Script Diversity PA-Occupancy

Gonzalo

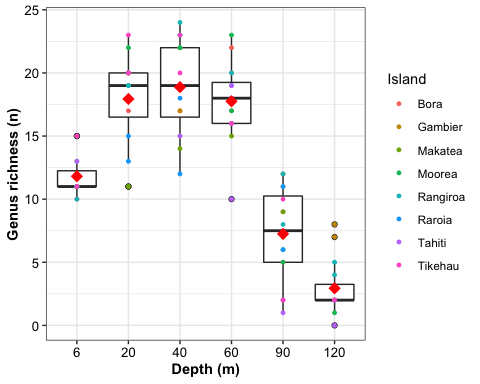
1/19/2021

# rm (list = ls())  
# Set working directory etc.  
setwd("~/Documents/AAASea\_Science/AAA\_PhD\_Thesis/Photoquadrats/Data\_Scripts/Dataframes")  
PA\_df <- read.csv(file = "photoquad\_sppcount\_DEEPHOPE\_genus\_form.csv", header = T, dec = ".", sep = ",", row.names = 1)  
  
PA\_df$Island <- gsub("Mangareva", "Gambier", PA\_df$Island)  
PA\_df <- subset (PA\_df,Coral\_genus!="NA\_Coral")

All stats below come from PA\_df. I re-start every time for simplicity

## Generic Richness profile

# How many quadrats  
nb\_genera <- ddply(PA\_df, ~ Island + Island\_Site + Depth + Coral\_genus ,function(x){  
 c(nobserv=nrow(x)) })  
  
  
# Transform values bigger than 30, because several "Porites branching" "Porites massive"....  
nb\_genera$nobserv[nb\_genera$nobserv>30] <- 30  
  
# Generic richness, unique number of genera  
# length (unique (nb\_genera$Coral\_genus))  
nb\_genera <- ddply(nb\_genera, ~ Island + Island\_Site + Depth ,function(x){  
 c(nobserv=nrow(unique (x))) })  
  
# Complete rows for all Island, site, Depth, nbgenera = 0  
nb\_genera <- nb\_genera %>% complete( Island,Island\_Site,Depth,fill = list(nobserv = 0))  
  
# Transform depth as a Qualitative variable   
nb\_genera$Depth <- as.factor(as.character(nb\_genera$Depth))  
nb\_genera$Depth = factor(nb\_genera$Depth,levels = c ("6", "20", "40", "60", "90", "120"))  
  
# ggplot with islands  
ggplot(nb\_genera, aes(x=Depth, y=nobserv)) +  
 geom\_boxplot() + geom\_point(aes (colour = Island),size = 1) +   
 stat\_summary(fun=mean, geom="point", shape=18, color="red", size=4) +   
 theme\_bw() + ylab ("Genus richness (n)") + xlab ("Depth (m)") +  
 theme(plot.title = element\_text(hjust=0.5, size=12, face="bold"),  
 axis.text = element\_text(size=10, colour="black"),  
 axis.title = element\_text(size=11, face="bold", colour="black"))



## Mid-domain effect

Ideally represent with Genus richness profile

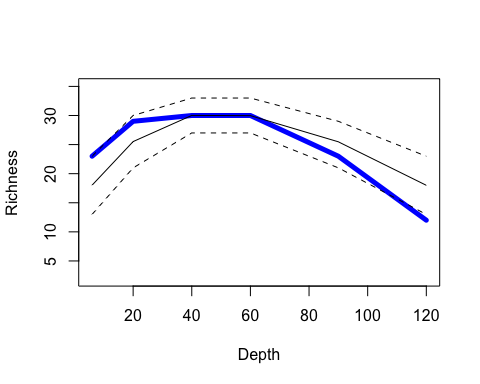
all\_short <- data.matrix(dcast(PA\_df, Coral\_genus ~ Depth))

## Warning in dcast(PA\_df, Coral\_genus ~ Depth): The dcast generic in  
## data.table has been passed a data.frame and will attempt to redirect to  
## the reshape2::dcast; please note that reshape2 is deprecated, and this  
## redirection is now deprecated as well. Please do this redirection yourself like  
## reshape2::dcast(PA\_df). In the next version, this warning will become an error.

