

## Data S1

R script:

```
# PRC: Principal Response Curve
if(!require(vegan)){install.packages("vegan"); library(vegan)}
if(!require(quantreg)){install.packages("quantreg"); library(quantreg)}

# species data loading (abundance data are ln(0.1x+1) transformed)
PRCdata <-read.table("C:/Users/aauber/Desktop/Papier PRC/PRCtest.txt",header=TRUE,
sep="", na.strings="NA", dec=".", strip.white=TRUE)
head(PRCdata)
nbofyears<-length(levels(as.factor(PRCdata$year)))
nbofsites<-length(levels(as.factor(PRCdata$site)))
siteslabels<-seq(1,nbofsites,1);PRCdata$year;sitesfunction<-
gl(nbofsites,nbofyears,labels=siteslabels)
time_period<-factor(PRCdata$time_period)

# cleaning table of NA items
na_lines <-which(is.na(PRCdata[,6]));length(na_lines)
species<-PRCdata[-na_lines,-c(1:5)];length(species)
time<-time_period[-na_lines];length(time)
sites<-sitesfunction[-na_lines];length(sites)

# PRC
PRCresults <- prc(species,time,sites)
PRCresults
plot(PRCresults,ylim=c(-1,1))

# test for PRC significance
testPRC<-anova.cca(PRCresults,step=1000)
testPRC

# significance of the difference between the two compared periods in terms of the community'
structure (permutation for multiple testing correction)
# cbind between tables, each table being a sampling site
listsite<-seq(1,nrow(PRCdata),nbofyears)
for (i in listsite) {
  if (i==listsite[1]) {tablo<-PRCdata[seq(i,i+23,1),]}
  else {tablo<-cbind(tablo,PRCdata[seq(i,i+23,1),]}
}
head(tablo)

# creating a matrix with random line numbers (for outer permutations)
# defining the number of permutations
nbpermut<-1000
# defining the number of species
nbsp<-ncol(PRCdata)-5
matricenum<-matrix(0,nbofyears,nbpermut)
listcol<-seq(1,nbpermut,1)
for (i in listcol) { matricenum[,i]<-sample(1:nrow(tablo)) }
matricenum<-as.data.frame(matricenum)

# creating the empirical null distribution
```

```

listpermut<-seq(1,nbpermut,1)
listsites<-seq(6,ncol(tablo),nbsp+5)
for (i in listpermut) {
  if (i==listpermut[1]){
    for (j in listsites) {
      if (j==listsites[1]) { tabpermutee<-
cbind(tablo[,seq(1,5,1)],tablo[matricenum[,i],6:ncol(tablo)]);soustable<-tabpermutee[,j];lignok<-
which(is.na(soustable)==FALSE);spe.rda<-rda(tabpermutee[lignok,seq(j,j+(nbsp-
1),1)],as.numeric(tablo[lignok,2]));testresult<-anova.cca(spe.rda,step=nbpermut); pval<-
as.numeric(testresult$Pr[1])}
      else { soustable<-tabpermutee[,j];lignok<-
which(is.na(soustable)==FALSE);spe.rda<-rda(tabpermutee[lignok,seq(j,j+(nbsp-
1),1)],as.numeric(tablo[lignok,2]));testresult<-anova.cca(spe.rda,step=nbpermut); pval<-
rbind(pval,as.numeric(testresult$Pr[1]))}
      allperm<-min(pval)
    }
  }
  else {
    for (j in listsites) {
      if (j==listsites[1]) { tabpermutee<-
cbind(tablo[,seq(1,5,1)],tablo[matricenum[,i],6:ncol(tablo)]);soustable<-tabpermutee[,j];lignok<-
which(is.na(soustable)==FALSE);spe.rda<-rda(tabpermutee[lignok,seq(j,j+(nbsp-
1),1)],as.numeric(tablo[lignok,2]));testresult<-anova.cca(spe.rda,step=nbpermut); pval<-
as.numeric(testresult$Pr[1])}
      else { soustable<-tabpermutee[,j];lignok<-
which(is.na(soustable)==FALSE);spe.rda<-rda(tabpermutee[lignok,seq(j,j+(nbsp-
1),1)],as.numeric(tablo[lignok,2]));testresult<-anova.cca(spe.rda,step=nbpermut); pval<-
rbind(pval,as.numeric(testresult$Pr[1]))}
    }
    allperm<-cbind(allperm,min(pval))
  }
}
allperm

# p-values computation from non permuted table
for (j in listsites) {
  if (j==listsites[1]) { soustable<-tablo[,j];lignok<-
which(is.na(soustable)==FALSE);spe.rda<-rda(tablo[lignok,seq(j,j+(nbsp-
1),1)],as.numeric(tablo[lignok,2]));testresult<-anova.cca(spe.rda,step=nbpermut); obspval<-
as.numeric(testresult$Pr[1])}
  else { soustable<-tablo[,j];lignok<-
which(is.na(soustable)==FALSE);spe.rda<-rda(tablo[lignok,seq(j,j+(nbsp-
1),1)],as.numeric(tablo[lignok,2]));testresult<-anova.cca(spe.rda,step=nbpermut); obspval<-
rbind(obspsval,as.numeric(testresult$Pr[1]))}
}
obspsval

# for each sampling site : Comparison between the observed pvalue and the
distribution of minimum permuted pvalues

