**S1 File. Annotated R script and R functions.**

**Part 1: Main script**

#-------------------------------------------------------------------------------------------

# Bathymetric analysis of foraminiferal assemblages

#-------------------------------------------------------------------------------------------

# R version 3.2.0

# written by M. Kowalewski (University of Florida), kowalewski@ufl.edu

# Last updated: April 4, 2016

# NOTES:

# In all csv datasets, species are rows and samples are columns

#-------------------------------------------------------------------------------------------

rm(list=ls(all=TRUE)) # remove all preexisting objects

library(vegan) # upload vegan library

source(file='foramF.R') # upload custom functions

runDCA <- 0 # Set to 1 to run DCA, Set to 0 for nMDS

options(scipen=100) # report small numbers in fixed notation

# - upload individual dataset

SBAr <- read.csv("SB\_Agg.csv")[-1,-1] # SB data matrix (aggutinated forams)

SBCr <- read.csv("SB\_Cal.csv")[-1,-1] # SB data matrix (calcareous forams)

MSAr <- read.csv("MS\_Agg.csv")[-1,-1] # MS data matrix (aggutinated forams)

MSCr <- read.csv("MS\_Cal.csv")[-1,-1] # MS data matrix (calcareous forams)

SBATr <- read.csv("SB\_Agg.csv")[-1,1] # SB species list (aggutinated forams)

SBCTr <- read.csv("SB\_Cal.csv")[-1,1] # SB species list (calcareous forams)

MSATr <- read.csv("MS\_Agg.csv")[-1,1] # MS species list(aggutinated forams)

MSCTr <- read.csv("MS\_Cal.csv")[-1,1] # MS species list(calcareous forams)

SBL <- read.csv("SB\_Loc.csv") # SB Locality Info

MSL <- read.csv("MS\_Loc.csv") # MS Locality Info

MSALL <- rbind(MSAr, MSCr) # SB data matrix (all formas)

SBALL <- rbind(SBAr, SBCr) # SB data matrix (all formas)

MSALLT <- c(MSATr,MSCTr) # SB species list (all forams)

SBALLT <- c(SBATr,SBCTr) # SB species list (all forams)

# upload the pooled dataset

ALL <- read.csv('totalforams.csv') # Data pooled across basins

ALLD <- t(ALL[-nrow(ALL),-1]) # Data matrix for pooled data

wall <- as.numeric(ALL[nrow(ALL),-1]) # 1=agglutinated, 2=calcareous

basin <- c(rep('SB',18),rep('MS',17)) # basin

# assemble a list with four individual datasets

dat1 <- list('Saros Bay (Agglutinated)'=SBAr,'Saros Bay (Calcaerous)'=SBCr,

'Marmara Sea (Agglutinated)'=MSAr, 'Marmara Sea (Calcareous)'=MSCr)

# assemble a corresponding list with species info

spec1 <- list('Saros Bay (Agglutinated)'=SBATr,'Saros Bay (Calcaerous)'=SBCTr,

'Marmara Sea (Agglutinated)'=MSATr, 'Marmara Sea (Calcareous)'=MSCTr)

sapply(dat1,overview, T) # check datasets

sapply(spec1, length) # check species lists

# output repository lists of species for each basin (disabled)

#write.csv(data.frame('taxon'=spec1[[1]], 'number of specimens'=rowSums(dat1[[1]])), 'Saros Bay (aggl).csv')

#write.csv(data.frame('taxon'=spec1[[2]], 'number of specimens'=rowSums(dat1[[2]])), 'Saros Bay (calc).csv')

#write.csv(data.frame('taxon'=spec1[[3]], 'number of specimens'=rowSums(dat1[[3]])), 'Marmara Sea (aggl).csv')

#write.csv(data.frame('taxon'=spec1[[4]], 'number of specimens'=rowSums(dat1[[4]])), 'Marmara Sea (calc).csv')