## Using 'Coral\_form' as value column. Use 'value.var' to override

## Aggregation function missing: defaulting to length

all\_short[which(all\_short > 0)] = 1  
  
all\_short <- t(all\_short[,2:7])  
  
rm <- rangemod1d(all\_short,cohesion = TRUE,var = NULL,rsize = "observed",reps = 10000)  
ric <- rowSums(all\_short)   
depth <- c(6,20,40,60,90,120)  
  
plot(depth, ric, ylim = c(2,35), xlab = "Depth", ylab="Richness", type="n", main = "")  
lines(depth, ric, lwd = 5, col="blue")  
lines(depth, rm$mod.rich, lwd=1)  
lines(depth, rm$q2.5, lty = 2)  
lines(depth, rm$q97.5, lty = 2)



print (rm)

## mod.rich mod.sd q2.5 q97.5  
## 6 18.0221 2.522310 13 23  
## 20 25.5177 2.086701 21 30  
## 40 30.0127 1.333910 27 33  
## 60 29.9760 1.323707 27 33  
## 90 25.4740 2.089392 21 29  
## 120 17.9975 2.476511 13 23

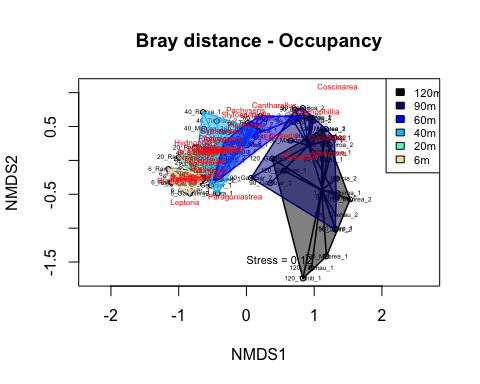
## NMDS

Question 1: I cannot work with all Quadrats if we keep the PA-Occuppancy “Frequency = Nb Quadrats with genus / Total Nb Quadrats”

resume\_df <- ddply(PA\_df, ~ Island + Island\_Site + Depth + Coral\_genus ,function(x){  
 c(nobserv=nrow(x)) })  
  
melt\_temp = melt(resume\_df, id=c("Island","Island\_Site","Depth","Coral\_genus"), measure.vars="nobserv", na.rm=FALSE)  
  
### Not the PA, the occupancy "Frequency"   
# Measure the Occupancy "Frecuency" f = (nº Quadrats gen (n) present / total n Quadrats)  
melt\_temp$value[melt\_temp$value > 30] <- 30  
melt\_temp$value <- melt\_temp$value / 30  
  
cast\_temp = dcast(melt\_temp, Depth + Island + Island\_Site ~ Coral\_genus, mean, add.missing = T)  
cast\_temp[is.na(cast\_temp)] <- 0  
  
cast\_temp$Island <- factor(cast\_temp$Island, levels = c("Bora","Makatea","Gambier","Moorea","Rangiroa","Raroia","Tahiti","Tikehau"))  
cast\_database <- cast\_temp  
  
# Preparing them for format matrix  
cast\_temp$ID<- with(cast\_temp, paste0(Depth, sep = "\_", Island, sep = "\_", Island\_Site))  
rownames (cast\_temp) <- cast\_temp$ID  
# Re get database  
final <- melt (cast\_temp, id = c ("Island","Island\_Site","Depth"), measure.vars=c(names(cast\_temp[,-c(1,2,3,39)])  
), na.rm = FALSE)  
  
cast\_temp <- cast\_temp[,-c(1,2,3,38)]  
  
  
# columns where sum is 0  
cast\_temp <- cast\_temp %>% select\_if(colSums(.) != 0)  
# rows where sum is 0  
cast\_temp <- cast\_temp[rowSums(cast\_temp[,])>0, ]  
  
# View (cast\_temp)  
  
# As I am working with the occupancy, "Frequency"  
PA\_NMDS <- metaMDS(cast\_temp, k=2, trymax = 1000, distance = "bray")

