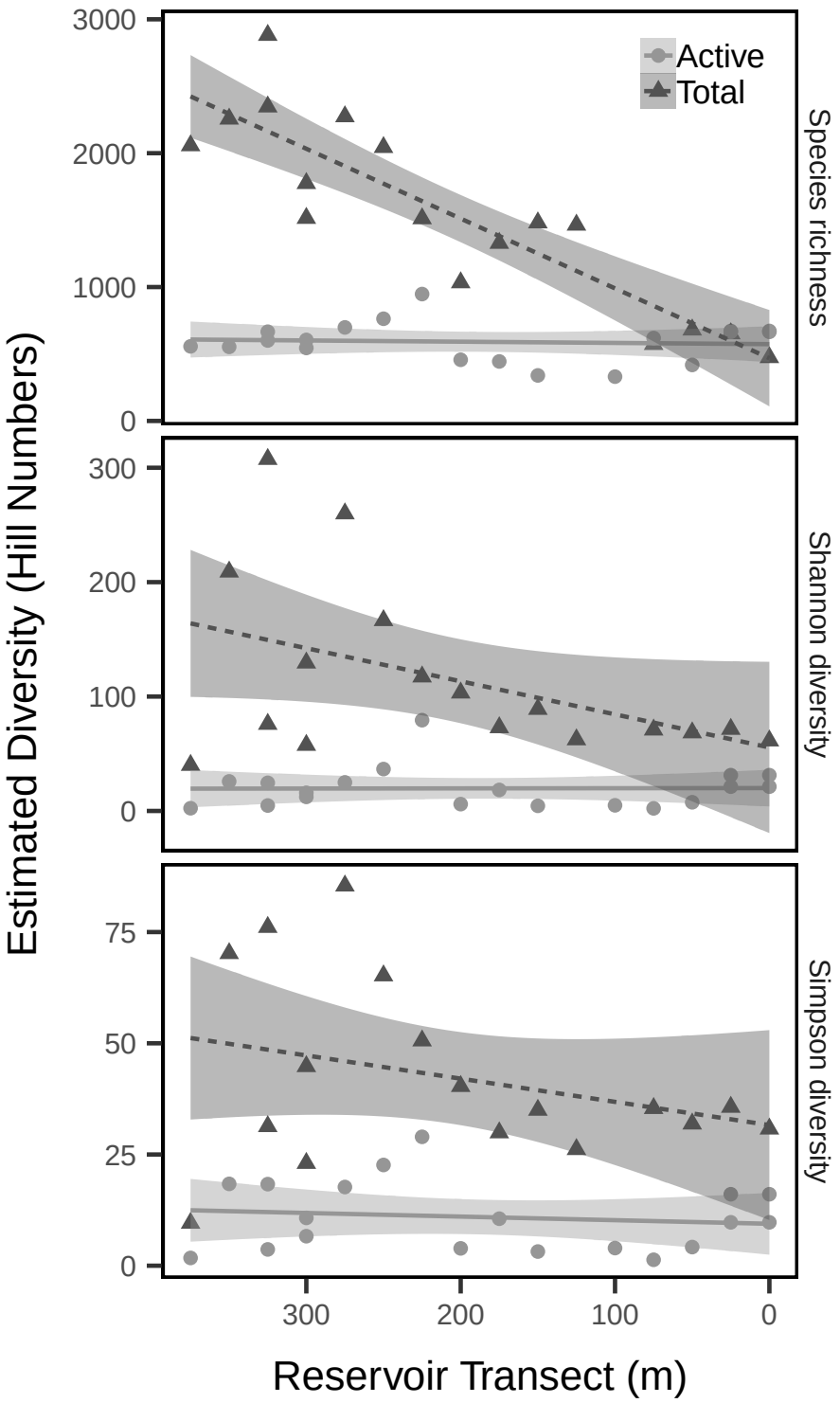
Table 1: Table continues below

Diversity	Term	Estimate	Std. Error	Statistic
Species richness	distance	5.21	0.54	9.66
Species richness	moleculeRNA	105	166	0.632
Species richness	distance:moleculeRNA	-5.12	0.716	-7.15
Shannon diversity	distance	0.289	0.105	2.75
Shannon diversity	moleculeRNA	-35.5	32.3	-1.1
Shannon diversity	distance:moleculeRNA	-0.291	0.139	-2.08
Simpson diversity	distance	0.0522	0.0313	1.66
Simpson diversity	moleculeRNA	-22.2	9.65	-2.3
Simpson diversity	distance:moleculeRNA	-0.0441	0.0416	-1.06

p-value
5.22e-11
0.532
4.07e-08
0.00965
0.28
0.0453
0.106
0.0282
0.297

```
hill.estim %>% filter(type == "water") %>%
  mutate(molecule = ifelse(molecule == "DNA", "Total", "Active")) %>%
  ggplot(aes(x = distance, y = Estimator,
             ymin = LCL, ymax = UCL,
             color = molecule, fill = molecule, shape = molecule)) +
  geom_point(size =3) +
  # geom_errorbar(size = .5, aes(ymin = Estimator - s.e., ymax = Estimator + s.e.),
  #                width = 10, alpha = 0.5) +
  geom_smooth(method = "lm", aes(linetype = molecule)) +
  labs(x = "Reservoir Transect (m)",
       y = "Estimated Diversity (Hill Numbers)") +
  scale_color_manual(values = my.cols) +
  scale_fill_manual(values = my.cols) +
  theme(legend.position = c(.85,.95),
        strip.text = element_text(size = 14),
```

```
      strip.background = element_blank()) +
scale_x_reverse() +
facet_grid(Diversity ~ ., scales = "free")
```



So, from the basis of these results, we can make the following conclusions. First, we note that diversity in the total community decays from the stream inlet to the dam of the reservoir. That is, all the lines have a negative slope. However, we do not see this decay in the metabolically active community. Second, we note

that the metabolically actively community has much lower diversity than the total community near the soils, but this difference decreases toward the dam. Last, because we quantified diversity across three orders of Hill numbers (q = 0, 1, and 2), we can also say something about the relative importance of rare versus common taxa along the reservoir transect. We see the the significance of the distance-by-molecule interaction term decrease as rare taxa are downweighted in favor of common taxa. This suggests that the differences between the active and total communities along the transect is driven primarily by rare taxa. However, the general trend of higher Simpson diversity across the whole transect suggests that low-activity, but relatively common, taxa are maintained in the reservoir.

Alpha Diversity Across Gradient

```
alpha.div <- divestim$AsyEst %>% filter(Diversity == "Species richness") %>%
  left_join(rownames_to_column(alpha.div), by = c("Observed" = "S.obs"))
# Seperate data based on lake and soil samples
lake <- alpha.div[alpha.div$type == "water",]
soil <- alpha.div[alpha.div$type == "soil", ]


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

```
##
## Call:
## lm(formula = lake$Estimator ~ lake$distance * lake$molecule)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -518.38 -137.60    0.71   98.61  718.25
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   468.6274   128.2862   3.653 0.000918 ***
## lake$distance                   5.2143     0.5397   9.662 5.22e-11 ***
## lake$moleculeRNA              104.9450   166.1694   0.632 0.532164
## lake$distance:lake$moleculeRNA  -5.1222     0.7163  -7.151 4.07e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 258.7 on 32 degrees of freedom
## Multiple R-squared:  0.8714, Adjusted R-squared:  0.8593
## F-statistic: 72.26 on 3 and 32 DF,  p-value: 2.431e-14
```

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

# Average Richess in Terrestrial Habitat
mean(soil$Estimator)
```

```
## [1] 7276.258
```

```
# Dummy Variables Regression Model ("Species Richness")
D1 <- (lake$molecule == "RNA")*1
```

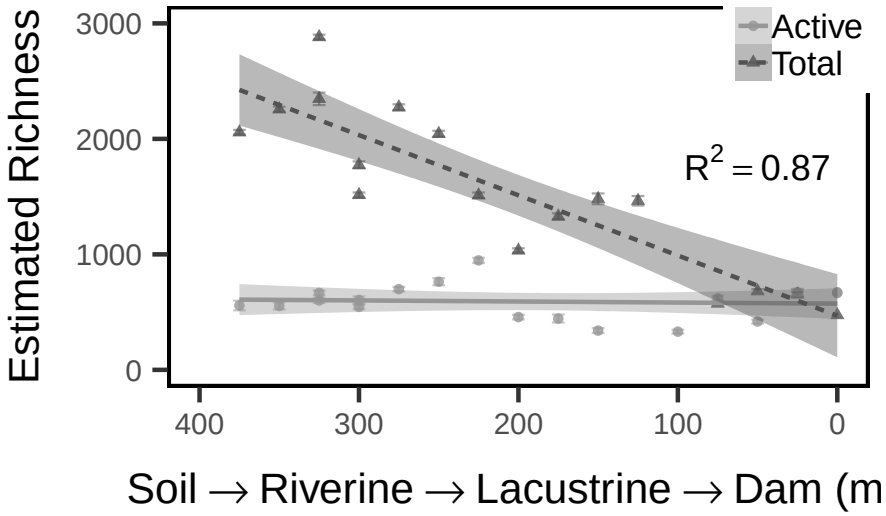
```r
fit.Fig.3a <- lm(lake$Estimator ~ lake$distance + D1 + lake$distance*D1)
summary(fit.Fig.3a)
```

```
##
## Call:
## lm(formula = lake$Estimator ~ lake$distance + D1 + lake$distance *
##     D1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -518.38 -137.60    0.71   98.61  718.25
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      468.6274   128.2862   3.653 0.000918 ***
## lake$distance      5.2143     0.5397   9.662 5.22e-11 ***
## D1               104.9450   166.1694   0.632 0.532164
## lake$distance:D1  -5.1222     0.7163  -7.151 4.07e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 258.7 on 32 degrees of freedom
## Multiple R-squared:  0.8714, Adjusted R-squared:  0.8593
## F-statistic: 72.26 on 3 and 32 DF,  p-value: 2.431e-14
```

```r
D1.R2 <- round(summary(fit.Fig.3a)$r.squared, 2)

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

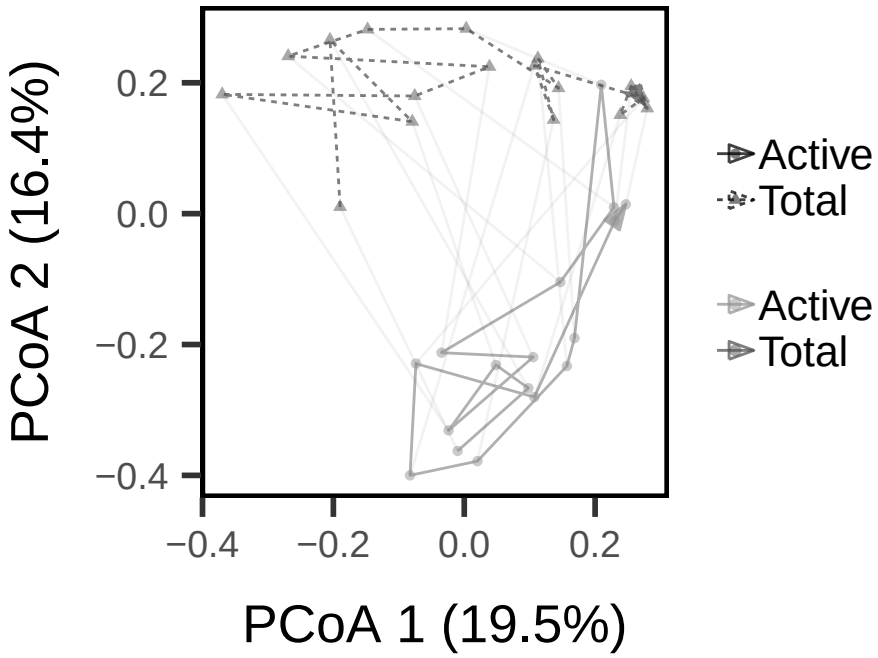
Figure 2: Estimated Richness

```r
lake %>%
  mutate(molecule = ifelse(molecule == "DNA", "Total", "Active")) %>%
  ggplot(aes(x = distance, y = Estimator, color = molecule, fill = molecule, shape = molecule)) +
  geom_point(size = 2, alpha = 0.8, show.legend = T) +
  geom_errorbar(aes(ymin = Estimator - s.e., ymax = Estimator + s.e.), width = 8, alpha = 0.5) +
  geom_smooth(method = 'lm', show.legend = T, aes(linetype = molecule)) +
  annotate(geom = "text", x = 50, y = 1800, size = 6,
           label = paste0("R^2== ",D1.R2), parse = T) +
  scale_x_reverse(limits = c(400,0)) +
  labs(x = (expression("Soil" %->% "Riverine" %->% "Lacustrine" %->% "Dam (m.)")),
       y = "Estimated Richness") +
  theme(legend.position = c(0.9, 0.9)) +
  scale_color_manual(values = my.cols) +
  scale_fill_manual(values = my.cols) +
  scale_y_continuous(limits = c(0, 3000)) +
  ggsave("figures/02_Richness.pdf", bg = "white", width = 7, height = 6) +
  ggsave("figures/02_Richness.png", width = 7, height = 6, dpi = 500)
```



Track structure across space

```
ul.pcoa <- cmdscale(vegdist(OTUsREL.log, method="bray"), 2, eig = T, add = T)
explainvars <- round(eigenvals(ul.pcoa)[c(1,2)]/sum(eigenvals(ul.pcoa)),3) *100
water.pcvals <- data.frame(scores(ul.pcoa)) %>%
  rownames_to_column("name") %>%
  left_join(rownames_to_column(design, "name")) %>%
  arrange(desc(distance)) %>% filter(type == "water")
pc_dists <- tibble(
  DNA_dim1 = subset(water.pcvals, molecule == "DNA")$Dim1,
  DNA_dim2 = subset(water.pcvals, molecule == "DNA")$Dim2,
  RNA_dim1 = subset(water.pcvals, molecule == "RNA")$Dim1,
  RNA_dim2 = subset(water.pcvals, molecule == "RNA")$Dim2)
data.frame(scores(ul.pcoa)) %>%
  rownames_to_column("name") %>%
  left_join(rownames_to_column(design, "name")) %>%
  arrange(desc(distance)) %>% filter(type == "water") %>%
  mutate(molecule = ifelse(molecule == "DNA", "Total", "Active")) %>%
  ggplot(aes(x = Dim1, y = Dim2)) +
  geom_point(alpha = 0.5, aes(color = molecule, shape = molecule)) +
  geom_path(alpha = 0.75, arrow = arrow(angle = 20,
                          length = unit(0.35, "cm"),
                          type = "closed"), aes(color = molecule, linetype = molecule)) +
  scale_color_manual("Community Subset", values = my.cols) +
  geom_segment(data = pc_dists,
               aes(x = DNA_dim1, y = DNA_dim2,
                   xend = RNA_dim1, yend = RNA_dim2),
               alpha = 0.05) +
  coord_fixed() +
  labs(x = paste0("PCoA 1 (", explainvars[1],"%)"),
       y = paste0("PCoA 2 (", explainvars[2],"%)")) +
  ggsave("figures/active-tot-pcoa-trajectories.pdf", bg = "white", width = 8, height = 8) +
  ggsave("figures/active-tot-pcoa-trajectories.png", width = 8, height = 8)
```