#----- pool foraminfera (both wall types) within each basin --------------------------------

dat2 <- list('Saros Bay'=rbind(dat1[[1]],dat1[[2]]), 'Marmara Sea' = rbind(dat1[[3]],dat1[[4]]))

spec2 <- list('Saros Bay'=c(spec1[[1]],spec1[[2]]), 'Marmara Sea'=c(spec1[[3]],spec1[[4]]))

sapply(dat2,overview, T) # check datasets

sapply(spec2, length) # check species lists

#----- filter datasets to remove rare or sparsely occurring species ------------------------

n\_crit <- 6 # minimum required number of specimens per species

o\_crit <- 4 # minimum required number of occurrences per species

dat1F <- sapply(dat1, filter1, n\_crit, o\_crit) # filtered list of datasets

spec1F <- filter2(spec1, dat1, n\_crit, o\_crit) # remove names from species list accordingly

sapply(dat1F,overview, T) # check datasets

sapply(spec1F, length) # check species lists

dat2F <- sapply(dat2, filter1, n\_crit, o\_crit) # filtered list of datasets

spec2F <- filter2(spec2, dat2, n\_crit, o\_crit) # remove names from species list accordingly

sapply(dat2F,overview, T) # check datasets

sapply(spec2F, length) # check species lists

#----- plotting parameters -----------------------------------------------------------------

aN <- 0.8 # font size for axis labels

aL <- 0.9 # font size for axis title

psc <- 0.01 # scale bubbles for samples

ssc <- 0.4 # scale bubbles for species

sps <- 0.5 # size of species symbols

col1 <- 'black' # color of symbols for samples

col2 <- 'white' # color of symbols for species (pch2 must be set to 21)

pch1 <- 16 # symbol for samples

pch2 <- 21 # symbol for species

myorder <- 1 # plotting order (1=species on top, 2=samples on top, 3=samples only, 4=species only)

lheight <- -1.2 # vertical placement of inset label

vheight <- 0.02 # horizontal placement of inset label

tck1 <- -0.02 # length of ticks

minspec <- 100 # minimum sample size for species to be plotted

#----- plot 4 datasets by basin and wall type ----------------------------------------------

x1 <- dat1F # use 4 datasets by basin and wall type

depth <- list(SBL[,2],SBL[,2],MSL[,2],MSL[,2]) # water depth values

#PLOT NMDS ORDINATIONS

op <- par(mfrow = c(2, 2), mai = c(0.5,0.5,0.2,0), omi=c(0.1,0.1,0,0), pty="square")

labels <- c('A', 'B', 'C', 'D')

for (i in 1:length(x1)) {

foram.plot1(x1[[i]],depth[[i]], runDCA=F, label=labels[i], axisL=aL, axisN=aN, pscale=psc,

sscale=ssc, spsize=sps, col1=col1, col2=col2, pch1=pch1, myorder=myorder, tck1=tck1,

lheight=lheight, vheight=vheight, minspec=minspec) }

par(op)

#PLOT DCA ORDINATIONS

op <- par(mfrow = c(2, 2), mai = c(0.5,0.5,0.2,0), omi=c(0.1,0.1,0,0), pty="square")

labels <- c('A', 'B', 'C', 'D')

for (i in 1:length(x1)) {

foram.plot1(x1[[i]],depth[[i]], runDCA=T, label=labels[i], axisL=aL, axisN=aN, pscale=psc,

sscale=ssc, spsize=sps, col1=col1, col2=col2, pch1=pch1, myorder=myorder, tck1=tck1,

lheight=lheight, vheight=vheight, minspec=minspec) }

par(op)

#----- plot 2 basins (all forams) ----------------------------------------------------------

x1 <- dat2F # use 4 datasets by basin and wall type

minspec <- 200 # minimum sample size for a species to be plotted

depth <- list(SBL[,2],MSL[,2]) # water depth values

op <- par(mfrow = c(2, 1), mai = c(0.5,0.5,0.2,0), omi=c(0.1,0.1,0,0), pty="square")

# PLOT NMDS ORDINATIONS

labels <- c('A', 'B')

for (i in 1:length(x1)) {

foram.plot1(x1[[i]],depth[[i]], runDCA=F, label=labels[i], axisL=aL, axisN=aN, pscale=psc,

sscale=ssc, spsize=sps, col1=col1, col2=col2, pch1=pch1, myorder=myorder, tck1=tck1,

lheight=lheight, vheight=vheight, minspec=minspec) }

# PLOT DCA ORDINATIONS

labels <- c('A', 'B')

for (i in 1:length(x1)) {

foram.plot1(x1[[i]],depth[[i]], runDCA=T, label=labels[i], axisL=aL, axisN=aN, pscale=psc,

sscale=ssc, spsize=sps, col1=col1, col2=col2, pch1=pch1, myorder=myorder, tck1=tck1,

lheight=lheight, vheight=vheight, minspec=minspec) }

par(op)

#----- plot all datasets pooled (both basins and both wall types) --------------------------

