Illumina data, Fungi: Q1

Marissa Lee

12/16/2019

Q1. Given within-plant habitat, do fungal communities diverge/converge based on proximity in the landscape?

Table of contents

Load packages, functions, paths

```
knitr::opts_chunk$set(echo = T)

# paths
merged_path <- "data_intermediates/Illum_analyses/FUN-merged"
out_path <- "output/illumina/Q1"

# custom functions
source("code/helpers.R") # misc helpful fxns
sourceDir("code") # loads all the custom functions in this folder
```

```
## estim_plantGPScoords_bySite.R :
## estim_plantGPScoords.R :
## fxn_dada2.R :
## fxn_rdp.R :
## helpers.R :
```

```
## load_bgc.R :
## load_siteinfo.R :
```

```
# formatting
require("tidyverse"); packageVersion("tidyverse")
```

```
## Loading required package: tidyverse

## -- Attaching packages ------------------------------------------------------------------------- tidyve

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts --------------------------------------------------------------------------- tidyverse_c
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

## [1] '1.3.0'
```

```
require("readxl"); packageVersion("readxl") # to read in excel files
```

```
## Loading required package: readxl

## [1] '1.3.1'
```

```
#library("ggsci"); packageVersion("ggsci") # pretty colors
library("gridExtra"); packageVersion("gridExtra")
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

## [1] '2.3'
```

```
# stats
#library("compositions"); packageVersion("compositions") # clr()
#library("philr") #
library("RRPP"); packageVersion("RRPP")
```

```
## [1] '0.6.1'
```

```
library("vegan"); packageVersion("vegan") #-- needed?
```

```
## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.5-6

## [1] '2.5.6'
```

```
#library("gdm")

# bioinformatics
#library("DESeq2")
library("phyloseq");  packageVersion("phyloseq")
```

```
##
```

```
## Attaching package: 'phyloseq'

## The following object is masked from 'package:RRPP':
##
##     ordinate

## [1] '1.32.0'
```

```r
library("speedyseq")
```

```
##
## Attaching package: 'speedyseq'

## The following objects are masked from 'package:phyloseq':
##
##     filter_taxa, plot_bar, plot_heatmap, plot_tree, psmelt, tax_glom,
##     tip_glom, transform_sample_counts
```

```r
#library("DECIPHER"); packageVersion("DECIPHER")
#library("ape"); packageVersion("ape")
```

Custom functions

```r
# calc geometric mean of each ASV
gm_mean = function(x, na.rm=TRUE){ exp(sum(log(x[x > 0]), na.rm=na.rm) / length(x))}

extract_uniquePairDists<-function(dist.mat){

  x<-as.matrix(dist.mat)
  rowCol <- expand.grid(rownames(x), colnames(x))
  labs <- rowCol[as.vector(upper.tri(x,diag=F)),]
  df <- cbind(labs, x[upper.tri(x,diag=F)])
  colnames(df) <- c("sp1","sp2","dist")

  return(df)
}

make_dist_df <- function(ps, vst, raodis, bray.dist, physor.dist, env.dist.df){

  ####
  # calculate pairwise community distances w/ vst
  asv.dist <- dist(vst) # calculate pairwise distances
  asv.dist.l <- extract_uniquePairDists(asv.dist) # put into a dataframe
  asv.dist.l %>%
    dplyr::rename('vst.comm.dist'='dist') -> asv.dist.l.vst

  # format pairwise Rao distances
  rao.mat <- as.matrix(raodis)
  rao.dist.l <- extract_uniquePairDists(rao.mat) # put into a dataframe
  rao.dist.l %>%
    dplyr::rename('rao.comm.dist'='dist') -> asv.dist.l.rao

  # format pairwise Bray distances
  b.mat <- as.matrix(bray.dist)
  b.dist.l <- extract_uniquePairDists(b.mat) # put into a dataframe
  b.dist.l %>%
    dplyr::rename('bray.comm.dist'='dist') -> asv.dist.l.b
```

```r
# format pairwise phylosor distances
p.mat <- as.matrix(physor.dist)
p.dist.l <- extract_uniquePairDists(p.mat) # put into a dataframe
p.dist.l %>%
  dplyr::rename('physor.comm.dist'='dist') -> asv.dist.l.p

# calculate pairwise spatial distances
sam <- data.frame(sample_data(ps))
sam %>%
  dplyr::select(samp.lon, samp.lat) -> samp.gps
require(geodist)
hav.dist <- geodist(samp.gps,
                    paired = TRUE,
                    sequential = FALSE, pad = FALSE,
                    measure = "haversine")
colnames(hav.dist) <- row.names(samp.gps)
row.names(hav.dist) <- row.names(samp.gps)
hav.dist.df <- extract_uniquePairDists(hav.dist)
hav.dist.df %>%
  dplyr::rename('hav.dist.m'='dist') -> hav.dist.df

# load pairwise environmental distances
#env.dist.df

####
# put it all together
asv.dist.l.vst %>%
  left_join(asv.dist.l.rao) %>%
  left_join(asv.dist.l.b) %>%
  left_join(asv.dist.l.p) %>%
  left_join(hav.dist.df) %>%
  left_join(env.dist.df) -> dist.df
sam %>%
  dplyr::select(sample.name.match, Site, Samp, Tissue) -> sam.indx
dist.df %>%
  dplyr::rename('sample.name.match'= 'sp1') %>%
  left_join(sam.indx) %>%
  dplyr::rename('Site_samp1'= 'Site',
                'Samp_samp1'= 'Samp',
                'Tissue_samp1'='Tissue',
                'samp1'='sample.name.match') %>%
  dplyr::rename('sample.name.match'= 'sp2') -> dist.df
dist.df %>%
  left_join(sam.indx) %>%
  dplyr::rename('Site_samp2'= 'Site',
                'Samp_samp2'= 'Samp',
                'Tissue_samp2'='Tissue',
                'samp2'='sample.name.match') -> dist.df

# code the types of site and tissue comparisons
dist.df %>%
  mutate(sameSite = ifelse(Site_samp1 == Site_samp2, TRUE, FALSE)) %>%
  mutate(sameTissue = ifelse(Tissue_samp1 == Tissue_samp2, TRUE, FALSE)) -> dist.df
  # remove distances between difference tissue types
  dist.df %>%
    filter(sameTissue == T) %>%
    mutate(hav.dist.km = hav.dist.m/1000) -> dist.df

  return(dist.df)

}
```

Set plotting parameters

```
tissue.colors <- c("#288737", "#4678a8", "#cbba4e")
names(tissue.colors) <- c("L","R","S")
```

Print sample data, ASV matrix, and taxonomy table [commented out so it doesn't get overwritten unnessarily]

```
# ps <- readRDS(file = file.path(merged_path, "phyloseq_samps_env_trimTreeASVs.RData"))
# ps
#
# # ASV matrix
# otu <- otu_table(ps)
# otu.df <- data.frame(otu)
# dim(otu.df)
# write.csv(otu.df, file = "data/ASVmatrix.csv")
#
# # taxonomy table
# tax <- tax_table(ps)
# tax.df <- data.frame(tax)
# write.csv(tax.df, file = "data/TAXmatrix.csv")
#
# # sample table
# sam <- sample_data(ps)
# sam.df <- data.frame(sam)
# dim(sam.df)
# colnames(sam.df)
#
# sam.df %>%
#   separate(Site, into = c("t1","t2",NA)) %>%
#   mutate(Site = paste0(t1, "-",t2)) %>%
#   select(sample.name.match, Site, mono.mixed, cultivar, plotarea.m2,
#          max.height.m, basal.area.m2, stand.age.yrs,
#          ph, perc.C, watercontent, doc,
#          p.resin, TIN, SOM, W.V, mbc, Cu, K, Mg, Mn, P, Zn, Ca, perc.N, S,
#          perc.clay, perc.sand, MAP.mm, MAT.C,
#          lat, lon, samp.lat, samp.lon) -> sam.out
# write.csv(sam.out, file = "data/SAMmatrix.csv")
```

A. Do and save calculations for VST and DPCoA objects

1. VST

```
# ps <- readRDS(file = file.path(merged_path, "phyloseq_samps_env_trimTreeASVs.RData"))
# ps
#
# # # calculate vst
# ps_ds <- phyloseq_to_deseq2(ps, ~1) # convert phyloseq to DeSeq object
# geoMeans = apply(counts(ps_ds), 1, gm_mean) # calc geometric mean of each ASV
# ps_ds = estimateSizeFactors(ps_ds, type="ratio", geoMeans = geoMeans)
# ps_ds = estimateDispersions(ps_ds, fitType = "parametric")
# #plotDispEsts(ps_ds) # plot the dispersion estimates
# vst <- getVarianceStabilizedData(ps_ds)
# vst <- t(vst) # need to make the rows samples
# saveRDS(vst, file = file.path("output/illumina/QO", "vst_all.RData"))
# vst <- readRDS(file = file.path("output/illumina/QO", "vst_all.RData"))
```

2. DPCoA and Rao distance matrix

```
ps <- readRDS(file = file.path(merged_path, "phyloseq_samps_env_trimTreeASVs.RData"))
ps
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 932 taxa and 332 samples ]
## sample_data() Sample Data:       [ 332 samples by 75 sample variables ]
## tax_table()   Taxonomy Table:    [ 932 taxa by 9 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 932 tips and 254 internal nodes ]
## refseq()      DNAStringSet:      [ 932 reference sequences ]
```

```
# DPCoA distance matrix
#tree <- phy_tree(ps)
#asv <- data.frame(otu_table(ps), stringsAsFactors = F)
# # square root of the cophenetic/patristic (cophenetic.phylo)
# # cophenetic.phylo = pairwise distances between the pairs of tips from a phylogenetic tree using its
#library(ade4); packageVersion("ade4")
#library(ape); packageVersion("ape")
#phylo.dist <- cophenetic.phylo(tree)
#phylo.dist <- as.dist(phylo.dist)
#sqrt.phylo.dist <- sqrt(phylo.dist)
# # #is.euclid(sqrt.phylo.dist)
#ps.dpcoa <- dpcoa(df = asv, dis = sqrt.phylo.dist, scannf = FALSE, nf = 2, RaoDecomp = TRUE)
#ps.raodis <- ps.dpcoa$RaoDis
#DPCoA seeks to represent the relationship between the locations and species with meaningful mea- sures
#saveRDS(ps.dpcoa, file = file.path("output/illumina/QO", "dpcoa_all.RData"))
#saveRDS(ps.raodis, file = file.path("output/illumina/QO", "dpcoa_all_raodist.RData"))

#ps.dpcoa <- readRDS("output/illumina/QO/dpcoa_all.RData")
#raodis <- readRDS("output/illumina/QO/dpcoa_all_raodist.RData")
```

B. Does fungal community composition differ between within-plant habitat, sites, and their interaction?

1. RRPP w/ VST and RaoDist – Full model

```
# package.version("rrpp")
# package.version("ade4")
# package.version("DESeq2")
ps <- readRDS(file = file.path(merged_path, "phyloseq_samps_env_trimTreeASVs.RData"))
ps
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 932 taxa and 332 samples ]
## sample_data() Sample Data:       [ 332 samples by 75 sample variables ]
## tax_table()   Taxonomy Table:    [ 932 taxa by 9 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 932 tips and 254 internal nodes ]
## refseq()      DNAStringSet:      [ 932 reference sequences ]
```

```
# load VST matrix
vst <- readRDS(file = file.path("output/illumina/Q0", "vst_all.RData"))

# load Rao distance matrix
raodis <- readRDS("output/illumina/Q0/dpcoa_all_raodist.RData")
sum(row.names(vst) != row.names(raodis)) # this needs to be 0, samples in the same order
```

```
## [1] 0
```

```
# add sample data
sam <- data.frame(sample_data(ps))
sam$Tissue <- factor(sam$Tissue)
sam$Samp <- factor(sam$Samp)
sam$Site <- factor(sam$Site)

library(RRPP)

### VST
fit.vst <- lm.rrpp(vst ~ Site + Tissue + Site:Tissue, data = sam, SS.type = "III", iter = 99)
```

```
##
## Preliminary Model Fit...
##
##
## Coefficients estimation: 100 permutations.
##   |                                                                                |
```

```
fit.vst$LM$term.labels  #check order of model terms
```

```
## [1] "Site"        "Tissue"      "Site:Tissue"
```

```
anova.fit.vst <- anova(fit.vst, effect.type = "F",
                       error = c("Site:Tissue","Site:Tissue","Residuals"))
summary(anova.fit.vst, formula = false)
```

```
##
```

```
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals
## Number of permutations: 100
## Estimation method: Ordinary Least Squares
## Sums of Squares and Cross-products: Type III
## Effect sizes (Z) based on F distributions
##
##              Df      SS    MS     Rsq      F      Z Pr(>F)
## Site         13  232943 17919 0.08968 1.0483 3.5786   0.01 *
## Tissue        2   63864 31932 0.02459 1.8681 5.9825   0.01 *
## Site:Tissue  26  444415 17093 0.17110 3.5314 9.7073   0.01 *
## Residuals   290 1403688  4840 0.54042
## Total       331 2597399
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call: lm.rrpp(f1 = vst ~ Site + Tissue + Site:Tissue, iter = 99, SS.type = "III",
##     data = sam)
```

```
capture.output(summary(anova.fit.vst, formula = false),
               file = file.path(out_path,"rrppVST_Site_x_Tissue.txt"))

### Rao
rdf <- rrpp.data.frame(d = raodis,
                       Site = factor(sam$Site),
                       Samp = factor(sam$Samp),
                       Tissue = factor(sam$Tissue))
fit.rao <- lm.rrpp(d ~ Site + Tissue + Site:Tissue, data = rdf, SS.type = "III", iter = 99)
```

```
##
## Preliminary Model Fit...
##
##
## Coefficients estimation: 100 permutations.
##   |                                                                              |
```

```
fit.rao$LM$term.labels  #check order of model terms
```

```
## [1] "Site"        "Tissue"      "Site:Tissue"
```

```
anova.fit.rao <- anova(fit.rao, effect.type = "F",
                       error = c("Site:Tissue","Site:Tissue","Residuals"))
summary(anova.fit.rao, formula = false)
```

```
##
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals
## Number of permutations: 100
## Estimation method: Ordinary Least Squares
## Sums of Squares and Cross-products: Type III
## Effect sizes (Z) based on F distributions
##
##              Df    SS     MS     Rsq      F      Z Pr(>F)
## Site          13  2785 214.22 0.08154 0.9669 0.9457   0.18
## Tissue         2  1106 552.97 0.03238 2.4960 4.1899   0.01 *
## Site:Tissue   26  5760 221.54 0.16866 3.6500 8.6460   0.01 *
## Residuals    290 17602  60.70 0.51540
## Total        331 34151
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call: lm.rrpp(f1 = d ~ Site + Tissue + Site:Tissue, iter = 99, SS.type = "III",
##     data = rdf)
```

```
capture.output(summary(anova.fit.rao, formula = false),
               file = file.path(out_path,"rrppRAO_Site_x_Tissue.txt"))


# #---------------#
# # Plot the prediction ordination
# pred.df <- data.frame(unique(sam[,c("Site","Tissue")]), row.names = NULL)
# pred <- predict(fit, pred.df, confidence = 0.95)
# plot(pred, PC = TRUE)
#
# pc.mean <- pred$pc.mean[,1:2]
# pc.ucl<- pred$pc.ucl[,1:2]
# pc.lcl<- pred$pc.lcl[,1:2]
# plot.df <- data.frame(Site = pred.df$Site, Tissue = pred.df$Tissue,
#                       mean = pc.mean, ucl = pc.ucl, lcl = pc.lcl)
# plot.df %>%
#     separate(Site, into = c("thing1","thing2","thing3")) %>%
#     mutate(Site.pretty = paste(thing1, thing2, sep = "-")) -> plot.df
#
# p.all <- ggplot(plot.df, aes(x = mean.PC1, y = mean.PC2, color = Tissue)) +
#   geom_point() +
#   theme_classic() +
#   xlab("PC1 (20%)") + ylab("PC2 (6.44%)") +
#   geom_errorbar(aes(ymin = lcl.PC2, ymax = ucl.PC2)) +
#   geom_errorbarh(aes(xmin = lcl.PC1, xmax = ucl.PC1))
# p.all
# ggsave(p.all, filename = file.path(out_path,"rrpp_SiteTissue.pdf"), width = 6, height = 4)
```

Yes, there is a strong Site x Tissue interaction

2. Examine each within-plant habitat individually – don't re-calculate VST or Rao

Leaf

```
library(tidyverse)
sam %>%
  filter(Tissue == "L") -> curr.sam
curr.vst <- vst[row.names(vst) %in% curr.sam$sample.name.match,]
sum(row.names(curr.vst) != curr.sam$sample.name.match) # this needs to be 0
```

```
## [1] 0
```

```
curr.sam$Samp <- factor(curr.sam$Samp)
curr.sam$Site <- factor(curr.sam$Site)
```

```
library("usedist")
raodis.curr <- dist_subset(raodis, curr.sam$sample.name.match)
rdf.curr <- rrpp.data.frame(d = raodis.curr,
                        Site = factor(curr.sam$Site),
                        Samp = factor(curr.sam$Samp))

### VST
fit.vst <- lm.rrpp(curr.vst ~ Site, data = curr.sam,
               SS.type = "III", iter = 99, print.progress = T)
```

```
##
## Preliminary Model Fit...

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(iOpt): no non-missing arguments to max; returning -Inf

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(iOpt): no non-missing arguments to max; returning -Inf

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(iOpt): no non-missing arguments to max; returning -Inf

##
## Coefficients estimation: 100 permutations.
##   |                                                                       |
```

```
fit.vst$LM$term.labels  #check order of model terms
```

```
## [1] "Site"
```

```
anova.fit.vst <- anova(fit.vst, effect.type = "F", error = c("Residuals"))
summary(anova.fit.vst, formula = false)
```

```
##
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals
## Number of permutations: 100
## Estimation method: Ordinary Least Squares
## Sums of Squares and Cross-products: Type III
## Effect sizes (Z) based on F distributions
##
##            Df     SS    MS     Rsq     F      Z Pr(>F)
## Site       13 232943 17919 0.48278 6.821 9.2493   0.01 *
## Residuals  95 249563  2627 0.51722
## Total     108 482506
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call: lm.rrpp(f1 = curr.vst ~ Site, iter = 99, SS.type = "III", data = curr.sam,
##     print.progress = T)
```

```
capture.output(summary(anova.fit.vst, formula = false), file = file.path(out_path,"rrppVST_Site_LEAF.tx
### Rao
fit.rao <- lm.rrpp(d ~ Site, data = rdf.curr,
                SS.type = "III", iter = 99, print.progress = T)
```

```
##
## Preliminary Model Fit...

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(iOpt): no non-missing arguments to max; returning -Inf

##
## Coefficients estimation: 100 permutations.
##   |                                                                                |
```

```
fit.rao$LM$term.labels  #check order of model terms
```

```
## [1] "Site"
```

```
anova.fit.rao <- anova(fit.rao, effect.type = "F", error = c("Residuals"))
summary(anova.fit.rao, formula = false)
```

```
##
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals
## Number of permutations: 100
## Estimation method: Ordinary Least Squares
## Sums of Squares and Cross-products: Type III
## Effect sizes (Z) based on F distributions
##
##            Df     SS      MS     Rsq      F      Z Pr(>F)
## Site       13 2784.8 214.218 0.46411 6.3288 7.2834   0.01 *
## Residuals  95 3215.6  33.848 0.53589
## Total     108 6000.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call: lm.rrpp(f1 = d ~ Site, iter = 99, SS.type = "III", data = rdf.curr,
##     print.progress = T)
```

```
capture.output(summary(anova.fit.rao, formula = false), file = file.path(out_path,"rrppRAO_Site_LEAF.tx

#---------------#
# # Plot the prediction ordination
# pred.df <- data.frame(Site = unique(curr.sam[,c("Site")]), row.names = NULL)
# pred <- predict(fit, pred.df, confidence = 0.95)
# plot(pred, PC = TRUE)
#
# pc.mean <- pred$pc.mean[,1:2]
# pc.ucl<- pred$pc.ucl[,1:2]
# pc.lcl<- pred$pc.lcl[,1:2]
# plot.df <- data.frame(Site = pred.df$Site,
#                        mean = pc.mean, ucl = pc.ucl, lcl = pc.lcl)
# plot.df %>%
#     separate(Site, into = c("thing1","thing2","thing3")) %>%
#     mutate(Site.pretty = paste(thing1, thing2, sep = "-")) -> plot.df
#
# p.l <- ggplot(plot.df, aes(x = mean.PC1, y = mean.PC2)) +
#   geom_point() +
#   geom_text(aes(label = Site.pretty), hjust = -0.1, vjust = 1.1, size = 3) +
#   theme_classic() +
#   xlab("PC1 (17.65%)") + ylab("PC2 (14.5%)") +
#   geom_errorbar(aes(ymin = lcl.PC2, ymax = ucl.PC2)) +
#   geom_errorbarh(aes(xmin = lcl.PC1, xmax = ucl.PC1)) +
#   ggtitle("a. Leaf")
# p.l
# ggsave(p.l, filename = file.path(out_path,"rrpp_Site_l.pdf"), width = 5, height = 4)
```

Yes, leaf communities differ more between than within sites.