```
# animation
# traj.animate <- data.frame(scores(ul.pcoa)) %>%
#   rownames_to_column("name") %>%
#   left_join(rownames_to_column(design, "name")) %>%
#   arrange(desc(distance)) %>% filter(type == "water", station != "UL17", station != "UL18") %>%
#   mutate(molecule = ifelse(molecule == "DNA", "Total", "Active")) %>%
#   mutate(transect = desc(distance)) %>%
#   ggplot(aes(x = Dim1, y = Dim2, frame = transect, cumulative = TRUE)) +
#   geom_point(alpha = 0.75, size = 10, aes(color = molecule)) +
#   geom_path(alpha = 1, size = 1, arrow = arrow(angle = 20,
#                           length = unit(0.35, "cm"),
#                           type = "closed"), aes(color = molecule)) +
#   scale_color_manual(values = my.cols) +
#   theme(legend.title = element_blank()) +
#   coord_fixed() +
#   labs(x = paste0("PCoA 1 (", explainvars[1],"%)"),
#        y = paste0("PCoA 2 (", explainvars[2],"%)"))
#
# gganimate(traj.animate, filename = "../figures/trajectory-animation.mp4", ani.width = 800, ani.height
```

Similarity To Terrestrial Habitat Across Gradient (Terrestrial Influence)

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

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

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

# Dummy Variables Regression Model ("Terrestrial Influence")
D2 <- (UL.sim$molecule == "RNA")*1
fit.Fig.3b <- lm(UL.sim$bray.mean ~ UL.sim$distance + D2 + UL.sim$distance*D2)
D2.R2 <- round(summary(fit.Fig.3b)$r.squared, 2)
summary(fit.Fig.3b)
```

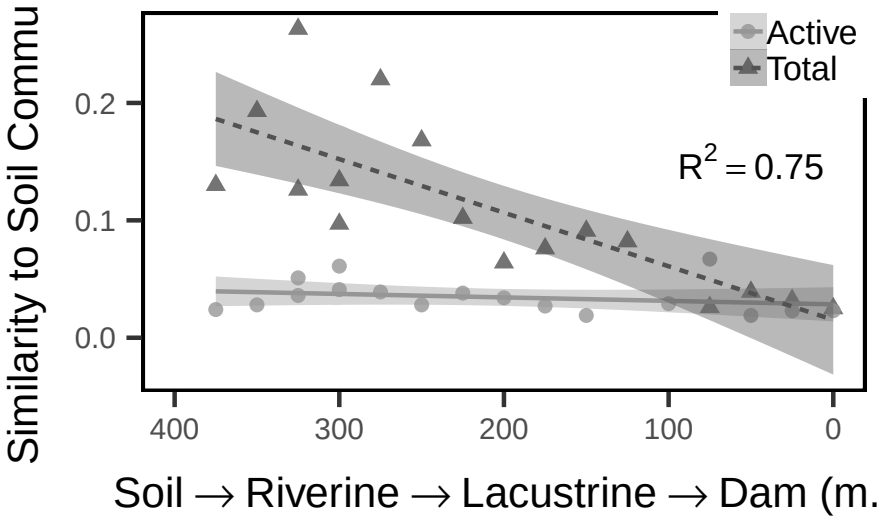
```
##
## Call:
## lm(formula = UL.sim$bray.mean ~ UL.sim$distance + D2 + UL.sim$distance *
##     D2)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.056398 -0.015112 -0.002225  0.009102  0.099423
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.524e-02  1.623e-02   0.939 0.355128
## UL.sim$distance      4.564e-04  6.828e-05   6.684  2.1e-07 ***
## D2                   1.321e-02  2.281e-02   0.579 0.566904
## UL.sim$distance:D2 -4.269e-04  9.608e-05  -4.443 0.000112 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03273 on 30 degrees of freedom
## Multiple R-squared:  0.7499, Adjusted R-squared:  0.7248
## F-statistic: 29.98 on 3 and 30 DF,  p-value: 3.68e-09
```

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

Figure 3: Similarity to Soils

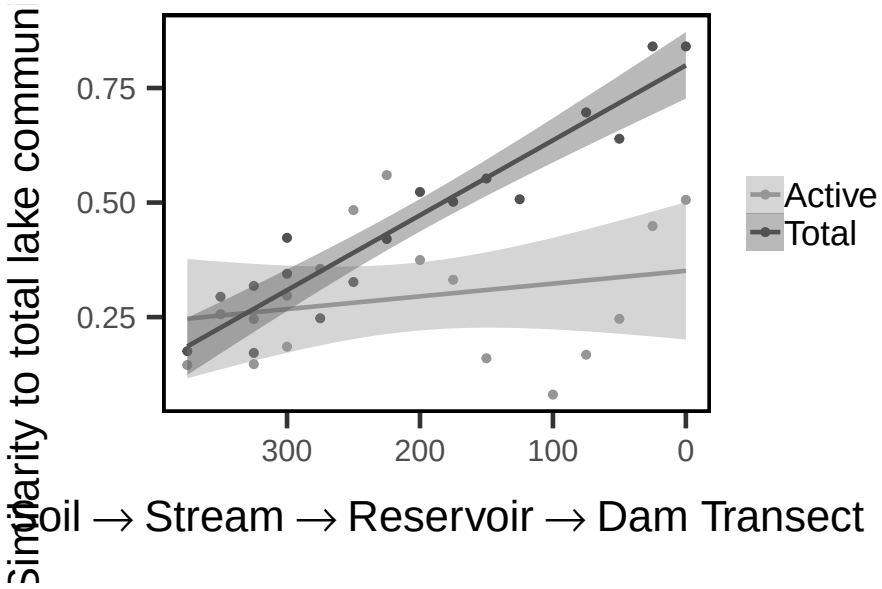
```
UL.sim %>%
  mutate(molecule = ifelse(UL.sim$molecule == "DNA", "Total", "Active")) %>%
  ggplot(aes(x = distance, y = bray.mean, color = molecule, fill = molecule, shape = molecule)) +
  geom_point(alpha = 0.8, size = 3, show.legend = T) +
  geom_smooth(method = "lm", show.legend = T, aes(linetype = molecule)) +
  labs(y = "Similarity to Soil Community",
       x = (expression("Soil" %->% "Riverine" %->% "Lacustrine" %->% "Dam (m.)"))) +
  scale_color_manual(values = my.cols) +
  scale_fill_manual(values = my.cols) +
  theme(legend.position = c(0.9, 0.9)) +
  scale_x_reverse(limits = c(400,0)) +
  annotate(geom = "text", x = 50, y = 0.15, size = 6,
           label = paste0("R^2== ",round(summary(fit.Fig.3b)$r.squared, 2)), parse = T) +
```

```
  ggsave("figures/03_similarity-to-soils.pdf", bg = "white", height = 6, width = 7) +
  ggsave("figures/03_similarity-to-soils.png", dpi = 500, height = 6, width = 7)
```

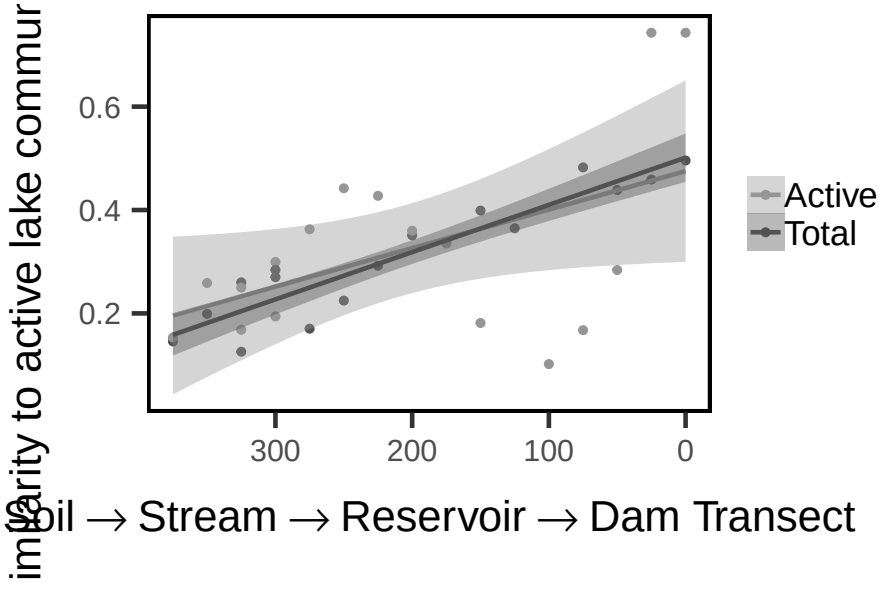


Similarity To Lake Habitat Across Gradient

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

UL.sim2 %>%
  mutate(molecule = ifelse(molecule == "DNA", "Total", "Active")) %>%
  ggplot(aes(x = distance, y = DNA, color = molecule, fill = molecule)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(y = "Similarity to total lake community (DNA)",
       x = (expression("Soil" %->% "Stream" %->% "Reservoir" %->% "Dam Transect"))) +
  scale_x_reverse() +
  scale_color_manual("Community Subset", values = my.cols) +
  scale_fill_manual("Community Subset", values = my.cols)
```




```
UL.sim2 %>%
  mutate(molecule = ifelse(molecule == "DNA", "Total", "Active")) %>%
  ggplot(aes(x = distance, y = RNA, color = molecule, fill = molecule)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(y = "Similarity to active lake community (RNA)",
       x = (expression("Soil" %->% "Stream" %->% "Reservoir" %->% "Dam Transect"))) +
  scale_x_reverse() +
  scale_color_manual("Community Subset", values = my.cols) +
  scale_fill_manual("Community Subset", values = my.cols)
```




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

```
##
```

```
## Call:
## lm(formula = UL.sim2$DNA ~ UL.sim2$distance * UL.sim2$molecule)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.242239 -0.086327  0.000567  0.063327  0.271642
##
## Coefficients:
##                                       Estimate Std. Error t value
## (Intercept)                           0.7994111  0.0558603  14.311
## UL.sim2$distance                     -0.0016356  0.0002350  -6.960
## UL.sim2$moleculeRNA                  -0.4484387  0.0784869  -5.714
## UL.sim2$distance:UL.sim2$moleculeRNA  0.0013571  0.0003307   4.104
##                                      Pr(>|t|)
## (Intercept)                          6.08e-15 ***
## UL.sim2$distance                     9.88e-08 ***
## UL.sim2$moleculeRNA                  3.11e-06 ***
## UL.sim2$distance:UL.sim2$moleculeRNA 0.000286 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1126 on 30 degrees of freedom
## Multiple R-squared:  0.6954, Adjusted R-squared:  0.6649
## F-statistic: 22.83 on 3 and 30 DF,  p-value: 6.829e-08
```

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

```
##
## Call:
## lm(formula = UL.sim2$RNA ~ UL.sim2$distance * UL.sim2$molecule)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.298879 -0.046626 -0.003357  0.046799  0.286274
##
## Coefficients:
##                                       Estimate Std. Error t value
## (Intercept)                           0.5013467  0.0610169   8.217
## UL.sim2$distance                     -0.0009140  0.0002567  -3.561
## UL.sim2$moleculeRNA                  -0.0259821  0.0857323  -0.303
## UL.sim2$distance:UL.sim2$moleculeRNA  0.0001688  0.0003612   0.467
##                                      Pr(>|t|)
## (Intercept)                          3.59e-09 ***
## UL.sim2$distance                      0.00126 **
## UL.sim2$moleculeRNA                   0.76393
## UL.sim2$distance:UL.sim2$moleculeRNA  0.64353
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.123 on 30 degrees of freedom
## Multiple R-squared:  0.4156, Adjusted R-squared:  0.3572
## F-statistic: 7.112 on 3 and 30 DF,  p-value: 0.0009525
```

```
# Calculate Confidance Intervals of Model
newdata.lake <- data.frame(cbind(UL.sim2$molecule, UL.sim2$distance))
```

```
conf95.lake <- predict(model.lake1, newdata.lake, interval="confidence")

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

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

Identifying the Soil Bacteria

```
soil.only <- OTUs[, which(colSums(OTUs[-c(1:3),]) == 0)] # what is present only in soil
lake.n.soil <- OTUs[, setdiff(colnames(OTUs),colnames(soil.only))] # what is present across all samples


# separate lake and soil samples
lake.total <- OTUs[which(design$molecule == "DNA", design$type == "water"),]
soil.total <- OTUs[which(design$molecule == "DNA", design$type == "soil"),]

# which otus are present in both lake and soil samples
lake.and.soil.total <- OTUs[which(design$molecule == "DNA", design$type == "water"),
                            which(colSums(lake.total) > 0 & colSums(soil.total) > 0)]

# isolate just the dna and rna lake communities
w.dna <- OTUs[which(design$molecule == "DNA" & design$type == "water"), ]
w.rna <- OTUs[which(design$molecule == "RNA" & design$type == "water"), ]

# pull out the lake rna counds for otus found in lake and soil
lake.and.soil.act <- w.rna[,colnames(lake.and.soil.total)]

# of these lake and soil taxa, which are never active?
nvr.act <- which(colSums(lake.and.soil.act) == 0)

# pull out their dna abundances and calculate richness
terr.lake <- w.dna[ , c(names(nvr.act))]
terr.rich <- rowSums((terr.lake > 0) * 1)
terr.REL <- rowSums(terr.lake) / rowSums(w.dna)
design.dna <- design[which(design$molecule == "DNA" & design$type == "water"), ]
terr.rich.log <- log10(terr.rich)
terr.REL.log <- log10(terr.REL)

terr.mod1 <- lm(terr.rich.log ~ design.dna$distance)
summary(terr.mod1)
```

```
##
## Call:
## lm(formula = terr.rich.log ~ design.dna$distance)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.23774 -0.13355  0.05749  0.10655  0.22647
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2.0158897  0.0771478   26.13 6.36e-14 ***
## design.dna$distance 0.0031610  0.0003245    9.74 7.06e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1555 on 15 degrees of freedom
## Multiple R-squared:  0.8635, Adjusted R-squared:  0.8544
## F-statistic: 94.87 on 1 and 15 DF,  p-value: 7.059e-08
```

```
T1.R2 <- round(summary(terr.mod1)$r.squared, 2)
T1.int <- terr.mod1$coefficients[1]
T1.slp <- terr.mod1$coefficients[2]

terr.mod2 <- lm(terr.REL.log ~ design.dna$distance)
summary(terr.mod2)
```