# NMDS ALL FORMAS BOTH BASINS

goodALL<- which(rowSums(ALLD)>=n\_crit & rowSums(ALLD>0)>=o\_crit)

ALLF<- ALLD[goodALL,]

WALL<- wall[goodALL]

depth <- c(SBL[,2],MSL[,2]) # water depth

minspec <- 500

sps <- 0.7

foram.plot1(ALLF, depth, basin, runDCA=runDCA, axisL=aL, axisN=aN, pscale=psc,

sscale=ssc, spsize=sps, col1=col1, col2=col2, pch1=pch1, myorder=myorder, tck1=tck1,

lheight=lheight, vheight=vheight, minspec=minspec)

absp <- sort(which(rowSums(ALLF)>minspec), decreasing=T)

taxnames <- gsub("[.]", " ", names(absp))

taxnames <- gsub("Spirorutilus sp ", "Spirorutilus sp.", taxnames)

text(rep(-1.3,length(absp)),seq(-2.5,-1,length.out=length(absp)),absp, cex=0.7)

text(rep(-1.2,length(absp)),seq(-2.5,-1,length.out=length(absp)),taxnames, font=3, cex=0.7, adj=0)

#----- plot ordination scores versus Depth -------------------------------------------------

x1 <- c(dat1F,dat2F, list(ALLF)) # use 4 datasets by basin and wall type

depth <- list(SBL[,2],SBL[,2],MSL[,2],MSL[,2], SBL[,2],MSL[,2], c(SBL[,2],MSL[,2])) # water depth values

op <- par(mfrow = c(7, 1), mai = c(0.15,0.3,0,0.1), omi=c(0.15,0,0.1,0))

labels <- c('A', 'B', 'C', 'D', 'E', 'F', 'G')

for (i in 1:length(x1)) {

if (i < length(x1)) {

foram.plot2(x1[[i]],sites=depth[[i]],label=labels[i], AXIS=1, runDCA=F)

}

if (i==length(x1)) {

foram.plot2(x1[[i]],sites=depth[[i]],label=labels[i], AXIS=2, runDCA=F)

mtext("water depth [m]", side=1, line=1, cex=0.7)

}

}

par(op)

#----- Report RMA Regression Models --------------------------------------------------------

table1 <- NULL

for (i in 1:length(x1)) {

if (i < length(x1)) {

table1 <- rbind(table1, reg.foram(x1[[i]], depth[[i]], AXIS=1))

ifelse(i==1, dataset <- names(x1[i]), dataset <- c(dataset, names(x1[i])))

}

if (i == length(x1)) {

table1 <- rbind(table1, reg.foram(x1[[i]], depth[[i]], AXIS=2))

dataset <- c(dataset, "ALL DATA")

}

}

(table1 <- data.frame(dataset, table1))

#write.csv(table1, 'table1forams.csv')

#----- Methods (Flitering Effects) ---------------------------------------------------------

nmin <- 50 # - maximum filter for total number of specimens per species

omin <- 12 # - maximum filter for total number of species occurrences

out1 <- NULL

for (i in 1:nmin) {

for (j in 1:omin) {

goodSBA<- which(rowSums(SBAr)>=i & rowSums(SBAr>0)>=j)

SBA<- SBAr[goodSBA,]; SBAT<- SBATr[goodSBA]

goodSBC<- which(rowSums(SBCr)>=i & rowSums(SBCr>0)>=j)

SBC<- SBCr[goodSBC,]; SBCT<- SBCTr[goodSBC]

goodMSA<- which(rowSums(MSAr)>=i & rowSums(MSAr>0)>=j)

MSA<- MSAr[goodMSA,]; MSAT<- MSATr[goodMSA]

goodMSC<- which(rowSums(MSCr)>=i & rowSums(MSCr>0)>=j)

MSC<- MSCr[goodMSC,]; MSCT<- MSCTr[goodMSC]

SB<- rbind(SBA,SBC)

MS<- rbind(MSA,MSC)

sites<- MSL

usethis<- t(MS)

d1<- decostand(usethis, "total")

Dcorr<- as.data.frame(decorana(d1)$rproj)

rd1<- cor(log(sites[,2],10),-Dcorr$DCA1)