Root

```
sam %>%
  filter(Tissue == "R") -> curr.sam
curr.vst <- vst[row.names(vst) %in% curr.sam$sample.name.match,]
sum(row.names(curr.vst) != curr.sam$sample.name.match) # this needs to be 0
```

```
## [1] 0
```

```
curr.sam$Samp <- factor(curr.sam$Samp)
curr.sam$Site <- factor(curr.sam$Site)
#library("usedist")
raodis.curr <- dist_subset(raodis, curr.sam$sample.name.match)
rdf.curr <- rrpp.data.frame(d = raodis.curr,
                       Site = factor(curr.sam$Site),
                       Samp = factor(curr.sam$Samp))

### VST
fit.vst <- lm.rrpp(curr.vst ~ Site, data = curr.sam,
               SS.type = "III", iter = 99, print.progress = T)
```

```
##
## Preliminary Model Fit...

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(iOpt): no non-missing arguments to max; returning -Inf

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(iOpt): no non-missing arguments to max; returning -Inf

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(iOpt): no non-missing arguments to max; returning -Inf

##
## Coefficients estimation: 100 permutations.
##   |                                                                              |
## Sums of Squares calculations: 100 permutations.
##   |                                                                          |
```

```
fit.vst$LM$term.labels  #check order of model terms
```

```
## [1] "Site"
```

```
anova.fit.vst <- anova(fit.vst, effect.type = "F", error = c("Residuals"))
summary(anova.fit.vst, formula = false)
```

```
##
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals
## Number of permutations: 100
## Estimation method: Ordinary Least Squares
## Sums of Squares and Cross-products: Type III
## Effect sizes (Z) based on F distributions
##
##             Df     SS      MS     Rsq     F      Z Pr(>F)
## Site        13 192057 14773.6 0.30568 3.285 9.3721   0.01 *
## Residuals   97 436244  4497.4 0.69432
## Total      110 628302
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call: lm.rrpp(f1 = curr.vst ~ Site, iter = 99, SS.type = "III", data = curr.sam,
##     print.progress = T)
```

```
capture.output(summary(anova.fit.vst, formula = false), file = file.path(out_path,"rrppVST_Site_ROOT.tx

### Rao
fit.rao <- lm.rrpp(d ~ Site, data = rdf.curr,
               SS.type = "III", iter = 99, print.progress = T)
```

```
##
## Preliminary Model Fit...

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(iOpt): no non-missing arguments to max; returning -Inf

##
## Coefficients estimation: 100 permutations.
##   |                                                                          |
```

```
fit.rao$LM$term.labels  #check order of model terms
```

```
## [1] "Site"
```

```
anova.fit.rao <- anova(fit.rao, effect.type = "F", error = c("Residuals"))
summary(anova.fit.rao, formula = false)
```

```
##
```

```
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals
## Number of permutations: 100
## Estimation method: Ordinary Least Squares
## Sums of Squares and Cross-products: Type III
## Effect sizes (Z) based on F distributions
##
##            Df      SS     MS     Rsq      F      Z Pr(>F)
## Site       13  4995.6 384.28 0.34635 3.9536 7.4638   0.01 *
## Residuals  97  9428.1  97.20 0.65365
## Total     110 14423.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call: lm.rrpp(f1 = d ~ Site, iter = 99, SS.type = "III", data = rdf.curr,
##     print.progress = T)
```

```
capture.output(summary(anova.fit.rao, formula = false), file = file.path(out_path,"rrppRAO_Site_ROOT.tx


# #---------------#
# # Plot the prediction ordination
# pred.df <- data.frame(Site = unique(curr.sam[,c("Site")]), row.names = NULL)
# pred <- predict(fit, pred.df, confidence = 0.95)
# plot(pred, PC = TRUE)
#
# pc.mean <- pred$pc.mean[,1:2]
# pc.ucl<- pred$pc.ucl[,1:2]
# pc.lcl<- pred$pc.lcl[,1:2]
# plot.df <- data.frame(Site = pred.df$Site,
#                       mean = pc.mean, ucl = pc.ucl, lcl = pc.lcl)
# plot.df %>%
#     separate(Site, into = c("thing1","thing2","thing3")) %>%
#     mutate(Site.pretty = paste(thing1, thing2, sep = "-")) -> plot.df
#
# p.r <- ggplot(plot.df, aes(x = mean.PC1, y = mean.PC2)) +
#   geom_point() +
#   geom_text(aes(label = Site.pretty), hjust = -0.1, vjust = 1.1, size = 3) +
#   theme_classic() +
#   xlab("PC1 (13.03%)") + ylab("PC2 (13.84%)") +
#   geom_errorbar(aes(ymin = lcl.PC2, ymax = ucl.PC2)) +
#   geom_errorbarh(aes(xmin = lcl.PC1, xmax = ucl.PC1)) +
#   ggtitle("b. Root")
# p.r
# ggsave(p.r, filename = file.path(out_path,"rrpp_Site_r.pdf"), width = 5, height = 4)
```

Yes, root communities differ more between than within sites

Soil

```
sam %>%
  filter(Tissue == "S") -> curr.sam
curr.vst <- vst[row.names(vst) %in% curr.sam$sample.name.match,]
sum(row.names(curr.vst) != curr.sam$sample.name.match) # this needs to be 0
```

```
## [1] 0
```

```
curr.sam$Samp <- factor(curr.sam$Samp)
curr.sam$Site <- factor(curr.sam$Site)
#library("usedist")
raodis.curr <- dist_subset(raodis, curr.sam$sample.name.match)
rdf.curr <- rrpp.data.frame(d = raodis.curr,
                       Site = factor(curr.sam$Site),
                       Samp = factor(curr.sam$Samp))

### VST
fit.vst <- lm.rrpp(curr.vst ~ Site, data = curr.sam,
               SS.type = "III", iter = 99, print.progress = T)
```

```
##
## Preliminary Model Fit...

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(iOpt): no non-missing arguments to max; returning -Inf

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(iOpt): no non-missing arguments to max; returning -Inf

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(iOpt): no non-missing arguments to max; returning -Inf

##
## Coefficients estimation: 100 permutations.
##   |                                                                       |
```

```
fit.vst$LM$term.labels  #check order of model terms
```

```
## [1] "Site"
```

```
anova.fit.vst <- anova(fit.vst, effect.type = "F", error = c("Residuals"))
summary(anova.fit.vst, formula = false)
```

```
##
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals
## Number of permutations: 100
## Estimation method: Ordinary Least Squares
## Sums of Squares and Cross-products: Type III
## Effect sizes (Z) based on F distributions
##
##            Df      SS    MS     Rsq     F      Z Pr(>F)
## Site       13  421105 32393 0.36972 4.422 9.4401   0.01 *
## Residuals  98  717881  7325 0.63028
## Total     111 1138986
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call: lm.rrpp(f1 = curr.vst ~ Site, iter = 99, SS.type = "III", data = curr.sam,
##     print.progress = T)
```

```
capture.output(summary(anova.fit.vst, formula = false), file = file.path(out_path,"rrppVST_Site_SOIL.tx

### Rao
fit.rao <- lm.rrpp(d ~ Site, data = rdf.curr,
               SS.type = "III", iter = 99, print.progress = T)
```

```
##
## Preliminary Model Fit...

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(i): no non-missing arguments to max; returning -Inf

## Warning in max(iOpt): no non-missing arguments to max; returning -Inf

##
## Coefficients estimation: 100 permutations.
##   |                                                                              |
```

```
fit.rao$LM$term.labels  #check order of model terms
```

```
## [1] "Site"
```

```
anova.fit.rao <- anova(fit.rao, effect.type = "F", error = c("Residuals"))
summary(anova.fit.rao, formula = false)
```

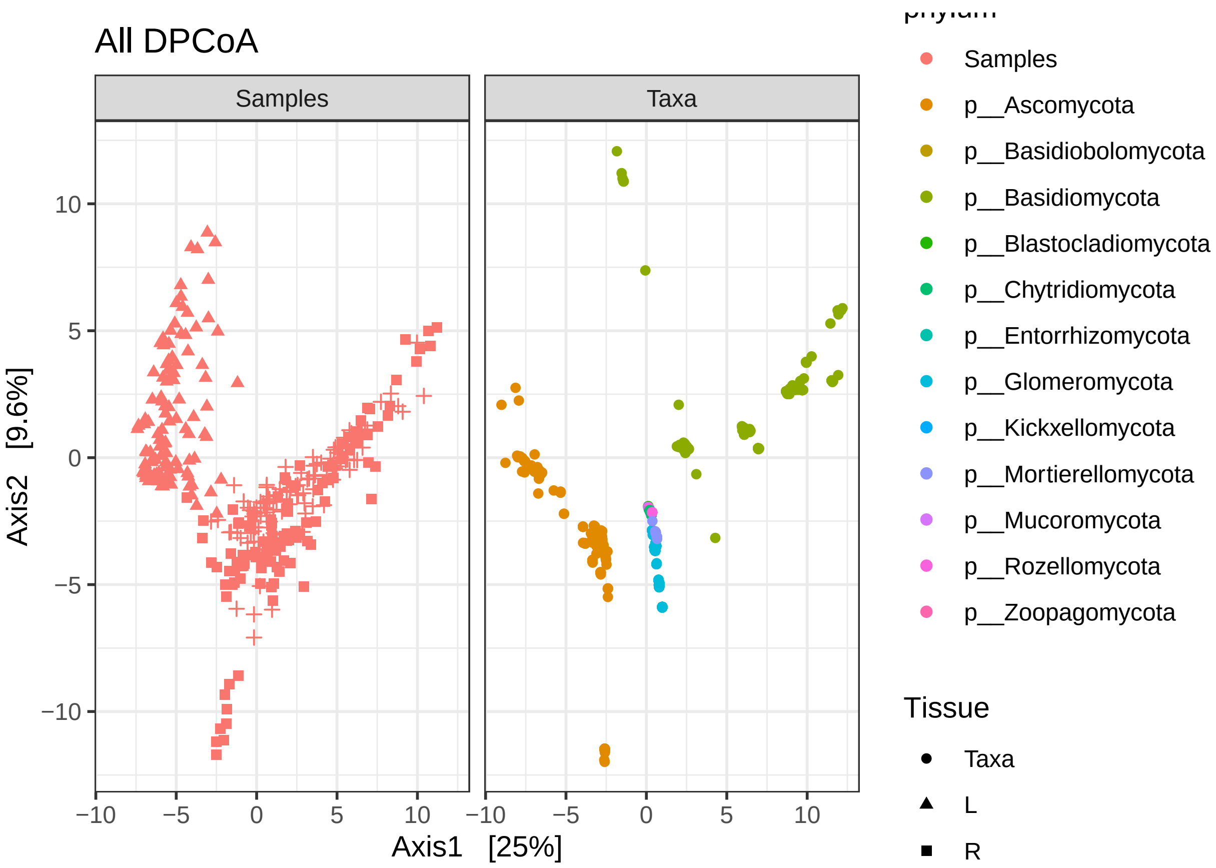
```
##
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals
## Number of permutations: 100
## Estimation method: Ordinary Least Squares
## Sums of Squares and Cross-products: Type III
## Effect sizes (Z) based on F distributions
##
##            Df     SS      MS     Rsq      F      Z Pr(>F)
## Site       13 2448.2 188.325 0.33056 3.7224 7.7524   0.01 *
## Residuals  98 4958.1  50.593 0.66944
## Total     111 7406.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call: lm.rrpp(f1 = d ~ Site, iter = 99, SS.type = "III", data = rdf.curr,
##     print.progress = T)
```

```
capture.output(summary(anova.fit.rao, formula = false), file = file.path(out_path,"rrppRAO_Site_SOIL.tx


#
# ###
# #---------------#
# # Plot the prediction ordination
# pred.df <- data.frame(Site = unique(curr.sam[,c("Site")]), row.names = NULL)
# pred <- predict(fit, pred.df, confidence = 0.95)
# plot(pred, PC = TRUE)
#
# pc.mean <- pred$pc.mean[,1:2]
# pc.ucl<- pred$pc.ucl[,1:2]
# pc.lcl<- pred$pc.lcl[,1:2]
# plot.df <- data.frame(Site = pred.df$Site,
#                        mean = pc.mean, ucl = pc.ucl, lcl = pc.lcl)
# plot.df %>%
#     separate(Site, into = c("thing1","thing2","thing3")) %>%
#     mutate(Site.pretty = paste(thing1, thing2, sep = "-")) -> plot.df
#
# p.s <- ggplot(plot.df, aes(x = mean.PC1, y = mean.PC2)) +
#   geom_point() +
#   geom_text(aes(label = Site.pretty), hjust = -0.1, vjust = 1.1, size = 3) +
#   theme_classic() +
#   xlab("PC1 (13.2%)") + ylab("PC2 (12.28%)") +
#   geom_errorbar(aes(ymin = lcl.PC2, ymax = ucl.PC2)) +
#   geom_errorbarh(aes(xmin = lcl.PC1, xmax = ucl.PC1)) +
#   ggtitle("c. Soil")
# p.s
# ggsave(p.s, filename = file.path(out_path,"rrpp_Site_s.pdf"), width = 5, height = 4)
```

Yes, soil communities differ more between than within sites

3. Make DPCoA plot

```
ps <- readRDS(file = file.path(merged_path, "phyloseq_samps_env_trimTreeASVs.RData"))
tree <- phy_tree(ps)
asv <- data.frame(otu_table(ps), stringsAsFactors = F)
# # # square root of the cophenetic/patristic (cophenetic.phylo)
# # # cophenetic.phylo = pairwise distances between the pairs of tips from a phylogenetic tree using its
# # #detach("package:compositions", unload = TRUE)
# library(ade4); packageVersion("ade4")
# phylo.dist <- cophenetic.phylo(tree)
# phylo.dist <- as.dist(phylo.dist)
# sqrt.phylo.dist <- sqrt(phylo.dist)
# mod.all <- dpcoa(df = asv, dis = sqrt.phylo.dist, scannf = FALSE, nf = 2, RaoDecomp = TRUE)
#saveRDS(mod.all, file = file.path(out_path, "dpcoa_all.RData"))
mod.all <- readRDS(file = file.path("output/illumina/Q0", "dpcoa_all.RData"))
plot_ordination(ps, mod.all, type="split",
                color = "phylum", shape = "Tissue") +
  ggplot2::scale_colour_discrete() +
  ggplot2::theme_bw() +
  ggtitle("All DPCoA")
```



```r
# ggsave(filename = file.path(out_path, "dpcoa_all.pdf"),
#        width = 6, height = 4)
#

# export the ASV scores
df <- data.frame(ASV = row.names(mod.all$dls),
                 DPCoA1 = mod.all$dls$CS1,
                 DPCoA2 = mod.all$dls$CS2)
tax <- data.frame(tax_table(ps), stringsAsFactors = F)
tax %>%
  left_join(df) %>%
  select(ASV, DPCoA1, DPCoA2, kingdom, phylum, class, order, family, genus, species) -> df.tax
```

```
## Joining, by = "ASV"
```

```r
write.csv(df.tax, file = file.path(out_path, "asv_dpcoaScores.csv"), row.names = F)

# export the sample scores
sam <- data.frame(sample_data(ps), stringsAsFactors = F)
df.sam <- data.frame(sample.name.match = row.names(mod.all$li),
           DPCoA1 = mod.all$li$Axis1,
           DPCoA2 = mod.all$li$Axis2)
df.sam %>%
  left_join(sam) %>%
  dplyr::rename('sample'='sample.name.match') %>%
  select(sample, DPCoA1, DPCoA2, Tissue, mono.mixed) -> df.sam
```

```
## Joining, by = "sample.name.match"
```

```
write.csv(df.sam, file = file.path(out_path, "sample_dpcoaScores.csv"), row.names = F)
```