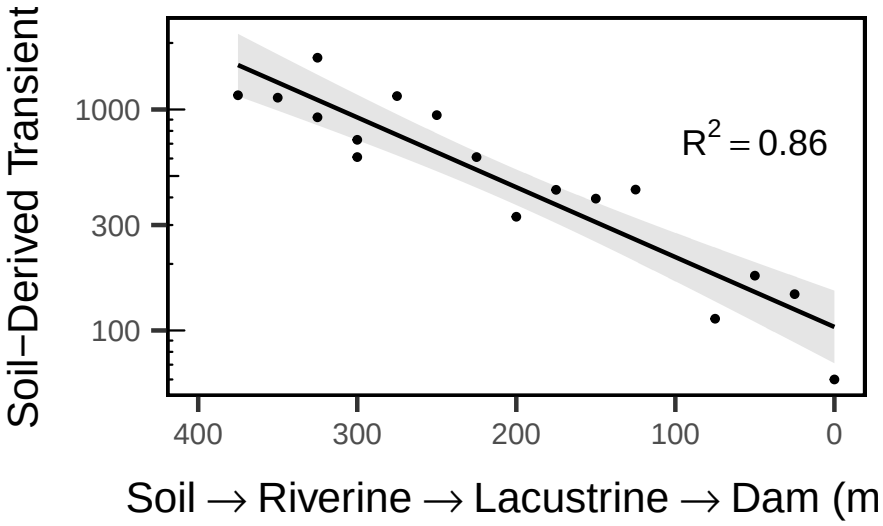
```
##
## Call:
## lm(formula = terr.REL.log ~ design.dna$distance)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44191 -0.07791  0.01510  0.08124  0.42185
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -2.1675684  0.1151258 -18.828 7.56e-12 ***
## design.dna$distance  0.0029814  0.0004843   6.156 1.84e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2321 on 15 degrees of freedom
## Multiple R-squared:  0.7164, Adjusted R-squared:  0.6975
## F-statistic:  37.9 on 1 and 15 DF,  p-value: 1.838e-05
```

```
T2.R2 <- round(summary(terr.mod2)$r.squared, 2)
T2.int <- terr.mod2$coefficients[1]
T2.slp <- terr.mod2$coefficients[2]
```

Figure 4: Transient decay

```
tibble(transient_rich = terr.rich, distance = design.dna$distance) %>%
  ggplot(aes(x = distance, y = transient_rich)) +
  geom_smooth(method = "lm", color = "black", fill = "grey") +
  geom_point(alpha = 1, color = "black") +
  scale_x_reverse(limits = c(400,0)) +
  scale_y_log10() +
```

```
  annotation_logticks(sides = "l") +
  labs(x = (expression("Soil" %->% "Riverine" %->% "Lacustrine" %->% "Dam (m.)")),
       y = "Soil-Derived Transients") +
  annotate("text", x = 50, y = 750, size = 6, label = paste0("R^2== ",T1.R2), parse = T) +
  ggsave("figures/04_transient-rich.pdf", bg = "white", height = 6, width = 7) +
  ggsave("figures/04_transient-rich.png", dpi = 500, height = 6, width = 7)
```



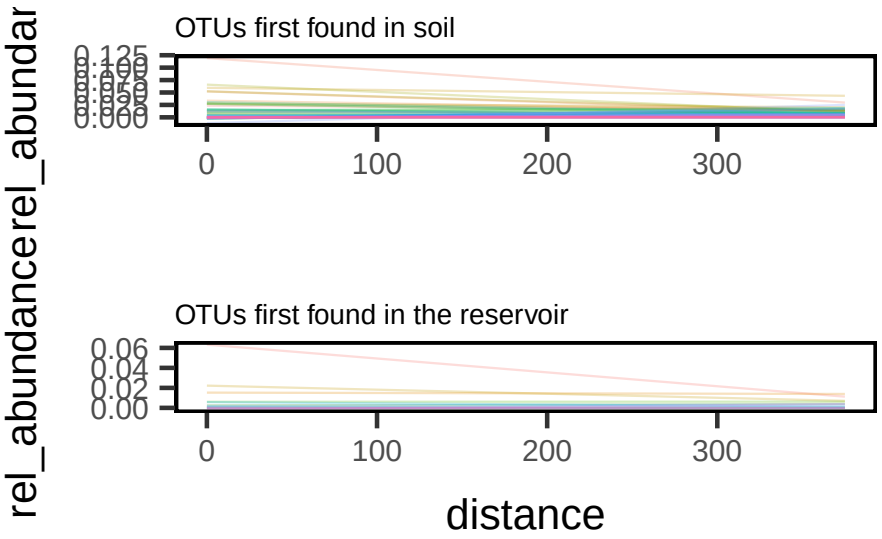
Define Core Lake Taxa

```
# identify otus in soil samples and lake samples
in.soil <- OTUs[, which(colSums(OTUs[c(1:3),]) > 0 )]
in.lake <- OTUs[, which(colSums(OTUs[-c(1:3),]) > 0)]

# isolate just the rna water samples and convert to presence-absence
in.lake.rna <- in.lake[which(design$molecule == "RNA" & design$type == "water"), ]
in.lake.rna.pa <- (in.lake.rna > 0) * 1

# define the 'core' taxa as otus present in 90% of samples
in.lake.core <- w.dna[, which((colSums(in.lake.rna.pa) / nrow(in.lake.rna.pa)) >= 0.8)]

# of the core, how many are also in the soil samples?
in.lake.core.from.soils <- in.lake.core[, intersect(colnames(in.lake.core), colnames(in.soil))]

# of the core which are not in the soil samples
in.lake.core.not.soils <- in.lake.core[, setdiff(colnames(in.lake.core), colnames(in.soil))]


in.lake.core.from.soils.REL <- in.lake.core.from.soils / rowSums(w.dna)
in.soil.plot <- as.data.frame(in.lake.core.from.soils.REL) %>%
  rownames_to_column("sample_ID") %>%
  gather(otu_id, rel_abundance, -sample_ID) %>%
  left_join(rownames_to_column(design.dna, "sample_ID")) %>%
  ggplot(aes(x = distance, y = rel_abundance, color = otu_id)) +
  labs(title = "OTUs first found in soil") +
```

```
  geom_line(alpha = 0.25, stat = "smooth", method = "lm", se = F, show.legend = F)

in.lake.core.not.soils.REL <- in.lake.core.not.soils / rowSums(w.dna)
in.lake.plot <- as.data.frame(in.lake.core.not.soils.REL) %>%
  rownames_to_column("sample_ID") %>%
  gather(OTU, rel_abundance, -sample_ID) %>%
  left_join(rownames_to_column(design.dna, "sample_ID")) %>%
  left_join(OTU.tax) %>%
  ggplot(aes(x = distance, y = rel_abundance, color = OTU)) +
  geom_line(alpha = 0.25, stat = "smooth", method = "lm", se = F, show.legend = F) +
  labs(title = "OTUs first found in the reservoir")
```

```
## Warning: Column `OTU` joining character vector and factor, coercing into
## character vector
```

```
plot_grid(in.soil.plot + theme(axis.title.x = element_blank()), in.lake.plot, align = "hv", ncol = 1) +
```




```
data.frame(mean_relabund = colMeans(in.lake.core.from.soils.REL)) %>%
  rownames_to_column(var = "OTU") %>% left_join(OTU.tax) %>%
  mutate(Taxon = paste(Phylum, Class, Order)) %>%
  arrange(desc(mean_relabund)) %>%
  ggplot() +
  geom_bar(aes(x = Taxon, y = mean_relabund), stat = "identity") +
  coord_flip()
```

```
## Warning: Column `OTU` joining character vector and factor, coercing into
## character vector
```

m

```
soil.core.mods <- apply(in.lake.core.from.soils.REL, MARGIN = 2,
    FUN = function(x) summary(lm(x ~ design.dna$distance))$coefficients[2,c(1,4)])
rownames(soil.core.mods) <- c("slope", "pval")
soil.core.decresing <- as.data.frame(t(soil.core.mods)) %>%
  rownames_to_column("OTU") %>%
  filter(pval < 0.05 & slope > 0) %>%   # rel abund decreases toward dam
  left_join(OTU.tax)
```

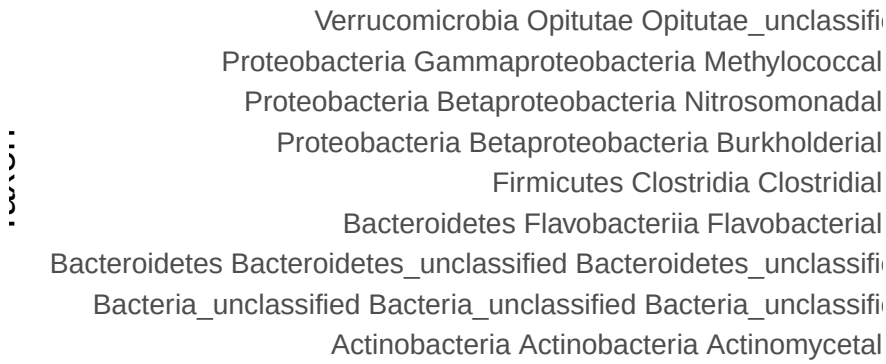
```
## Warning: Column `OTU` joining character vector and factor, coercing into
## character vector
```

```
soil.core.increasing <- as.data.frame(t(soil.core.mods)) %>%
  rownames_to_column("OTU") %>%
  filter(pval < 0.05 & slope < 0) %>%   # rel abund increases toward dam
  left_join(OTU.tax)
```

```
## Warning: Column `OTU` joining character vector and factor, coercing into
## character vector
```

```
data.frame(mean_relabund = colMeans(in.lake.core.not.soils.REL)) %>%
  rownames_to_column(var = "OTU") %>% left_join(OTU.tax) %>%
  mutate(Taxon = paste(Phylum, Class, Order)) %>%
  arrange(desc(mean_relabund)) %>%
  ggplot() +
  geom_bar(aes(x = Taxon, y = mean_relabund), stat = "identity") +
  coord_flip()
```

```
## Warning: Column `OTU` joining character vector and factor, coercing into
## character vector
```



mear

```
nonsoil.core.mods <- apply(in.lake.core.not.soils.REL, MARGIN = 2, FUN = function(x) summary(lm(x ~ des
rownames(nonsoil.core.mods) <- c("slope", "pval")
nonsoil.core.decreasing <- as.data.frame(t(nonsoil.core.mods)) %>%
  rownames_to_column("OTU") %>%
  filter(pval < 0.05 & slope > 0) %>%   # rel abund decreases toward dam
  left_join(OTU.tax)
```

```
## Warning: Column `OTU` joining character vector and factor, coercing into
## character vector
```

```
nonsoil.core.increasing <- as.data.frame(t(nonsoil.core.mods)) %>%
  rownames_to_column("OTU") %>%
  filter(pval < 0.05 & slope < 0) %>%   # rel abund increases toward dam
  left_join(OTU.tax)
```

```
## Warning: Column `OTU` joining character vector and factor, coercing into
## character vector
```

```
as.data.frame(OTUsREL[,nonsoil.core.increasing$OTU]) %>% rownames_to_column("sampleID") %>%
  left_join(rownames_to_column(design, "sampleID")) %>%
  gather(OTU, rel_abund, -station, -molecule, -type, -distance, -sampleID) %>%
  filter(molecule == "DNA") %>% left_join(OTU.tax) %>%
  mutate(taxon = paste(Class, Order, Family)) %>%
  ggplot(aes(x = distance, y = rel_abund, color = taxon)) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_x_reverse()
```

```
## Warning: Column `OTU` joining character vector and factor, coercing into
## character vector
```

```
## Warning: Removed 15 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 15 rows containing missing values (geom_point).
```

:tinobacteria Actinomycetales Actinomycetales_unclassified
ıcteroidetes_unclassified Bacteroidetes_unclassified Bacteroi

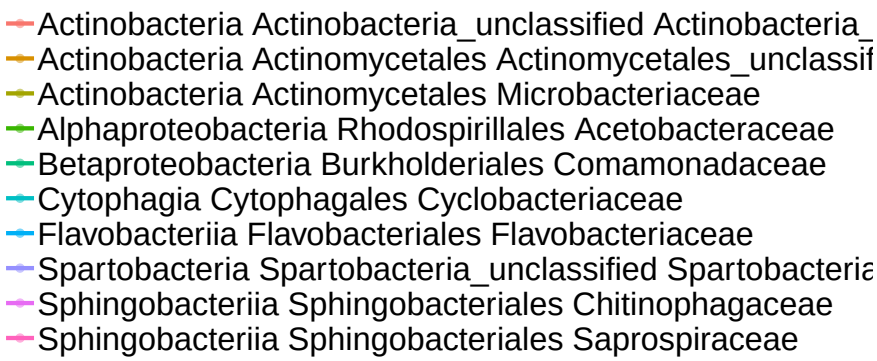
```
as.data.frame(OTUsREL[,soil.core.increasing$OTU]) %>% rownames_to_column("sampleID") %>%
  left_join(rownames_to_column(design, "sampleID")) %>%
  gather(OTU, rel_abund, -station, -molecule, -type, -distance, -sampleID) %>%
  filter(molecule == "DNA") %>% left_join(OTU.tax) %>%
```

```
  mutate(taxon = paste(Class, Order, Family)) %>%
  ggplot(aes(x = distance, y = rel_abund, color = taxon)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_reverse()
```

```
## Warning: Column `OTU` joining character vector and factor, coercing into
## character vector
```

```
## Warning: Removed 42 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 42 rows containing missing values (geom_point).
```



ce

```
# how much do the different core components contribute to total abundances
in.lake.core.soil.REL <- rowSums(in.lake.core.from.soils) / rowSums(w.dna)
in.lake.core.water.REL <- rowSums(in.lake.core.not.soils) / rowSums(w.dna)
```