Mydata<- decostand(d1, "max")

initial\_nMDS<- metaMDS(Mydata, distance="bray", k=2, trymax=50)

nMDS <- metaMDS(Mydata, previous.best = initial\_nMDS, k=2)

rn1<- cor(log(sites[,2],10),nMDS$points[,1])\*\*2

out1 <- rbind(out1,c(i,j,sum(usethis),dim(usethis),rn1\*\*2,rd1\*\*2,nMDS$stress))

}

}

out1 <- out1[order(out1[,3], decreasing=T),]

#----- plot filter effects -----------------------------------------------------------------

plot(seq(1,nrow(out1),1),out1[,3]/max(out1[,3]),pch=16,type='o', cex=0.6,

xlab='', ylab='',

ylim=c(0,1), xlim=c(0,700), axes=F)

axis(1, labels=F, tick=F)

axis(2)

abline(v=which(out1[,1]==6 & out1[,2]==4), col='gray', lwd=2, lty=2)

text(600,min(out1[,3]/max(out1[,3])),'% specimens', cex=0.6, pos=4)

points(seq(1,nrow(out1),1),out1[,5]/max(out1[,5]), type='o',pch=16,cex=0.3, col='blue4', lwd=0.7)

text(600,min(out1[,5]/max(out1[,5])),'% species', cex=0.6, col='blue4', pos=4)

points(seq(1,nrow(out1),1),out1[,6], type='o',pch=16,cex=0.3,col='gray', lwd=0.7)

text(600,min(out1[,6]),'r-square NMDS', cex=0.6, col='gray', pos=4)

points(seq(1,nrow(out1),1),out1[,7], type='o',pch=16,cex=0.3,col='green4', lwd=0.7)

text(600,min(out1[,7]),'r-square DCA', cex=0.6, col='green4', pos=4)

points(seq(1,nrow(out1),1),out1[,8], type='o',pch=16,cex=0.3,col='red4', lwd=0.7)

text(600,min(out1[,8]),'NMDS stress', cex=0.6, col='red4', pos=4)

text(seq(1,nrow(out1),10), 0.035, out1[,1][seq(1,nrow(out1),10)], cex=0.4)

text(600, 0.035, 'minimum no. specimens', cex=0.4, pos=4)

text(seq(1,nrow(out1),10), 0, out1[,2][seq(1,nrow(out1),10)], cex=0.4)

text(600, 0, 'minimum no. occurrences', cex=0.4, pos=4)

mtext(side=1, 'filter')

mtext(side=2, 'variables', line=2.5)

#----- END ---------------------------------------------------------------------------------

**Part 2: Custom R functions used in the main script**

#-------------------------------------------------------------------------------------------

# Bathymetric analysis of foraminferal assemblages

#

# Custom functions used in the main script

#-------------------------------------------------------------------------------------------

# R version 3.2.0

# written by M. Kowalewski (University of Florida), kowalewski@ufl.edu

# Last updated: April 4, 2016

# NOTES:

# In all csv datasets, species are rows and samples are columns

#-------------------------------------------------------------------------------------------

# summary function

overview <- function(x, transpose=F) {

if (transpose) x <- t(x)

c(samples=nrow(x), species=ncol(x), "total specimens"=sum(x),

"smallest sample"=min(rowSums(x)), "rarest species"=min(colSums(x)),

"empty species"=sum(colSums(x)==0), "empty samples"=sum(rowSums(x)==0))

}

# filter function

filter1 <- function(x, n\_crit=2, o\_crit=2) {

x[which(rowSums(x)>=n\_crit & rowSums(x>0)>=o\_crit),]

}

# filter function 2

filter2 <- function (x, y, n\_crit=2, o\_crit=2) {

out <- vector('list', length(x))

for (i in 1:length(x)) {

good <- which(rowSums(y[[i]])>=n\_crit & rowSums(y[[i]]>0)>=o\_crit)

out[[i]] <- x[[i]][good]

}

return(out)

}

# plot function for plotting multiple ordinations

foram.plot1 <- function(x, depth, groups=1, runDCA=F, dist="bray", k=2, pch1=16, pch2=21, lheight=0,

label="", xann=T, yann=T, axisL=.75, axisN=.8, myorder=2, vheight=.05, minspec=100,

tck1=-.02, pscale=.01, sscale=.5, spsize=.5, col1='black', col2='white', col3='black',

col4='gray', species.label=T) {

usethis<- t(x)

d1<- decostand(usethis, "total")