## Run 0 stress 0.1173584   
## Run 1 stress 0.1168841   
## ... New best solution  
## ... Procrustes: rmse 0.02098676 max resid 0.1577118   
## Run 2 stress 0.1217939   
## Run 3 stress 0.1201403   
## Run 4 stress 0.1196843   
## Run 5 stress 0.1217911   
## Run 6 stress 0.1252715   
## Run 7 stress 0.1176962   
## Run 8 stress 0.1234314   
## Run 9 stress 0.1265737   
## Run 10 stress 0.1169164   
## ... Procrustes: rmse 0.005909167 max resid 0.05495875   
## Run 11 stress 0.1177527   
## Run 12 stress 0.1169164   
## ... Procrustes: rmse 0.00590577 max resid 0.05494838   
## Run 13 stress 0.1290843   
## Run 14 stress 0.124781   
## Run 15 stress 0.117422   
## Run 16 stress 0.1248152   
## Run 17 stress 0.1169164   
## ... Procrustes: rmse 0.005912729 max resid 0.05496965   
## Run 18 stress 0.1255359   
## Run 19 stress 0.1264576   
## Run 20 stress 0.1261422   
## Run 21 stress 0.1250232   
## Run 22 stress 0.1210752   
## Run 23 stress 0.117229   
## ... Procrustes: rmse 0.0129144 max resid 0.1108399   
## Run 24 stress 0.1214901   
## Run 25 stress 0.1192924   
## Run 26 stress 0.1205407   
## Run 27 stress 0.1187523   
## Run 28 stress 0.1178768   
## Run 29 stress 0.1176822   
## Run 30 stress 0.1179354   
## Run 31 stress 0.1227163   
## Run 32 stress 0.1178581   
## Run 33 stress 0.1197616   
## Run 34 stress 0.1181112   
## Run 35 stress 0.1252575   
## Run 36 stress 0.11718   
## ... Procrustes: rmse 0.01236403 max resid 0.09564481   
## Run 37 stress 0.117422   
## Run 38 stress 0.1202357   
## Run 39 stress 0.1203734   
## Run 40 stress 0.1206779   
## Run 41 stress 0.1172847   
## ... Procrustes: rmse 0.01316286 max resid 0.1107883   
## Run 42 stress 0.1181077   
## Run 43 stress 0.1219453   
## Run 44 stress 0.1248307   
## Run 45 stress 0.1172982   
## ... Procrustes: rmse 0.01081009 max resid 0.09650915   
## Run 46 stress 0.1200178   
## Run 47 stress 0.1275252   
## Run 48 stress 0.1171806   
## ... Procrustes: rmse 0.01236618 max resid 0.09565305   
## Run 49 stress 0.1276059   
## Run 50 stress 0.1179354   
## Run 51 stress 0.1181112   
## Run 52 stress 0.1181077   
## Run 53 stress 0.117587   
## Run 54 stress 0.1234017   
## Run 55 stress 0.1205403   
## Run 56 stress 0.1195218   
## Run 57 stress 0.1181942   
## Run 58 stress 0.1181077   
## Run 59 stress 0.1271483   
## Run 60 stress 0.1199283   
## Run 61 stress 0.1247952   
## Run 62 stress 0.1187264   
## Run 63 stress 0.1206837   
## Run 64 stress 0.1281629   
## Run 65 stress 0.1253223   
## Run 66 stress 0.1168842   
## ... Procrustes: rmse 4.621785e-05 max resid 0.0003494222   
## ... Similar to previous best  
## \*\*\* Solution reached

# Prepare to plot  
treat=c(rep("6M",16),rep("20M",16), rep("40M",16), rep("60M",16), rep("90M",16), rep("120M",15)) # Alphabetic order for colours  
colours <- c("black", "aquamarine2","deepskyblue","blue","wheat","navyblue")  
# 120, 20, 40, 60, 6, 90)  
  
par (mfrow =c(1,1))  
  
ordiplot(PA\_NMDS,type="n", choices = c(1,2)) # Check with numbers and see if I can get three axes  
orditorp(PA\_NMDS,display="sites",labels =T)  
orditorp(PA\_NMDS,display="sites",cex=0.4,air=0.01)  
ordispider(PA\_NMDS, groups=treat, col = colours,cex=0.6, lwd = 1.5)  
ordihull(PA\_NMDS,groups=treat,border = colours, col = colours, draw="polygon",label=F, lwd = 1.5)  
orditorp(PA\_NMDS,display="species",col="red",air=0.01, cex =0.5)  
legend(x="topright", y="top", legend=c ("120m","90m","60m", "40m","20m","6m"), col=c("black", "navyblue","blue","deepskyblue","aquamarine2","wheat"), fill = c("black", "navyblue","blue","deepskyblue","aquamarine2","wheat"), pt.cex=0.8, cex = 0.70, horiz = F, text.width = 0.3, text.font = 20)  
title ("Bray distance - Occupancy")  
mysubtitle <- paste0("Stress = ", format(round(PA\_NMDS$stress, 2)))  
mtext(mysubtitle, side=1, line=-2, at=0, adj=0, cex=0.7)



## Beta.pair per depth

Same question as 1: I cannot work with all Quadrats if we keep the PA-Occuppancy “Frequency = Nb Quadrats with genus / Total Nb Quadrats”

# Resume to keep only coral genus  
resume\_df <- ddply(PA\_df, ~ Island + Island\_Site + Depth + Coral\_genus ,function(x){  
 c(nobserv=nrow(x)) })  
  
# Complete rows for all genera with Coral cover = 0 for all genera  
resume\_df <- resume\_df %>% complete( Island,Island\_Site,Depth, Coral\_genus,fill = list(nobserv = 0))  
  
# Prepare columns and rows (by now keeping depth)  
melt\_depth = melt(resume\_df, id=c("Depth","Island", "Island\_Site","Coral\_genus"), measure.vars="nobserv", na.rm=FALSE)  
  
