#################################loading data#################################

library(readxl)

echino <- read\_excel ("echino.xlsx", sheet=1, col\_names=TRUE)

enviro <- read\_excel ("enviro. xlsx ", sheet=1, col\_names=TRUE)

dat.pca <- read\_excel ("dat\_pca. xlsx ", sheet=1, col\_names=TRUE)

#Bathymetric data for map in figure 1

library(raster)

bat <- raster("ETOPO1\_Bed\_c\_geotiff.tif") #Etopo data downloaded from https://www.ngdc.noaa.gov/mgg/global/

bat.c <- crop(bat, extent(-115,-75,-37,-17))

dem.p <- rasterToPoints(bat.c)

df <- data.frame(dem.p)

#############################PCA############################

library(vegan)

library(tidyverse)

sca.dat <- scale(dat.pca[,c(2:8)], scale=T, center=T)

pca <- rda(sca.dat)

summary(pca)

sco.pca <- scores(pca)

pca.plot <- cbind(dat.pca, sco.pca$sites)

pca.vec <- as.data.frame(sco.pca$species)

hull.pca <- pca.plot %>% group\_by(station) %>% slice(chull(PC1,PC2))

gg.pca <- pca.plot[pca.plot$time=="A",]

gg.pca$station <- as.factor(gg.pca$station)

gg.pca$area <- enviro$area

hull.pca$station <- as.factor(hull.pca$station)

#function for a bidimensional color space

fun\_xy <- function(x, y){

 R <- (x+1)/2

 G <- (1-x)/2

 B <- (y+1)/2

 A <- 1- 0.5\*exp(-(x^2+y^2)/0.2)

 rgb(R, G, B, A)

}

library(scales)

var1 <- rescale(gg.pca$lon, to=c(-1,1))

var2 <- rescale(gg.pca$lat, to=c(-1,1))

t <- cbind.data.frame(var1, var2)

cols <- c(apply(t, 1, function(x) fun\_xy(x[1],x[2])))

dev.off()

pc1 <- ggplot() + geom\_point(data=gg.pca, aes(x=PC1, y=PC2, col=station, shape=area), size=3) +

 theme\_bw() + theme(legend.position="none", panel.grid=element\_blank()) +

 stat\_ellipse() + coord\_fixed(ylim=c(-0.9,0.7), xlim=c(-0.7,0.9)) + scale\_color\_manual(values=cols) +

 geom\_polygon(data=hull.pca, aes(x=PC1, y=PC2, fill=station), alpha = 0.3) + scale\_fill\_manual(values=cols) +

 xlab("PCA1(46.2%)") + ylab("PCA2(25.1%)") + scale\_shape\_manual(values=c(0,1,2,5,6,7))

pc2<- ggplot() + geom\_segment(data=pca.vec, aes(x=0, y=0, xend=PC1, yend=PC2)) +

 coord\_fixed(xlim=c(-4,4), ylim=c(-4,4)) + geom\_point(data=pca.vec, aes(x=PC1, y=PC2), size=1) +

 geom\_text(data=pca.vec, aes(x=PC1, y=PC2, label=row.names(pca.vec))) + theme\_bw() +

 theme(panel.grid=element\_blank()) + xlab("PCA1") + ylab("PCA2")

pc3<- ggplot() + geom\_point(data=gg.pca, aes(x=lon, y=lat, col=station, shape=area), size=3) +

 theme\_bw() + theme(legend.position="none", panel.grid=element\_blank()) +

 stat\_ellipse() + coord\_fixed(xlim=c(-115,-75), ylim=c(-35,-20)) + scale\_color\_manual(values=cols) +

 scale\_fill\_manual(values=cols) + xlab("Longitude") + ylab("Latitude") +

 scale\_shape\_manual(values=c(0,1,2,5,6,7))

library(Rmisc)

mat <- matrix(c(rep(1,9),2,2,3,2,2,3), byrow=F, ncol=5)

multiplot(pc1,pc2,pc3, layout=mat)