C. Does alpha diversity differ between within-plant habitat?

Summarize the number of ASVs per phylum in each compartment

```
ps <- readRDS(file = file.path(merged_path, "phyloseq_samps_env_trimTreeASVs.RData"))
ps
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 932 taxa and 332 samples ]
## sample_data() Sample Data:       [ 332 samples by 75 sample variables ]
## tax_table()   Taxonomy Table:    [ 932 taxa by 9 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 932 tips and 254 internal nodes ]
## refseq()      DNAStringSet:      [ 932 reference sequences ]
```

```
otu.df <- data.frame(otu_table(ps), stringsAsFactors = F)
otu.df <- data.frame(sample.name.match = row.names(otu.df), otu.df, stringsAsFactors = F)
otu.df %>%
  gather(key = "ASV", value = "abund", -sample.name.match) -> otu.l
sam <- data.frame(sample_data(ps), stringsAsFactors = F)
sam %>%
  select(sample.name.match, Tissue) -> sam.indx

sam %>%
  filter(Tissue == "L") %>%
  dim()
```

```
## [1] 109  75
```

```
sam %>%
  filter(Tissue == "R") %>%
  dim()
```

```
## [1] 111  75
```

```
sam %>%
  filter(Tissue == "S") %>%
  dim()
```

```
## [1] 112  75
```

```
tax <- data.frame(tax_table(ps), stringsAsFactors = F)
tax %>%
  select(ASV, phylum) -> tax.indx
otu.l %>%
  left_join(sam.indx) %>%
  left_join(tax.indx) -> otu.l
```

```
## Joining, by = "sample.name.match"
```

```
## Joining, by = "ASV"
```

```
otu.l %>%
  filter(abund > 0) %>%
  group_by(phylum) %>%
  summarize(n = length(unique(ASV))) -> all
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
all
```

```
## # A tibble: 12 x 2
##    phylum                    n
##    <chr>                 <int>
##  1 p__Ascomycota           277
##  2 p__Basidiobolomycota      1
##  3 p__Basidiomycota        249
##  4 p__Blastocladiomycota     2
##  5 p__Chytridiomycota       76
##  6 p__Entorrhizomycota       1
##  7 p__Glomeromycota        253
##  8 p__Kickxellomycota        1
##  9 p__Mortierellomycota     26
## 10 p__Mucoromycota           9
## 11 p__Rozellomycota         36
## 12 p__Zoopagomycota          1
```

```
sum(all$n)
```

```
## [1] 932
```

```
all[all$phylum == "p__Glomeromycota","n"]
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   253
```

```
76/sum(all$n)
```

```
## [1] 0.08154506
```

```
108/253
```

```
## [1] 0.4268775
```

```
otu.l %>%
  filter(abund > 0) %>%
  group_by(Tissue, phylum) %>%
  summarize(n = length(unique(ASV))) %>%
  spread(key = Tissue, value = n) -> summ.phy
```

```
## `summarise()` regrouping output by 'Tissue' (override with `.groups` argument)
```

```
summ.phy
```

```
## # A tibble: 12 x 4
##    phylum                    L     R     S
##    <chr>                 <int> <int> <int>
##  1 p__Ascomycota           105   205   232
##  2 p__Basidiobolomycota     NA     1     1
##  3 p__Basidiomycota        108   157   173
##  4 p__Blastocladiomycota    NA     2     2
##  5 p__Chytridiomycota        1    48    76
##  6 p__Entorrhizomycota      NA     1     1
##  7 p__Glomeromycota          1   250   245
##  8 p__Kickxellomycota       NA     1     1
##  9 p__Mortierellomycota     NA    24    26
## 10 p__Mucoromycota           1     8     9
## 11 p__Rozellomycota         NA    34    36
## 12 p__Zoopagomycota         NA     1     1
```

```
total.l <- sum(summ.phy$L, na.rm = T)
total.r <- sum(summ.phy$R, na.rm = T)
total.s <- sum(summ.phy$S, na.rm = T)

summ.phy %>%
  mutate(L.perc = L / total.l)
```

```
## # A tibble: 12 x 5
##    phylum                    L     R     S   L.perc
##    <chr>                 <int> <int> <int>    <dbl>
##  1 p__Ascomycota           105   205   232  0.486
##  2 p__Basidiobolymycota     NA     1     1 NA
##  3 p__Basidiomycota        108   157   173  0.5
##  4 p__Blastocladiomycota    NA     2     2 NA
##  5 p__Chytridiomycota        1    48    76  0.00463
##  6 p__Entorrhizomycota      NA     1     1 NA
##  7 p__Glomeromycota          1   250   245  0.00463
##  8 p__Kickxellomycota       NA     1     1 NA
##  9 p__Mortierellomycota     NA    24    26 NA
## 10 p__Mucoromycota           1     8     9  0.00463
## 11 p__Rozellomycota         NA    34    36 NA
## 12 p__Zoopagomycota         NA     1     1 NA
```

```
# most and least cosmopolitan ASVs
head(otu.l)
```

```
##   sample.name.match      ASV abund Tissue           phylum
## 1               L18 ASV_3509     0      L p__Rozellomycota
## 2              L105 ASV_3509     0      L p__Rozellomycota
## 3               L31 ASV_3509     0      L p__Rozellomycota
## 4               L12 ASV_3509     0      L p__Rozellomycota
## 5               L42 ASV_3509     0      L p__Rozellomycota
## 6                L6 ASV_3509     0      L p__Rozellomycota
```

```
otu.l %>%
  mutate(pres = abund > 0) %>%
  group_by(ASV, Tissue) %>%
  summarize(n.samps = sum(pres),
            n.total = length(pres),
            n.perc = n.samps/n.total) %>%
  arrange(-n.samps) -> df.n
```

```
## `summarise()` regrouping output by 'ASV' (override with `.groups` argument)
```

```
df.n %>%
  left_join(tax) -> df.n
```

```
## Joining, by = "ASV"
```

```
df.n %>%
  mutate(Tissue1 = ifelse(Tissue == "L", "Leaf",
                          ifelse(Tissue == "R", "Root", "Soil"))) %>%
  separate(phylum, into = c(NA, "phylum1"), remove=F) -> df.n

# p.cosmo <- ggplot(tmp, aes(x = Tissue1, y = n.perc*100, color = phylum1)) +
#   geom_point(position = "jitter", alpha = .8) +
#   ylab("ASV plant occupancy (%)") +
#   xlab("Plant-associated habitat") +
#   scale_color_discrete(name = "Phylum") +
#   theme_bw()
# p.cosmo

# ggsave(p.cosmo, filename = file.path(out_path, "cosmoASVs.png"),
#        dpi = 600, width = 6, height = 5)
```

1. Calculate alpha diversity

```
ps <- readRDS(file = file.path(merged_path, "phyloseq_samps_env_trimTreeASVs.RData"))
ps
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 932 taxa and 332 samples ]
## sample_data() Sample Data:       [ 332 samples by 75 sample variables ]
## tax_table()   Taxonomy Table:    [ 932 taxa by 9 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 932 tips and 254 internal nodes ]
## refseq()      DNAStringSet:      [ 932 reference sequences ]
```

```
asv <- otu_table(ps)
asv.df <- data.frame(asv, stringsAsFactors = F)
asv.mat <- as.matrix(asv.df)

library(picante)
```

```
## Loading required package: ape

## Loading required package: nlme

##
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
library(lme4)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

##
## Attaching package: 'lme4'
## The following object is masked from 'package:nlme':
##
##     lmList
```

```
library(lmerTest)
```

```
##
## Attaching package: 'lmerTest'

## The following object is masked from 'package:lme4':
##
##     lmer

## The following object is masked from 'package:stats':
##
##     step
```

```
library(emmeans)

# calculate Faith's PD
df.pd <- pd(asv.mat, phy_tree(ps), include.root = F)
df.pd$sample.name.match <- row.names(df.pd)
sam <- data.frame(sample_data(ps))
df.pd %>%
  left_join(sam) -> alpha
```

```
## Joining, by = "sample.name.match"
```

```
alpha %>%
  group_by(Tissue) %>%
  summarize(n = length(SR),
            SR.mean = mean(SR),
            SR.se = sd(SR)/sqrt(n),
            PD.mean = mean(PD),
            PD.se = sd(PD)/sqrt(n)) -> alpha.tab
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
alpha.tab
```

```
## # A tibble: 3 x 6
##   Tissue     n SR.mean SR.se PD.mean PD.se
##   <chr>  <int>   <dbl> <dbl>   <dbl> <dbl>
## 1 L        109    38.8 0.994  10247.  221.
## 2 R        111    35.1 1.35    9492.  305.
## 3 S        112    67.5 2.16   17924.  451.
```

```
sum(alpha.tab$n)
```

```
## [1] 332
```

```
# summarize by mono.mixed
colnames(alpha)
```

```
##  [1] "PD"                     "SR"                     "sample.name.match"
##  [4] "sample.type"            "SiteSamp"               "Site"
##  [7] "Tissue"                 "Site.name"              "Short.Site"
## [10] "sampling.day"           "sampling.month"         "sampling.year"
## [13] "Ecoregion"              "mono.mixed"             "stand.age.yrs"
## [16] "stand.age.yrs.num"   "stand.age.yrs.cat"   "num.cultivars"
## [19] "cultivar"            "other.veg"           "pasture.yn"
## [22] "harvest.mow.burn.yn" "fert.yn"             "mow.burn.notes"
## [25] "fert.notes"          "numberOfplots"       "plotarea.m2"
## [28] "plotarea.m2.se"      "plotarea.cat"        "lat"
## [31] "lon"                 "MAP.mm"              "MALT.C"
## [34] "MAT.C"               "MAHT.C"              "Site.address"
## [37] "County"              "Land.owner"          "Site.access.contact"
## [40] "Site.access.email"   "Site.access.phone"   "Samp"
## [43] "SOM"                 "W.V"                 "BS."
## [46] "Ac"                  "CEC"                 "ph"
## [49] "watercontent"        "P"                   "K"
## [52] "Na"                  "Ca"                  "Cu"
## [55] "Mg"                  "Mn"                  "S"
## [58] "Zn"                  "nh4"                 "no3"
## [61] "TIN"                 "perc.C"              "perc.N"
## [64] "mbc"                 "doc"                 "p.resin"
## [67] "perc.sand"           "perc.clay"           "perc.silt"
## [70] "usda.class"          "max.height.m"        "max.basalwidth.m"
## [73] "max.basallength.m"   "samp.lat"            "samp.lon"
## [76] "samp.plot"           "basal.area.m2"
```

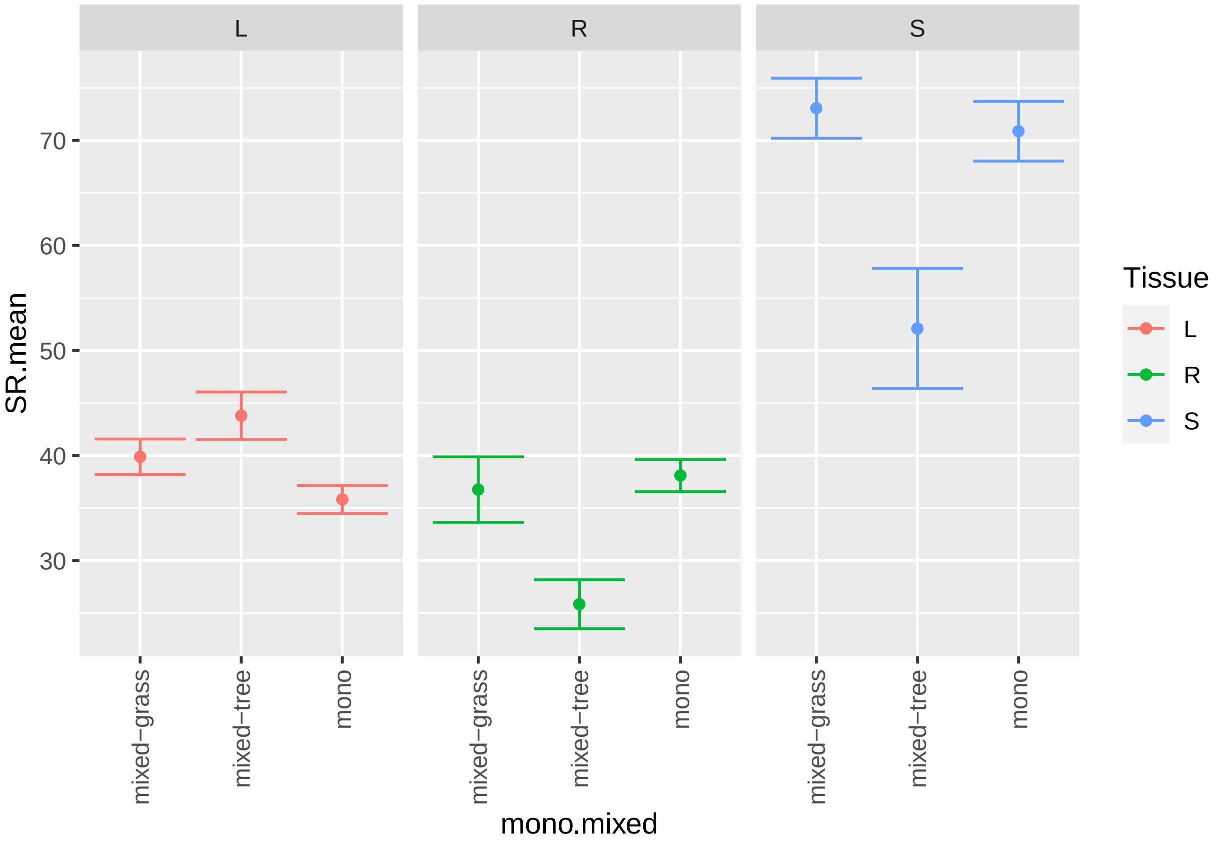
```
alpha %>%
  group_by(mono.mixed, Tissue) %>%
  summarize(n = length(SR),
            SR.mean = mean(SR),
            SR.se = sd(SR)/sqrt(n),
            PD.mean = mean(PD),
            PD.se = sd(PD)/sqrt(n)) -> alpha.tab.mono
```

```
## `summarise()` regrouping output by 'mono.mixed' (override with `.groups` argument)
```

```
alpha.tab.mono
```

```
## # A tibble: 9 x 7
## # Groups:   mono.mixed [3]
##   mono.mixed  Tissue     n SR.mean SR.se PD.mean PD.se
##   <chr>       <chr>  <int>   <dbl> <dbl>   <dbl> <dbl>
## 1 mixed-grass L         32    39.9  1.69  10457.  362.
## 2 mixed-grass R         32    36.8  3.12   9741.  683.
## 3 mixed-grass S         32    73.1  2.86  18896.  553.
## 4 mixed-tree  L         24    43.8  2.25  11645   459.
## 5 mixed-tree  R         24    25.8  2.33   7668.  581.
## 6 mixed-tree  S         24    52.1  5.71  14982. 1239.
## 7 mono        L         53    35.8  1.33   9487.  300.
## 8 mono        R         55    38.1  1.54  10144.  360.
## 9 mono        S         56    70.9  2.84  18629.  600.
```

```
p <- ggplot(alpha.tab.mono, aes(x = mono.mixed, y = SR.mean, color = Tissue)) +
  geom_point() +
  geom_errorbar(aes(ymin = SR.mean - SR.se, ymax = SR.mean + SR.se)) +
  facet_grid(~Tissue) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
p
```



```
# ggsave(p, filename = file.path(out_path, "SR_monomix.png"),
#        width = 5, height = 4, dpi = 300)


mod.sr <- lmer(SR ~ mono.mixed * Tissue + (1|Site), data = alpha)
an.sr <- anova(mod.sr)
an.sr
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                   Sum Sq Mean Sq NumDF   DenDF  F value    Pr(>F)
## mono.mixed           848   423.9     2  11.016   1.9043    0.1948
## Tissue             56049 28024.3     2 312.077 125.9083 < 2.2e-16 ***
## mono.mixed:Tissue   7011  1752.6     4 312.083   7.8743 4.665e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
tuk.sr <- emmeans(mod.sr, list(pairwise ~ Tissue * mono.mixed), adjust = "tukey")
tuk.sr
```

```
## $`emmeans of Tissue, mono.mixed`
##  Tissue mono.mixed  emmean   SE   df lower.CL upper.CL
##  L      mixed-grass   39.9 3.96 21.9     27.7     52.0
##  R      mixed-grass   36.8 3.96 21.9     24.6     48.9
##  S      mixed-grass   73.1 3.96 21.9     60.9     85.2
##  L      mixed-tree    43.8 4.58 21.9     29.8     57.8
##  R      mixed-tree    25.8 4.58 21.9     11.8     39.9
##  S      mixed-tree    52.1 4.58 21.9     38.0     66.1
##  L      mono          35.8 3.04 23.0     26.6     45.1
##  R      mono          38.1 3.01 22.2     28.9     47.3
##  S      mono          70.9 3.00 21.9     61.7     80.1
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## Conf-level adjustment: sidak method for 9 estimates
##
## $`pairwise differences of Tissue, mono.mixed`
##  contrast                          estimate   SE    df t.ratio p.value
##  (L mixed-grass) - (R mixed-grass)    3.125 3.73 312.0   0.838 0.9956
##  (L mixed-grass) - (S mixed-grass)  -33.188 3.73 312.0  -8.898 <.0001
##  (L mixed-grass) - (L mixed-tree)    -3.917 6.06  21.9  -0.647 0.9990
##  (L mixed-grass) - (R mixed-tree)    14.042 6.06  21.9   2.318 0.3734
##  (L mixed-grass) - (S mixed-tree)   -12.208 6.06  21.9  -2.016 0.5492
##  (L mixed-grass) - L mono             4.044 4.99  22.3   0.810 0.9953
##  (L mixed-grass) - R mono             1.749 4.98  22.0   0.351 1.0000
##  (L mixed-grass) - S mono           -31.000 4.97  21.9  -6.237 0.0001
##  (R mixed-grass) - (S mixed-grass)  -36.312 3.73 312.0  -9.736 <.0001
##  (R mixed-grass) - (L mixed-tree)    -7.042 6.06  21.9  -1.163 0.9563
##  (R mixed-grass) - (R mixed-tree)    10.917 6.06  21.9   1.802 0.6803
##  (R mixed-grass) - (S mixed-tree)   -15.333 6.06  21.9  -2.532 0.2704
##  (R mixed-grass) - L mono             0.919 4.99  22.3   0.184 1.0000
##  (R mixed-grass) - R mono            -1.376 4.98  22.0  -0.276 1.0000
##  (R mixed-grass) - S mono           -34.125 4.97  21.9  -6.866 <.0001
##  (S mixed-grass) - (L mixed-tree)    29.271 6.06  21.9   4.833 0.0021
##  (S mixed-grass) - (R mixed-tree)    47.229 6.06  21.9   7.798 <.0001
##  (S mixed-grass) - (S mixed-tree)    20.979 6.06  21.9   3.464 0.0465
##  (S mixed-grass) - L mono            37.232 4.99  22.3   7.455 <.0001
##  (S mixed-grass) - R mono            34.936 4.98  22.0   7.018 <.0001
##  (S mixed-grass) - S mono             2.188 4.97  21.9   0.440 0.9999
##  (L mixed-tree) - (R mixed-tree)     17.958 4.31 312.0   4.170 0.0013
##  (L mixed-tree) - (S mixed-tree)     -8.292 4.31 312.0  -1.925 0.5967
##  (L mixed-tree) - L mono              7.961 5.49  22.2   1.449 0.8664
##  (L mixed-tree) - R mono              5.665 5.48  22.0   1.034 0.9778
##  (L mixed-tree) - S mono            -27.083 5.47  21.9  -4.949 0.0016
##  (R mixed-tree) - (S mixed-tree)    -26.250 4.31 312.0  -6.095 <.0001
##  (R mixed-tree) - L mono             -9.997 5.49  22.2  -1.820 0.6698
##  (R mixed-tree) - R mono            -12.293 5.48  22.0  -2.244 0.4140
##  (R mixed-tree) - S mono            -45.042 5.47  21.9  -8.231 <.0001
##  (S mixed-tree) - L mono             16.253 5.49  22.2   2.958 0.1274
##  (S mixed-tree) - R mono             13.957 5.48  22.0   2.547 0.2635
##  (S mixed-tree) - S mono            -18.792 5.47  21.9  -3.434 0.0495
##  L mono - R mono                     -2.295 2.87 312.1  -0.799 0.9968
##  L mono - S mono                    -35.044 2.86 312.2 -12.249 <.0001
##  R mono - S mono                    -32.749 2.83 312.1 -11.561 <.0001
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 9 estimates
```

```
# summarize by cultivar
alpha %>%
  group_by(cultivar, Tissue) %>%
  summarize(n = length(SR),
            SR.mean = mean(SR),
            SR.se = sd(SR)/sqrt(n),
            PD.mean = mean(PD),
            PD.se = sd(PD)/sqrt(n)) -> alpha.tab.cult
```

```
## `summarise()` regrouping output by 'cultivar' (override with `.groups` argument)
```

```
alpha.tab.cult
```