### Not the PA, the occupancy "Frequency"  
# Measure the Occupancy "Frecuency" f = (nº Quadrats gen (n) present / total n Quadrats)  
melt\_depth$value[melt\_depth$value > 30] <- 30  
melt\_depth$value <- melt\_depth$value / 30  
  
melt\_depth$ID<- with(melt\_depth, paste0(Island, sep = "\_", Island\_Site))  
  
cast\_depth = dcast(melt\_depth, Depth + Coral\_genus ~ ID, mean, add.missing = T)  
# cast\_depth = dcast(melt\_depth, Coral\_genus ~ Depth, mean, add.missing = T)  
  
### For 6m #View(Depth\_1) - Necessary rows as sites and columns as species (genera)  
Depth\_1 <- filter (cast\_depth, Depth == 6)  
rownames (Depth\_1) <- Depth\_1$Coral\_genus  
Depth\_1 <- subset (Depth\_1, select = - c(Depth, Coral\_genus))  
  
Depth\_1\_Beta <- as.data.frame (t(Depth\_1))  
#I think, I need to delete columns where all genus are 0   
Depth\_1\_Beta <- Depth\_1\_Beta %>% select\_if(colSums(.) != 0)   
  
### Working with Occupancy - "Frequency"  
# To do simply beta.pair, necessary to Really working with PA - Jaccard Sorensentransform to 0-1 . Otherwise, beta.pair.abund  
# Depth\_1\_Beta[Depth\_1\_Beta > 0] <- 1  
  
coral.matrices\_Depth\_1 <- beta.pair.abund(Depth\_1\_Beta, index.family = "bray")  
  
mean (coral.matrices\_Depth\_1$beta.bray.bal) # This is species replacement

## [1] 0.1756681

mean (coral.matrices\_Depth\_1$beta.bray.gra) # This is species nestedness

## [1] 0.1135596

mean (coral.matrices\_Depth\_1$beta.bray) # PArtially divided between species replacement and nestedness. Accounts for the two of them.

## [1] 0.2892277

# Turnover is higher than nestedness in the overall dissimilarity   
  
  
### For 20m #View(Depth\_2) - Necessary rows as sites and columns as species (genera)  
Depth\_2 <- filter (cast\_depth, Depth == 20)  
rownames (Depth\_2) <- Depth\_2$Coral\_genus  
Depth\_2 <- subset (Depth\_2, select = - c(Depth, Coral\_genus))  
  
Depth\_2\_Beta <- as.data.frame (t(Depth\_2))  
#I think, I need to delete columns where all genus are 0   
Depth\_2\_Beta <- Depth\_2\_Beta %>% select\_if(colSums(.) != 0)   
coral.matrices\_Depth\_2 <- beta.pair.abund(Depth\_2\_Beta, index.family = "bray")  
mean (coral.matrices\_Depth\_2$beta.bray)

## [1] 0.3268643

# So on with the remaining depths

## Betadisper

Here is where I am confused… I can make it for all depths together, considering groups as different depths. I obtain (1) the average distance to median (“b-dissimilarity” per depth ?), (2) anova and (3)permutest pair-wise differences between (depths)

Working with all depths, either I work with the mother matrix ## 1 ## or straight from beta.pair.abund matrix ## 2 ##

However, for making a betadisper per depth and check differences between islands is not possible. I need replicates. - Valeriano: you said using all quadrats per depth. However, impossible if we work with the index occupancy-frequency.

# betadisper using all depths

## 1 ## Betadisper from the entire mother dissimilarity matrix...  
resume\_df <- ddply(PA\_df, ~ Island + Island\_Site + Depth + Coral\_genus ,function(x){  
 c(nobserv=nrow(x)) })  
  
melt\_temp = melt(resume\_df, id=c("Island","Island\_Site","Depth","Coral\_genus"), measure.vars="nobserv", na.rm=FALSE)  
  
### Not the PA, the occupancy "Frequency"   
# Measure the Occupancy "Frecuency" f = (nº Quadrats gen (n) present / total n Quadrats)  
melt\_temp$value[melt\_temp$value > 30] <- 30  
melt\_temp$value <- melt\_temp$value / 30  
  
cast\_temp = dcast(melt\_temp, Depth + Island + Island\_Site ~ Coral\_genus, mean, add.missing = T)  
cast\_temp[is.na(cast\_temp)] <- 0  
  
cast\_temp$Island <- factor(cast\_temp$Island, levels = c("Bora","Makatea","Gambier","Moorea","Rangiroa","Raroia","Tahiti","Tikehau"))  
cast\_database <- cast\_temp  
  
# Preparing them for format matrix  
cast\_temp$ID<- with(cast\_temp, paste0(Depth, sep = "\_", Island, sep = "\_", Island\_Site))  
rownames (cast\_temp) <- cast\_temp$ID  
  
cast\_temp <- cast\_temp[,-c(1,2,3,38)]  
  
