

## Text S1

### ###pollen diagram

```
library(rioja)
library(dplyr)
library(vegan)
```

###we prepared the data using the max function in Excel to remove all species that did not reach 5% abundance in at least one sample. The imported csv file should include species as columns and the samples as the first row. Optional, to change the order that species appear on the diagram add a number before the species name.

```
data = read.csv("FileName.csv")
```

```
data.clean <- data[,-c(1)]
```

```
###cluster analysis
```

```
data.dist <- vegdist(data.clean, method="bray", binary=FALSE, diag=FALSE, upper=FALSE, na.rm = FALSE)
```

```
data.chclust <- chclust(data.dist, method="coniss")
```

```
###plot figure
```

```
p.col.group <- c(rep("black", times=40))
```

```
y.scale <- 1:40
```

```
pol.plot <- strat.plot(data.clean, yvar=y.scale, y.tks=y.scale, y.rev=FALSE, plot.line=FALSE,
plot.poly=FALSE, plot.bar=TRUE, col.bar=p.col.group, lwd.bar=10, sep.bar=TRUE, scale.percent=TRUE,
xSpace=0.01, x.pc.lab=TRUE, x.pc.omit0=TRUE, srt.xlabel=45, las=2, clust=data.chclust)
```

### ###DCA

```
library(vegan)
```

###we set up the csv file with a matrix that is comprised of sites as rows and columns as species.

```
mydata <- read.csv("FileName.csv", header=TRUE,
row.names=1, sep= ",")
```

```
mydata.ra <- decorana(mydata, ira=1)
```

###perform DCA on the data set, preceded by transformations to correct for differences in sample size/abundance among taxa

```
mydata.t1 <- decostand(mydata, "total")
```

```
mydata.t2 <- decostand(mydata.t1, "max")
```

```
mydata.t2.dca <- decorana(mydata.t2)
```

```
mydata.t2.dca.DW <- decorana(mydata.t2, iweigh=1)
```

```
####view sample scores for the first 4 axes produced by decorana
mydata.t2.dca$rproj
```

```
####view summary of DCA results
summary(mydata.t2.dca)
```

```
####extract scores from decorana output
mydata.t2.dca.taxoncores <- scores(mydata.t2.dca,
                                   display=c("species"), choices=c(1,2))
```

```
####plot scores with customization of symbols and labels
plot(mydata.t2.dca, xlim=c(-4,4), ylim=c(-4,4), display=c("none"), cols=c(1,2))
```

```
points(mydata.t2.dca, display=c("sites"), choices=1:2,
       pch=3, col="green")
```

```
text(mydata.t2.dca, display=c("species"), choices=1:2,
     cex=0.5)
```

```
####plot sample scores
```

```
####note, the file below is comprised of two columns, one for the site name and the second column for the
age bin (i.e., Paleocene, PETM Body, PETM Rec and Eocene)
```

```
myFacies <- read.csv(file="FileName_Order.csv", header=TRUE,
                    row.names=1, sep=",")
```

```
t2.dca.scores <- mydata.t2.dca$rproj[, 1:2]
```

```
t2.dca.scoresFacies <- data.frame(t2.dca1=t2.dca.scores[,1],
                                 t2.dca2=t2.dca.scores[,2], facies=myFacies)
```

```
attach(t2.dca.scoresFacies)
```

```
plot(t2.dca1, t2.dca2, xlim=c(-1,5), ylim=c(-1,3), type="n", xlab="DCA Axis 1",
     ylab="DCA Axis 2", main="Sample Scores, by Age")
```

```
points(t2.dca1[facies=="Paleocene"], t2.dca2[facies=="Paleocene"],
      col="royalblue2", pch=16)
points(t2.dca1[facies=="PETM Body"], t2.dca2[facies=="PETM Body"],
      col="red3", pch=16)
points(t2.dca1[facies=="PETM Rec"], t2.dca2[facies=="PETM Rec"],
      col="sienna1", pch=16)
points(t2.dca1[facies=="Eocene"], t2.dca2[facies=="Eocene"],
      col="seagreen3", pch=16)
```