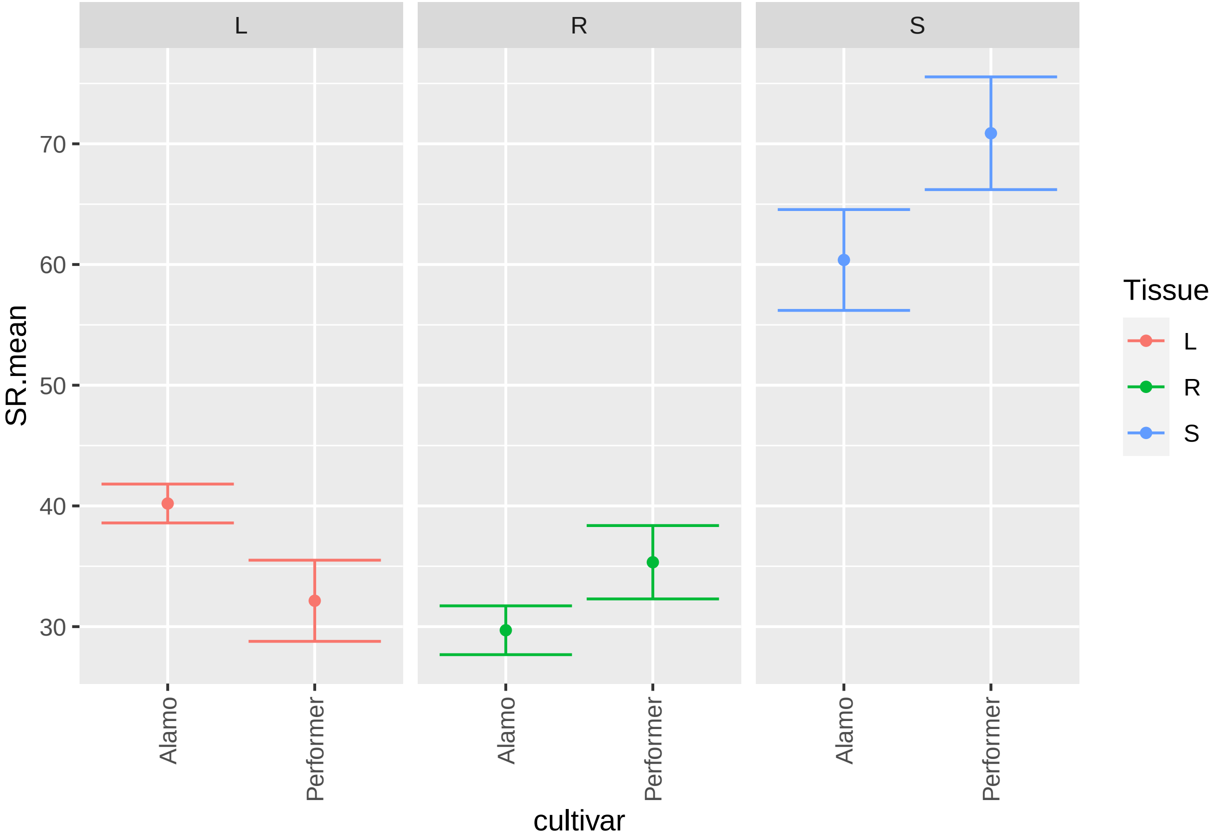
```
## # A tibble: 15 x 7
## # Groups:   cultivar [5]
##    cultivar      Tissue     n SR.mean SR.se PD.mean PD.se
##    <chr>         <chr>  <int>   <dbl> <dbl>   <dbl> <dbl>
##  1 Alamo         L         40    40.2  1.61  10780.  371.
##  2 Alamo         R         40    29.7  2.02   8834.  500.
##  3 Alamo         S         40    60.4  4.18  16472.  881.
##  4 mixed         L          8    46.5  2.61  11602.  542.
##  5 mixed         R          8    34.9  3.36   8940   767.
##  6 mixed         S          8    79.5  7.79  21030  1597.
##  7 mixed unknown L          7    38.7  3.82   9883.  748.
##  8 mixed unknown R          8    33.5  2.24   9195   645.
##  9 mixed unknown S          8    62.5  6.34  16418. 1276.
## 10 Performer     L         14    32.1  3.36   8717.  798.
## 11 Performer     R         15    35.3  3.04   9448   683.
## 12 Performer     S         16    70.9  4.67  19492. 1034.
## 13 unknown       L         40    38.1  1.47  10041   301.
## 14 unknown       R         40    40.6  2.62  10338   592.
## 15 unknown       S         40    71.8  3.18  18429   619.
```

```
alpha %>%
  select(Site, cultivar) %>%
  unique()
```

```
##           Site      cultivar
## 1  CGF-MON-PRO         Alamo
## 2  UCP-MXG-NCD         Alamo
## 3  CGF-MXG-PRO       unknown
## 4  CCR-ONE-NCD     Performer
## 5  CRE-MXT-NCD         mixed
## 6  BRF-ONE-COM       unknown
## 9  LWR-BHO-NCS       unknown
## 11 WBI-NRT-NCS     Performer
## 12 SFA-ONE-PRO mixed unknown
## 15 MHC-ONE-NCD       unknown
## 20 CRE-MXG-NCD       unknown
## 21 LCO-MXT-COM         Alamo
## 25 OTO-MXT-NCD         Alamo
## 32 OTO-MON-NCD         Alamo
```

```
alpha.tab.cult %>%
  filter(cultivar %in% c("Alamo","Performer")) -> tmp

p <- ggplot(tmp, aes(x = cultivar, y = SR.mean, color = Tissue)) +
  geom_point() +
  geom_errorbar(aes(ymin = SR.mean - SR.se, ymax = SR.mean + SR.se)) +
  facet_grid(~Tissue) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

```
p
```



```
# ggsave(p, filename = file.path(out_path, "SR_cult.png"),
#        width = 5, height = 4, dpi = 300)

alpha %>%
  filter(cultivar %in% c("Alamo","Performer")) -> tmp
mod.sr <- lmer(SR ~ cultivar * Tissue + (1|Site), data = tmp)
an.sr <- anova(mod.sr)
an.sr
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                  Sum Sq Mean Sq NumDF   DenDF F value  Pr(>F)
## cultivar           32.7    32.7     1   5.052  0.1391 0.72433
## Tissue          29093.2 14546.6     2 154.054 61.8022 < 2e-16 ***
## cultivar:Tissue  1875.7   937.9     2 154.054  3.9846 0.02055 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
tuk.sr <- emmeans(mod.sr, list(pairwise ~ Tissue * cultivar), adjust = "tukey")
tuk.sr
```

```
## $`emmeans of Tissue, cultivar`
##  Tissue cultivar  emmean   SE   df lower.CL upper.CL
##  L      Alamo       40.2 4.66 7.41    23.65     56.8
##  R      Alamo       29.7 4.66 7.41    13.15     46.3
##  S      Alamo       60.4 4.66 7.41    43.82     76.9
##  L      Performer   32.7 7.52 8.01     6.63     58.7
##  R      Performer   35.6 7.44 7.68     9.48     61.7
##  S      Performer   70.9 7.37 7.41    44.70     97.0
## 
## Degrees-of-freedom method: kenward-roger 
## Confidence level used: 0.95 
## Conf-level adjustment: sidak method for 6 estimates 
## 
## $`pairwise differences of Tissue, cultivar`
##  contrast                  estimate   SE     df t.ratio p.value
##  L Alamo - R Alamo            10.50 3.43 154.00  3.061  0.0307 
##  L Alamo - S Alamo           -20.18 3.43 154.00 -5.881  <.0001 
##  L Alamo - L Performer         7.52 8.85   7.83  0.850  0.9485 
##  L Alamo - R Performer         4.62 8.78   7.60  0.526  0.9933 
##  L Alamo - S Performer       -30.68 8.72   7.41 -3.516  0.0654 
##  R Alamo - S Alamo           -30.68 3.43 154.00 -8.942  <.0001 
##  R Alamo - L Performer        -2.98 8.85   7.83 -0.337  0.9992 
##  R Alamo - R Performer        -5.88 8.78   7.60 -0.670  0.9804 
##  R Alamo - S Performer       -41.17 8.72   7.41 -4.720  0.0151 
##  S Alamo - L Performer        27.70 8.85   7.83  3.130  0.1021 
##  S Alamo - R Performer        24.79 8.78   7.60  2.823  0.1553 
##  S Alamo - S Performer       -10.50 8.72   7.41 -1.204  0.8236 
##  L Performer - R Performer    -2.90 5.70 154.03 -0.509  0.9958 
##  L Performer - S Performer   -38.20 5.62 154.12 -6.792  <.0001 
##  R Performer - S Performer   -35.29 5.52 154.03 -6.398  <.0001 
## 
## Degrees-of-freedom method: kenward-roger 
## P value adjustment: tukey method for comparing a family of 6 estimates
```

```
mod.sr <- lmer(SR ~ Tissue + (1|Site), data = alpha)
sum(resid(mod.sr)^2)
```

```
## [1] 77086.09
```

```
an.sr <- anova(mod.sr)
an.sr
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##        Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)    
## Tissue  69990   34995     2 316.12  144.64 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
su.sr <- summary(mod.sr)
su.sr
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: SR ~ Tissue + (1 | Site)
##    Data: alpha
## 
## REML criterion at convergence: 2774.4
## 
## Scaled residuals: 
##     Min      1Q  Median      3Q     Max 
## -2.3557 -0.6134 -0.0630  0.5716  4.3410 
## 
## Random effects:
```

```
##  Groups   Name        Variance Std.Dev.
##  Site     (Intercept)  40.54    6.367
##  Residual             241.94   15.554
## Number of obs: 332, groups:  Site, 14
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   38.802      2.262  25.817  17.150 1.25e-15 ***
## TissueR       -3.720      2.098 316.097  -1.773   0.0772 .
## TissueS       28.672      2.094 316.174  13.695  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##         (Intr) TissuR
## TissueR -0.468
## TissueS -0.469  0.506
```

```
tuk.sr <- emmeans(mod.sr, list(pairwise ~ Tissue), adjust = "tukey")
tuk.sr
```

```
## $`emmeans of Tissue`
##  Tissue emmean   SE   df lower.CL upper.CL
##  L        38.8 2.26 25.7     33.0     44.6
##  R        35.1 2.25 25.3     29.3     40.8
##  S        67.5 2.25 25.1     61.7     73.2
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## Conf-level adjustment: sidak method for 3 estimates
##
## $`pairwise differences of Tissue`
##  contrast estimate   SE  df t.ratio p.value
##  L - R        3.72 2.10 316   1.773 0.1803
##  L - S      -28.67 2.09 316 -13.695 <.0001
##  R - S      -32.39 2.08 316 -15.547 <.0001
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 3 estimates
```

```
# capture.output(an.sr, file = file.path(out_path, "alphaDiv.txt"))
# capture.output(su.sr, file = file.path(out_path, "alphaDiv.txt"), append = T)
# capture.output(tuk.sr, file = file.path(out_path, "alphaDiv.txt"), append = T)

mod.pd <- lmer(PD ~ Tissue + (1|Site), data = alpha)
mod.pd
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: PD ~ Tissue + (1 | Site)
##    Data: alpha
## REML criterion at convergence: 6308.901
## Random effects:
##  Groups   Name        Std.Dev.
##  Site     (Intercept) 1328
##  Residual             3351
## Number of obs: 332, groups:  Site, 14
## Fixed Effects:
## (Intercept)      TissueR      TissueS
##     10261.7       -759.3       7662.2
```

```
sum(resid(mod.pd)^2)
```

```
## [1] 3578650234
```

```
an.pd <- anova(mod.pd)
an.pd
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##            Sum Sq    Mean Sq NumDF  DenDF F value    Pr(>F)
## Tissue 4833458797 2416729399     2 316.08  215.26 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
su.pd <- summary(mod.pd)
su.pd
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: PD ~ Tissue + (1 | Site)
##    Data: alpha
##
## REML criterion at convergence: 6308.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6440 -0.6286 -0.0411  0.5718  4.5914
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Site     (Intercept)  1764623 1328
##  Residual             11227058 3351
## Number of obs: 332, groups:  Site, 14
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept) 10261.72     478.74    26.52   21.43   <2e-16 ***
## TissueR      -759.32     451.89   316.07   -1.68   0.0939 .
## TissueS      7662.21     450.99   316.15   16.99   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##         (Intr) TissuR
## TissueR -0.477
## TissueS -0.478  0.506
```

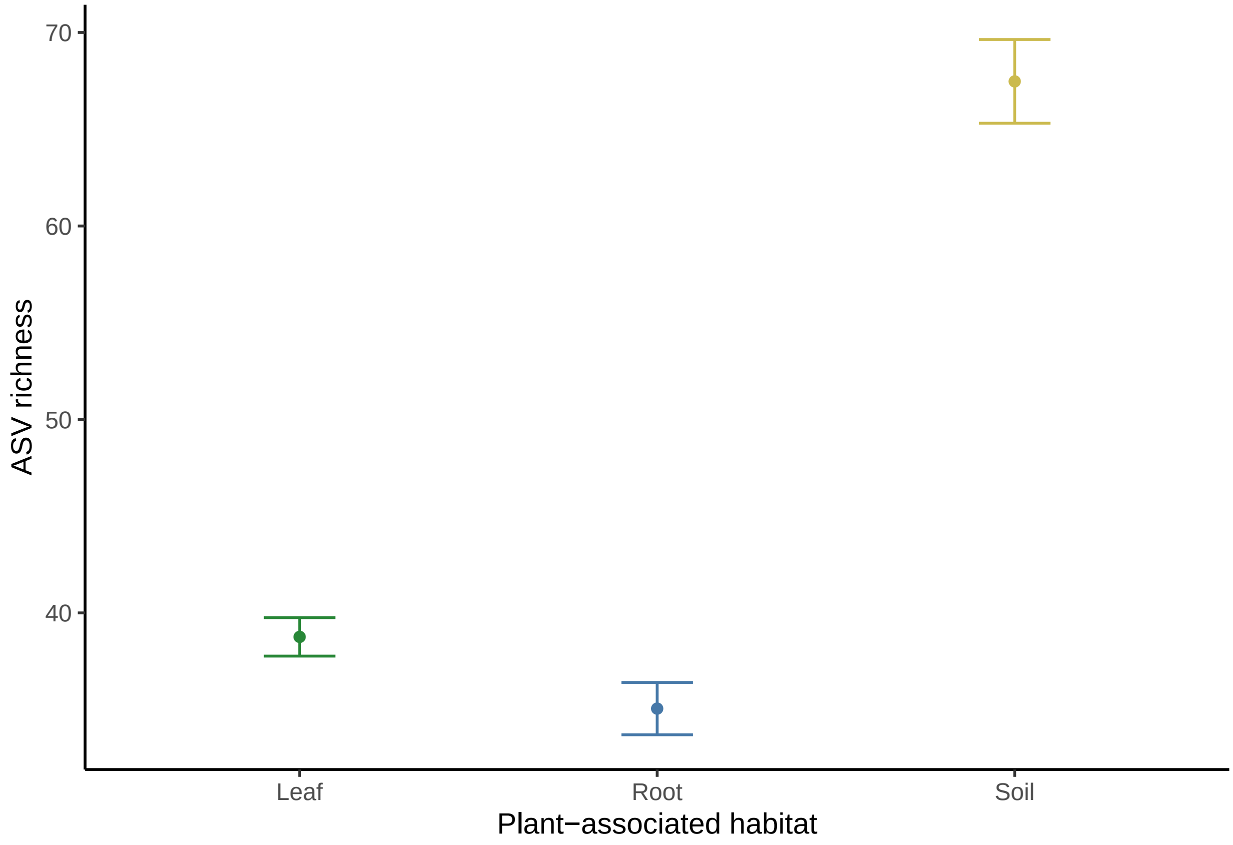
```
tuk.pd <- emmeans(mod.pd, list(pairwise ~ Tissue), adjust = "tukey")
tuk.pd
```

```
## $`emmeans of Tissue`
##  Tissue emmean  SE   df lower.CL upper.CL
##  L       10262 479 26.5     9042    11482
##  R        9502 477 26.1     8286    10718
##  S       17924 476 25.9    16710    19138
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## Conf-level adjustment: sidak method for 3 estimates
##
## $`pairwise differences of Tissue`
##  contrast estimate  SE  df t.ratio p.value
##  L - R         759 452 316   1.680 0.2143
##  L - S       -7662 451 316 -16.989 <.0001
##  R - S       -8422 449 316 -18.765 <.0001
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 3 estimates
```

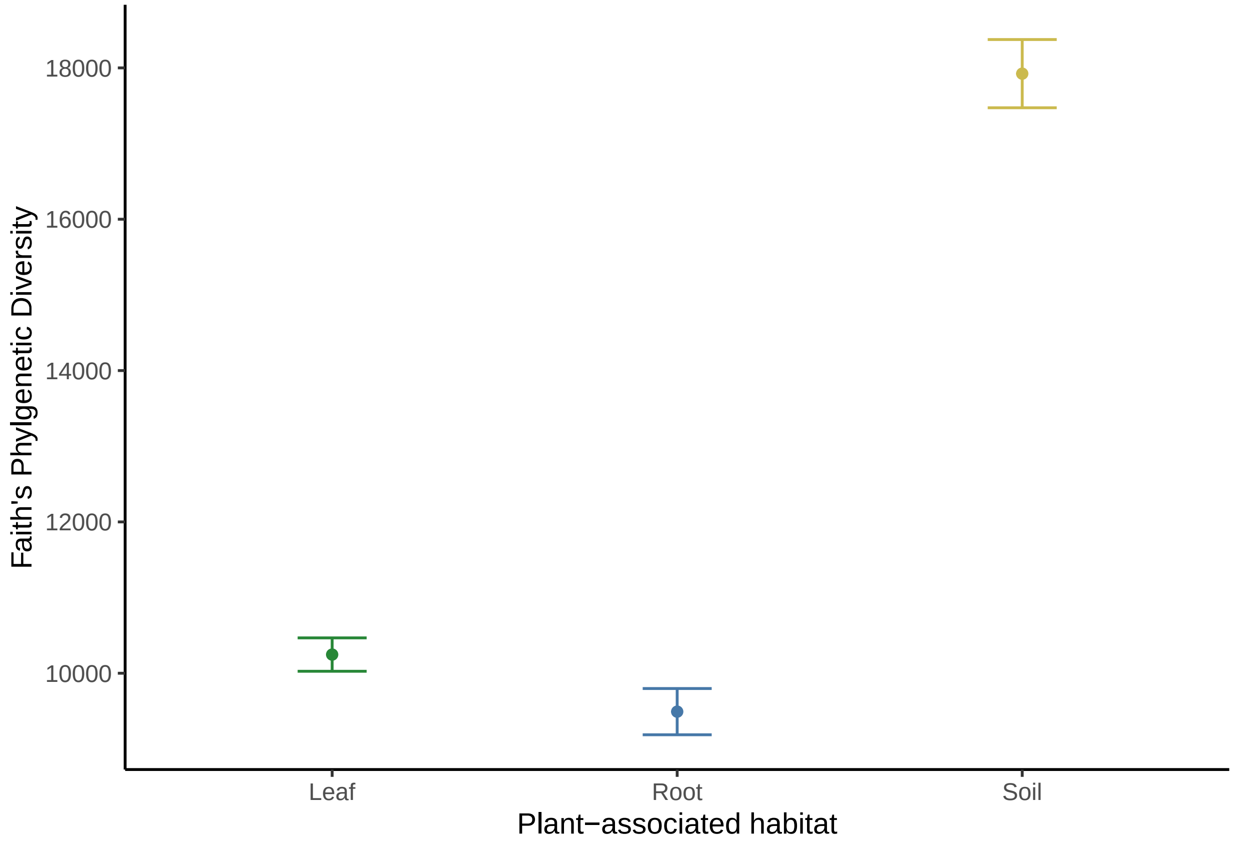
```
# capture.output(an.pd, file = file.path(out_path, "alphaDiv.txt"), append = T)
# capture.output(su.pd, file = file.path(out_path, "alphaDiv.txt"), append = T)
# capture.output(tuk.pd, file = file.path(out_path, "alphaDiv.txt"), append = T)

alpha.tab
```

```
## # A tibble: 3 x 6
##   Tissue     n SR.mean SR.se PD.mean PD.se
##   <chr>  <int>   <dbl> <dbl>   <dbl> <dbl>
## 1 L        109    38.8 0.994  10247.  221.
## 2 R        111    35.1 1.35    9492.  305.
## 3 S        112    67.5 2.16   17924.  451.
```