# columns where sum is 0  
cast\_temp <- cast\_temp %>% select\_if(colSums(.) != 0)  
# rows where sum is 0  
cast\_temp <- cast\_temp[rowSums(cast\_temp[,])>0, ]  
  
# Distance of the entire dissimilarity matrix  
dis <- vegdist(cast\_temp)  
# dis <- vegdist(cast\_depth)  
groups <- c(rep("6M",16),rep("20M",16), rep("40M",16), rep("60M",16), rep("90M",16), rep("120M",15))  
  
# Calculate multivariate dispersions  
mod <- betadisper(dis, groups)  
mod

##   
## Homogeneity of multivariate dispersions  
##   
## Call: betadisper(d = dis, group = groups)  
##   
## No. of Positive Eigenvalues: 46  
## No. of Negative Eigenvalues: 48  
##   
## Average distance to median:  
## 120M 20M 40M 60M 6M 90M   
## 0.4484 0.2243 0.3139 0.2988 0.1973 0.3682   
##   
## Eigenvalues for PCoA axes:  
## (Showing 8 of 94 eigenvalues)  
## PCoA1 PCoA2 PCoA3 PCoA4 PCoA5 PCoA6 PCoA7 PCoA8   
## 10.3503 3.8844 1.8291 1.1541 0.9625 0.8460 0.8331 0.7019

# Perform test  
anova(mod)

## Analysis of Variance Table  
##   
## Response: Distances  
## Df Sum Sq Mean Sq F value Pr(>F)   
## Groups 5 0.66344 0.132688 12.319 4.369e-09 \*\*\*  
## Residuals 89 0.95862 0.010771   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Permutation test for F  
permutest(mod, pairwise = TRUE, permutations = 99)

##   
## Permutation test for homogeneity of multivariate dispersions  
## Permutation: free  
## Number of permutations: 99  
##   
## Response: Distances  
## Df Sum Sq Mean Sq F N.Perm Pr(>F)   
## Groups 5 0.66344 0.132688 12.319 99 0.01 \*\*  
## Residuals 89 0.95862 0.010771   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Pairwise comparisons:  
## (Observed p-value below diagonal, permuted p-value above diagonal)  
## 120M 20M 40M 60M 6M 90M  
## 120M 1.0000e-02 1.0000e-02 1.0000e-02 1.0000e-02 0.14  
## 20M 4.9842e-07 2.0000e-02 3.0000e-02 2.1000e-01 0.01  
## 40M 1.4382e-03 4.1189e-04 6.4000e-01 1.0000e-02 0.15  
## 60M 1.5721e-03 1.6867e-02 6.4992e-01 1.0000e-02 0.15  
## 6M 1.1270e-07 1.6338e-01 3.7132e-05 2.4265e-03 0.01  
## 90M 1.3494e-01 1.4882e-03 2.2532e-01 1.5627e-01 3.1772e-04

# Tukey's Honest Significant Differences  
(mod.HSD <- TukeyHSD(mod))

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = distances ~ group, data = df)  
##   
## $group  
## diff lwr upr p adj  
## 20M-120M -0.22407564 -0.33271954 -0.115431748 0.0000006  
## 40M-120M -0.13442625 -0.24307014 -0.025782355 0.0066442  
## 60M-120M -0.14957551 -0.25821940 -0.040931612 0.0017091  
## 6M-120M -0.25106612 -0.35971002 -0.142422229 0.0000000  
## 90M-120M -0.08021731 -0.18886121 0.028426584 0.2715053  
## 40M-20M 0.08964939 -0.01722782 0.196526602 0.1528626  
## 60M-20M 0.07450014 -0.03237707 0.181377345 0.3337544  
## 6M-20M -0.02699048 -0.13386769 0.079886728 0.9769481  
## 90M-20M 0.14385833 0.03698112 0.250735541 0.0023277  
## 60M-40M -0.01514926 -0.12202647 0.091727952 0.9984145  
## 6M-40M -0.11663987 -0.22351708 -0.009762665 0.0241257  
## 90M-40M 0.05420894 -0.05266827 0.161086148 0.6794363  
## 6M-60M -0.10149062 -0.20836783 0.005386592 0.0726500  
## 90M-60M 0.06935820 -0.03751901 0.176235405 0.4149241  
## 90M-6M 0.17084881 0.06397160 0.277726022 0.0001595

## 2 ## Betadisper from the beta.pair bray distance matrix   
  
resume\_df <- ddply(PA\_df, ~ Island + Island\_Site + Depth + Coral\_genus ,function(x){  
 c(nobserv=nrow(x)) })  
  
melt\_temp = melt(resume\_df, id=c("Island","Island\_Site","Depth","Coral\_genus"), measure.vars="nobserv", na.rm=FALSE)  
  