Taxonomic Analysis

```
# Taxa comprising total lake 'core', those from soils, and those not from soils
core.taxa <- OTU.tax[OTU.tax$OTU %in% colnames(in.lake.core),]

core.soil.taxa <- OTU.tax[OTU.tax$OTU %in% colnames(in.lake.core.from.soils),]
core.water.taxa <- OTU.tax[OTU.tax$OTU %in% colnames(in.lake.core.not.soils),]

# Get relative abundances for each of the core taxa
core.soil.taxa.DNA.REL <- OTUsREL[which(design$molecule == "DNA" & design$type == "water"),
                                  as.numeric(rownames(core.soil.taxa))]
core.water.taxa.DNA.REL <- OTUsREL[which(design$molecule == "DNA" & design$type == "water"),
                                   as.numeric(rownames(core.water.taxa))]
core.soil.taxa.RNA.REL <- OTUsREL[which(design$molecule == "RNA" & design$type == "water"),
                                  as.numeric(rownames(core.soil.taxa))]
core.water.taxa.RNA.REL <- OTUsREL[which(design$molecule == "RNA" & design$type == "water"),
                                   as.numeric(rownames(core.water.taxa))]

core.soil.taxa.DNA.REL.max <- as.matrix(apply(core.soil.taxa.DNA.REL, 2, max))
core.soil.taxa.RNA.REL.max <- as.matrix(apply(core.soil.taxa.RNA.REL, 2, max))
core.water.taxa.DNA.REL.max <- as.matrix(apply(core.water.taxa.DNA.REL, 2, max))
```

```r
core.water.taxa.RNA.REL.max <- as.matrix(apply(core.water.taxa.RNA.REL, 2, max))

core.soil.taxa.DNA.REL.min <- as.matrix(apply(core.soil.taxa.DNA.REL, 2, min))
core.soil.taxa.RNA.REL.min <- as.matrix(apply(core.soil.taxa.RNA.REL, 2, min))
core.water.taxa.DNA.REL.min <- as.matrix(apply(core.water.taxa.DNA.REL, 2, min))
core.water.taxa.RNA.REL.min <- as.matrix(apply(core.water.taxa.RNA.REL, 2, min))

core.soil.taxa.DNA.REL.mean <- as.matrix(apply(core.soil.taxa.DNA.REL, 2, mean))
core.soil.taxa.RNA.REL.mean <- as.matrix(apply(core.soil.taxa.RNA.REL, 2, mean))
core.water.taxa.DNA.REL.mean <- as.matrix(apply(core.water.taxa.DNA.REL, 2, mean))
core.water.taxa.RNA.REL.mean <- as.matrix(apply(core.water.taxa.RNA.REL, 2, mean))

core.soil.taxa.soil.max <- as.matrix(apply(OTUsREL[which(design$type == "soil"), rownames(core.soil.taxa

core.soil.taxa.DNA.REL.bounds <- cbind(core.soil.taxa.DNA.REL.min, core.soil.taxa.DNA.REL.max,
                                       core.soil.taxa.RNA.REL.min, core.soil.taxa.RNA.REL.max,
                                       core.soil.taxa.DNA.REL.mean, core.soil.taxa.RNA.REL.mean,
                                       core.soil.taxa.soil.max)

colnames(core.soil.taxa.DNA.REL.bounds) <- c("DNA.min", "DNA.max", "RNA.min", "RNA.max", "DNA.mean", "R




core.water.taxa.DNA.REL.bounds <- cbind(core.water.taxa.DNA.REL.min, core.water.taxa.DNA.REL.max,
                                       core.water.taxa.RNA.REL.min, core.water.taxa.RNA.REL.max,
                                       core.water.taxa.DNA.REL.mean, core.water.taxa.RNA.REL.mean)
colnames(core.water.taxa.DNA.REL.bounds) <- c("DNA.min", "DNA.max", "RNA.min", "RNA.max", "DNA.mean", "

# core.soil and core.water are summaries of lake core
core.soil <- as.data.frame(cbind(core.soil.taxa$Family, core.soil.taxa$Genus,
                                 signif(core.soil.taxa.DNA.REL.bounds[,c(1:4, 7)], digits = 3)))
colnames(core.soil) <- c("Family", "Genus", "DNA.min", "DNA.max", "RNA.min", "RNA.max", "Soil.max")
core.water <- as.data.frame(cbind(core.water.taxa$Family, core.water.taxa$Genus,
                                  signif(core.water.taxa.DNA.REL.bounds[,1:4], digits = 3)))
colnames(core.water) <- c("Family", "Genus", "DNA.min", "DNA.max", "RNA.min", "RNA.max")

# Core Soil LaTeX Table
addtorow <- list()
addtorow$pos <- list(0, 0)
addtorow$command <- c("& \\multicolumn{1}{c}{Class} & \\multicolumn{1}{c}{Order} &
                      \\multicolumn{2}{c}{DNA} & \\multicolumn{2}{c}{RNA} \\\\\n",
                      "& &  & min & max & min & max \\\\\n")
core.soil.tab <- xtable(core.soil)
align(core.soil.tab) <- "crrrrrrr"
print(core.soil.tab, add.to.row = addtorow, include.colnames = FALSE,
      type= "latex", file="tables/table1.tex")
print(core.soil.tab, add.to.row = addtorow, include.colnames = FALSE, comment = FALSE)
```

```r
core.water.tab <- xtable(core.water)
align(core.water.tab) <- "crrrrrr"
print(core.water.tab, add.to.row = addtorow, include.colnames = FALSE,
      type= "latex", file="tables/table2.tex")
print(core.water.tab, add.to.row = addtorow, include.colnames = FALSE, comment = FALSE)
```

Comparisons of relabunds

```
high.activity.soil.core <- as.data.frame(core.soil.taxa.DNA.REL.bounds) %>%
  rownames_to_column("OTU") %>%
  filter(RNA.max > 0) %>% arrange(desc(RNA.max)) %>%
  left_join(OTU.tax)
high.activity.water.core <- as.data.frame(core.water.taxa.DNA.REL.bounds) %>%
  rownames_to_column("OTU") %>%
  filter(RNA.max > 0) %>% arrange(desc(RNA.max)) %>%
  left_join(OTU.tax)

mean.soil.abunds.soil.core <- OTUsREL[which(design$type == "soil"), high.activity.soil.core$OTU] %>%
  colMeans %>% data.frame(mean_soil_relabund = .) %>%
  rownames_to_column("OTU") %>% arrange(desc(mean_soil_relabund))
max.soil.abunds.soil.core <- OTUsREL[which(design$type == "soil"), high.activity.soil.core$OTU] %>%
  apply(X = ., MARGIN = 2, max) %>% data.frame(max_soil_relabund = .) %>%
  rownames_to_column("OTU") %>% arrange(desc(max_soil_relabund))

mean.soil.abunds.water.core <- OTUsREL[which(design$type == "soil"), high.activity.water.core$OTU] %>%
  colMeans %>% data.frame(mean_soil_relabund = .) %>%
  rownames_to_column("OTU") %>% arrange(desc(mean_soil_relabund))
max.soil.abunds.water.core <- OTUsREL[which(design$type == "soil"), high.activity.water.core$OTU] %>%
  apply(X = ., MARGIN = 2, max) %>% data.frame(max_soil_relabund = .) %>%
  rownames_to_column("OTU") %>% arrange(desc(max_soil_relabund))


soil.vs.lake.abunds <- high.activity.soil.core %>%
  left_join(mean.soil.abunds.soil.core) %>% left_join(max.soil.abunds.soil.core) %>%
  mutate(soil_is_source = ifelse(max_soil_relabund > 1e-3 & RNA.max > 1e-3, T, F)) %>%
  mutate(Taxon = ifelse(Genus == "unclassified", paste(Family, "sp."), Genus))

combined.relabunds <- max.soil.abunds.soil.core %>%
  left_join(rownames_to_column(as.data.frame(t(in.lake.core.from.soils.REL)), "OTU"))
rownames(combined.relabunds) <- combined.relabunds$OTU
combined.relabunds <- combined.relabunds[,-1]

otus.fold.change <- combined.relabunds / combined.relabunds$max_soil_relabund # Calculate fold changes

fold_change_summary <- otus.fold.change %>% rownames_to_column("OTU") %>%
  select(-max_soil_relabund) %>%
  gather("sample", "fold_change", -OTU) %>%
  left_join(select(rownames_to_column(design.dna, "sample"), -station, -molecule, -type)) %>%
  group_by(OTU) %>%
  summarize(max_change = max(fold_change), min_change = min(fold_change))

otus.fold.change %>% rownames_to_column("OTU") %>%
  select(-max_soil_relabund) %>%
  gather("sample", "fold_change", -OTU) %>%
  left_join(select(rownames_to_column(design.dna, "sample"), -station, -molecule, -type)) %>%
  ggplot(aes(x = distance, y = fold_change, color = OTU)) +
```

```
  geom_jitter(alpha = 0.05) +
  geom_smooth(alpha = 0.5, method = "lm", se = F) +
  scale_y_log10()

otus.fold.change %>% rownames_to_column("OTU") %>%
  select(-max_soil_relabund) %>%
  gather("sample", "fold_change", -OTU) %>%
  left_join(select(rownames_to_column(design.dna, "sample"), -station, -molecule, -type))

foldchanges <- t(otus.fold.change)[-1,]
foldchangelms <- apply(foldchanges, MARGIN = 2,
    FUN = function(x) summary(lm(x ~ design.dna$distance))$coefficients[c(1,2,8)])
rownames(foldchangelms) <- c("intercept", "slope", "pval")

soil.core.decresing <- as.data.frame(t(foldchangelms)) %>%
  rownames_to_column("OTU") %>%
  filter( slope > 0) %>%   # rel abund decreases toward dam
  left_join(OTU.tax) %>% select(-intercept, -slope, -pval, everything()) %>%
  arrange(desc(slope))
soil.core.increasing <- as.data.frame(t(foldchangelms)) %>%
  rownames_to_column("OTU") %>%
  filter( slope < 0) %>%   # rel abund increases toward dam
  left_join(OTU.tax) %>% select(-intercept, -slope, -pval, everything()) %>%
  arrange((slope))

soil.decrease.tab <- soil.core.decresing %>% select(-OTU, -Domain) %>%  flextable()
soil.increase.tab <- soil.core.increasing %>% select(-OTU, -Domain) %>% flextable()

read_docx() %>%
  body_end_section_continuous() %>%
  body_add_par("Increasing away from stream inlet", style = "heading 2") %>%
  body_add_flextable(soil.increase.tab) %>%
  body_add_par("Decreasing away from stream inlet", style = "heading 2") %>%
  body_add_flextable(soil.decrease.tab) %>%
  body_end_section_landscape() %>%
  print(target = "tables/soil-core-change-tables.docx")
```

Word Table

```
soil.tab <- core.soil %>% arrange(desc(RNA.max)) %>% flextable() %>% autofit()
water.tab <- core.water %>% arrange(desc(RNA.max)) %>% flextable() %>% autofit()

read_docx() %>%
  body_add_par("Table S1", style = "heading 1") %>%
  body_end_section_continuous() %>%
  body_add_par("Core Reservoir Microbiome (present in soils)", style = "heading 2") %>%
  body_add_flextable(soil.tab) %>%
  body_add_par("Core Reservoir Microbiome (absent from soils)", style = "heading 2") %>%
  body_add_flextable(water.tab) %>%
  body_end_section_landscape() %>%
  print(target = "tables/core_tables.docx")
```

```
## [1] "/Users/nawis/GitHub/ReservoirGradient/tables/core_tables.docx"
```

Figure 5: Soil vs. Lake Comparisons

```
soil.vs.lake.abunds %>%
  ggplot(aes(x = max_soil_relabund, y = RNA.max)) +
  geom_vline(xintercept = 1e-3, alpha = 0.1) +
  geom_hline(yintercept = 1e-3, alpha = 0.1) +
  geom_jitter(size = 3, aes(alpha = soil_is_source,color = soil_is_source), show.legend = F) +
  scale_x_log10() +
  scale_y_log10() +
  scale_alpha_discrete(range = c(.3,.8)) +
  annotation_logticks(long = unit(.1, "in")) +
  scale_color_manual(values = my.cols) +
  labs(x = "Max Soil Relative Abundance", y = "Max Lake RNA \nRelative Abundance") +
  geom_text_repel(size = 5, data = subset(soil.vs.lake.abunds, soil_is_source == T),
                  aes(label = Taxon), force = 1.5, alpha = 0.9, segment.alpha = 0.8, box.padding = .75
  ggsave("figures/05_soil-contribs-to-active-water-otus.pdf", bg = "white", height = 6, width = 7) +
  ggsave("figures/05_soil-contribs-to-active-water-otus.png", dpi = 500, height = 6, width = 7)
```

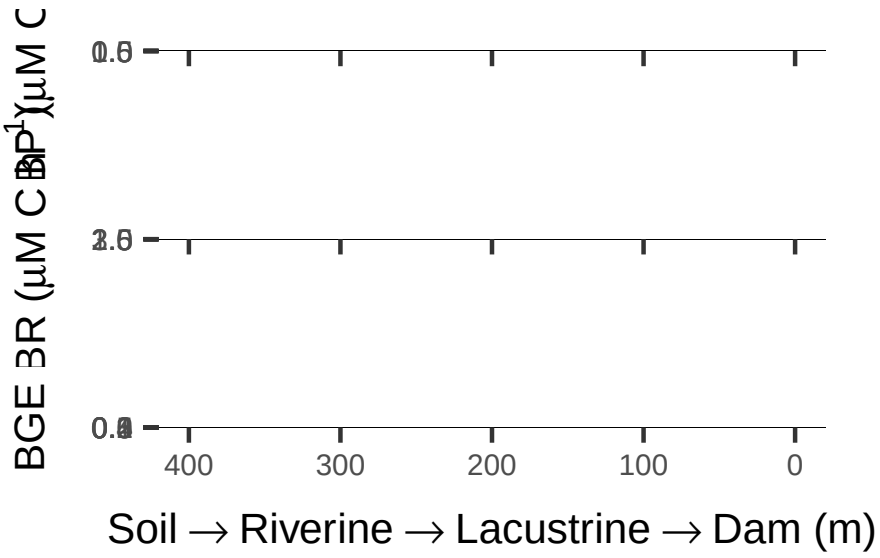
Ecosystem Functioning

Fig 1: Microbial metabolism along reservoir gradient

Read in data

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


# Quadratic regression for BP
dist <- metab$dist
dist2 <- metab$dist^2
BP.fit <- lm(metab$BP ~ dist + dist2)
BP.R2 <- round(summary(BP.fit)$r.squared, 2)

# Simple linear regression for BR
BR.fit <- lm(metab$BR ~ metab$dist)
BR.R2 <- round(summary(BR.fit)$r.squared, 2)
BR.int <- BR.fit$coefficients[1]
BR.slp <- BR.fit$coefficients[2]

# Simple linear regression for BGE
BGE.fit <- lm(metab$BGE ~ metab$dist)
BGE.R2 <- round(summary(BGE.fit)$r.squared, 2)
BGE.int <- BGE.fit$coefficients[1]
BGE.slp <- BGE.fit$coefficients[2]