```
p1 <- ggplot(alpha.tab, aes(x = Tissue, y = SR.mean, color = Tissue)) +
  geom_point() +
  geom_errorbar(aes(ymin = SR.mean - SR.se,
                    ymax = SR.mean + SR.se), width = .2) +
  ylab("ASV richness") +
  xlab("Plant-associated habitat") +
  theme_classic() +
  scale_x_discrete(labels = c("Leaf","Root","Soil")) +
  scale_color_manual(values= tissue.colors) +
  guides(color = F)
p1
```



```
p2 <- ggplot(alpha.tab, aes(x = Tissue, y = PD.mean, color = Tissue)) +
  geom_point() +
  geom_errorbar(aes(ymin = PD.mean - PD.se,
                    ymax = PD.mean + PD.se), width = .2) +
  ylab("Faith's Phylgenetic Diversity") +
  xlab("Plant-associated habitat") +
  theme_classic()+
  scale_x_discrete(labels = c("Leaf","Root","Soil")) +
  scale_color_manual(values= tissue.colors) +
  guides(color = F)
p2
```



```
library(gridExtra)
# ggsave(
#   file.path(out_path, "alphaDiv.png"),
#   grid.arrange(p1 + ggtitle("a"),
#                p2 + ggtitle("b"), ncol = 1),
#   width = 4,
#   height = 6,
#   dpi = 600
# )
```

2. Venn diagram of ASVs shared/unique to leaf, root, soil [commented out]

```
# ps <- readRDS(file = file.path(merged_path, "phyloseq_samps_env_trimTreeASVs.RData"))
#
# ps.l <- subset_samples(ps, Tissue == "L")
# l.asvs <- names(colSums(otu_table(ps.l))[colSums(otu_table(ps.l)) != 0])
#
# ps.r <- subset_samples(ps, Tissue == "R")
# r.asvs <- names(colSums(otu_table(ps.r))[colSums(otu_table(ps.r)) != 0])
#
# ps.s <- subset_samples(ps, Tissue == "S")
# s.asvs <- names(colSums(otu_table(ps.s))[colSums(otu_table(ps.s)) != 0])

# ps.l <- readRDS(file = file.path(merged_path, "phyloseq_samps_env_trimTreeASVs_leaf.RData"))
# ps.r <- readRDS(file = file.path(merged_path, "phyloseq_samps_env_trimTreeASVs_root.RData"))
# ps.s <- readRDS(file = file.path(merged_path, "phyloseq_samps_env_trimTreeASVs_soil.RData"))
```

```
#
# ps.l


# library(ggVennDiagram)
# x <- list(Leaf=taxa_names(ps.l),
#           Root=taxa_names(ps.r),
#           Soil=taxa_names(ps.s))
# p <- ggVennDiagram(x)
# p
# ggsave(p, filename = file.path(out_path, "ASVs_Tissue_venn.png"),
#        width = 4, height = 4, dpi = 300)
```

D. Is there a critical distance where communities diverge/converge?

1. Generate pairwise dataframes, save, and plot
2. Breakpoint regression with spatial distance
3. Breakpoint regression with environmental distance

1. Generate pairwise dataframes and plot

```
sameSite.colors <- c("gray","black")
names(sameSite.colors) <- c(FALSE, TRUE)

#load the data
require(phyloseq)

ps <- readRDS(file = file.path(merged_path, "phyloseq_samps_env_trimTreeASVs.RData"))
ps
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 932 taxa and 332 samples ]
## sample_data() Sample Data:       [ 332 samples by 75 sample variables ]
## tax_table()   Taxonomy Table:    [ 932 taxa by 9 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 932 tips and 254 internal nodes ]
## refseq()      DNAStringSet:      [ 932 reference sequences ]
```

```
vst <- readRDS(file = file.path("output/illumina/Q0", "vst_all.RData"))
raodis <- readRDS("output/illumina/Q0/dpcoa_all_raodist.RData")

# add similarity distances
library(vegan)
bray.dist <- distance(ps, method="bray") # careful! this doesn't work if DESeq2 is loaded
#library(picante)
# tree <- phy_tree(ps)
# tree$root.edge <- 0
# asv <- data.frame(otu_table(ps), stringsAsFactors = F)
# phy.sor<- phylosor(samp = asv, tree = tree)
# saveRDS(phy.sor, file = file.path(out_path, "physor_dist.RData"))
physor.dist <- readRDS(file = file.path(out_path, "physor_dist.RData"))
```

```
# add environmental distances
sam <- data.frame(sample_data(ps))
# load transformed environmental variables (prior to lasso filter)
mat.t <- read.csv(file = "output/illumina/Q2/normTransformed_contvars_trim.csv",
                  row.names = 1)
sam %>%
    dplyr::select(sample.name.match, SiteSamp, Site, Tissue) -> samp.tmp
samp.tmp %>%
  left_join(mat.t) -> samp.tmp
```

```
## Joining, by = c("SiteSamp", "Site")
```

```
samp.tmp %>%
  dplyr::select(-c(sample.name.match, SiteSamp, Site, Tissue)) -> samp.env
row.names(samp.env)<- samp.tmp$sample.name.match
dist.env <- dist(samp.env, method = "euclidean")
mat.env <- as.matrix(dist.env)

env.dist.df <- extract_uniquePairDists(mat.env)
env.dist.df %>%
    dplyr::rename('env.dist.m'='dist') -> env.dist.df

# make dataframe
#dist.df <- make_dist_df(ps, vst, raodis, bray.dist, physor.dist, env.dist.df)
#saveRDS(dist.df, file = file.path(out_path, "dist_df.RData"))
dist.df <- readRDS(file = file.path(out_path, "dist_df.RData"))
#head(dist.df)
range(dist.df$hav.dist.km)
```

```
## [1] 1.190274e-03 4.624840e+02
```

```
# # what makes leaf communities vary different at small scales?
# colnames(dist.df)
# dist.df %>%
#   filter(Tissue_samp1 == "L") %>%
#   filter(hav.dist.km < 0.38) -> tmp

# check out clustered distances
# # these are because of sites that are very close: CRE-MXG to CRE-MXT, OTO-MON to OTO-MXT
# dist.df %>%
#   mutate(hav.dist.km = hav.dist.m / 1000) %>%
#   filter(hav.dist.km < 0.33) %>%
#   filter(hav.dist.km > 0.300) -> sub
# sub %>%
#   group_by(sameSite) %>%
#   summarize(n = length(samp1)) # all are comparisons between sites (not within)
# sub %>%
#   group_by(Site_samp1, Site_samp2) %>%
#   summarize(n = length(samp1))
# 16*16 # maximum number of pairs between 2 sites
# 111+81
# sub %>%
#   mutate(site.pair = paste0(Site_samp1, Site_samp2)) -> sub
# ggplot(sub, aes(x = hav.dist.km, y = vst.comm.dist, color = site.pair)) +
#   geom_point() +
#   scale_color_manual(values = c(1,1,2,2))
```

Add sitePairs

```
#dist.df

# color by site comparisons
all.sites <- unique(c(dist.df$Site_samp1,dist.df$Site_samp2))
all.site.pairs <- data.frame(t(combn(all.sites, 2)))
all.site.pairs$pairs <- paste0(all.site.pairs$X1, "__", all.site.pairs$X2)
pairs <- all.site.pairs$pairs

dist.df %>%
  mutate(site.pairs = paste0(Site_samp1,"__", Site_samp2)) %>%
  mutate(site.pairs.rev = paste0(Site_samp2, "__", Site_samp1)) %>%
  mutate(site.pairs = ifelse(site.pairs %in% pairs, site.pairs, site.pairs.rev)) %>%
  mutate(site.pairs = ifelse(sameSite == TRUE, Site_samp1, site.pairs)) %>%
  dplyr::select(-site.pairs.rev) -> dist.df
```

2. Breakpoint regression with spatial distance

Fit segmented regression models – Bray

```
#library("segmented")
#str(dist.df)
dist.df$Tissue <- factor(dist.df$Tissue_samp1)
dist.df$bray.sim <- 1- dist.df$bray.comm.dist

# build the dummy variables for the Tissue x distance interaction
require(segmented)
```

```
## Loading required package: segmented
```

```
X <- model.matrix(~ 0 + dist.df$Tissue) * dist.df$hav.dist.km
max(which(dist.df$Tissue == "L"))
```

```
## [1] 5886
```

```
min(which(dist.df$Tissue == "R"))
```

```
## [1] 5887
```

```
hav.L <- X[,1]
hav.R <- X[,2]
hav.S <- X[,3]
mod <- lm(bray.sim ~ 0 + Tissue + hav.L + hav.R + hav.S,
          data = dist.df)
mod.seg <- segmented(mod, seg.Z = ~hav.L + hav.R + hav.S,
                           psi = list(hav.L = 1,
                                      hav.R = 1,
                                      hav.S = 1))
summary(mod.seg)
```

```
##
##  ***Regression Model with Segmented Relationship(s)***
##
## Call:
```

```
## segmented.lm(obj = mod, seg.Z = ~hav.L + hav.R + hav.S, psi = list(hav.L = 1,
##     hav.R = 1, hav.S = 1))
##
## Estimated Break-Point(s):
##              Est. St.Err
## psi1.hav.L 0.251  0.014
## psi1.hav.R 0.363  0.033
## psi1.hav.S 0.291  0.025
##
## Meaningful coefficients of the linear terms:
##           Estimate Std. Error t value Pr(>|t|)
## TissueL   0.483971   0.006822  70.941   <2e-16 ***
## TissueR   0.192803   0.005945  32.432   <2e-16 ***
## TissueS   0.249726   0.006588  37.905   <2e-16 ***
## hav.L    -1.124714   0.076520 -14.698   <2e-16 ***
## hav.R    -0.356767   0.040319  -8.849   <2e-16 ***
## hav.S    -0.672543   0.071527  -9.403   <2e-16 ***
## U1.hav.L  1.124683   0.076520  14.698       NA
## U1.hav.R  0.356666   0.040319   8.846       NA
## U1.hav.S  0.672525   0.071527   9.402       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09744 on 18195 degrees of freedom
## Multiple R-Squared: 0.6665,  Adjusted R-squared: 0.6662
##
## Convergence attained in 2 iter. (rel. change 5.151e-06)
```

```
coef(mod.seg)
```

```
##    TissueL    TissueR    TissueS      hav.L      hav.R      hav.S   U1.hav.L
##  0.4839709  0.1928027  0.2497258 -1.1247144 -0.3567670 -0.6725429  1.1246834
##   U1.hav.R   U1.hav.S psi1.hav.L psi1.hav.R psi1.hav.S
##  0.3566664  0.6725246  0.0000000  0.0000000  0.0000000
```

```
capture.output(summary(mod.seg), file = file.path(out_path,"segBray.txt"))
tmp <- summary(mod.seg)
tmp$psi[1,2]
```

```
## [1] 0.2506083
```

```
#U1 = difference-in-slope parameter of the variable hav.L
# to test the significance of difference in slopes for each Tissue...
dt.l <- davies.test(mod, seg.Z = ~hav.L, k = 10, values = tmp$psi[1,2])
dt.r <- davies.test(mod, seg.Z = ~hav.R, k = 10, values = tmp$psi[2,2])
dt.s <- davies.test(mod, seg.Z = ~hav.S, k = 10, values = tmp$psi[3,2])
capture.output(dt.l, file = file.path(out_path,"segBray.txt"), append = T)
capture.output(dt.r, file = file.path(out_path,"segBray.txt"), append = T)
capture.output(dt.s, file = file.path(out_path,"segBray.txt"), append = T)
# yes, all signif different

# save the CIs for breakpoints
brks <- rbind(confint.segmented(mod.seg, "hav.L"),
      confint.segmented(mod.seg, "hav.R"),
      confint.segmented(mod.seg, "hav.S"))
brks <- data.frame(brks, stringsAsFactors = F)
```

```
brks
```

```
##                  Est. CI.95...low CI.95...up
## psi1.hav.L 0.250608    0.224059   0.277158
## psi1.hav.R 0.362648    0.298786   0.426510
## psi1.hav.S 0.290763    0.241495   0.340031
```

```
brks$Tissue <- c("L","R","S")
brks
```

```
##                  Est. CI.95...low CI.95...up Tissue
## psi1.hav.L 0.250608    0.224059   0.277158      L
## psi1.hav.R 0.362648    0.298786   0.426510      R
## psi1.hav.S 0.290763    0.241495   0.340031      S
```

```
capture.output(brks, file = file.path(out_path,"segBray.txt"), append = T)
# save the CIs for slopes
slopes <- list_to_df(slope(mod.seg))
capture.output(slopes, file = file.path(out_path,"segBray.txt"), append = T)

# break the regression
#library(lsmeans)
break.here <- mean(brks[,"Est."])
break.here
```

```
## [1] 0.3013397
```

```
dist.df %>%
  filter(hav.dist.km < break.here) -> dist.dfa
dist.df %>%
  filter(hav.dist.km > break.here) -> dist.dfb

# posthoc t-test to test difference in means - lower
moda <- lm(bray.sim ~ Tissue * hav.dist.km, data = dist.dfa)
summary(moda)
```

```
##
## Call:
## lm(formula = bray.sim ~ Tissue * hav.dist.km, data = dist.dfa)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34402 -0.10390 -0.02235  0.08073  0.69851
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.48206    0.01009  47.798  < 2e-16 ***
## TissueR               -0.28101    0.01415 -19.854  < 2e-16 ***
## TissueS               -0.23233    0.01412 -16.449  < 2e-16 ***
## hav.dist.km           -1.08249    0.10802 -10.021  < 2e-16 ***
## TissueR:hav.dist.km    0.55505    0.15242   3.642 0.000282 ***
## TissueS:hav.dist.km    0.40995    0.15230   2.692 0.007199 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1463 on 1305 degrees of freedom
## Multiple R-squared:  0.3956, Adjusted R-squared:  0.3933
## F-statistic: 170.9 on 5 and 1305 DF,  p-value: < 2.2e-16
```

```
capture.output(summary(moda),
               file = file.path(out_path,"segBray.txt"), append = T)

library(emmeans)
moda.lst <- lstrends(moda, ~ Tissue, var = "hav.dist.km")
pairs(moda.lst)   # comparisons of slopes
```

```
##  contrast estimate    SE   df t.ratio p.value
##  L - R      -0.555 0.152 1305 -3.642  0.0008
##  L - S      -0.410 0.152 1305 -2.692  0.0197
##  R - S       0.145 0.152 1305  0.955  0.6056
##
## P value adjustment: tukey method for comparing a family of 3 estimates
```

```
TukeyHSD(aov(moda), which = "Tissue") # comparison of intercepts
```

```
## Warning in replications(paste("~", xx), data = mf): non-factors ignored:
## hav.dist.km

## Warning in replications(paste("~", xx), data = mf): non-factors ignored: Tissue,
## hav.dist.km

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = moda)
##
## $Tissue
##            diff         lwr         upr    p adj
## R-L -0.24373963 -0.26707792 -0.22040135 0.000000
## S-L -0.20463267 -0.22788058 -0.18138475 0.000000
## S-R  0.03910697  0.01603578  0.06217815 0.000217
```

```
capture.output(summary(moda), file = file.path(out_path,"segBray.txt"), append = T)
capture.output(pairs(moda.lst), file = file.path(out_path,"segBray.txt"), append = T)
capture.output(TukeyHSD(aov(moda), which = "Tissue"),
               file = file.path(out_path,"segBray.txt"), append = T)
```

```
## Warning in replications(paste("~", xx), data = mf): non-factors ignored:
## hav.dist.km

## Warning in replications(paste("~", xx), data = mf): non-factors ignored: Tissue,
## hav.dist.km
```

```
# posthoc t-test to test difference in means - upper
modb <- lm(bray.sim ~ Tissue * hav.dist.km, data = dist.dfb)
summary(modb)
```

```
##
## Call:
## lm(formula = bray.sim ~ Tissue * hav.dist.km, data = dist.dfb)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19953 -0.04479 -0.02117  0.02671  0.75036
```

```
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           2.020e-01  2.263e-03  89.233  < 2e-16 ***
## TissueR              -1.371e-01  3.181e-03 -43.094  < 2e-16 ***
## TissueS              -1.478e-01  3.168e-03 -46.641  < 2e-16 ***
## hav.dist.km          -3.037e-05  1.069e-05  -2.840  0.00452 **
## TissueR:hav.dist.km -7.586e-05  1.498e-05  -5.063 4.16e-07 ***
## TissueS:hav.dist.km  1.208e-05  1.493e-05   0.809  0.41844
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09261 on 16890 degrees of freedom
## Multiple R-squared:  0.3618, Adjusted R-squared:  0.3616
## F-statistic:  1915 on 5 and 16890 DF,  p-value: < 2.2e-16
```

```
capture.output(summary(modb),
               file = file.path(out_path,"segBray.txt"), append = T)

modb.lst <- lstrends(modb, ~ Tissue, var = "hav.dist.km")
pairs(modb.lst)   # comparisons of slopes
```