### Not the PA, the occupancy "Frequency"   
# Measure the Occupancy "Frecuency" f = (nº Quadrats gen (n) present / total n Quadrats)  
melt\_temp$value[melt\_temp$value > 30] <- 30  
melt\_temp$value <- melt\_temp$value / 30  
  
cast\_temp = dcast(melt\_temp, Depth + Island + Island\_Site ~ Coral\_genus, mean, add.missing = T)  
cast\_temp[is.na(cast\_temp)] <- 0  
  
cast\_temp$Island <- factor(cast\_temp$Island, levels = c("Bora","Makatea","Gambier","Moorea","Rangiroa","Raroia","Tahiti","Tikehau"))  
  
cast\_all\_depth <- cast\_temp  
cast\_all\_depth$ID<- with(cast\_all\_depth, paste0(Depth, sep = "\_", Island, sep = "\_", Island\_Site))  
rownames (cast\_all\_depth) <- cast\_all\_depth$ID  
  
cast\_all\_depth <- cast\_all\_depth[,-c(1,2,3,38)]  
# columns where sum is 0  
cast\_all\_depth <- cast\_all\_depth %>% select\_if(colSums(.) != 0)  
# rows where sum is 0  
cast\_all\_depth <- cast\_all\_depth[rowSums(cast\_all\_depth[,])>0, ]  
  
# Measure pair.abund  
coral.matrices\_Depth <- beta.pair.abund(cast\_all\_depth, index.family = "bray")  
mean (coral.matrices\_Depth$beta.bray)

## [1] 0.6512673

Beta\_Depth <- dist (coral.matrices\_Depth$beta.bray)  
  
mod\_Depth <- betadisper (Beta\_Depth,groups)  
mod\_Depth

##   
## Homogeneity of multivariate dispersions  
##   
## Call: betadisper(d = Beta\_Depth, group = groups)  
##   
## No. of Positive Eigenvalues: 94  
## No. of Negative Eigenvalues: 0  
##   
## Average distance to median:  
## 120M 20M 40M 60M 6M 90M   
## 1.4711 0.6538 0.9755 1.0815 0.5984 1.1804   
##   
## Eigenvalues for PCoA axes:  
## (Showing 8 of 94 eigenvalues)  
## PCoA1 PCoA2 PCoA3 PCoA4 PCoA5 PCoA6 PCoA7 PCoA8   
## 309.245 73.777 16.565 8.623 5.319 4.343 4.171 3.059

# Perform test  
anova(mod\_Depth)

## Analysis of Variance Table  
##   
## Response: Distances  
## Df Sum Sq Mean Sq F value Pr(>F)   
## Groups 5 8.4515 1.69030 8.0429 2.706e-06 \*\*\*  
## Residuals 89 18.7043 0.21016   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Permutation test for F  
permutest(mod\_Depth, pairwise = TRUE, permutations = 99)

##   
## Permutation test for homogeneity of multivariate dispersions  
## Permutation: free  
## Number of permutations: 99  
##   
## Response: Distances  
## Df Sum Sq Mean Sq F N.Perm Pr(>F)   
## Groups 5 8.4515 1.69030 8.0429 99 0.01 \*\*  
## Residuals 89 18.7043 0.21016   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Pairwise comparisons:  
## (Observed p-value below diagonal, permuted p-value above diagonal)  
## 120M 20M 40M 60M 6M 90M  
## 120M 1.0000e-02 2.0000e-02 8.0000e-02 1.0000e-02 0.24  
## 20M 1.9495e-06 1.0000e-02 1.0000e-02 4.0000e-01 0.02  
## 40M 2.4746e-03 7.6623e-04 5.6000e-01 1.0000e-02 0.33  
## 60M 5.7767e-02 8.2197e-03 5.1561e-01 1.0000e-02 0.72  
## 6M 6.2481e-07 3.9744e-01 1.2104e-04 3.2157e-03 0.01  
## 90M 2.0857e-01 7.9267e-03 2.9773e-01 6.7014e-01 3.6994e-03