BP.R2
```

```
BR.R2
BGE.R2

BP.plot <- ggplot(metab, aes(x = dist, y = BP)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x + I(x^2), color = "black") +
  annotate(geom = "text", x = 50, y = 1.5, size = 5, label = paste0("R^2== ",BP.R2), parse = T) +
  labs(y = expression(paste('BP (', mu ,'M C h'^-1* ')')),
       x = (expression("Soil" %->% "Riverine" %->% "Lacustrine" %->% "Dam (m)"))) +
  scale_x_reverse(limits = c(400,0))
BR.plot <- ggplot(metab, aes(x = dist, y = BR)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, color = "black") +
  annotate("text", x = 50, y = 1.5, size = 5, label = paste0("R^2== ",BR.R2), parse = T ) +
  labs(y = expression(paste('BR (', mu ,'M C h'^-1* ')')),
       x = (expression("Soil" %->% "Riverine" %->% "Lacustrine" %->% "Dam (m)"))) +
  scale_x_reverse(limits = c(400,0))
BGE.plot <- ggplot(metab, aes(x = dist, y = BGE)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x + I(x^2), color = "black") +
  annotate("text", x = 50, y = .5, size = 5, label = paste0("R^2== ",BGE.R2), parse = T ) +
  labs(y = "BGE",
       x = (expression("Soil" %->% "Riverine" %->% "Lacustrine" %->% "Dam (m)"))) +
  scale_x_reverse(limits = c(400,0))
plot_grid(BP.plot + theme(axis.title.x = element_blank(), axis.text.x = element_blank(),
                          plot.margin = unit(c(1, 1, -1, 0), "cm")),
          BR.plot + theme(axis.title.x = element_blank(), axis.text.x = element_blank(),
                          plot.margin = unit(c(-1, 1, -1, 0), "cm")),
          BGE.plot + theme(plot.margin = unit(c(-1, 1, 0, 0), "cm")), align = "hv", ncol = 1) +
  ggsave("figures/06_ecosystem-functions.pdf", bg = "white", width = 6, height = 11)
```



Relation of ecosystem functions and community structure

```
# detrend the spatial signal
bp.resid <- resid(lm(BP ~ dist + I(dist)^2, data = metab))
br.resid <- resid(lm(BR ~ dist, data = metab))


metab.resids <- metab
metab.resids$BR_resid <- br.resid + mean(metab$BR)
metab.resids$BP_resid <- bp.resid + mean(metab$BP)

transient.metabolism <- data.frame(transients = terr.REL, dist = design.dna$distance) %>%
  left_join(metab.resids)


bp.mod.quad <- lm(BP_resid ~ transients + I(transients^2), data = transient.metabolism)
bp.mod.lin <- lm(BP_resid ~ transients, data = transient.metabolism)
bp.mod.int <- lm(BP_resid ~ 1, data = transient.metabolism)
anova(bp.mod.int, bp.mod.lin, bp.mod.quad)
AIC(bp.mod.quad, bp.mod.lin, bp.mod.int)

br.mod.quad <- lm(BR_resid ~ transients + I(transients^2), data = transient.metabolism)
br.mod.lin <- lm(BR_resid ~ transients, data = transient.metabolism)
br.mod.int <- lm(BR_resid ~ 1, data = transient.metabolism)
anova(br.mod.int, br.mod.lin, br.mod.quad)
AIC(br.mod.int, br.mod.lin, br.mod.quad)

bge.mod.quad <- lm(BGE ~ transients + I(transients^2), data = transient.metabolism)
bge.mod.lin <- lm(BGE ~ transients, data = transient.metabolism)
bge.mod.int <- lm(BGE ~ 1, data = transient.metabolism)
anova(bge.mod.int, bge.mod.lin, bge.mod.quad)
AIC(bge.mod.int, bge.mod.lin, bge.mod.quad)

round(summary(br.mod.quad)$r.squared, 2)
round(summary(bp.mod.quad)$r.squared, 2)


total_core <- rowSums(OTUsREL[design$molecule == "DNA" & design$type == "water",
                              subset(rbind.data.frame(high.activity.water.core,
                                                      high.activity.soil.core), RNA.max > .01)$OTU])

summary(lm(BP ~ transients * dist, transient.metabolism))
summary(lm(BR ~ transients * dist, transient.metabolism))


data.frame(
  soil_core = rowSums(OTUsREL[design$molecule == "DNA" & design$type == "water",
           subset(soil.vs.lake.abunds, RNA.max > .01)$OTU]),
  dist = design.dna$distance) %>%
  left_join(metab.resids) %>% select(-BGE, -BP, -BR) %>% gather(metab, value, -soil_core, -dist) %>%
  ggplot(aes(x = soil_core, y = value, color = metab, fill = metab)) +
  geom_point(size = 2) +
  geom_smooth(alpha = .25, method = 'lm', formula = y ~ x + I(x^2)) +
```

```
  labs(x = "Relative Abundance of Soil-derived Core",
       y = expression(paste('Metabolism (', mu ,'M C h'^-1* ')'))) +
  scale_color_viridis("Ecosystem Function", discrete = T, begin = .1, end = .6, option = "D") +
  scale_fill_viridis("Ecosystem Function", discrete = T, begin = .1, end = .6, option = "D") +
  ggsave("figures/06_soilcore-function.pdf", bg = "white", width = 7, height = 6)

data.frame(
  water_core = rowSums(OTUsREL[design$molecule == "DNA" & design$type == "water",
                               subset(high.activity.water.core, RNA.max > .01)$OTU]),
  dist = design.dna$distance) %>%
  left_join(metab.resids) %>% select(-BGE,-BR,-BP) %>% gather(metab, value, -water_core, -dist) %>%
  ggplot(aes(x = water_core, y = value, color = metab, fill = metab)) +
  geom_point(size = 2) +
  geom_smooth(alpha = .25, method = 'lm', formula = y ~ x + I(x^2)) +
  labs(x = "Relative Abundance of non-soil-derived Core",
       y = expression(paste('Metabolism (', mu ,'M C h'^-1* ')'))) +
  scale_color_viridis("Ecosystem Function", discrete = T, begin = .1, end = .6, option = "D") +
  scale_fill_viridis("Ecosystem Function", discrete = T, begin = .1, end = .6, option = "D") +
  ggsave("figures/06_nonsoilcore-function.pdf", bg = "white", width = 7, height = 6)



data.frame(transients = resid(lm(terr.REL ~ design.dna$distance)) + mean(terr.REL), dist = design.dna$di
  left_join(metab.resids) %>% select(-BGE, -BP, -BR) %>% gather(metab, value, -transients, -dist) %>%
  ggplot(aes(x = transients, y = value, color = metab, fill = metab)) +
  geom_point(size = 2, show.legend = F) +
  geom_smooth(alpha = .25, method = 'lm', formula = y ~ x, show.legend = F) +
  annotation_logticks(sides = "b") +
  labs(x = "Relative Abundance of Transient Taxa",
       y = expression(paste('Metabolism (', mu ,'M C h'^-1* ')'))) +
  scale_color_viridis("Ecosystem Function", discrete = T, begin = .1, end = .6, option = "D") +
  scale_fill_viridis("Ecosystem Function", discrete = T, begin = .1, end = .6, option = "D") +
  scale_y_continuous(limits = c(0,3)) +
  theme(plot.margin = unit(c(1,1,0,0), "cm")) +
  ggsave("figures/06_transients-function.pdf", bg = "white", width = 7, height = 6)


core.metab <- data.frame(
  total_core = rowSums(OTUsREL[design$molecule == "DNA" & design$type == "water",
                               subset(rbind.data.frame(high.activity.water.core,
                                                       high.activity.soil.core), RNA.max > .01)$OTU]),
  dist = design.dna$distance) %>%
  left_join(metab.resids)

summary(lm(BP ~ total_core * dist, core.metab))
summary(lm(BR ~ total_core + dist, core.metab))


core.metab <- data.frame(
  total_core = rowSums(OTUsREL[design$molecule == "DNA" & design$type == "water",
                               subset(rbind.data.frame(high.activity.water.core,
                                                       high.activity.soil.core), RNA.max > .01)$OTU]),
  dist = design.dna$distance) %>%
```

```
  left_join(metab.resids)
core.metab$total_core_resid <- resid(lm(total_core ~ dist + I(dist^2), core.metab)) + mean(core.metab$t
summary(lm(BP_resid ~ total_core, core.metab))
summary(lm(BR_resid ~ total_core + I(total_core^2), core.metab))


core.metab %>% select(-BGE, -BP, -BR, -total_core) %>% gather(metab, value, -total_core_resid, -dist) %
  ggplot(aes(x = total_core_resid, y = value, color = metab, fill = metab)) +
  geom_point(size = 2, show.legend = F) +
  geom_smooth(alpha = .25, method = 'lm', formula = y ~ x, show.legend = F) +
  labs(x = "Relative Abundance of Core Taxa",
       y = expression(paste('Metabolism (', mu ,'M C h'^-1* ')'))) +
  scale_color_viridis("Ecosystem Function", discrete = T, begin = .1, end = .6, option = "D") +
  scale_fill_viridis("Ecosystem Function", discrete = T, begin = .1, end = .6, option = "D") +
  scale_y_continuous(limits = c(0,3)) +
  theme(plot.margin = unit(c(1,1,0,0), "cm")) +
  ggsave("figures/06_core-function.pdf", bg = "white", width = 7, height = 6)
```

Analyze environmental controls

```
# library(AEM)
# # pull out the in-lake samples
# in.lake.dna.samples <- which(design$type == "water" & design$molecule == "DNA" & design$distance < 30
# env <- env.dat[which(env.dat$sample.ID %in% design$station[in.lake.dna.samples]),c("temp", "pH", "DO"
# env <- scale(env)
# geo_dist <- as.vector(dist(design[in.lake.dna.samples,]$distance))
# env_dist <- as.vector(dist(env))
# com_dist <- as.vector(vegdist(decostand(lake.tot, "total")))
# data.frame(env_dist = resid(lm(env_dist ~ geo_dist)),
#            com_dist = com_dist,
#            geo_dist = resid(lm(geo_dist ~ env_dist))) %>%
#   ggplot(aes(x = env_dist, y = com_dist)) +
#   geom_point(alpha = 0.5) +
#   geom_smooth(method = 'lm')
# data.frame(env_dist = resid(lm(env_dist ~ geo_dist)),
#            com_dist = com_dist,
#            geo_dist = resid(lm(geo_dist ~ env_dist))) %>%
#   ggplot(aes(x = geo_dist, y = com_dist)) +
#   geom_point(alpha = 0.5) +
#   geom_smooth(method = 'lm')
# data.frame(env_dist = resid(lm(env_dist ~ geo_dist)),
#            com_dist = com_dist,
#            geo_dist = resid(lm(geo_dist ~ env_dist))) %>%
#   ggplot(aes(x = env_dist, y = geo_dist)) +
#   geom_point(alpha = 0.5) +
#   geom_smooth(method = 'lm')
#
# # construct asymetric eigenvector maps along transect
# n <- nrow(env)
# aem.out <- aem.time(n, moran = T, plot.moran = T)
# colnames(aem.out$aem) <- paste0("AEM", seq(1, n-1))
```

```
# aems <- aem.out$aem[,which(aem.out$Moran$p.value < 0.05)]
#
#
# prcomp(env, scale. = T)
# round(cor(cbind(aems, env)),2)
# lake.tot <- OTUs[in.lake.dna.samples,]
#
# vp.tot.pos <- varpart(lake.tot, aems[,c("AEM1", "AEM2", "AEM3")], env, transfo = "hellinger")
# vp.tot.pos
# plot(vp.tot.pos)
#
# vp.tot.neg <- varpart(lake.tot, aems[,c("AEM8", "AEM9", "AEM10")], env, transfo = "hellinger")
# vp.tot.neg
# plot(vp.tot.neg)
#
# # rna
# in.lake.rna.samples <- which(design$type == "water" & design$molecule == "RNA" & design$distance < 30
# env <- env.dat[which(env.dat$sample.ID %in% design$station[in.lake.rna.samples]),c("temp", "pH", "DO"
# env <- scale(env)
#
# # construct asymetric eigenvector maps along transect
# n <- nrow(env)
# aem.out <- AEM::aem.time(n, moran = T, plot.moran = T)
# colnames(aem.out$aem) <- paste0("AEM", seq(1, n-1))
# aems <- aem.out$aem[,which(aem.out$Moran$p.value < 0.05)]
#
# prcomp(env, scale. = T)
# round(cor(cbind(aems, env)),2)
# lake.act <- OTUs[in.lake.rna.samples,]
#
# geo_dist <- as.vector(dist(design[in.lake.rna.samples,]$distance))
# env_dist <- as.vector(dist(env))
# com_dist <- as.vector(vegdist(decostand(lake.act, "total")))
# data.frame(env_dist = resid(lm(env_dist ~ geo_dist)),
#            com_dist = com_dist,
#            geo_dist = resid(lm(geo_dist ~ env_dist))) %>%
#   ggplot(aes(x = env_dist, y = com_dist)) +
#   geom_point(alpha = 0.5) +
#   geom_smooth(method = 'lm')
# data.frame(env_dist = resid(lm(env_dist ~ geo_dist)),
#            com_dist = com_dist,
#            geo_dist = resid(lm(geo_dist ~ env_dist))) %>%
#   ggplot(aes(x = geo_dist, y = com_dist)) +
#   geom_point(alpha = 0.5) +
#   geom_smooth(method = 'lm')
# data.frame(env_dist = resid(lm(env_dist ~ geo_dist)),
#            com_dist = com_dist,
#            geo_dist = resid(lm(geo_dist ~ env_dist))) %>%
#   ggplot(aes(x = env_dist, y = geo_dist)) +
#   geom_point(alpha = 0.5) +
#   geom_smooth(method = 'lm')
#
# vp.act.pos <- varpart(lake.act, aems[,c("AEM1", "AEM2", "AEM3")], env, transfo = "hellinger")
```

```
# vp.act.pos
# plot(vp.act.pos)
#
# vp.act.neg <- varpart(lake.act, aems[,c("AEM8", "AEM9", "AEM10")], env, transfo = "hellinger")
# vp.act.neg
# plot(vp.act.neg)
```

null model to control for richness

```
# require("progress")
#
# Calculate abundance-weighted Raup-Crick dissimilarities
# rcbraycurtis <- function(OTUs){
#   regional.abunds <- t(as.matrix(colSums(OTUs)))
#   regional.relabunds <- decostand(regional.abunds, method = "total")
#   occupancy.probs <- t(as.matrix(colSums(decostand(OTUs, method = "pa")) / nrow(OTUs)))
#   site.abunds <- rowSums(OTUs)
#   site.rich <- specnumber(OTUs)
#   a <- regional.relabunds * occupancy.probs
#
#   # Create a null community based on Stegen et al. 2015
#   r <- nrow(OTUs)
#   c <- ncol(OTUs)
#   spec.vec <- 1:ncol(OTUs)
#   RCbc.nulls <- array(NA, c(r, r, 999))
#
#   # stochastic community assembly nulls
#   for(i in 1:999){
#     if(i == 1) pb <- progress_bar$new(total = 999, force = T)
#     pb$update(ratio = i/999)
#
#     null.comm <- OTUs * 0
#     # for first simulation:
#     for(row.i in 1:nrow(null.comm)){
#       #print(paste("run :", i, " -> ", row.i, " : ", site.abunds[row.i], " inds"))
#
#       while(rowSums(null.comm)[row.i] < site.abunds[row.i]){
#
#
#         # choose a species based on its occupancy
#         local.specs <- sample(x = spec.vec, size = site.rich[row.i],
#                               prob = as.vector(occupancy.probs), replace = FALSE)
#
#         local.probs <- decostand(t(as.matrix(regional.abunds[,local.specs])), method = "total")
#
#         local.inds <- sample(x = local.specs, size = site.abunds[row.i],
#                              prob = as.vector(local.probs), replace = TRUE)
#
#         local.abunds <- rle(sort(local.inds))
#
#         # add an individual to the local community
```