```
##  contrast  estimate       SE    df t.ratio p.value
##  L - R     7.59e-05 1.50e-05 16890   5.063  <.0001
##  L - S    -1.21e-05 1.49e-05 16890  -0.809  0.6974
##  R - S    -8.79e-05 1.48e-05 16890  -5.947  <.0001
##
## P value adjustment: tukey method for comparing a family of 3 estimates
```

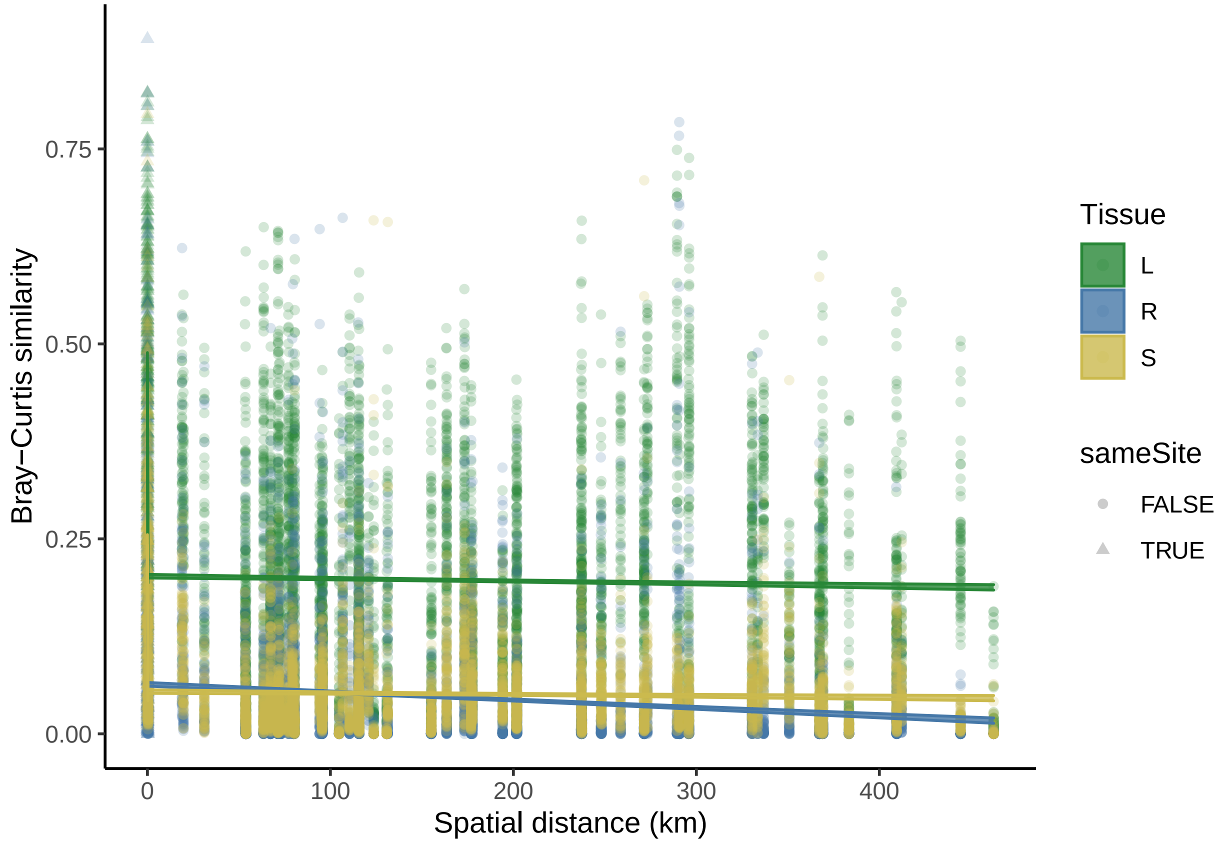
```
TukeyHSD(aov(modb), which = "Tissue") # comparison of intercepts
```

```
## Warning in replications(paste("~", xx), data = mf): non-factors ignored:
## hav.dist.km

## Warning in replications(paste("~", xx), data = mf): non-factors ignored: Tissue,
## hav.dist.km

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = modb)
##
## $Tissue
##             diff           lwr          upr    p adj
## R-L -0.150596660 -0.1547131023 -0.146480218 0.000000
## S-L -0.145675203 -0.1497733989 -0.141577006 0.000000
## S-R  0.004921457  0.0008617196  0.008981195 0.012499
```

```
capture.output(summary(modb), file = file.path(out_path,"segBray.txt"), append = T)
capture.output(pairs(modb.lst), file = file.path(out_path,"segBray.txt"), append = T)
capture.output(TukeyHSD(aov(modb), which = "Tissue"),
               file = file.path(out_path,"segBray.txt"), append = T)
```

```
## Warning in replications(paste("~", xx), data = mf): non-factors ignored:
## hav.dist.km

## Warning in replications(paste("~", xx), data = mf): non-factors ignored: Tissue,
## hav.dist.km
```

```
# use predict to show the fitted model
pred <- predict(mod.seg, se.fit = TRUE)
dist.df$pred <- pred$fit
dist.df$pred.se <- pred$se.fit

p <- ggplot(dist.df, aes(x = hav.dist.km, y = pred,
                    fill = Tissue, color = Tissue, shape = sameSite)) +
  geom_point(aes(y = bray.sim), alpha = .2) +
  geom_ribbon(aes(ymin = pred - pred.se, ymax = pred + pred.se),
              alpha = .8) +
  #geom_line()+
  theme_classic() +
  ylab("Bray-Curtis similarity") +
  xlab("Spatial distance (km)") +
  scale_color_manual(values = tissue.colors) +
  scale_fill_manual(values = tissue.colors)
p
```



```
# add error around breaks
brks <- rbind(confint.segmented(mod.seg, "hav.L"),
      confint.segmented(mod.seg, "hav.R"),
      confint.segmented(mod.seg, "hav.S"))
brks <- data.frame(brks, stringsAsFactors = F)
brks$Tissue <- c("L","R","S")
brks
```

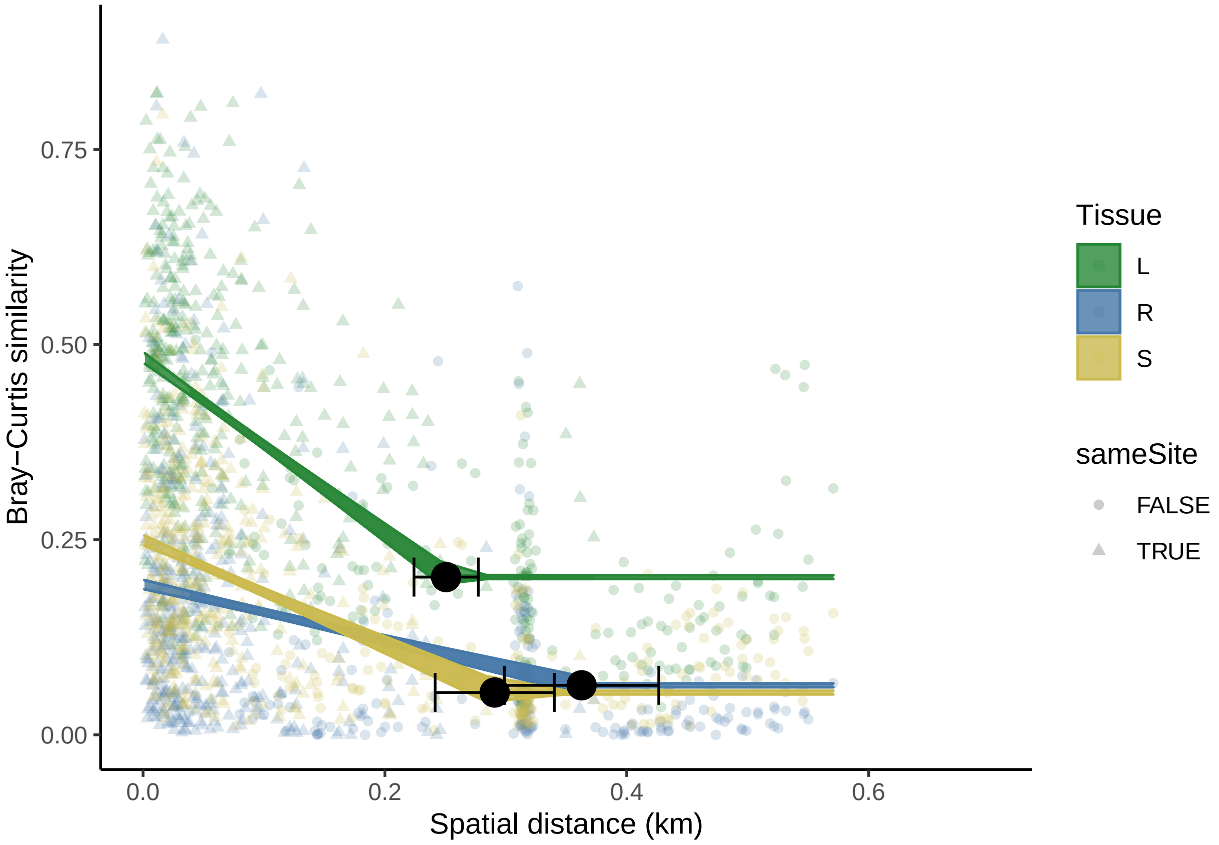
```
##                   Est. CI.95...low CI.95...up Tissue
## psi1.hav.L 0.250608    0.224059   0.277158      L
## psi1.hav.R 0.362648    0.298786   0.426510      R
## psi1.hav.S 0.290763    0.241495   0.340031      S
```

```
hav.l<- brks[1,'Est.']
hav.r<- brks[2,'Est.']
hav.s<- brks[3,'Est.']
y.l <- coef(mod.seg)['TissueL'] + hav.l *coef(mod.seg)['hav.L']
y.r <- coef(mod.seg)['TissueR'] + hav.r * coef(mod.seg)['hav.R']
y.s <- coef(mod.seg)['TissueS'] + hav.s * coef(mod.seg)['hav.S']
brks$y <- c(y.l, y.r, y.s)
#colnames(brks)
brks
```

```
##                   Est. CI.95...low CI.95...up Tissue          y
## psi1.hav.L 0.250608    0.224059   0.277158      L 0.20210852
## psi1.hav.R 0.362648    0.298786   0.426510      R 0.06342181
## psi1.hav.S 0.290763    0.241495   0.340031      S 0.05417516
```

```
p +
  xlim(c(0,.7)) +
  geom_errorbarh(data = brks,
                 aes(xmin = CI.95...low,
                     xmax = CI.95...up,
                     y = y), color = "black", height = .05,
                 inherit.aes = F) +
  geom_point(data = brks,
             aes(x = Est., y = y),
             size = 5, pch = 16, fill = "white",
             inherit.aes = F) -> p.sub
p.sub
```

```
## Warning: Removed 16482 rows containing missing values (geom_point).
```



```r
library(gridExtra)

# ggsave(p + guides(fill = F, shape = F, color = F),
#        filename = file.path(out_path,"dist_breaks_bray_full.png"),
#        width = 5, height = 4,
#        dpi = 600)
#
# ggsave(p.sub + guides(fill = F, shape = F, color = F),
#        filename = file.path(out_path,"dist_breaks_bray_inset.png"),
#        width = 5, height = 4,
#        dpi = 600)
#
# library(cowplot)
# p.leg<- get_legend(p)
# ggsave(plot_grid(p.leg),
#        filename = file.path(out_path,"dist_breaks_bray_legend.png"),
#        width = 5, height = 4,
#        dpi = 300)
```

Fit segmented regression models – Phylosor

```r
#library("segmented")
#str(dist.df)
dist.df$Tissue <- factor(dist.df$Tissue_samp1)

# build the dummy variables for the Tissue x distance interaction
require(segmented)
X <- model.matrix(~ 0 + dist.df$Tissue) * dist.df$hav.dist.km
```

```r
max(which(dist.df$Tissue == "L"))
```

```
## [1] 5886
```

```r
min(which(dist.df$Tissue == "R"))
```

```
## [1] 5887
```

```r
hav.L <- X[,1]
hav.R <- X[,2]
hav.S <- X[,3]
#colnames(dist.df)
mod <- lm(physor.comm.dist ~ 0 + Tissue + hav.L + hav.R + hav.S,
          data = dist.df)
mod
```

```
##
## Call:
## lm(formula = physor.comm.dist ~ 0 + Tissue + hav.L + hav.R +
##     hav.S, data = dist.df)
##
## Coefficients:
##    TissueL     TissueR     TissueS       hav.L       hav.R       hav.S
##  0.5864890   0.3477656   0.4429063  -0.0002326  -0.0001834  -0.0001913
```

```r
mod.seg <- segmented(mod, seg.Z = ~hav.L + hav.R + hav.S,
                         psi = list(hav.L = 1,
                                    hav.R = 1,
                                    hav.S = 1))
summary(mod.seg)
```

```
##
##  ***Regression Model with Segmented Relationship(s)***
##
## Call:
## segmented.lm(obj = mod, seg.Z = ~hav.L + hav.R + hav.S, psi = list(hav.L = 1,
##     hav.R = 1, hav.S = 1))
##
## Estimated Break-Point(s):
##              Est. St.Err
## psi1.hav.L 0.313  0.023
## psi1.hav.R 0.383  0.027
## psi1.hav.S 0.456  0.027
##
## Meaningful coefficients of the linear terms:
##            Estimate Std. Error t value Pr(>|t|)
## TissueL    0.716569   0.005541  129.32   <2e-16 ***
## TissueR    0.476201   0.005274   90.29   <2e-16 ***
## TissueS    0.590959   0.005068  116.60   <2e-16 ***
## hav.L     -0.494793   0.044605  -11.09   <2e-16 ***
## hav.R     -0.406105   0.035211  -11.53   <2e-16 ***
## hav.S     -0.398553   0.028832  -13.82   <2e-16 ***
## U1.hav.L   0.494658   0.044605   11.09       NA
## U1.hav.R   0.406028   0.035211   11.53       NA
## U1.hav.S   0.398494   0.028832   13.82       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08671 on 18195 degrees of freedom
## Multiple R-Squared: 0.9621,  Adjusted R-squared: 0.962
##
## Convergence attained in 1 iter. (rel. change 1.6829e-06)
```

```
tmp <- summary(mod.seg)
tmp$psi[1,2]
```

```
## [1] 0.3129977
```

```
capture.output(summary(mod.seg), file = file.path(out_path,"segPhysor.txt"))
#U1 = difference-in-slope parameter of the variable hav.L

# to test the significance of difference in slopes for each Tissue...
dt.l <- davies.test(mod, seg.Z = ~hav.L, k = 10, values = tmp$psi[1,2])
dt.r <- davies.test(mod, seg.Z = ~hav.R, k = 10, values = tmp$psi[2,2])
dt.s <- davies.test(mod, seg.Z = ~hav.S, k = 10, values = tmp$psi[3,2])
capture.output(dt.l, file = file.path(out_path,"segPhysor.txt"), append = T)
capture.output(dt.r, file = file.path(out_path,"segPhysor.txt"), append = T)
capture.output(dt.s, file = file.path(out_path,"segPhysor.txt"), append = T)

# save the CIs for breakpoints
brks <- rbind(confint.segmented(mod.seg, "hav.L"),
      confint.segmented(mod.seg, "hav.R"),
      confint.segmented(mod.seg, "hav.S"))
brks <- data.frame(brks, stringsAsFactors = F)
brks$Tissue <- c("L","R","S")
brks
```

```
##                 Est. CI.95...low CI.95...up Tissue
## psi1.hav.L 0.312998    0.268827   0.357169      L
## psi1.hav.R 0.383160    0.331098   0.435222      R
## psi1.hav.S 0.456253    0.404239   0.508267      S
```

```
brks$Est. - brks$CI.95...up
```

```
## [1] -0.044171 -0.052062 -0.052014
```

```
brks$Est. - brks$CI.95...low
```

```
## [1] 0.044171 0.052062 0.052014
```

```
capture.output(brks, file = file.path(out_path,"segPhysor.txt"), append = T)
# save the CIs for slopes
slopes <- list_to_df(slope(mod.seg))
capture.output(slopes, file = file.path(out_path,"segPhysor.txt"), append = T)

# break the regression
#library(lsmeans)
break.here <- mean(brks[,"Est."])
break.here
```

```
## [1] 0.384137
```

```
dist.df %>%
  filter(hav.dist.km < break.here) -> dist.dfa
```

```r
dist.df %>%
  filter(hav.dist.km > break.here) -> dist.dfb

# posthoc t-test to test difference in means - lower
moda <- lm(physor.comm.dist ~ Tissue * hav.dist.km, data = dist.dfa)
summary(moda)
```

```
##
## Call:
## lm(formula = physor.comm.dist ~ Tissue * hav.dist.km, data = dist.dfa)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38967 -0.04921  0.00535  0.05329  0.27955
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.714955   0.005095 140.333   <2e-16 ***
## TissueR              -0.238588   0.007146 -33.385   <2e-16 ***
## TissueS              -0.121517   0.007125 -17.055   <2e-16 ***
## hav.dist.km          -0.468074   0.033626 -13.920   <2e-16 ***
## TissueR:hav.dist.km   0.059257   0.047436   1.249    0.212
## TissueS:hav.dist.km   0.033003   0.047410   0.696    0.486
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0824 on 1533 degrees of freedom
## Multiple R-squared:  0.622,  Adjusted R-squared:  0.6208
## F-statistic: 504.6 on 5 and 1533 DF,  p-value: < 2.2e-16
```

```r
capture.output(summary(moda),
               file = file.path(out_path,"segPhysor.txt"), append = T)

pred.a <- predict(moda)
#pred.a
moda.lst <- lstrends(moda, ~ Tissue, var = "hav.dist.km")
moda.lst
```

```
##  Tissue hav.dist.km.trend     SE   df lower.CL upper.CL
##  L                 -0.468 0.0336 1533   -0.534   -0.402
##  R                 -0.409 0.0335 1533   -0.474   -0.343
##  S                 -0.435 0.0334 1533   -0.501   -0.370
##
## Confidence level used: 0.95
```

```r
pairs(moda.lst)   # comparisons of slopes
```

```
##  contrast estimate     SE   df t.ratio p.value
##  L - R     -0.0593 0.0474 1533 -1.249  0.4244
##  L - S     -0.0330 0.0474 1533 -0.696  0.7658
##  R - S      0.0263 0.0473 1533  0.555  0.8438
##
## P value adjustment: tukey method for comparing a family of 3 estimates
```

```r
TukeyHSD(aov(moda), which = "Tissue") # comparison of intercepts
```

```
## Warning in replications(paste("~", xx), data = mf): non-factors ignored:
## hav.dist.km

## Warning in replications(paste("~", xx), data = mf): non-factors ignored: Tissue,
## hav.dist.km

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = moda)
##
## $Tissue
##           diff        lwr        upr p adj
## R-L -0.2317164 -0.2438400 -0.2195927     0
## S-L -0.1171604 -0.1292439 -0.1050770     0
## S-R  0.1145559  0.1025505  0.1265614     0
```

```
capture.output(summary(moda), file = file.path(out_path,"segPhysor.txt"), append = T)
capture.output(pairs(moda.lst), file = file.path(out_path,"segPhysor.txt"), append = T)
capture.output(TukeyHSD(aov(moda), which = "Tissue"),
               file = file.path(out_path,"segPhysor.txt"), append = T)
```

```
## Warning in replications(paste("~", xx), data = mf): non-factors ignored:
## hav.dist.km

## Warning in replications(paste("~", xx), data = mf): non-factors ignored: Tissue,
## hav.dist.km
```

```
# posthoc t-test to test difference in means - upper
modb <- lm(physor.comm.dist ~ Tissue * hav.dist.km, data = dist.dfb)
summary(modb)
```