### **###sample richness and evenness**

```
library(ggplot2)
library(iNEXT)
library(devtools)
```

####to calculate time bin richness the observed species abundances was arranged as a species (in rows) by time bin (in columns) matrix. Note, we added additional 0's to make all assemblages have the same number of rows. Refer to Chao et al. (2022) for additional detail on data preparation.

```
data = read.csv("FileName.csv")
```

```
str(data)
```

```
m <- c(1, 5, 50)
```

```
iNEXT(data, q=0, datatype="abundance", size=m)
```

####to plot species richness

```
out <- iNEXT(data, q=c(0), datatype="abundance")
```

```
ggiNEXT(out, type=1, se=TRUE, facet.var="None", color.var="Assemblage", grey=FALSE)
```

####to calculate rarefied time bin richness the observed species abundances was arranged as a species (in rows) by time bin (in columns) matrix. Note, we added additional 0's to make all assemblages have the same number of rows. Refer to Chao et al. (2022) for additional detail on data preparation.

```
data = read.csv("FileName.csv")
```

```
str(data)
```

```
iNEXT(data, q=0, datatype="abundance", size=NULL, endpoint = 1100, nboot = 1000)
```

####to calculate sample richness the observed species abundances was arranged as a species (in rows) by site (in columns) matrix. Note, we added additional 0's to make all assemblages have the same number of rows. Refer to Chao et al. (2022) for additional detail on data preparation.

```
data = read.csv("FileName.csv")
```

```
str(data)
```

```
m <- c(1, 5, 50)
```

```
iNEXT(data, q=0, datatype="abundance", size=m)
```

####to calculate the Shannon Index the observed species abundances were arranged as a species (in rows) by site (in columns) matrix. Note, we added additional 0's to make all assemblages have the same number of rows. Refer to Chao et al. (2022) for additional detail on data preparation.

```
data = read.csv("FileName.csv")
```

```
str(data)
```

```
iNEXT(data, q=1, datatype="abundance", size=NULL, endpoint = 50)
```

### **###species accumulation**

```
library(vegan)
```

###we arranged the data as a site (in rows) by species (in columns) matrix. Note, we prepared 4 cvs files, with 1 for each time bin of interest.

```
data1 = read.csv("FileName.csv")
```

```
data2 <- data1[,-c(1)]
```

```
sp1 <- specaccum(data2, "exact")
```

```
sp2 <- specaccum(data2, "random")
```

###estimators

```
sp1_pool <- poolaccum(data2, permutations = 1000)
```

```
plot(sp1_pool)
```

###plot

```
plot(sp1_pool, log = "x", xlim=c(1,20), ylim=c(1,80), display = "boot", col = "blue", auto.key = FALSE,
     grid = F, strip = FALSE, xlab = "Sample",
     par.settings = list(axis.line = list(col = 0)),
     scales = list(col=1, tck=c(1,0)),
     panel = function(...){
       lims <- current.panel.limits()
       panel.xyplot(...)
       panel.abline(h=lims$ylim[1], v=lims$xlim[1],)
     })
```

###repeat for each time bin of interest

### **###ANOVA**

###the data should be arranged with two columns, the first column should be the site time bin (i.e., Paleocene, PETM, Recovery or Eocene) and the second should contain either 1) the species richness value, or 2) the evenness value, for that same site. Note, you should prepare 2 .cvs files, with 1 for species richness and 1 for evenness.

```
library(tidyverse)
```

```
library(ggplot2)
```

```
dat = read.csv("FileName.csv")
```

```
str(dat)
```

###analysis for richness

```
oneway.test(Richness ~ Epoch,
            data = dat,
            var.equal = TRUE)
```

###analysis for evenness

```
oneway.test(Shannon ~ Epoch,
            data = dat,
            var.equal = TRUE)
```

### **###ANOSIM**

```
library(vegan)
```

###we set up the data matrix with site as the first column, age as the second column, the environment as the third column and the remaining columns as the species. The rows should contain the abundance data for each species expressed as a percentage of the sample.

```
data = read.csv("FileName", header = TRUE)
```

```
###analysis
```

```
com = data[,4:ncol(data)]
```

```
m_com = as.matrix(com)
```

```
ano = anosim(m_com, data$Age, distance = "bray", permutations = 9999)  
ano
```

```
ano = anosim(m_com, data$Enviro, distance = "bray", permutations = 9999)  
ano
```