```
#         null.comm[row.i, local.abunds$values] <- local.abunds$lengths
#       }
#     }
#     null.bc <- as.matrix(vegdist(decostand(null.comm, method = "log"), method = "bray"))
#     RCbc.nulls[,,i] <- null.bc
#   }
# }
#
# RCbc.nulls.total <- rcbraycurtis(OTUs = OTUs[which(design$molecule == "DNA"),])
# RCbc.nulls.active <- rcbraycurtis(OTUs = OTUs[which(design$molecule == "RNA"),])
#
# saveRDS(RCbc.nulls.total, file = "intermediate-data/RCbc.null.total.rda")
# saveRDS(RCbc.nulls.active, file = "intermediate-data/RCbc.null.active.rda")
#
# RCbc.nulls.total <- readRDS(file = "intermediate-data/RCbc.null.total.rda")
# RCbc.nulls.active <- readRDS(file = "intermediate-data/RCbc.null.active.rda")
#
# obs.bc.total <- as.matrix(vegdist(OTUsREL[which(design$molecule == "DNA"),], method = "bray"))
# obs.bc.active <- as.matrix(vegdist(OTUsREL[which(design$molecule == "RNA"),], method = "bray"))
#
# get.rcbrayvals <- function(obs.bc, RCbc.nulls){
#   r <- nrow(as.matrix(obs.bc))
#   site.compares <- expand.grid(site1 = 1:r, site2 = 1:r)
#   RC.bray <- matrix(NA, nrow = r, ncol = r)
#
#   for(row.i in 1:nrow(site.compares)){
#     site1 <- site.compares[row.i,1]
#     site2 <- site.compares[row.i,2]
#     pairwise.null <- RCbc.nulls[site1,site2,]
#     pairwise.bray <- obs.bc[site1,site2]
#     num.greater <- sum(pairwise.null > pairwise.bray)
#     num.ties <- sum(pairwise.null == pairwise.bray)
#     val <- (((1 * num.greater) + (0.5 * num.ties))/1000 - 0.5) * 2
#     RC.bray[site1, site2] <- val
#   }
#   return(RC.bray)
# }
#
# RC.bray.active <- get.rcbrayvals(obs.bc.active, RCbc.nulls.active)
# rownames(RC.bray.active) <- rownames(subset(design, molecule == "RNA"))
# colnames(RC.bray.active) <- rownames(subset(design, molecule == "RNA"))
# RC.bray.active.dist <- as.dist(RC.bray.active)
#
# RC.bray.total <- get.rcbrayvals(obs.bc.total, RCbc.nulls.total)
# rownames(RC.bray.total) <- rownames(subset(design, molecule == "DNA"))
# colnames(RC.bray.total) <- rownames(subset(design, molecule == "DNA"))
# RC.bray.total.dist <- as.dist(RC.bray.total)
#
# # Similarity to Soil Sample
# UL.bray      <- 1-as.matrix(vegdist(OTUsREL.log, method="bray"))
# UL.bray.lake <- UL.bray[-c(1:3), 1:3]
# bray.mean    <- round(apply(UL.bray.lake, 1, mean), 3)
# bray.se      <- round(apply(UL.bray.lake, 1, se), 3)
```

```
# UL.sim       <- cbind(design[-c(1:3), ], bray.mean, bray.se)
#
#
# # Calculate Linear Model
# model.terr <- lm(UL.sim$bray.mean ~ UL.sim$distance * UL.sim$molecule)
#
# # Calculate Confidance Intervals of Model
# newdata.terr <- data.frame(cbind(UL.sim$molecule, UL.sim$distance))
# conf95.terr <- predict(model.terr, newdata.terr, interval="confidence")
#
# # Dummy Variables Regression Model ("Terrestrial Influence")
# D2 <- (UL.sim$molecule == "RNA")*1
# fit.Fig.3b <- lm(UL.sim$bray.mean ~ UL.sim$distance + D2 + UL.sim$distance*D2)
# D2.R2 <- round(summary(fit.Fig.3b)$r.squared, 2)
# summary(fit.Fig.3b)
#
# DNA.int.3b <- fit.Fig.3b$coefficients[1]
# DNA.slp.3b <- fit.Fig.3b$coefficients[2]
# RNA.int.3b <- DNA.int.3b + fit.Fig.3b$coefficients[3]
# RNA.slp.3b <- DNA.slp.3b + fit.Fig.3b$coefficients[4]
#
#
# UL.sim %>%
#   mutate(molecule = ifelse(UL.sim$molecule == "DNA", "Total", "Active")) %>%
#   ggplot(aes(x = distance, y = bray.mean, color = molecule, fill = molecule)) +
#   geom_point() +
#   geom_smooth(method = "lm") +
#   labs(y = "Community Similarity to Soils",
#        x = (expression("Soil" %->% "Riverine" %->% "Lacustrine" %->% "Dam (m)"))) +
#   scale_color_manual("Community Subset", values = my.cols) +
#   scale_fill_manual("Community Subset", values = my.cols) +
#   scale_x_reverse(limits = c(400,0)) +
#   annotate(geom = "text", x = 50, y = 0.2, size = 5,
#            label = paste0("R^2== ",round(summary(fit.Fig.3b)$r.squared, 2)), parse = T) +
#   ggsave("../figures/03_similarity-to-soils.pdf", bg = "white", height = 6, width = 8)
#
# RC.bray.total.dist
```

Import Phototrophs

```
# # The phototrophs
# cyanos.in <- "../data/UL.cyano.final.shared"
# phytos.in <- "../data/UL.euks.final.shared"
#
# cyanos <- read.otu(shared = cyanos.in, cutoff = "0.03")
# phytos <- read.otu(shared = phytos.in, cutoff = "0.03")
#
# # Remove OTUs with less than two occurences across all sites
# cyanos <- cyanos[, which(colSums(cyanos) >= 2)]
# phytos <- phytos[, which(colSums(phytos) >= 2)]
#
# # Remove sites where we have low coverage
```

```
# cyanos <- cyanos[-which(coverage < 10000), ]
# phytos <- phytos[-which(coverage < 10000), ]
#
# # Remove Non Intersecting Sites
# ratio.sites <- intersect(intersect(rownames(cyanos), rownames(phytos)), rownames(OTUs))
# cyanos <- cyanos[ratio.sites, ]
# phytos <- phytos[ratio.sites, ]
# heteros <- OTUs[ratio.sites, ]
# design.int <- design[ratio.sites, ]
#
# # Remove RNA Sites
# DNA.samps <- which(design.int$molecule == "DNA")
# cyanos <- cyanos[DNA.samps, ]
# phytos <- phytos[DNA.samps, ]
# heteros <- OTUs[DNA.samps, ]
# design.dna <- design[DNA.samps, ]
#
# # Observed Richness
# S.cyano <- rowSums((cyanos > 0) * 1)
# S.phyto <- rowSums((phytos > 0) * 1)
# S.hetero <- rowSums((heteros > 0) * 1)
#
# N.cyano <- rowSums(cyanos)
# N.phyto <- rowSums(phytos)
# N.hetero <- rowSums(heteros) - rowSums(cyanos)
#
# HtoC <- N.hetero / N.cyano
# HtoP <- N.hetero / N.phyto
# HtoBoth <- N.hetero / (N.cyano + N.phyto)
#
# data_frame(S.cyano, S.phyto, S.hetero,
#            metric = "richness",
#            location = design.dna$distance) %>%
#   gather(group, richness, -metric, -location) %>%
#   filter(!is.na(location)) %>%
#   ggplot(aes(x = location, y = richness, color = group)) +
#   geom_jitter(alpha = 0.5) +
#   geom_smooth() +
#   scale_y_log10() +
#   scale_x_reverse()
#
# data_frame(N.cyano, N.phyto, N.hetero,
#            metric = "abundance",
#            location = design.dna$distance) %>%
#   gather(group, abundance, -metric, -location) %>%
#   filter(!is.na(location)) %>%
#   ggplot(aes(x = location, y = abundance, color = group)) +
#   geom_jitter(alpha = 0.5) +
#   geom_smooth() +
#   scale_y_log10() +
#   scale_x_reverse()
#
# data_frame(HtoC, HtoP, HtoBoth,
```

```
#            metric = "ratio",
#            location = design.dna$distance) %>%
#   gather(group, ratio, -metric, -location) %>%
#   filter(!is.na(location)) %>%
#   ggplot(aes(x = location, y = ratio, color = group)) +
#   geom_jitter(alpha = 0.5) +
#   geom_smooth() +
#   scale_y_log10() +
#   scale_x_reverse()
```

Old Figures

Figure 2: Bacterial Richness and Terrestrial Influence Across Gradient

```
# # Set Plot Symbol Parameters
# mol <- rep(NA, length(lake$molecule))
#   for (i in 1:length(mol)){
#     if (lake$molecule[i] == "DNA"){
#       mol[i] <- 22
#     } else {
#       mol[i] <- 24
#     }
#   }
# cols <- rep(NA, length(lake$molecule))
#   for (i in 1:length(cols)){
#     if (lake$molecule[i] == "DNA"){
#       cols[i] <- "gray15"
#     } else {
#       cols[i] <- "gray75"
#     }
#   }
#
# # Initial Plot
# png(filename="../figures/Figure2.png",
#     width = 1200, height =1200, res = 96*2)
#
# par(mfrow = c(1,1), mar = c(0, 5, 0, 1) + 0.5, oma = c(4, 2, 0, 0) + 0.5)
# bar.layout <- layout(rbind(1, 2), height = c(4, 4))
#
# # Richness Across Gradient Plot
# plot(lake$S.obs ~ lake$distance, col= "black", bg = cols, pch=mol, las = 1,
#      xlim = c(400, -15), ylim = c(0, 2750), cex = 1.5,
#      xlab="", ylab="", xaxt="n")
#
# #   matlines(lake$distance[lake$molecule == "DNA"], conf95.rich[lake$molecule == "DNA", ],
# #          lty = c(1, 0, 0), col=c("black", "gray50", "gray50"), lwd=c(2, 1, 1))
# #   matlines(lake$distance[lake$molecule == "RNA"], conf95.rich[lake$molecule == "RNA", ],
# #          lty = c(1, 0, 0), col=c("black", "gray50", "gray50"), lwd=c(2, 1, 1))
#
# # Add multiple regression lines
# clip(400, 0, 0, 2800)
```

```
# abline(a = DNA.int.3a, b = DNA.slp.3a, col = "black", lwd = 2.5, lty = 6)
# abline(a = RNA.int.3a, b = RNA.slp.3a, col = "black", lwd = 2.5, lty = 4)
# text(40, 1500, labels = bquote(italic(R)^2 == .(D1.R2)), cex = 1)
#
# axis(side = 1, lwd.ticks = 2, tck=-0.02, labels = F, cex.axis = 1, las = 1,
#      at = c(0, 100, 200, 300, 400))
# axis(side = 2, lwd.ticks = 2, cex.axis = 1, las = 1)
# axis(side = 3, lwd.ticks = 2, tck=-0.01, labels = F,  cex.axis = 2, las = 1,
#      at = c(0, 100, 200, 300, 400))
# axis(side = 4, lwd.ticks = 2, tck=-0.01, labels = F,  cex.axis = 2, las = 1)
# axis(side = 1, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1,
#      at = c(0, 100, 200, 300, 400))
# axis(side = 2, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)
# axis(side = 3, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1,
#      at = c(0, 100, 200, 300, 400))
# axis(side = 4, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)
#
# mtext("Richness\n(S)" , side = 2, line = 4, cex=1.5)
#
# legend("topright", legend = levels(lake$molecule), pch=c(22, 24),
#        pt.bg = c("gray15", "gray75"), bty='n', cex = 1, ncol=2)
#
# box(lwd=2)
#
# # Terrestrial Influence Plot
#
# plot(UL.sim$bray.mean ~ UL.sim$distance, col= "black", bg = cols, pch=mol,
#      las = 1, xlim = c(400, -15), ylim = c(0, 0.25), cex = 1.5,
#      xlab="", ylab="", xaxt="n")
#
# #   matlines(lake$distance[lake$molecule == "DNA"], conf95.terr[lake$molecule == "DNA", ],
# #          lty = c(1, 0, 0), col=c("black", "gray50", "gray50"), lwd=c(2, 1, 1))
# #   matlines(lake$distance[lake$molecule == "RNA"], conf95.terr[lake$molecule == "RNA", ],
# #          lty = c(1, 0, 0), col=c("black", "gray50", "gray50"), lwd=c(2, 1, 1))
#
# # Add multiple regression lines
# clip(400, 0, 0, 0.27)
# abline(a = DNA.int.3b, b = DNA.slp.3b, col = "black", lwd = 2.5, lty = 6)
# abline(a = RNA.int.3b, b = RNA.slp.3b, col = "black", lwd = 2.5, lty = 4)
# text(40, 0.125, labels = bquote(italic(R)^2 == .(D2.R2)), cex = 1)
#
# axis(side = 1, lwd.ticks = 2, cex.axis = 1, las = 1,
#      labels = c("400", "300", "200", "100", "0"), at = c(400, 300, 200, 100, 000))
# axis(side = 2, lwd.ticks = 2, cex.axis = 1, las = 1)
# axis(side = 3, lwd.ticks = 2, tck=-0.02, labels = F,  cex.axis = 2, las = 1,
#      at = c(400, 300, 200, 100, 000))
# axis(side = 4, lwd.ticks = 2, tck=-0.01, labels = F,  cex.axis = 2, las = 1)
# axis(side = 1, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1,
#      at = c(400, 300, 200, 100, 000))
# axis(side = 2, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)
# axis(side = 3, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1,
#      at = c(400, 300, 200, 100, 000))
# axis(side = 4, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)
```