```

```

listsites2<-seq(1,nrow(ospval),1)
for (i in listsites2) {
  if (i==listsites2[1]) {significance<-length(allperm[allperm <
ospval[i,]]/length(allperm))
  else {significance<-
rbind(significance,length(allperm[allperm < ospval[i,]]/length(allperm))}
  }
  rownames(significance)<-
levels(as.factor(PRCdata$site));colnames(significance)<-c("significance")
significance<=0.05

```

```

# Regression between Cdt and distance to the nearest coast
distcote <-read.table("C:/Users/aauber/Desktop/Papier PRC/version
3/reg_distcote_cdt.txt",header=TRUE, sep=" ", na.strings="NA", dec=".", strip.white=TRUE)
head(distcote)
cor.test(distcote$dist,distcote$cdt)
lm(distcote$cdt~distcote$dist)
# quantile regressions
fit1<-rq(distcote$cdt~distcote$dist,tau=(0.1))
summary(fit1)
#smooth curve
smooth<-scatter.smooth(distcote$dist, distcote$cdt,pch=19,xlab="Distance to the nearest
coast (km)",ylab="Cdt")
# first decile
fitfirst<-rq(distcote$cdt~distcote$dist,tau=0.1)
summary(fitfirst,se="boot")
# last decile
fitlast<-rq(distcote$cdt~distcote$dist,tau=0.9)
summary(fitlast,se="boot")
#first decile
abline(-0.52248,0.00136)
#last decile
abline(-0.1824,-0.00049)

```

```

# Regression between Cdt and longitude
# quantile regressions
fit1<-rq(distcote$cdt~distcote$lon,tau=(0.1))
summary(fit1)
#smooth curve
smooth<-scatter.smooth(distcote$lon, distcote$cdt,pch=19,xlab="Longitude
(°)",ylab="Cdt")
# first decile
fitfirst<-rq(distcote$cdt~distcote$lon,tau=0.1)
summary(fitfirst,se="boot")
# last decile
fitlast<-rq(distcote$cdt~distcote$lon,tau=0.9)
summary(fitlast,se="boot")
#first decile
abline(-0.52711,0.01028)
#last decile
abline(-0.19920,-0.01205)

```

```
# Regression between Cdt and latitude
# quantile regressions
fit1<-rq(distcote$cdt~distcote$lat,tau=(0.1))
summary(fit1)
#smooth curve
smooth<-scatter.smooth(distcote$lat, distcote$cdt,pch=19,xlab="Latitude (°)",ylab="Cdt")
# first decile
fitfirst<-rq(distcote$cdt~distcote$lat,tau=0.1)
summary(fitfirst,se="boot")
# last decile
fitlast<-rq(distcote$cdt~distcote$lat,tau=0.9)
summary(fitlast,se="boot")
#first decile
abline(-2.12649,0.03215)
#last decile
abline(0.71884,-0.01830)
```