```
##
## Call:
## lm(formula = physor.comm.dist ~ Tissue * hav.dist.km, data = dist.dfb)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41673 -0.05057  0.01113  0.06032  0.40205
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.616e-01  2.179e-03 257.740  < 2e-16 ***
## TissueR             -2.409e-01  3.061e-03 -78.699  < 2e-16 ***
## TissueS             -1.517e-01  3.049e-03 -49.775  < 2e-16 ***
## hav.dist.km         -1.345e-04  1.022e-05 -13.155  < 2e-16 ***
## TissueR:hav.dist.km  5.728e-05  1.432e-05   4.000 6.35e-05 ***
## TissueS:hav.dist.km  7.313e-05  1.427e-05   5.125 3.01e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08711 on 16662 degrees of freedom
## Multiple R-squared:  0.5423, Adjusted R-squared:  0.5421
## F-statistic:  3948 on 5 and 16662 DF,  p-value: < 2.2e-16
```

```
capture.output(summary(modb),
               file = file.path(out_path,"segPhysor.txt"), append = T)
```

```
modb.lst <- lstrends(modb, ~ Tissue, var = "hav.dist.km")
pairs(modb.lst)   # comparisons of slopes
```

```
##  contrast  estimate       SE    df t.ratio p.value
##  L - R    -5.73e-05 1.43e-05 16662 -4.000  0.0002
##  L - S    -7.31e-05 1.43e-05 16662 -5.125  <.0001
##  R - S    -1.58e-05 1.41e-05 16662 -1.122  0.5007
##
## P value adjustment: tukey method for comparing a family of 3 estimates
```

```
TukeyHSD(aov(modb), which = "Tissue") # comparison of intercepts
```

```
## Warning in replications(paste("~", xx), data = mf): non-factors ignored:
## hav.dist.km

## Warning in replications(paste("~", xx), data = mf): non-factors ignored: Tissue,
## hav.dist.km

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = modb)
##
## $Tissue
##            diff         lwr         upr p adj
## R-L -0.23076720 -0.23466595 -0.22686844     0
## S-L -0.13876260 -0.14264385 -0.13488136     0
## S-R  0.09200459  0.08816028  0.09584891     0
```

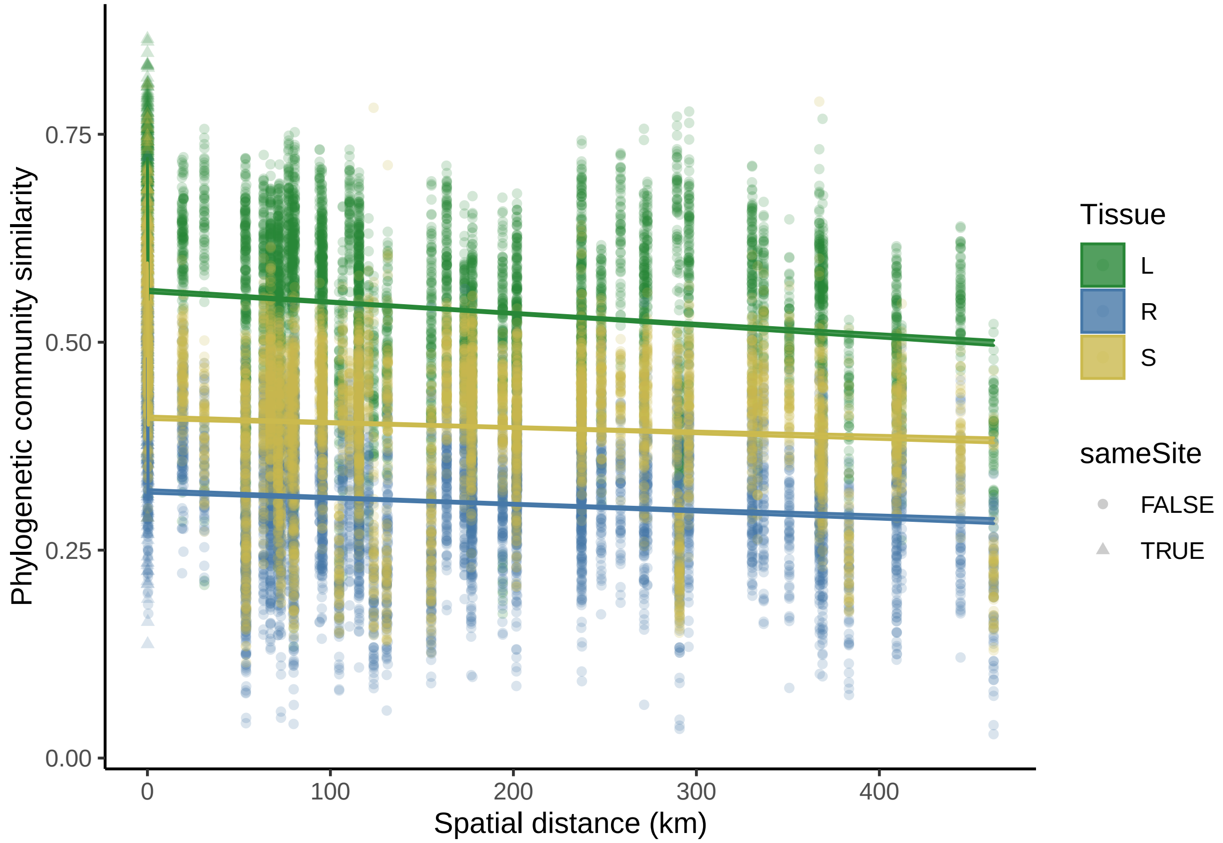
```
capture.output(summary(modb), file = file.path(out_path,"segPhysor.txt"), append = T)
capture.output(pairs(modb.lst), file = file.path(out_path,"segPhysor.txt"), append = T)
capture.output(TukeyHSD(aov(modb), which = "Tissue"),
               file = file.path(out_path,"segPhysor.txt"), append = T)
```

```
## Warning in replications(paste("~", xx), data = mf): non-factors ignored:
## hav.dist.km

## Warning in replications(paste("~", xx), data = mf): non-factors ignored: Tissue,
## hav.dist.km
```

```
# use predict to show the fitted model
pred <- predict(mod.seg, se.fit = TRUE)
mod.seg
```

```
## Call: segmented.lm(obj = mod, seg.Z = ~hav.L + hav.R + hav.S, psi = list(hav.L = 1,
##     hav.R = 1, hav.S = 1))
##
## Meaningful coefficients of the linear terms:
##  TissueL   TissueR   TissueS     hav.L     hav.R     hav.S  U1.hav.L  U1.hav.R
##   0.7166    0.4762    0.5910   -0.4948   -0.4061   -0.3986    0.4947    0.4060
## U1.hav.S
##   0.3985
##
## Estimated Break-Point(s):
## psi1.hav.L  psi1.hav.R  psi1.hav.S
##     0.3130      0.3832      0.4563
```

```r
dist.df$pred <- pred$fit
dist.df$pred.se <- pred$se.fit
dist.df$pred.before <- NA
dist.df[dist.df$hav.dist.km < break.here,"pred.before"] <- pred.a

p <- ggplot(dist.df, aes(x = hav.dist.km, y = pred,
                    fill = Tissue, color = Tissue, shape = sameSite)) +
  geom_point(aes(y = physor.comm.dist), alpha = .2) +
  geom_ribbon(aes(ymin = pred - pred.se, ymax = pred + pred.se),
              alpha = .8) +
  #geom_line()+
  theme_classic() +
  ylab("Phylogenetic community similarity") +
  xlab("Spatial distance (km)") +
  scale_color_manual(values = tissue.colors) +
  scale_fill_manual(values = tissue.colors)
p
```



```r
# p + xlim(c(0,.7)) +
#   geom_line(aes(y = pred.before, x = hav.dist.km, color = Tissue), inherit.aes = F)


# add error around breaks
brks <- rbind(confint.segmented(mod.seg, "hav.L"),
      confint.segmented(mod.seg, "hav.R"),
      confint.segmented(mod.seg, "hav.S"))
brks <- data.frame(brks, stringsAsFactors = F)
```

```
brks$Tissue <- c("L","R","S")
brks
```

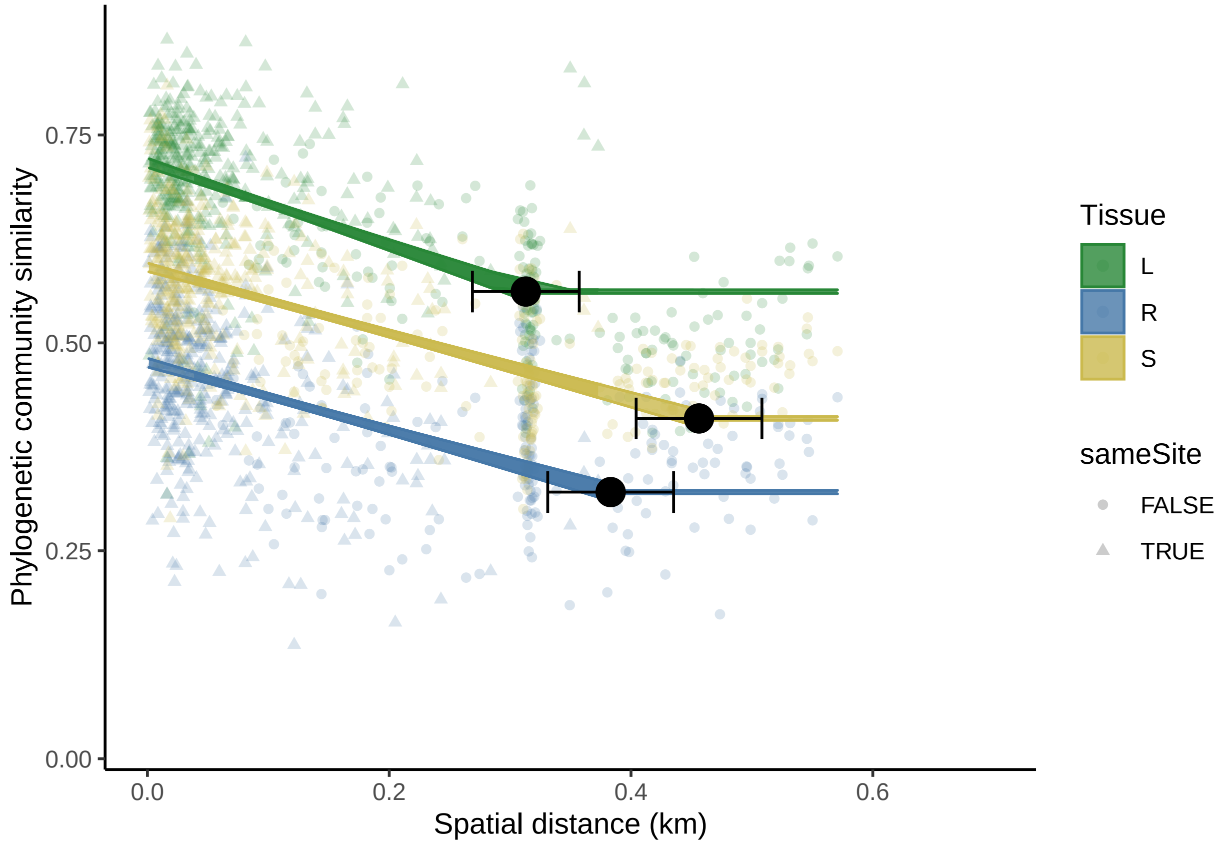
```
##                  Est. CI.95...low CI.95...up Tissue
## psi1.hav.L 0.312998    0.268827   0.357169      L
## psi1.hav.R 0.383160    0.331098   0.435222      R
## psi1.hav.S 0.456253    0.404239   0.508267      S
```

```
hav.l<- brks[1,'Est.']
hav.r<- brks[2,'Est.']
hav.s<- brks[3,'Est.']

y.l <- coef(mod.seg)['TissueL'] + hav.l *coef(mod.seg)['hav.L']
y.r <- coef(mod.seg)['TissueR'] + hav.r * coef(mod.seg)['hav.R']
y.s <- coef(mod.seg)['TissueS'] + hav.s * coef(mod.seg)['hav.S']
brks$y <- c(y.l, y.r, y.s)
#colnames(brks)
brks
```

```
##                  Est. CI.95...low CI.95...up Tissue         y
## psi1.hav.L 0.312998    0.268827   0.357169      L 0.5617001
## psi1.hav.R 0.383160    0.331098   0.435222      R 0.3205973
## psi1.hav.S 0.456253    0.404239   0.508267      S 0.4091176
```

```
p +
  xlim(c(0,.7)) +
  geom_errorbarh(data = brks,
                 aes(xmin = CI.95...low,
                     xmax = CI.95...up,
                     y = y), color = "black", height = .05,
                 inherit.aes = F) +
  geom_point(data = brks,
             aes(x = Est., y = y),
             size = 5, pch = 16, fill = "white",
             inherit.aes = F) -> p.sub
p.sub
```

```
## Warning: Removed 16482 rows containing missing values (geom_point).
```



```
#
# ggsave(p + guides(fill = F, shape = F, color = F),
#        filename = file.path(out_path,"dist_breaks_physor_full.png"),
#        width = 5, height = 4,
#        dpi = 600)
#
# ggsave(p.sub + guides(fill = F, shape = F, color = F),
#        filename = file.path(out_path,"dist_breaks_physor_inset.png"),
#        width = 5, height = 4,
#        dpi = 600)
#
# library(cowplot)
# p.leg<- get_legend(p)
# ggsave(plot_grid(p.leg),
#        filename = file.path(out_path,"dist_breaks_physor_legend.png"),
#        width = 5, height = 4,
#        dpi = 300)
```

Follow-ups

Examine just within-site distances [commented out]

```
# dist.df %>%
#   filter(sameSite == TRUE) -> dist.df.s
#
# ggplot(dist.df.s, aes(x = hav.dist.km, y = physor.comm.dist)) +
#   geom_point() +
#   facet_grid(~Tissue_samp1)
#
# library(lme4)
# library(lmerTest)
# library(lsmeans)
# mod <- lmer(physor.comm.dist ~ Tissue_samp1 * hav.dist.km + (1|Site_samp1), data = dist.df.s)
# summary(mod)
# capture.output(summary(mod),
#                file = file.path(out_path,"segPhysor_withinSite.txt"))
# capture.output(anova(mod),
#                file = file.path(out_path,"segPhysor_withinSite.txt"),
#                append = T)
# capture.output(emmeans(mod, list(pairwise ~ Tissue_samp1), adjust = "tukey"),
#                file = file.path(out_path,"segPhysor_withinSite.txt"),
#                append = T)
# emmeans(mod, list(pairwise ~ Tissue_samp1), adjust = "tukey")
# trend <- lstrends(mod, ~ Tissue_samp1, var = "hav.dist.km")
# capture.output(trend,
#                file = file.path(out_path,"segPhysor_withinSite.txt"),
#                append = T)
# capture.output(pairs(trend),
#                file = file.path(out_path,"segPhysor_withinSite.txt"),
#                append = T)
#
# sd(residuals(mod))/sqrt(length(residuals(mod)))
# mod
# summary(mod)
#
# library(MuMIn)
# r.squaredGLMM(mod)
#
#
# # add environmental distances
# sam <- data.frame(sample_data(ps))
# # load transformed environmental variables (prior to lasso filter)
# mat.t <- read.csv(file = "output/illumina/Q2/normTransformed_contvars_trim.csv",
#                   row.names = 1)
# sam %>%
#     dplyr::select(sample.name.match, SiteSamp, Site, Tissue) -> samp.tmp
# samp.tmp %>%
#   left_join(mat.t) -> samp.tmp
# samp.tmp %>%
#   dplyr::select(-c(sample.name.match, SiteSamp, Site, Tissue)) -> samp.env
# row.names(samp.env)<- samp.tmp$sample.name.match
# dist.env <- dist(samp.env, method = "euclidean")
# mat.env <- as.matrix(dist.env)
# env.dist.df <- extract_uniquePairDists(mat.env)
# env.dist.df %>%
#     dplyr::rename('env.dist.m'='dist') -> env.dist.df
```

Examine leaf communities at long distances… potential environmental drivers? [commented out]

```
# dist.df %>%
#   filter(Tissue_samp1 == "L") -> dist.df.l
```

```
#
# # color by site comparisons
# all.sites <- unique(c(dist.df$Site_samp1,dist.df$Site_samp2))
# all.site.pairs <- data.frame(t(combn(all.sites, 2)))
# all.site.pairs$pairs <- paste0(all.site.pairs$X1, "__", all.site.pairs$X2)
# pairs <- all.site.pairs$pairs
#
# dist.df.l %>%
#   mutate(site.pairs = paste0(Site_samp1,"__", Site_samp2)) %>%
#   mutate(site.pairs.rev = paste0(Site_samp2, "__", Site_samp1)) %>%
#   mutate(site.pairs = ifelse(site.pairs %in% pairs, site.pairs, site.pairs.rev)) %>%
#   mutate(site.pairs = ifelse(sameSite == TRUE, Site_samp1, site.pairs)) %>%
#   dplyr::select(-site.pairs.rev) -> tmp
#
# ggplot(tmp, aes(x = hav.dist.km, y = physor.comm.dist, color = site.pairs)) +
#   geom_point() +
#   guides(color = F)
# ggplot(tmp, aes(x = hav.dist.km, y = reorder(site.pairs, hav.dist.km),
#                 color = physor.comm.dist)) +
#   geom_point()
#
# ggplot(tmp, aes(x = hav.dist.km, y = reorder(site.pairs, hav.dist.km),
#                 color = env.dist.m)) +
#   geom_point()
# ggplot(tmp, aes(x = hav.dist.km, y = env.dist.m,
#                 color = site.pairs)) +
#   geom_point() +
#   guides(color = F)
#
# #### what drives environmental distances at long distances?
# sam <- data.frame(sample_data(ps))
# # load transformed environmental variables (prior to lasso filter)
# mat.t <- read.csv(file = "output/illumina/Q2/normTransformed_contvars_trim.csv",
#                   row.names = 1)
# sam %>%
#     dplyr::select(sample.name.match, SiteSamp, Site, Tissue) %>%
#   left_join(mat.t) %>%
#   filter(Tissue == "L") -> samp.tmp
#
# #K
# dist <- dist(samp.tmp$K, method = "euclidean")
# mat <- as.matrix(dist)
# row.names(mat) <- samp.tmp$sample.name.match
# colnames(mat) <- samp.tmp$sample.name.match
# mat.df.k <- extract_uniquePairDists(mat)
# colnames(mat.df.k) <- c("samp1","samp2","k.dist")
#
# #P
# dist <- dist(samp.tmp$P, method = "euclidean")
# mat <- as.matrix(dist)
# row.names(mat) <- samp.tmp$sample.name.match
# colnames(mat) <- samp.tmp$sample.name.match
# mat.df.p <- extract_uniquePairDists(mat)
# colnames(mat.df.p) <- c("samp1","samp2","p.dist")
#
# #pH
# dist <- dist(samp.tmp$ph, method = "euclidean")
# mat <- as.matrix(dist)
# row.names(mat) <- samp.tmp$sample.name.match
# colnames(mat) <- samp.tmp$sample.name.match
# mat.df.ph <- extract_uniquePairDists(mat)
# colnames(mat.df.ph) <- c("samp1","samp2","ph.dist")
#
# #height
# dist <- dist(samp.tmp$max.height.m, method = "euclidean")
# mat <- as.matrix(dist)
# row.names(mat) <- samp.tmp$sample.name.match
# colnames(mat) <- samp.tmp$sample.name.match
# mat.df.h <- extract_uniquePairDists(mat)
# colnames(mat.df.h) <- c("samp1","samp2","height.dist")
#
# # combine
# tmp %>%
#   left_join(mat.df.k) %>%
#   left_join(mat.df.p) %>%
#   left_join(mat.df.ph) %>%
#   left_join(mat.df.h) -> tmp.env
#
# ggplot(tmp.env, aes(x = hav.dist.km, y = k.dist,
#                  color = site.pairs)) +
#   geom_point() +
#   guides(color = F)
# tmp.env %>%
#   filter(hav.dist.km > 400) -> sub
# tmp.env %>%
#   group_by(site.pairs) %>%
#   summarize(n = length(k.dist),
#             mean = mean(k.dist),
#             sd = sd(k.dist)) %>%
#   arrange(-mean)
#
# tmp.env %>%
#   group_by(site.pairs) %>%
#   summarize(n = length(hav.dist.km),
#             mean = mean(hav.dist.km)) %>%
#   arrange(-mean)
#
# samp.tmp %>%
#   group_by(Site) %>%
#   summarize(n = length(K),
#             mean = mean(K),
#             sd = sd(K),
#             se = sd/sqrt(n)) %>%
#   arrange(mean)
#
#
```