##########################Species richness - analyses##########################

#######################Standardizing Species Richness##########################

library(iNEXT)

library(vegan)

x0 <- aggregate(.~site, data=echino[,c(2,13:length(echino))], FUN=sum)

x2 <- aggregate(Amphio\_sp ~ site, data=echino, FUN=length)

names(x2) <- c("site", "n")

x3 <- cbind.data.frame(x2, x0[,-1])

x4 <- x3[x3$n>1,]

i <- colSums(x4[,-c(1:2)])>0

x5 <- x4[,c(T,T,i)]

x6 <- x5[specnumber(x5[,-c(1:2)])>1,]

x7<- split(x6[,-1], x6$site)

x8 <- lapply(x7, function(x) {x[x!=0]})

rare <- iNEXT(x8, q=0, datatype="incidence\_freq")

spe\_rare0 <- rare$AsyEst

spe\_rare00 <- spe\_rare0[spe\_rare0$Diversity=="Species richness",]

ggiNEXT(rare, type=1, facet.var="order")

ggiNEXT(rare, type=3, facet.var="none", color.var="site") #MS.3 has small sample completness. Estimates might be highly biased

spe\_rare <- spe\_rare[!spe\_rare00$Site=="MS.3",]

enviro.aves <- aggregate(. ~ site, data=enviro[,c(3,11:length(enviro))], FUN=mean)

data.richness <- cbind.data.frame(spe\_rare[,-c(2,5:7)],

 enviro.aves[enviro.aves$site %in%

 spe\_rare$Site,])

ggplot() + geom\_raster(data=df, aes(x=x, y=y, fill=ETOPO1\_Bed\_c\_geotiff)) +

 geom\_point(data=data.richness, aes(x=lon, y=lat, size=Estimator), shape=1, col="red") +

 theme\_bw() + theme(panel.grid=element\_blank()) +

 stat\_ellipse() + coord\_fixed(xlim=c(-113,-77), ylim=c(-35,-20)) + scale\_color\_manual(values=cols) +

 xlab("Longitude") + ylab("Latitude")

#GLMs

data.richness$chloro <- aggregate(chloro ~ site, enviro[enviro$site %in% data.richness$Site,], FUN=mean)[,2]

library(MuMIn)

library(MASS)

data.glm <- data.richness[,-c(1:2,4)]

glm.glob <- glm(Estimator ~ ., data=data.glm, family=gaussian(link="log"), na.action = "na.fail")

result <- dredge(glm.glob, extra="adjR^2")

result

mod.ave <- model.avg(result, subset=weight>0.05, fit=T)

summary(mod.ave)

mod.ave$sw

p <- gaussian()

pred.s <- predict(mod.ave, full=T)

work.resid.ave.s <- (data.glm$Estimator - p$linkinv(pred.s))/p$mu.eta(pred.s)

coeff.ave.s <- coef(mod.ave)

coeff.ave.s

res.depth.s0 <- work.resid.ave.s + coeff.ave.s["depth.m"]\*data.glm$depth.m

res.depth.s <- res.depth.s0 - mean(res.depth.s0)

res.o2.s0 <- work.resid.ave.s + coeff.ave.s["o2"]\*data.glm$o2

res.o2.s <- res.o2.s0 - mean(res.o2.s0)

res.sal.s0 <- work.resid.ave.s + coeff.ave.s["salinity"]\*data.glm$salinity

res.sal.s <- res.sal.s0 - mean(res.sal.s0)

resid.S <- cbind.data.frame(data.glm, res.depth.s, res.o2.s, res.sal.s)

p1 <- ggplot(resid.S, aes(x=depth.m, y=res.depth.s)) + geom\_point(col="darkgrey") + theme\_bw() + theme(panel.grid=element\_blank()) + stat\_smooth(method="lm", col="black", se=F) + ggtitle("b)") + xlab("Depth (m)")