# Tukey's Honest Significant Differences  
(mod.HSD <- TukeyHSD(mod\_Depth))

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = distances ~ group, data = df)  
##   
## $group  
## diff lwr upr p adj  
## 20M-120M -0.81730996 -1.29721165 -0.33740827 0.0000483  
## 40M-120M -0.49562801 -0.97552970 -0.01572632 0.0387843  
## 60M-120M -0.38961263 -0.86951432 0.09028906 0.1800742  
## 6M-120M -0.87273978 -1.35264147 -0.39283809 0.0000123  
## 90M-120M -0.29072381 -0.77062550 0.18917788 0.4936856  
## 40M-20M 0.32168195 -0.15041594 0.79377984 0.3592659  
## 60M-20M 0.42769733 -0.04440056 0.89979522 0.0986751  
## 6M-20M -0.05542982 -0.52752771 0.41666807 0.9993623  
## 90M-20M 0.52658615 0.05448826 0.99868404 0.0197031  
## 60M-40M 0.10601539 -0.36608251 0.57811328 0.9863411  
## 6M-40M -0.37711177 -0.84920966 0.09498612 0.1944627  
## 90M-40M 0.20490420 -0.26719369 0.67700209 0.8034535  
## 6M-60M -0.48312715 -0.95522505 -0.01102926 0.0417545  
## 90M-60M 0.09888882 -0.37320907 0.57098671 0.9900463  
## 90M-6M 0.58201597 0.10991808 1.05411386 0.0069271

# betadisper separate per depths

As I said above, this really doesn’t work. I think I need replicates. Cannot use quadrats as replicates because we compute Occupancy - “Frequency” Either working from mother matrix (Depth\_1\_Beta) or from distance matrix from bray.pair.abund of Depth\_1 (=6m)

### Depth\_1  
beta\_Depth\_1 <- dist (coral.matrices\_Depth\_1$beta.bray)  
dis1 <- vegdist(Depth\_1\_Beta, method = "bray")  
  
# Only with Depth 1  
groups <- rownames (Depth\_1\_Beta)  
# groups <- c(rep("6M",16))  
  
mod1 <- betadisper (dis1,groups)  
# mod1 <- betadisper (beta\_Depth\_1,groups)  
mod1

##   
## Homogeneity of multivariate dispersions  
##   
## Call: betadisper(d = dis1, group = groups)  
##   
## No. of Positive Eigenvalues: 10  
## No. of Negative Eigenvalues: 5  
##   
## Average distance to median:  
## Bora\_1 Bora\_2 Gambier\_1 Gambier\_2 Makatea\_1 Makatea\_2 Moorea\_1   
## 0 0 0 0 0 0 0   
## Moorea\_2 Rangiroa\_1 Rangiroa\_2 Raroia\_1 Raroia\_2 Tahiti\_1 Tahiti\_2   
## 0 0 0 0 0 0 0   
## Tikehau\_1 Tikehau\_2   
## 0 0   
##   
## Eigenvalues for PCoA axes:  
## (Showing 8 of 15 eigenvalues)  
## PCoA1 PCoA2 PCoA3 PCoA4 PCoA5 PCoA6 PCoA7 PCoA8   
## 0.21734 0.14794 0.12229 0.07336 0.05289 0.03252 0.02995 0.01867

# Perform test  
anova(mod1)

## Analysis of Variance Table  
##   
## Response: Distances  
## Df Sum Sq Mean Sq F value Pr(>F)  
## Groups 15 0 0   
## Residuals 0 0

# Permutation test for F  
permutest(mod1, pairwise = TRUE, permutations = 99)

##   
## Permutation test for homogeneity of multivariate dispersions  
## Permutation: free  
## Number of permutations: 99  
##   
## Response: Distances  
## Df Sum Sq Mean Sq F N.Perm Pr(>F)  
## Groups 15 0 0 99   
## Residuals 0 0   
##   
## Pairwise comparisons:  
## (Observed p-value below diagonal, permuted p-value above diagonal)  
## Bora\_1 Bora\_2 Gambier\_1 Gambier\_2 Makatea\_1 Makatea\_2 Moorea\_1  
## Bora\_1   
## Bora\_2   
## Gambier\_1   
## Gambier\_2   
## Makatea\_1   
## Makatea\_2   
## Moorea\_1   
## Moorea\_2   
## Rangiroa\_1   
## Rangiroa\_2   
## Raroia\_1   
## Raroia\_2   
## Tahiti\_1   
## Tahiti\_2   
## Tikehau\_1   
## Tikehau\_2   
## Moorea\_2 Rangiroa\_1 Rangiroa\_2 Raroia\_1 Raroia\_2 Tahiti\_1 Tahiti\_2  
## Bora\_1   
## Bora\_2   
## Gambier\_1   
## Gambier\_2   
## Makatea\_1   
## Makatea\_2   
## Moorea\_1   
## Moorea\_2   
## Rangiroa\_1   
## Rangiroa\_2   
## Raroia\_1   
## Raroia\_2   
## Tahiti\_1   
## Tahiti\_2   
## Tikehau\_1   
## Tikehau\_2   
## Tikehau\_1 Tikehau\_2  
## Bora\_1   
## Bora\_2   
## Gambier\_1   
## Gambier\_2   
## Makatea\_1   
## Makatea\_2   
## Moorea\_1   
## Moorea\_2   
## Rangiroa\_1   
## Rangiroa\_2   
## Raroia\_1   
## Raroia\_2   
## Tahiti\_1   
## Tahiti\_2   
## Tikehau\_1   
## Tikehau\_2