```
#
# mtext("Terrestrial\nInfluence" , side = 2, line = 4, cex=1.5)
# mtext("Distance (m)" , side = 1, line = 3, cex=1.5)
#
# # legend("topright", legend = levels(UL.sim$molecule), pch=c(22, 24),
# #        pt.bg = c("gray15", "gray75"), bty='n', cex = 1.25)
#
# box(lwd=2)
#
# # # Lake Influence Plot
# # # plot(UL.sim2$DNA ~ UL.sim2$distance, col= "black", bg = cols, pch=mol, las = 1,
# # #      xlim = c(400, 0), ylim = c(0, 1), cex = 1.5,
# # #      xlab="", ylab="")
# #
# # #   matlines(lake$distance[lake$molecule == "DNA"], conf95.lake[lake$molecule == "DNA", ],
# # #           lty = c(1, 0, 0), col=c("black", "gray50", "gray50"), lwd=c(2, 1, 1))
# # #   matlines(lake$distance[lake$molecule == "RNA"], conf95.lake[lake$molecule == "RNA", ],
# # #           lty = c(1, 0, 0), col=c("black", "gray50", "gray50"), lwd=c(2, 1, 1))
# #
# # # Add multiple regression lines
# # clip(400, 0, 0, 1)
# # abline(a = DNA.int.3c, b = DNA.slp.3c, col = "black", lwd = 2.5, lty = 6)
# #
# # clip(400, 0, 0, 1)
# # abline(a = RNA.int.3c, b = RNA.slp.3c, col = "black", lwd = 2.5, lty = 4)
# #
# # axis(side = 1, lwd.ticks = 2, cex.axis = 1, las = 1)
# # axis(side = 2, lwd.ticks = 2, cex.axis = 1, las = 1)
# # axis(side = 3, lwd.ticks = 2, tck=-0.05, labels = F,  cex.axis = 2, las = 1)
# # axis(side = 4, lwd.ticks = 2, tck=-0.01, labels = F,  cex.axis = 2, las = 1)
# # axis(side = 1, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)
# # axis(side = 2, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)
# # axis(side = 3, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)
# # axis(side = 4, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)
# #
# # mtext("Distance (m)" , side = 1, line = 3, cex=1.5)
# # mtext("Lake\nInfluence", side = 2, line = 4, cex=1.5)
# #
# # legend("topleft", legend = levels(UL.sim$molecule), pch=c(22, 24),
# #        pt.bg = c("gray15", "gray75"), bty='n', cex = 1.25)
# #
# # box(lwd=2)
#
# # Close Plot Defice
# dev.off()
# graphics.off()
# par(opar)
```

Figure 3: Soil Organisms Plot

```
# # Initial Plot
# png(filename="../figures/Figure3.png",
```

```
#     width = 1200, height =1200, res = 96*2)
#
# par(mfrow = c(1,1), mar = c(0, 7, 0, 1) + 0.5, oma = c(4, 2, 0, 0) + 0.5)
# bar.layout <- layout(rbind(1, 2), height = c(4, 4))
#
# # Soil OTU Richness Across Gradient Plot
# plot(terr.rich.log ~ design.dna$distance, col= "black", pch=22, las = 1,
#      xlim = c(400, -15), ylim = c(1.5, 3.5), cex = 1.5,
#      xlab="", ylab="", xaxt="n", yaxt="n")
#
# clip(0, 375, 1.5, 3.4)
# abline(a = T1.int, b = T1.slp, col = "black", lwd = 2.5, lty = 6)
# text(40, 3, labels = bquote(italic(R)^2 == .(T1.R2)), cex = 1)
#
# axis(side = 1, lwd.ticks = 2, tck = -0.02, labels = F, cex.axis = 1, las = 1)
# axis(side = 1, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)
# axis(side = 3, lwd.ticks = 2, tck=-0.01, labels = F,  cex.axis = 2, las = 1)
# axis(side = 3, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)
# axis(side = 2, lwd.ticks = 2, at = c(2, 3), labels = c(10^2, 10^3), cex.axis = 1, las = 1)
# axis(side = 2, lwd.ticks = 2, tck = -0.02, at = log10(c(seq(10, 100, by = 10),
#      seq(100, 1000, by = 100), seq(1000, 10000, by = 1000))), labels = F, cex.axis = 1, las = 1)
# axis(side = 2, lwd.ticks = 2, at = c(2, 3), tck=0.01, labels = F,  cex.axis = 2, las = 1)
# axis(side = 2, lwd.ticks = 2, tck = 0.005, at = log10(c(seq(10, 100, by = 10),
#      seq(100, 1000, by = 100), seq(1000, 10000, by = 1000))), labels = F, cex.axis = 1, las = 1)
# axis(side = 4, lwd.ticks = 2, at = c(2, 3), tck=-0.01, labels = F,  cex.axis = 2, las = 1)
# axis(side = 4, lwd.ticks = 2, at = c(2, 3), tck=0.02, labels = F,  cex.axis = 2, las = 1)
# axis(side = 4, lwd.ticks = 2, tck = 0.01, at = log10(c(seq(10, 100, by = 10),
#      seq(100, 1000, by = 100), seq(1000, 10000, by = 1000))), labels = F, cex.axis = 1, las = 1)
#
# mtext("Transient\nRichness\n(S)" , side = 2, line = 4, cex=1.5)
# box(lwd=2)
#
# # Soil OTU Relative Abundance Across Gradient Plot
# plot(terr.REL.log ~ design.dna$distance, col= "black", pch=22, las = 1,
#      xlim = c(400, -15), ylim = c(-2.5, -.5), cex = 1.5,
#      xlab="", ylab="", xaxt="n", yaxt="n")
#
# clip(0, 375, -2.5, -0.5)
# abline(a = T2.int, b = T2.slp, col = "black", lwd = 2.5, lty = 6)
# text(40, -1, labels = bquote(italic(R)^2 == .(T2.R2)), cex = 1)
#
#   axis(side = 1, lwd.ticks = 2, labels = T, cex.axis = 1, las = 1)
#   axis(side = 1, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)
#   axis(side = 3, lwd.ticks = 2, tck=-0.02, labels = F,  cex.axis = 2, las = 1)
#   axis(side = 3, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)
#   axis(side = 2, lwd.ticks = 2, at = c(-2, -1), labels = c(0.01, 0.1), cex.axis = 1, las = 1)
#   axis(side = 2, lwd.ticks = 2, tck = -0.02, at = log10(c(seq(0.001, 0.01, by = 0.001),
#        seq(0.01, 0.1, by = 0.01), seq(0.1, 1, by = 0.1))), labels = F, cex.axis = 1, las = 1)
#   axis(side = 2, lwd.ticks = 2, at = c(-2, -1), tck=0.01, labels = F,  cex.axis = 2, las = 1)
#   axis(side = 2, lwd.ticks = 2, tck = 0.005, at = log10(c(seq(0.001, 0.01, by = 0.001),
#        seq(0.01, 0.1, by = 0.01), seq(0.1, 1, by = 0.1))), labels = F, cex.axis = 1, las = 1)
#   axis(side = 4, lwd.ticks = 2, at = c(-2, -1), tck=-0.01, labels = F,  cex.axis = 2, las = 1)
#   axis(side = 4, lwd.ticks = 2, at = c(-2, -1), tck=0.02, labels = F,  cex.axis = 2, las = 1)
```

```
#   axis(side = 4, lwd.ticks = 2, tck = 0.01, at = log10(c(seq(0.001, 0.01, by = 0.001),
#        seq(0.01, 0.1, by = 0.01), seq(0.1, 1, by = 0.1))), labels = F, cex.axis = 1, las = 1)
#
# mtext("Distance (m)" , side = 1, line = 3, cex=1.5)
# mtext("Transient\nRelative\nAbundance", side = 2, line = 4, cex=1.5)
#
# box(lwd=2)
#
#
# # Close Plot Defice
# dev.off()
# graphics.off()
```

Core

Model Fit

```
#
# terrestrial <- data.frame(in.lake.core.soil.REL, design.dna$distance)
# terrestrial$distance[1] <- 10^-8
# colnames(terrestrial) <- c("t.rel.abund", "distance")
#
# mm <- function(x,V,K){
#   (V * x)/(K + x)
# }
#
# fit.t.1 <- lm(terrestrial$t.rel.abund ~ terrestrial$distance)
# fit.t.2 <- lm(terrestrial$t.rel.abund ~ poly(terrestrial$distance,2,raw=TRUE))
# fit.t.3 <- lm(terrestrial$t.rel.abund ~ poly(terrestrial$distance,3,raw=TRUE))
# fit.t.mm <- nls(t.rel.abund ~ mm((400-distance[1:15]), max, halfsat), data = terrestrial[1:15,], star
# summary(fit.t.mm)$coefficients
#
# lake <- data.frame(in.lake.core.water.REL, design.dna$distance)
# lake$distance[1] <- 10^-8
# colnames(lake) <- c("l.rel.abund", "distance")
#
# fit.l.1 <- lm(lake$l.rel.abund ~ lake$distance)
# fit.l.2 <- lm(lake$l.rel.abund ~ poly(lake$distance,2,raw=TRUE))
# fit.l.3 <- lm(lake$l.rel.abund ~ poly(lake$distance,3,raw=TRUE))
# fit.l.mm <- nls(l.rel.abund ~ mm((400-distance), max, halfsat), data = lake[1:15,], start = list(max
# #fit.l.pow = nls(l.rel.abund ~ I(distance^power), data=lake, start=list(power = -0.06590247), trace =
```

Figure 4: Plot Core Community

```
# png(filename="../figures/Figure4.png",
#     width = 1000, height = 1000, res = 96*2)
# par(mar = c(5,7,4,1))
# plot(in.lake.core.soil.REL ~ design.dna$distance,
#      ylim = c(0, 1), xlim = c(400, -15), pch = 22, bg = "black",
#      ylab = "", xlab = "", xaxt = "n", yaxt = "n", cex = 2, cex.lab = 2)
```

```
# points(in.lake.core.water.REL ~ design.dna$distance, cex = 2, pch = 22)
# mtext("Distance (m)", side = 1, line = 3, cex = 1.5)
# mtext("Relative abundance of \ncore microbiome", side = 2, line = 3, cex = 1.5)
# axis(side = 1, lwd.ticks = 2, labels = T, cex.axis = 1, las = 1)
# axis(side = 1, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)
# axis(side = 3, lwd.ticks = 2, tck=-0.01, labels = F,  cex.axis = 2, las = 1)
# axis(side = 3, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1)
# axis(side = 2, lwd.ticks = 2, at = c(0, .25, .5, .75, 1), cex.axis = 1, las = 1)
# axis(side = 2, lwd.ticks = 2, at = c(0, .25, .5, .75, 1), tck=0.01, labels = F,  cex.axis = 2, las =
# axis(side = 4, lwd.ticks = 2, at = c(0, .25, .5, .75, 1), tck=-0.01, labels = F,  cex.axis = 2, las =
# axis(side = 4, lwd.ticks = 2, at = c(0, .25, .5, .75, 1), tck=0.01, labels = F,  cex.axis = 2, las =
#
# legend("topright", c("Terrestrial", "Lake"), col = c("black", "black"),
#          pt.bg = c("black", "white"), pch = 22, cex = 1.25, bty = "n")
#
# box(lwd=2)
#
# x <- lake$distance[1:15]
#
# #lines(terrestrial$distance, predict(fit.t.2, data.frame(x=x)))
#
# #lines(lake$distance, predict(fit.l.2, data.frame(x=x)))
#
# #power <- round(summary(fit.t.pow)$coefficients[1], 3)
# #power.se <- round(summary(m)$coefficients[2], 3)
#
# lines(x, predict(fit.t.mm, list(x = x)), lwd = 2.5, lty = 6)
# lines(x, predict(fit.l.mm, list(x = x)), lwd = 2.5, lty = 6)
#
# dev.off()
# graphics.off()
```

Figure S2: chemical and physical variables along reservoir gradient

```
#
# # Start Plotting File
# png(filename="../figures/FigureS2.png",
#     width = 1500, height = 1200, res = 96*2)
#
# par(mfrow = c(2,2))
# par(mar = c(5, 6, 1, 2) + 0.5)
#
# # Total Phosphorus
# TP <- plot(rev(env.dat$dist.dam), env.dat$TP,
#      ylab = "", xlab = "", cex.lab = 2, las = 1,
#      ylim = c(0,140), xlim = c(-15, 400),
#      pch = 22, cex = 2, bg = "white", col = "black", lwd = 2,
#      yaxt = "n", xaxt = "n")
# box(lwd = 2)
#
# axis(side = 2, lwd.ticks = 2, cex.axis = 1.5, las = 1,
#     labels = c("0", "40", "80", "120"), at = c(0, 40, 80, 120))
```

```
#
# axis(side = 1, lwd.ticks = 2,  labels = F, cex.axis = 2, las = 1, mgp = c(3, 1.5, 0),
#    #labels = c("0", "100", "200", "300", "400"),
#    at = c(0, 100, 200, 300, 400))
#
#
# mtext(expression(paste('Total Phosphorus (',mu,'g P L'^-1*')')), side = 2, line = 4, cex = 1)
#
# par(mar = c(5, 5, 1, 3) + 0.5)
#
# # Chlorophyll
# chla <- plot(rev(env.dat$dist.dam), env.dat$chla,
#      ylab = "", xlab = "", cex.lab = 2, las = 1,
#      ylim = c(0,30), xlim = c(-15, 400),
#      pch = 22, cex = 2, bg = "white", col = "black", lwd = 2,
#      yaxt = "n", xaxt = "n")
# box(lwd = 2)
#
# axis(side = 2, lwd.ticks = 2, cex.axis = 1.5, las = 1,
#     labels = c("0", "10", "20", "30"), at = c(0, 10, 20, 30))
#
# axis(side = 1, lwd.ticks = 2, labels = F, cex.axis = 2, las = 1, mgp = c(3, 1.5, 0),
#    #labels = c("0", "100", "200", "300", "400"),
#    at = c(0, 100, 200, 300, 400))
#
# mtext(expression(paste('Chlorophyll a (',mu,'g L'^-1*')')), side = 2, line = 4, cex = 1)
#
# par(mar = c(5, 6, 0, 2) + 0.5)
# #Dissolved Oxygen
# plot(rev(env.dat$dist.dam), env.dat$DO,
#      ylab = "", xlab = "", cex.lab = 2, las = 1,
#      ylim = c(5,10), xlim = c(-15, 400),
#      pch = 22, cex = 2, bg = "white", col = "black", lwd = 2,
#      yaxt = "n", xaxt = "n")
# box(lwd = 2)
#
# axis(side = 2, lwd.ticks = 2, cex.axis = 1.5, las = 1,
#     labels = c("5", "7.5", "10"), at = c(5,7.5, 10))
#
# axis(side = 1, lwd.ticks = 2, cex.axis = 1.5, las = 1, mgp = c(3, 1.5, 0),
#    labels = c("0", "100", "200", "300", "400"),
#    at = rev(c(0, 100, 200, 300, 400)))
#
# mtext(expression(paste('Dissolved Oxygen (mg L'^-1*')')), side = 2, line = 4, cex = 1)
#
# text(x = 35, y = 5.1, "STREAM", font = 2)
# text(x = 375, y = 5.1, "DAM", font = 2)
#
#
# #pH
# par(mar = c(5, 5, 0, 3) + 0.5)
# plot(rev(env.dat$dist.dam), env.dat$pH,
#      ylab = "", xlab = "", cex.lab = 2, las = 1,
```