p2 <- ggplot(resid.S, aes(x=o2, y=res.o2.s)) + geom\_point(col="darkgrey") + theme\_bw() + theme(panel.grid=element\_blank()) + stat\_smooth(method="lm", col="black", se=F) + ggtitle("c)") + xlab(expression(paste("Oxygen (", mu, "mol/l)")))

p3 <- ggplot(resid.S, aes(x=salinity, y=res.sal.s)) + geom\_point(col="darkgrey") + theme\_bw() + theme(panel.grid=element\_blank()) + stat\_smooth(method="lm", col="black", se=F) + ggtitle("d)") + xlab("Salinity (PSU)")

#plot

library(Rmisc)

m3 <- matrix(c(1,2,3), ncol=3)

multiplot(p1,p2,p3, layout=m3)

#######################Multivariate GLM#######################

echi.all <- echino[rowSums(echino[,13:length(echino)])>0, ]

Y0 <- echi.all[,13:length(echi.all)]

Y <- Y0[,colSums(Y0)>0]

X0 <- enviro[enviro$station %in% echi.all$station,]

X <- X0[,c(11:18)]

#model selection - multivariate GLM

library(mvabund)

occr.mv <- mvabund(Y)

full.multi.glm0 <- manyglm(occr.mv ~ depth.m + lon + lat + temp0 + temp300 + salinity + o2 + chloro, data=X, family="binomial")

drop1(full.multi.glm0)

multi.glm1 <- manyglm(occr.mv ~ depth.m + lon + temp0 + temp300 + salinity + o2 + chloro, data=X, family="binomial")

drop1(multi.glm1)

multi.glm2 <- manyglm(occr.mv ~ depth.m + lon + temp300 + salinity + o2 + chloro, data=X, family="binomial")

drop1(multi.glm2)

multi.glm3 <- manyglm(occr.mv ~ lon + temp300 + salinity + o2 + chloro, data=X, family="binomial")

drop1(multi.glm3)

multi.glm4 <- manyglm(occr.mv ~ lon + salinity + o2 + chloro, data=X, family="binomial")

drop1(multi.glm4)

multi.glm5 <- manyglm(occr.mv ~ lon + o2 + chloro, data=X, family="binomial")

drop1(multi.glm5)

sum.glm5 <- summary(multi.glm5)

dev.glm5 <- anova(multi.glm5)

sum.glm5

dev.glm5

#####################Model-based ordinations#####################

library(boral)

X2 <- X[,c(3,5,8)]

b0 <- boral(Y, lv.control=list(num.lv=2), family="binomial", save.model=T, calc.ics=T)

b1 <- boral(Y, X=X2, lv.control=list(num.lv=2), family="binomial", save.model=T, calc.ics=T)

tcross <- tcrossprod(b0$lv.median, b0$lv.coefs.median[,2:3])

svd.d <- svd(tcross, 2, 2)

scores.site <- as.data.frame(scale(svd.d$u \* matrix(svd.d$d[1:2]^0.5, nrow=b0$n, ncol=2, byrow=T), center=T, scale=F))

scores.sps <- as.data.frame(scale(svd.d$v \* matrix(svd.d$d[1:2]^0.5, nrow=b0$p, ncol=2, byrow=T), center=T, scale=F))

rownames(scores.sps) <- rownames(b0$lv.coefs.median)

names(scores.site) <- c("Lv.1", "Lv.2")

names(scores.sps) <- c("Lv.1", "Lv.2")

scores.sps <- cbind.data.frame(scores.sps, occ=colSums(Y))

lv0 <- cbind.data.frame(scores.site, echi.all)

lv01 <- rescale(lv0$Lv.1, to=c(-1,1))

lv02 <- rescale(lv0$Lv.2, to=c(-1,1))

t0 <- cbind.data.frame(lv01, lv02)

cols0 <- c(apply(t0, 1, function(x) fun\_xy(x[1],x[2])))

cols0

lv0$station <- factor(lv0$station, levels=c(lv0$station))