Mydata<- decostand(usethis, "max")

initial\_nMDS<- metaMDS(Mydata, distance=dist, k=k, trymax=50)

nMDS <- metaMDS(Mydata, previous.best = initial\_nMDS, k=k)

if (runDCA==1) {

DCA <- decorana(d1)

nMDS$points[,1] <- DCA$rproj[,1]

nMDS$points[,2] <- DCA$rproj[,2]

nMDS$species[,1] <- DCA$cproj[,1]

nMDS$species[,2] <- DCA$cproj[,2]

}

plot(nMDS$points[,1], nMDS$points[,2], pch=16, cex=0,

xlim=range(nMDS$points[,1], nMDS$species[,1]),

ylim=range(nMDS$points[,2], nMDS$species[,2]),

xlab="", ylab="", axes=F)

box(col='black')

mtext(label,line=lheight, adj=vheight, cex=1)

if (length(groups)>1) {

groups <- as.factor(groups)

gpcol <- c(col3, col4)

k <- NULL

for (i in levels(groups)) {

ifelse(is.null(k), k <- 1, k <- k +1)

gp <- which(groups==i)

points(nMDS$points[gp,1],nMDS$points[gp,2],pch=pch1,col=gpcol[k],

cex=0.5+pscale\*depth[gp],type="o")

}

points(nMDS$species[,1][which(colSums(usethis)>=minspec)],nMDS$species[,2][which(colSums(usethis)>=minspec)],

pch=pch2,col="black", bg=col2, cex=sscale\*log(colSums(usethis)[which(colSums(usethis)>=minspec)]))

text(nMDS$species[,1][which(colSums(usethis)>=minspec)],

nMDS$species[,2][which(colSums(usethis)>=minspec)],

as.character(which(colSums(usethis)>=minspec)),cex=spsize, col="black")

}

if (length(groups)==1) {

if (myorder==1) {

points(nMDS$points[,1],nMDS$points[,2],pch=pch1,col=col1[as.factor(groups)],cex=0.5+pscale\*depth,type="o")

points(nMDS$species[,1][which(colSums(usethis)>=minspec)],nMDS$species[,2][which(colSums(usethis)>=minspec)],

pch=pch2,col="black", bg=col2, cex=sscale\*log(colSums(usethis)[which(colSums(usethis)>=minspec)]))

text(nMDS$species[,1][which(colSums(usethis)>=minspec)],

nMDS$species[,2][which(colSums(usethis)>=minspec)],

colnames(usethis)[which(colSums(usethis)>=minspec)],cex=spsize, col="black")

}

if (myorder==2) {

points(nMDS$species[,1][which(colSums(usethis)>=minspec)],nMDS$species[,2][which(colSums(usethis)>=minspec)],

pch=pch2,col="black", bg=col2, cex=sscale\*log(colSums(usethis)[which(colSums(usethis)>=minspec)]))

text(nMDS$species[,1][which(colSums(usethis)>=minspec)],

nMDS$species[,2][which(colSums(usethis)>=minspec)],

colnames(usethis)[which(colSums(usethis)>=minspec)],cex=spsize, col="black")

mtext(label,line=lheight, adj=vheight, cex=1)

points(nMDS$points[,1], nMDS$points[,2], pch=pch1, col=col1[as.factor(groups)], cex=0.5+pscale\*depth,type="o")

}

if (myorder==3) {

points(nMDS$points[,1], nMDS$points[,2], pch=pch1, col=col1[as.factor(groups)], cex=0.5+pscale\*depth, type="o")

}

if (myorder==4) {

points(nMDS$species[,1][which(colSums(usethis)>=minspec)],nMDS$species[,2][which(colSums(usethis)>=minspec)],

pch=pch2,col="black", bg=col2, cex=sscale\*log(colSums(usethis)[which(colSums(usethis)>=minspec)]))

text(nMDS$species[,1][which(colSums(usethis)>=minspec)],

nMDS$species[,2][which(colSums(usethis)>=minspec)],

colnames(usethis)[which(colSums(usethis)>=minspec)],cex=spsize, col="black")

}

}

if (runDCA!=1) {

mtext(paste("stress=",round(nMDS$stress,3),sep=""), side=3, line=lheight,adj=0.95, cex=0.7, col="black")

if (yann) mtext("NMDS 2", side=2, line=2.5, cex=axisL)

if (xann) mtext("NMDS 1", side=1, line=1.8, cex=axisL)

}

if (runDCA==1) {

if (yann) mtext("DCA 2", side=2, line=2.5, cex=axisL)

if (xann) mtext("DCA 1", side=1, line=1.8, cex=axisL)