```
#      ylim = c(8,9), xlim = c(-15, 400),
#      pch = 22, cex = 2, bg = "white", col = "black", lwd = 2,
#      yaxt = "n", xaxt = "n")
# box(lwd = 2)
#
# axis(side = 2, lwd.ticks = 2, cex.axis = 1.5, las = 1,
#     labels = c("8", "8.5","9"), at = c(8, 8.5, 9))
#
# axis(side = 1, lwd.ticks = 2, cex.axis = 1.5, las = 1, mgp = c(3, 1.5, 0),
#    labels = c("0", "100", "200", "300", "400"),
#    at = rev(c(0, 100, 200, 300, 400)))
#
# mtext("pH", side = 2, line = 4, cex = 1)
#
# text(x = 35, y = 8.02, "STREAM", font = 2)
# text(x = 375, y = 8.02, "DAM", font = 2)
#
# dev.off() # this writes plot to folder
# graphics.off() # shuts down open devices
```

Figure 1: Microbial Processes Across the Gradient

```
# png(filename="../figures/Figure1.png",
#     width = 1200, height =1200, res = 96*2)
# par(mfrow = c(1,1), mar = c(0, 5, 0, 1) + 0.5, oma = c(6, 2, 0, 0) + 0.5)
# bar.layout <- layout(rbind(1, 2, 3), height = c(3, 3, 3))
# #layout.show(bar.layout)
#
# # Baterial Producivity (BP)
# plot(metab$dist, metab$BP, ylab = "", xlab = "", pch = 22, ylim = c(0, 2),
#      xlim = c(400, -15), cex = 2, bg = "white", col = "black", cex.lab = 2,
#      las = 1, lwd = 2, yaxt = "n", xaxt = "n")
#
# axis(side = 1, lwd.ticks = 2, tck=-0.02, labels = F, cex.axis = 1, las = 1,
#      at = c(0, 100, 200, 300, 400))
# axis(side = 2, lwd.ticks = 2, cex.axis = 1.5, las = 1,
#     labels = c("0.0", "1.0", "2.0"), at = c(0, 1.0, 2.0))
# axis(side = 3, lwd.ticks = 2, tck=-0.01, labels = F,  cex.axis = 2, las = 1,
#      at = c(0, 100, 200, 300, 400))
# axis(side = 4, lwd.ticks = 2, tck=-0.01, labels = F,  cex.axis = 2, las = 1,
#    at = c(0, 1.0, 2.0))
# axis(side = 1, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1,
#      at = c(0, 100, 200, 300, 400))
# axis(side = 2, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1,
#     at = c(0, 1.0, 2.0))
# axis(side = 3, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1,
#      at = c(0, 100, 200, 300, 400))
# axis(side = 4, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1,
#    at = c(0, 1.0, 2.0))
#
# box(lwd = 2)
```

```
#
# mtext(expression(paste('BP (', mu ,'M C h'^-1* ')')), side = 2, line = 4, cex = 1.25)
#
# # Quadratic regression for BP
# dist <- metab$dist
# dist2 <- metab$dist^2
# BP.fit <- lm(metab$BP ~ dist + dist2)
# BP.R2 <- round(summary(BP.fit)$r.squared, 2)
# dist.vals <- seq(0, 375, 25)
# BP.pred <- predict(BP.fit,list(dist = dist.vals, dist2 = dist.vals^2))
# lines(dist.vals, BP.pred, col = "black", lwd = 2.5, lty = 6)
# text(40, 1.8, labels = bquote(italic(R)^2 == .(BP.R2)), cex = 1.5)
#
# # Bacterial Respiration (BR)
# plot(metab$dist, metab$BR, ylab = "", xlab = "", pch = 22, ylim = c(0, 4),
#      xlim = c(400, -15), cex = 2, bg = "white", col = "black", cex.lab = 2,
#      las = 1, lwd = 2, yaxt = "n", xaxt = "n")
#
# axis(side = 1, lwd.ticks = 2, tck=-0.02, labels = F, cex.axis = 1, las = 1,
#      at = c(0, 100, 200, 300, 400))
# axis(side = 2, lwd.ticks = 2, cex.axis = 1.5, las = 1,
#     labels = c("0.0", "2.0", "4.0"), at = c(0, 2, 4))
# axis(side = 3, lwd.ticks = 2, tck=-0.02, labels = F,  cex.axis = 2, las = 1,
#      at = c(0, 100, 200, 300, 400))
# axis(side = 4, lwd.ticks = 2, tck=-0.01, labels = F,  cex.axis = 2, las = 1,
#    at = c(0, 2, 4))
# axis(side = 1, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1,
#      at = c(0, 100, 200, 300, 400))
# axis(side = 2, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1,
#     at = c(0, 2, 4))
# axis(side = 3, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1,
#      at = c(0, 100, 200, 300, 400))
# axis(side = 4, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1,
#    at = c(0, 2, 4))
#
# box(lwd = 2)
#
# mtext(expression(paste('BR (', mu ,'M C h'^-1* ')')), side = 2, line = 4, cex = 1.25)
#
# # Simple linear regression for BR
# BR.fit <- lm(metab$BR ~ metab$dist)
# BR.R2 <- round(summary(BR.fit)$r.squared, 2)
# BR.int <- BR.fit$coefficients[1]
# BR.slp <- BR.fit$coefficients[2]
# clip(0, 375, 0, 4.1)
# abline(a = BR.int, b = BR.slp, col = "black", lwd = 2.5, lty = 6)
# text(40, 3.75, labels = bquote(italic(R)^2 == .(BR.R2)), cex = 1.5)
#
# # Bacterial Growth Efficiency
# plot(metab$dist, metab$BGE, ylab = "", xlab = "", pch = 22, ylim = c(0, 0.6),
#      xlim = c(400, -15), cex = 2, bg = "white", col = "black", cex.lab = 2,
#      las = 1, lwd = 2, yaxt = "n", xaxt = "n")
#
```

```
# axis(side = 1, lwd.ticks = 2, cex.axis = 1.5, las = 1,
#      labels = c("400", "300", "200", "100", "0"), at = c(400, 300, 200, 100, 000))
# axis(side = 2, lwd.ticks = 2, cex.axis = 1.5, las = 1,
#     labels = c("0.0", "0.3", "0.6"), at = c(0, 0.3, 0.6))
# axis(side = 3, lwd.ticks = 2, tck=-0.02, labels = F,  cex.axis = 2, las = 1,
#      at = c(0, 100, 200, 300, 400))
# axis(side = 4, lwd.ticks = 2, tck=-0.01, labels = F,  cex.axis = 2, las = 1,
#    at = c(0, 0.3, 0.6))
# axis(side = 1, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1,
#      at = c(0, 100, 200, 300, 400))
# axis(side = 2, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1,
#     at = c(0, 0.3, 0.6))
# axis(side = 3, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1,
#      at = c(0, 100, 200, 300, 400))
# axis(side = 4, lwd.ticks = 2, tck=0.01, labels = F,  cex.axis = 2, las = 1,
#    at = c(0, 0.3, 0.6))
#
# box(lwd = 2)
#
# mtext("BGE", side = 2, line = 4, cex = 1.25)
# mtext("Distance (m)", side = 1, line = 4, cex = 1.25)
#
# # Simple linear regression for BGE
# BGE.fit <- lm(metab$BGE ~ metab$dist)
# BGE.R2 <- round(summary(BGE.fit)$r.squared, 2)
# BGE.int <- BGE.fit$coefficients[1]
# BGE.slp <- BGE.fit$coefficients[2]
# clip(0, 375, 0, 0.58)
# abline(a = BGE.int, b = BGE.slp, col = "black", lwd = 2.5, lty = 6)
# text(40, 0.535, labels = bquote(italic(R)^2 == .(BGE.R2)), cex = 1.5)
#
# dev.off() # this writes plot to folder
# graphics.off() # shuts down open devices
# par(opar)
```

	Class	Order	DNA		RNA	
			min	max	min	
Otu00001	Comamonadaceae	Comamonadaceae_unclassified	0.00465	0.026	5.1e-05	0.0
Otu00002	Actinomycetales_unclassified	Actinomycetales_unclassified	0.00327	0.127	0	0
Otu00003	Spartobacteria_unclassified	Spartobacteria_unclassified	0.0016	0.06	2.69e-05	0
Otu00005	Chitinophagaceae	Sediminibacterium	0.00155	0.0369	0	0.0
Otu00006	Saprospiraceae	Saprospiraceae_unclassified	0.000158	0.00806	0	0
Otu00008	Actinomycetales_unclassified	Actinomycetales_unclassified	0.000716	0.0288	0	0.0
Otu00009	Pseudomonadaceae	Pseudomonas	0	0.0412	3.1e-05	0
Otu00010	Proteobacteria_unclassified	Proteobacteria_unclassified	0.00297	0.134	4.25e-05	0.0
Otu00011	Betaproteobacteria_unclassified	Betaproteobacteria_unclassified	0.000108	0.0731	5.23e-06	0.0
Otu00012	Comamonadaceae	Comamonadaceae_unclassified	0.00616	0.0186	8.5e-06	
Otu00014	Actinomycetales_unclassified	Actinomycetales_unclassified	0.00108	0.0512	0	0.0
Otu00015	Actinobacteria_unclassified	Actinobacteria_unclassified	0.000363	0.0675	0	0.0
Otu00016	Microbacteriaceae	Microbacteriaceae_unclassified	0.000115	0.0268	0	
Otu00017	Actinomycetales_unclassified	Actinomycetales_unclassified	0.00103	0.0141	0	0
Otu00018	Pseudomonadaceae	Pseudomonas	4.21e-05	0.0328	3.12e-05	0
Otu00019	Cytophagaceae	Cytophagaceae_unclassified	0.000697	0.0844	0	0.0
Otu00020	Alcaligenaceae	Alcaligenaceae_unclassified	0.000777	0.0399	0	
Otu00022	Opitutae_unclassified	Opitutae_unclassified	0.00421	0.0332	5.23e-06	0
Otu00023	Moraxellaceae	Acinetobacter	0	0.00186	1.55e-05	0
Otu00024	Bacteroidetes_unclassified	Bacteroidetes_unclassified	0.000367	0.00679	0	0.0
Otu00025	Microbacteriaceae	Microbacteriaceae_unclassified	0.00233	0.0271	0	0.0
Otu00028	Pseudomonadaceae	Pseudomonas	0	0.0232	5.23e-06	0
Otu00030	Micrococcaceae	Micrococcus	6.84e-05	0.0215	1.56e-05	0
Otu00031	Cyclobacteriaceae	Algoriphagus	0.000735	0.0293	0	0.0
Otu00032	Bacteroidetes_unclassified	Bacteroidetes_unclassified	0.00101	0.0326	0	0
Otu00033	Rhizobiales_unclassified	Rhizobiales_unclassified	0.00136	0.0398	5.16e-06	0
Otu00039	Comamonadaceae	Comamonas	0.000143	0.0142	0	0.0
Otu00040	Acetobacteraceae	Roseomonas	0.00021	0.015	0	
Otu00042	Burkholderiaceae	Burkholderia	0	0.0129	0	0
Otu00045	Oxalobacteraceae	Oxalobacteraceae_unclassified	0.00103	0.0214	0	0.00
Otu00053	Clostridiales_Incertae_Sedis_XI	Finegoldia	0	0.00102	0	0
Otu00057	Methylococcaceae	Methylococcaceae_unclassified	0.000373	0.0179	0	0.0
Otu00059	Micrococcaceae	Arthrobacter	0	0.0435	0	0.00
Otu00063	Verrucomicrobia_unclassified	Verrucomicrobia_unclassified	0.000573	0.0317	0	0.0
Otu00065	Sphingobacteriaceae	Pedobacter	0	0.0344	0	0.0
Otu00069	Xanthomonadaceae	Stenotrophomonas	0	0.000679	0	0
Otu00072	Sphingomonadaceae	Sphingomonas	7.52e-05	0.118	0	0.0
Otu00078	Flavobacteriaceae	Flavobacterium	5.63e-06	0.00306	0	0.00
Otu00081	Flavobacteriaceae	Flavobacterium	0	0.0154	0	0.00
Otu00082	Oxalobacteraceae	Janthinobacterium	0.000957	0.0141	0	0.0
Otu00087	Bradyrhizobiaceae	Bradyrhizobium	7.74e-06	0.000906	0	0.00
Otu00089	Sphingobacteriales_unclassified	Sphingobacteriales_unclassified	3.82e-05	0.0163	0	0.0
Otu00094	Sphingobacteriaceae	Sphingobacterium	0	0.0142	0	0.00
Otu00095	Oxalobacteraceae	Duganella	4.56e-05	0.0269	5.23e-06	0.0
Otu00098	Sphingomonadaceae	Sphingomonadaceae_unclassified	0	0.00101	0	0.000
Otu00118	Comamonadaceae	Comamonadaceae_unclassified	0.00023	0.00495	0	0.0
Otu00144	Methylococcaceae	Methylobacter	0	0.000353	0	8.86
Otu00145	Caulobacteraceae	Phenylobacterium	0	0.00107	0	1.12
Otu00158	Sphingomonadaceae	Sphingomonas	0	0.000484	0	0.000
Otu00162	Aeromonadaceae	Aeromonas	0	0.000611	0	7.07
Otu00279	Rhizobiaceae	Rhizobiaceae_unclassified	7.74e-06	0.00201	0	0.0
Otu00838	Chitinophagaceae	Chitinophagaceae_unclassified	0	0.000162	0	0.000
Otu01248	Subdivision3_unclassified	Subdivision3_unclassified	0	2.87e-05	0	1.78

	Class	Order	DNA		RNA	
			min	max	min	max
Otu00004	Actinomycetales_unclassified	Actinomycetales_unclassified	0.00348	0.0602	0	0.0227
Otu00007	Burkholderiaceae	Polynucleobacter	0.000697	0.0207	0	0.0865
Otu00038	Actinomycetales_unclassified	Actinomycetales_unclassified	0.00153	0.0222	0	0.0986
Otu00080	Bacteroidetes_unclassified	Bacteroidetes_unclassified	1.91e-05	0.0189	0	0.0188
Otu00090	Opitutae_unclassified	Opitutae_unclassified	0	0.00123	0	0.187
Otu00136	Methylococcaceae	Methylomonas	0	0.00192	0	0.0121
Otu00140	Cryomorphaceae	Fluviicola	0	0.000679	0	0.15
Otu00142	Bacteroidetes_unclassified	Bacteroidetes_unclassified	0.000101	0.0053	0	0.0537
Otu00172	Bacteroidetes_unclassified	Bacteroidetes_unclassified	9.55e-06	0.00224	0	0.00231
Otu00173	Bacteria_unclassified	Bacteria_unclassified	0	0.000459	0	0
Otu00532	Bacteroidetes_unclassified	Bacteroidetes_unclassified	0	0.000772	0	0.000571
Otu00633	Nitrosomonadaceae	Nitrosomonas	0	0.000561	0	2.69e-05
Otu01046	Clostridiales_Incertae_Sedis_XI	Anaerococcus	0	6.54e-05	0	1.41e-05
Otu01198	Burkholderiales_unclassified	Burkholderiales_unclassified	0	0.000274	0	4.24e-05