### **###CCA**

```
library(vegan)
```

###we prepared three separate csv files. The first csv file contained the species occurrence data presented as columns. Note, the first row contained the species names. The second csv file contained the environmental data presented as 4 columns, these being the %TOC,  $\delta^{13}\text{C}$ , %sand and silt:clay ratio. The third csv file contains the spatial data presented as 2 columns, these being the latitude and longitude for each site. Note, ensure the order of samples is consistent in all three csv files.

```
spe <- read.csv("FileName.csv")
```

```
env <- read.csv("FileName.csv")
```

```
spatial <- read.csv("FileName.csv")
```

```
### Apply log+1 transformation to your species occurrences data (spe matrix)
```

```
spellog <- decostand(spe, "log")
```

```
### Perform CCA. Note, specify that spe (species distribution matrix) is explained by env (environmental matrix)
```

```
ccamodel <- cca(spe~., env)
```

```
###Perform partial CCA
```

```
envspatial<-cbind(env,spatial)
```

```
nams <- names(envspatial)
```

```
partialccamodel <- formula(paste("spe ~", paste(nams[1: (length(envspatial)-(length(spatial)) )], collapse = "
+ "),"+ Condition(", paste(nams[(length(envspatial)-(length(spatial)-1) ):length(envspatial)], collapse
="+"),")"))
```

```
partialccamodel<-cca(partialccamodel, envspatial)
```

```
finalmodel<- ordistep(ccamodel, scope=formula(ccamodel))
```

```
vif.cca(finalmodel)
```

```
partialccamodel <- formula(paste("spe ~", paste(nams[1: (length(envspatial)-(length(spatial)) )], collapse = "
+ "),"+ Condition(", paste(nams[(length(envspatial)-(length(spatial)-1) ):length(envspatial)], collapse
="+"),")"))
```

```
simplemodel<-cca(partialccamodel, envspatial)
```

```
finalmodelpartial<- ordistep(simplemodel, scope=formula(partialccamodel))
```

```
vif.cca(finalmodelpartial)
```

```
finalmodelpartial
```

```
###plot
```

```
plot(finalmodel, xlim=c(-2,2), ylim=c(-2,2), display=c("sp","cn","wa"))
```

```
points(finalmodel, display=c("site"), choices=1:2,pch=3, col="green")
```

### **###NMDS**

```
library(scales)
```

```
library(vegan)
```

```
library(plotly)
```

```
library(stringr)
```

###we set up the csv file with a matrix that is comprised of sites as columns and rows as species. Note, the first five rows were designated (in ascending order): Site, Lithology, Age.time (i.e., the age in Ma), age (i.e., the Epoch), Depth.m (in this instance the age in Ma), followed by the list of species.

```
data.in = read.csv("FileName.csv", header = T, row.names = 1)
```

```
env <- data.in[1:4, ]
```

```
env <- data.frame (t(env))
```

```
data.in <- data.in[-c(1:4), ]
```

```
for(i in 1:ncol(data.in) ){
  data.in[,i] <- as.numeric(as.character(data.in[,i]) )
}
```

```
site.data <- t(data.in)
```

```
###Choose Bray-Curtis
```

```
MDS.out <- metaMDS(site.data, distance="bray")
```

```
####Assign colors
Pal.col <- "blue"
PETM.col <- "red"
Rec.col <- "orange"
Eo.col <- "green"

site.data.col <- ifelse(env$Age == "Paleocene", Pal.col, ifelse(env$Age == "PETM Body", PETM.col,
ifelse(env$Age == "PETM Rec", Rec.col, Eo.col)))

ordiplot(MDS.out, display = "sites")
points(MDS.out$points[, "MDS1"], MDS.out$points[, "MDS2"], type = "b", col = site.data.col, pch = 16)
legend("topleft", pch = 5, legend = c( "Eo","PETM", "Rec", "Pal"), col = rev(c(Pal.col, PETM.col, Rec.col,
Eo.col)) )
```