#pure latent variable plot

library(ggplot2)

p1 <- ggplot() + geom\_point(data=lv0, aes(x=Lv.1, y=Lv.2, col=station, shape=area), size=3) +

 geom\_text(data=scores.sps[scores.sps$occ>4,], aes(x=Lv.1, y=Lv.2, label=row.names(scores.sps[scores.sps$occ>4,]))) +

 geom\_point(data=scores.sps[scores.sps$occ>4,], aes(x=Lv.1, y=Lv.2), col="black", pch=18) + ggtitle("a)") +

 theme\_bw() + theme(panel.grid=element\_blank(), legend.position="none") + scale\_color\_manual(values=c(cols0)) +

 scale\_shape\_manual(values=c(0,1,2,5,6,7)) + coord\_fixed(xlim=c(-2.1,2.3), ylim=c(-2.3,2.1))

p2 <- ggplot() + theme\_bw() + coord\_fixed() + geom\_point(data=lv0, aes(x=lon, y=lat, col=station, shape=area), size=4) +

 theme(panel.grid=element\_blank(), legend.position="none") + ggtitle("b)") +

 xlim(-112.5,-76.5) +

 ylim(-34.5,-20.7) + scale\_shape\_manual(values=c(0,1,2,5,6,7)) + scale\_color\_manual(values=c(cols0))

m2 <- matrix(1:2, ncol=1)

multiplot(p1, p2, layout=m2)

#residual ordination

tcross2 <- tcrossprod(b1$lv.median, b1$lv.coefs.median[,2:3])

svd.d2 <- svd(tcross2, 2, 2)

scores.site2 <- as.data.frame(scale(svd.d2$u \* matrix(svd.d2$d[1:2]^0.5, nrow=b1$n, ncol=2, byrow=T), center=T, scale=F))

scores.sps2 <- as.data.frame(scale(svd.d2$v \* matrix(svd.d2$d[1:2]^0.5, nrow=b1$p, ncol=2, byrow=T), center=T, scale=F))

rownames(scores.sps2) <- rownames(b1$lv.coefs.median)

names(scores.site2) <- c("Lv.1", "Lv.2")

names(scores.sps2) <- c("Lv.1", "Lv.2")

scores.sps2 <- cbind.data.frame(scores.sps2, occ=colSums(Y))

lv2 <- cbind.data.frame(scores.site2, echi.all)

lv2.1 <- rescale(lv2$lon, to=c(-1,1))

lv2.2 <- rescale(lv2$lat, to=c(-1,1))

t1 <- cbind.data.frame(lv2.1, lv2.2)

cols1 <- c(apply(t1, 1, function(x) fun\_xy(x[1],x[2])))

lv2$station <- factor(lv2$station, levels=c(lv2$station))

par1 <- ggplot() + geom\_point(data=lv0, aes(x=Lv.1, y=Lv.2, col=area, shape=area), size=3) +

 theme\_bw() + theme(panel.grid=element\_blank(), legend.position="none") +

 scale\_shape\_manual(values=c(0,1,2,5,6,7)) + coord\_fixed(xlim=c(-2.2,2.5), ylim=c(-2.5,2.2)) +

 ggtitle("a) Unconstrained ordination")

par2 <- ggplot() + geom\_point(data=lv2, aes(x=Lv.1, y=Lv.2\*-1, col=area, shape=area), size=3) +

 theme\_bw() + theme(panel.grid=element\_blank(), legend.position="none") +

 scale\_shape\_manual(values=c(0,1,2,5,6,7)) + coord\_fixed(xlim=c(-5,6), ylim=c(-5.5,5.5)) +

 ggtitle("b) Residual ordination") + ylab("Lv.2")

multiplot(par1, par2, layout=matrix(1:2, ncol=2))