## Tukey's Honest Significant Differences  
# (mod.HSD <- TukeyHSD(mod1))  
  
  
### Depth\_2  
beta\_Depth\_2 <- dist (coral.matrices\_Depth\_2$beta.bray)  
dis2 <- vegdist(Depth\_2\_Beta)  
  
# Only with Depth 1  
groups <- rownames (Depth\_2\_Beta)  
  
mod2 <- betadisper (beta\_Depth\_2,groups, )  
# mod2  
  
str(beta\_Depth\_1)

## 'dist' num [1:120] 0.525 0.404 0.61 0.369 0.459 ...  
## - attr(\*, "Size")= int 16  
## - attr(\*, "Labels")= chr [1:16] "Bora\_1" "Bora\_2" "Gambier\_1" "Gambier\_2" ...  
## - attr(\*, "Diag")= logi FALSE  
## - attr(\*, "Upper")= logi FALSE  
## - attr(\*, "method")= chr "euclidean"  
## - attr(\*, "call")= language dist(x = coral.matrices\_Depth\_1$beta.bray)

library (usedist)  
  
beta\_Depth\_1 <- dist\_setNames(beta\_Depth\_1, paste0 ("6",sep = "\_",rownames (Depth\_1\_Beta)))  
  
beta\_Depth\_2 <- dist\_setNames(beta\_Depth\_2, paste0 ("20",sep = "\_",rownames (Depth\_2\_Beta)))

## Mantel tests with distance - per depth

I can also plot bray-distance according to vertical depth distance measuring bray-distance per site and not per depth

# Mantel tests   
# It needs Beta-dissimilarity matrix and matrix distance  
  
library (geodist)  
  
Locations <- read.csv(file = "~/Documents/AAASea\_Science/AAA\_PhD\_Thesis/Photoquadrats/GIS\_MAP/Deephope\_sampling\_locations\_RAN\*.csv", header = T, dec = ".", sep = ";", row.names = 1)  
Locations$Island <- gsub("Mangareva", "Gambier", Locations$Island)  
Locations$Island <- gsub("Bora Bora", "Bora", Locations$Island)  
  
Locations$ID<- with(Locations, paste0(Island, sep = "\_", Site))  
  
# Keep only same ID as coral data   
unique (melt\_depth$ID)

## [1] "Bora\_1" "Bora\_2" "Gambier\_1" "Gambier\_2" "Makatea\_1"   
## [6] "Makatea\_2" "Moorea\_1" "Moorea\_2" "Rangiroa\_1" "Rangiroa\_2"  
## [11] "Raroia\_1" "Raroia\_2" "Tahiti\_1" "Tahiti\_2" "Tikehau\_1"   
## [16] "Tikehau\_2"

Locations <- Locations[Locations$ID %in% melt\_depth$ID, ]  
Locations <- Locations[order(Locations[,'ID']), ]  
  
rownames (Locations) <- Locations$ID  
Locations <- subset (Locations, select = c(Latitude, Longitude))  
  
dm <- geodist (Locations, measure = "geodesic", paired = T) /1000  
  
dm <- as.data.frame(dm)  
rownames (dm) <- rownames (Locations)  
names (dm) <- rownames (Locations)  
  
# Transform to object of class dist  
dm <- dist (dm)  
  
##### Make the mantel test #####   
# For Depth 1 - 6m   
mantel(coral.matrices\_Depth\_1$beta.bray, dm)

##   
## Mantel statistic based on Pearson's product-moment correlation   
##   
## Call:  
## mantel(xdis = coral.matrices\_Depth\_1$beta.bray, ydis = dm)   
##   
## Mantel statistic r: 0.2859   
## Significance: 0.05   
##   
## Upper quantiles of permutations (null model):  
## 90% 95% 97.5% 99%   
## 0.191 0.280 0.329 0.378   
## Permutation: free  
## Number of permutations: 999

mantel(coral.matrices\_Depth\_2$beta.bray, dm)

##   
## Mantel statistic based on Pearson's product-moment correlation   
##   
## Call:  
## mantel(xdis = coral.matrices\_Depth\_2$beta.bray, ydis = dm)   
##   
## Mantel statistic r: -0.0891   
## Significance: 0.638   
##   
## Upper quantiles of permutations (null model):  
## 90% 95% 97.5% 99%   
## 0.222 0.299 0.347 0.427   
## Permutation: free  
## Number of permutations: 999

# and so on!

Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.