```
# ggplot(tmp.env, aes(x = hav.dist.km, y = p.dist,
#                 color = site.pairs)) +
#   geom_point() +
#   guides(color = F)
# ggplot(tmp.env, aes(x = hav.dist.km, y = ph.dist,
#                 color = site.pairs)) +
#   geom_point() +
#   guides(color = F)
# ggplot(tmp.env, aes(x = hav.dist.km, y = height.dist,
#                 color = site.pairs)) +
#   geom_point() +
#   guides(color = F)
```

3. Breakpoint regression with environmental distance

Fit segmented regression models – Phylosor w/environmental distance [commented out]

```
# #library("segmented")
# str(dist.df)
# dist.df$Tissue <- factor(dist.df$Tissue_samp1)
#
# # build the dummy variables for the Tissue x distance interaction
# require(segmented)
# colnames(dist.df)
# X <- model.matrix(~ 0 + dist.df$Tissue) * dist.df$env.dist.m
# max(which(dist.df$Tissue == "L"))
# min(which(dist.df$Tissue == "R"))
# hav.L <- X[,1]
# hav.R <- X[,2]
# hav.S <- X[,3]
# mod <- lm(physor.comm.dist ~ 0 + Tissue + hav.L + hav.R + hav.S,
#           data = dist.df)
# mod
# mod.seg <- segmented(mod, seg.Z = ~hav.L + hav.R + hav.S,
#                          psi = list(hav.L = 1,
#                                     hav.R = 1,
#                                     hav.S = 1))
# summary(mod.seg)
# tmp <- summary(mod.seg)
# tmp$psi[1,2]
#
# capture.output(summary(mod.seg), file = file.path(out_path,"segPhysor_env.txt"))
# #U1 = difference-in-slope parameter of the variable hav.L
#
# # to test the significance of difference in slopes for each Tissue...
# dt.l <- davies.test(mod, seg.Z = ~hav.L, k = 10, values = tmp$psi[1,2])
# dt.r <- davies.test(mod, seg.Z = ~hav.R, k = 10, values = tmp$psi[2,2])
# dt.s <- davies.test(mod, seg.Z = ~hav.S, k = 10, values = tmp$psi[3,2])
# dt.l
# dt.r
# dt.s
# #r
# # rearrange terms
# # r2 for each separate mode
# # r2 for SEMs?
# #
# capture.output(dt.l, file = file.path(out_path,"segPhysor_env.txt"), append = T)
# capture.output(dt.r, file = file.path(out_path,"segPhysor_env.txt"), append = T)
# capture.output(dt.s, file = file.path(out_path,"segPhysor_env.txt"), append = T)
#
# # save the CIs for breakpoints
# brks <- rbind(confint.segmented(mod.seg, "hav.L"),
#       confint.segmented(mod.seg, "hav.R"),
#       confint.segmented(mod.seg, "hav.S"))
# brks <- data.frame(brks, stringsAsFactors = F)
# brks$Tissue <- c("L","R","S")
# brks
# brks$Est. - brks$CI.95...up
# brks$Est. - brks$CI.95...low
#
# capture.output(brks, file = file.path(out_path,"segPhysor_env.txt"), append = T)
# # save the CIs for slopes
# slopes <- list_to_df(slope(mod.seg))
# capture.output(slopes, file = file.path(out_path,"segPhysor_env.txt"), append = T)
#
# # break the regression
# #library(lsmeans)
# break.here <- mean(brks[,"Est."])
# break.here
# dist.df %>%
#   filter(env.dist.m < break.here) -> dist.dfa
# dist.df %>%
#   filter(env.dist.m > break.here) -> dist.dfb
#
# # posthoc t-test to test difference in means - lower
# moda <- lm(physor.comm.dist ~ Tissue * env.dist.m, data = dist.dfa)
# summary(moda)
# pred.a <- predict(moda)
# pred.a
# library(lsmeans)
# moda.lst <- lstrends(moda, ~ Tissue, var = "env.dist.m")
# moda.lst
# pairs(moda.lst)   # comparisons of slopes
# TukeyHSD(aov(moda), which = "Tissue") # comparison of intercepts
# capture.output(summary(moda), file = file.path(out_path,"segPhysor_env.txt"), append = T)
# capture.output(pairs(moda.lst), file = file.path(out_path,"segPhysor_env.txt"), append = T)
# capture.output(TukeyHSD(aov(moda), which = "Tissue"),
#                file = file.path(out_path,"segPhysor_env.txt"), append = T)
#
# # posthoc t-test to test difference in means - upper
# modb <- lm(physor.comm.dist ~ Tissue * env.dist.m, data = dist.dfb)
# summary(modb)
# modb.lst <- lstrends(modb, ~ Tissue, var = "env.dist.m")
# pairs(modb.lst)   # comparisons of slopes
# TukeyHSD(aov(modb), which = "Tissue") # comparison of intercepts
# capture.output(summary(modb), file = file.path(out_path,"segPhysor_env.txt"), append = T)
# capture.output(pairs(modb.lst), file = file.path(out_path,"segPhysor_env.txt"), append = T)
# capture.output(TukeyHSD(aov(modb), which = "Tissue"),
#                 file = file.path(out_path,"segPhysor_env.txt"), append = T)
#
# # use predict to show the fitted model
# pred <- predict(mod.seg, se.fit = TRUE)
# mod.seg
# dist.df$pred <- pred$fit
# dist.df$pred.se <- pred$se.fit
# dist.df$pred.before <- NA
# dist.df[dist.df$env.dist.m < break.here,"pred.before"] <- pred.a
#
# p <- ggplot(dist.df, aes(x = env.dist.m, y = pred,
#                      fill = Tissue, color = Tissue, shape = sameSite)) +
#   geom_point(aes(y = physor.comm.dist), alpha = .2) +
#   geom_ribbon(aes(ymin = pred - pred.se, ymax = pred + pred.se),
#               alpha = .8) +
#   #geom_line()+
#   theme_classic() +
#   ylab("Phylogenetic community similarity") +
#   xlab("Environmental distance (Euclidean)")
# p
#
# # p + xlim(c(0,.7)) +
# #   geom_line(aes(y = pred.before, x = hav.dist.km, color = Tissue), inherit.aes = F)
#
# # add error around breaks
# brks <- rbind(confint.segmented(mod.seg, "hav.L"),
#       confint.segmented(mod.seg, "hav.R"),
#       confint.segmented(mod.seg, "hav.S"))
# brks <- data.frame(brks, stringsAsFactors = F)
# brks$Tissue <- c("L","R","S")
# brks
# hav.l<- brks[1,'Est.']
# hav.r<- brks[2,'Est.']
# hav.s<- brks[3,'Est.']
#
# y.l <- coef(mod.seg)['TissueL'] + hav.l *coef(mod.seg)['hav.L']
# y.r <- coef(mod.seg)['TissueR'] + hav.r * coef(mod.seg)['hav.R']
# y.s <- coef(mod.seg)['TissueS'] + hav.s * coef(mod.seg)['hav.S']
# brks$y <- c(y.l, y.r, y.s)
# colnames(brks)
# brks
# p +
#   geom_errorbarh(data = brks,
#                  aes(xmin = CI.95...low,
#                      xmax = CI.95...up,
#                      y = y), color = "black", height = .05,
#                  inherit.aes = F) +
#   geom_point(data = brks,
#              aes(x = Est., y = y),
#              size = 5, pch = 16, fill = "white",
#              inherit.aes = F) -> p.sub
# p.sub
```

```
#
#
# pdf(file = file.path(out_path, "dist_breaks_physor.pdf"), width = 10, height = 4)
# grid.arrange(
#   p,
#   p.sub, ncol = 2
# )
# dev.off()
```

Fit segmented regression models – Bray w/environmental distance [commented out]

```
# #library("segmented")
# str(dist.df)
# dist.df$Tissue <- factor(dist.df$Tissue_samp1)
#
# # build the dummy variables for the Tissue x distance interaction
# require(segmented)
# colnames(dist.df)
# X <- model.matrix(~ 0 + dist.df$Tissue) * dist.df$env.dist.m
# max(which(dist.df$Tissue == "L"))
# min(which(dist.df$Tissue == "R"))
# hav.L <- X[,1]
# hav.R <- X[,2]
# hav.S <- X[,3]
# mod <- lm(bray.comm.dist ~ 0 + Tissue + hav.L + hav.R + hav.S,
#           data = dist.df)
# mod
# mod.seg <- segmented(mod, seg.Z = ~hav.L + hav.R + hav.S,
#                           psi = list(hav.L = 1,
#                                      hav.R = 1,
#                                      hav.S = 1))
# summary(mod.seg)
# tmp <- summary(mod.seg)
# tmp$psi[1,2]
#
# capture.output(summary(mod.seg), file = file.path(out_path,"segBray_env.txt"))
# #U1 = difference-in-slope parameter of the variable hav.L
#
# # to test the significance of difference in slopes for each Tissue...
# dt.l <- davies.test(mod, seg.Z = ~hav.L, k = 10, values = tmp$psi[1,2])
# dt.r <- davies.test(mod, seg.Z = ~hav.R, k = 10, values = tmp$psi[2,2])
# dt.s <- davies.test(mod, seg.Z = ~hav.S, k = 10, values = tmp$psi[3,2])
# dt.l
# dt.r
# dt.s
# #r
# # rearrange terms
# # r2 for each separate mode
# # r2 for SEMs?
# #
# capture.output(dt.l, file = file.path(out_path,"segBray_env.txt"), append = T)
# capture.output(dt.r, file = file.path(out_path,"segBray_env.txt"), append = T)
# capture.output(dt.s, file = file.path(out_path,"segBray_env.txt"), append = T)
#
```

```
# # save the CIs for breakpoints
# brks <- rbind(confint.segmented(mod.seg, "hav.L"),
#       confint.segmented(mod.seg, "hav.R"),
#       confint.segmented(mod.seg, "hav.S"))
# brks <- data.frame(brks, stringsAsFactors = F)
# brks$Tissue <- c("L","R","S")
# brks
# brks$Est. - brks$CI.95...up
# brks$Est. - brks$CI.95...low
#
# capture.output(brks, file = file.path(out_path,"segBray_env.txt"), append = T)
# # save the CIs for slopes
# slopes <- list_to_df(slope(mod.seg))
# capture.output(slopes, file = file.path(out_path,"segBray_env.txt"), append = T)
#
# # break the regression
# #library(lsmeans)
# break.here <- mean(brks[,"Est."])
# break.here
# dist.df %>%
#   filter(hav.dist.km < break.here) -> dist.dfa
# dist.df %>%
#   filter(hav.dist.km > break.here) -> dist.dfb
#
# # posthoc t-test to test difference in means - lower
# moda <- lm(physor.comm.dist ~ Tissue * env.dist.m, data = dist.dfa)
# summary(moda)
# pred.a <- predict(moda)
# pred.a
# library(lsmeans)
# moda.lst <- lstrends(moda, ~ Tissue, var = "env.dist.m")
# moda.lst
# pairs(moda.lst)   # comparisons of slopes
# TukeyHSD(aov(moda), which = "Tissue") # comparison of intercepts
# capture.output(summary(moda), file = file.path(out_path,"segPhysor_env.txt"), append = T)
# capture.output(pairs(moda.lst), file = file.path(out_path,"segPhysor_env.txt"), append = T)
# capture.output(TukeyHSD(aov(moda), which = "Tissue"),
#                file = file.path(out_path,"segPhysor_env.txt"), append = T)
#
# # posthoc t-test to test difference in means - upper
# modb <- lm(physor.comm.dist ~ Tissue * env.dist.m, data = dist.dfb)
# summary(modb)
# modb.lst <- lstrends(modb, ~ Tissue, var = "env.dist.m")
# pairs(modb.lst)   # comparisons of slopes
# TukeyHSD(aov(modb), which = "Tissue") # comparison of intercepts
# capture.output(summary(modb), file = file.path(out_path,"segPhysor_env.txt"), append = T)
# capture.output(pairs(modb.lst), file = file.path(out_path,"segPhysor_env.txt"), append = T)
# capture.output(TukeyHSD(aov(modb), which = "Tissue"),
#                file = file.path(out_path,"segPhysor_env.txt"), append = T)
#
# # use predict to show the fitted model
# pred <- predict(mod.seg, se.fit = TRUE)
# mod.seg
# dist.df$pred <- pred$fit
# dist.df$pred.se <- pred$se.fit
# dist.df$pred.before <- NA
# dist.df[dist.df$hav.dist.km < break.here,"pred.before"] <- pred.a
#
# p <- ggplot(dist.df, aes(x = env.dist.m, y = pred,
#                      fill = Tissue, color = Tissue, shape = sameSite)) +
#   geom_point(aes(y = physor.comm.dist), alpha = .2) +
#   geom_ribbon(aes(ymin = pred - pred.se, ymax = pred + pred.se),
#               alpha = .8) +
#   #geom_line()+
#   theme_classic() +
#   ylab("Phylogenetic community similarity") +
#   xlab("Environmental distance (Euclidean)")
# p
#
# # p + xlim(c(0,.7)) +
# #   geom_line(aes(y = pred.before, x = hav.dist.km, color = Tissue), inherit.aes = F)
#
#
# # add error around breaks
# brks <- rbind(confint.segmented(mod.seg, "hav.L"),
#       confint.segmented(mod.seg, "hav.R"),
#       confint.segmented(mod.seg, "hav.S"))
# brks <- data.frame(brks, stringsAsFactors = F)
# brks$Tissue <- c("L","R","S")
# brks
# hav.l<- brks[1,'Est.']
# hav.r<- brks[2,'Est.']
# hav.s<- brks[3,'Est.']
#
# y.l <- coef(mod.seg)['TissueL'] + hav.l *coef(mod.seg)['hav.L']
# y.r <- coef(mod.seg)['TissueR'] + hav.r * coef(mod.seg)['hav.R']
# y.s <- coef(mod.seg)['TissueS'] + hav.s * coef(mod.seg)['hav.S']
# brks$y <- c(y.l, y.r, y.s)
# colnames(brks)
# brks
# p +
#   xlim(c(0,.7)) +
#   geom_errorbarh(data = brks,
#                  aes(xmin = CI.95...low,
#                      xmax = CI.95...up,
#                      y = y), color = "black", height = .05,
#                  inherit.aes = F) +
#   geom_point(data = brks,
#              aes(x = Est., y = y),
#              size = 5, pch = 16, fill = "white",
#              inherit.aes = F) -> p.sub
# p.sub
#
#
# pdf(file = file.path(out_path, "dist_breaks_physor.pdf"), width = 10, height = 4)
# grid.arrange(
#   p,
#   p.sub, ncol = 2
# )
# dev.off()
```

Plot DPCoA of the mixed tree sites w/ nearby sites [commented out]

```
# ps <- readRDS(file = file.path(merged_path, "phyloseq_samps_env_trimTreeASVs.RData"))
# sam <- data.frame(sample_data(ps), stringsAsFactors = F)
# df.sam <- read.csv(file = file.path(out_path, "sample_dpcoaScores.csv"))
# colnames(df.sam)[1] <-"sample.name.match"
#
# sam %>%
#   left_join(df.sam) -> sam
#
# sam %>%
#   filter(Site %in% c("CRE-MXT-NCD","CRE-MXG-NCD",
#                      "OTO-MXT-NCD","OTO-MON-NCD")) %>%
#   mutate(loc = ifelse(Site %in% c("CRE-MXT-NCD","CRE-MXG-NCD"),
#   "CRE", "OTO")) %>%
#   mutate(tree = ifelse(Site %in% c("CRE-MXT-NCD", "OTO-MXT-NCD"),
#                        TRUE, FALSE))-> tmp
# ggplot(tmp, aes(x = DPCoA1, y = DPCoA2, color = tree)) +
#   geom_point() +
#   facet_grid(loc~Tissue)
#
#
# # also look at pairwise distances
# # CRE
# dist.df %>%
#   filter(grepl("CRE", Site_samp1)) %>%
#   filter(grepl("CRE", Site_samp2)) -> dist.cre
# colnames(dist.cre)
# dist.cre %>%
#   mutate(site.pair = paste0(Site_samp1, "__", Site_samp2)) %>%
#   mutate(site.pair = ifelse(sameSite == TRUE, Site_samp1, "Between")) -> dist.cre
#
# ggplot(dist.cre, aes(x = hav.dist.km, y = physor.comm.dist,
#                      color = site.pair)) +
#   geom_point() +
#   geom_smooth(method =  "lm") +
#   facet_grid(~Tissue_samp1)
#
# # OTO
# dist.df %>%
#   filter(grepl("OTO", Site_samp1)) %>%
#   filter(grepl("OTO", Site_samp2)) -> dist.cre
# dist.cre %>%
#   mutate(site.pair = paste0(Site_samp1, "__", Site_samp2)) %>%
#   mutate(site.pair = ifelse(sameSite == TRUE, Site_samp1, "Between")) -> dist.cre
# ggplot(dist.cre, aes(x = hav.dist.km, y = physor.comm.dist,
#                      color = site.pair)) +
#   geom_point() +
#   geom_smooth(method =  "lm") +
#   facet_grid(~Tissue_samp1) +
#   xlim(0, 0.05)
#
#
# # only within sites
# dist.df %>%
#   filter(sameSite == TRUE) %>%
#   filter(Tissue_samp1 == "L") -> tmp
# tmp$Site <- tmp$Site_samp1
# indx <- sam[,c("Site","mono.mixed")]
# tmp %>%
#   left_join(indx) -> tmp
#
# ggplot(tmp, aes(x = hav.dist.km, y = , color = Site)) +
#   geom_point() +
#   facet_wrap(~mono.mixed)
#
# mod <- lm(physor.comm.dist ~ hav.dist.km*Site, data = tmp)
# anova(mod)
#
# lstends(mod,"Site")
```