}

axis(2,tck=tck1,hadj=0.7,las=1, cex.axis=axisN)

axis(1,tck=tck1,padj=-0.8, cex.axis=axisN)

}

# plot function for plotting NMDS scores versus depth

foram.plot2 <- function(x, sites, groups=1, label="", runDCA=F, dist='bray', k=2, AXIS=1) {

usethis <- t(x)

d1<- decostand(usethis, "total")

Mydata<- decostand(usethis, "max")

initial\_nMDS<- metaMDS(Mydata, distance=dist, k=2, trymax=50)

nMDS <- metaMDS(Mydata, distance=dist, previous.best = initial\_nMDS, k=2)

nMDS$points[,1] <- nMDS$points[,AXIS]

if (runDCA==1) {

DCA <- decorana(d1)

nMDS$points[,1] <- DCA$rproj[,AXIS]

}

if (cor(log(sites),nMDS$points[,1])<0) nMDS$points[,1] <- -nMDS$points[,1]

slopeRMA=sign(cov(log(sites,10),nMDS$points[,1]))\*sd(nMDS$points[,1])/sd(log(sites,10))

interceptRMA=mean(nMDS$points[,1]-slopeRMA\*mean(log(sites,10)))

r2<- round(cor(log(sites,10),nMDS$points[,1])\*\*2,2)

plot(sites, nMDS$points[,1], pch=16, xlab="", ylab="",log="x", axes=F,

cex=0, xlim=c(10,max(sites)),

ylim=c(floor(min(nMDS$points[,1])),ceiling(max(nMDS$points[,1]))))

points(c(min(sites),max(sites)),interceptRMA + log(c(min(sites),max(sites)),10)\*slopeRMA,

lty=1, lwd=1, col='darkgray', type='l')

points(sites, nMDS$points[,1], pch=21, lwd=0.7, cex=0.05\*sqrt(rowSums(usethis)), col="black")

axis(1,tck=-.025,padj=-3.2, label=c(10,seq(50,50\*round(max(sites)/50),50)),

at=c(10,seq(50,50\*round(max(sites)/50),50)),cex.axis=0.5)

axis(2, tck=-.025, padj=2.2,

labels=as.character(seq(floor(min(nMDS$points[,1])),ceiling(max(nMDS$points[,1])),1)),

at=seq(floor(min(nMDS$points[,1])),ceiling(max(nMDS$points[,1])),1),cex.axis=0.5)

if (runDCA==1) {

mtext(paste("DCA",AXIS,sep=""), side=2, line=1, cex=0.5)

}

if (runDCA!=1) {

mtext(paste("NMDS",AXIS, sep=""), side=2, line=1, cex=0.5)

}

mtext(bquote(italic(r)^{' 2'} == .(r2)), cex=0.3, line=-0.8, adj=0.2, col="black")

box()

mtext(label,adj=0.015,line=-0.8,cex=0.4)

}

# regression function

reg.foram <- function(x, y, transpose=F, k=2, dist='bray', AXIS=1) {

if (transpose) x <- t(x)

usethis <- t(x)

d1<- decostand(usethis, "total")

Mydata<- decostand(usethis, "max")

initial\_nMDS<- metaMDS(Mydata, distance=dist, k=2, trymax=50)

nMDS <- metaMDS(Mydata, distance=dist, previous.best = initial\_nMDS, k=2)

nMDS$points[,1] <- nMDS$points[,AXIS]

if (cor(log(y),nMDS$points[,1])<0) nMDS$points[,1] <- -nMDS$points[,1]

slopeRMA=sign(cov(log(y,10),nMDS$points[,1]))\*sd(nMDS$points[,1])/sd(log(y,10))

interceptRMA=mean(nMDS$points[,1]-slopeRMA\*mean(log(y,10)))

r2<- round(cor(log(y,10),nMDS$points[,1])\*\*2,2)

rho <- cor.test(log(y,10),nMDS$points[,1], method='spearman')$p.value

ifelse(rho < 0.0001, p <- 0.0001, p <- round(rho, 4))

c("samples"=ncol(x), "species"=nrow(x), "total specimens"=sum(x),

"smallest sample"=min(colSums(x)), "rarest species"=min(rowSums(x)),

"mean sample size"=round(mean(colSums(x)),1), "slope"=slopeRMA,

"intercept"=interceptRMA, "r2"=r2, "p value"=p)

}