```
# Program to evaluate microbial communities using sequence data

# K. A. Garrett (1, kgarrett@ksu.edu), L. Gomez-Montano (1), and A. Jumpponen (2)

# (1) Department of Plant Pathology, Kansas State University
# (2) Division of Biology, Kansas State University

# This program is contribution no. 14-020-C of the Kansas Agricultural Experiment Station

# If you find this R script useful in your research, please cite the following paper, which includes examples of this type of analysis

# Gomez-Montano, L., A. Jumpponen, M. A. Gonzales, J. Cusicanqui, C. Valdivia, P.P. Motavalli, M. Herman, and K. A. Garrett. 2013. Do bacterial and fungal communities in soils of the Bolivian Altiplano change under shorter fallow periods? Soil Biology & Biochemistry, 65: 50-59.

### Introduction to R script ########################################

# This R script performs statistical analyses using sequence data. It goes through all the steps
# from reading the data into R, merging multiple datasets, calculating diversity metrics,
# performing statistical analysis such as linear regression and non-metric multidimensional
# scaling (NMDS), making graphs and saving and exporting them for use in presentations
# and publications.  It illustrates the analyses used in Gomez-Montano et al. 2013, and can be
# used as a template for other similar analyses.

# Sections

# 1. Preparing data

# 2. Diversity metrics

# 3. Analysis of individual OTU frequencies

# 4. Soil physical and chemical analyses

# 5. NMDS community analysis

### 1. Preparing data ########################################

# Read in file with operational taxonomic units (OTUs, or clusters) observed for each
# sample.  This file combines counts of each OTU in each sample and the taxonomic
# strings for each OTU. The latter is the result obtained from the submission of the most
# representative sequences (from your data) to databases such as BLAST or RDP to find the
# best match that can be assigned to each OTU. BLAST (Basic Local Alignment Search Tool) is
# often used for fungal OTU assignments, although using the Small SubUnit (SSU) or Large
# SubUnit (LSU) of the ribosomal RNA gene allows the use of Silva or RDP databases.
# RDP (Ribosomal Database Project) is often used for bacterial OTU assignments.

# A description of the columns found in the file is provided below along with
# some similarities and differences between the files obtained when using BLAST or RDP.
```

```
## Columns in data set

# (Not included here: additional useful information from a BLAST query could include
# the score or the e-values, analogous to confidence in RDP)

# Plot- sample ID
# Cluster – (output from Pyrotagger outputs ‘Cluster’ (OTU) and RDP)
# Count – the number of sequences per OTU
# Totalseq – number of sequences per plot-OTU combination
# Freq – frequency of sequences in plot-OTU combination
# Site – place where the samples were taken
# Length – length of sequence ID for OTU (only present in fungal data set)
# Accession – accession number assigned from BLAST (Not present when using RDP)
# Confidence – confidence value assigned from RDP (Not present when using BLAST)
## Taxonomic classification
# Genus
# Species
# Phylum
# Division
# Class
# Order
# Family

# The object created below is called ‘DataOTU’ and the first step is to open the file that has the
# OTU information. The command ‘read.table’ reads a table from an external file, where
# ‘file.choose’ opens up a window for selecting the OTU file. The header here is set explicitly,
# the first row in the field being read becomes the variable names. If you change to header=F
# then the first row will be treated like the rest of the data and not as a label. ‘sep’ stands for
# separation and indicates that columns are separated by a comma, as in CSV files. ‘fill=TRUE’
# indicates that trailing spaces should be filled in as blanks when rows are of unequal length.

DataOTU <- read.table(file.choose(),header=T,sep=",",fill=T)
# Fungal data set is 'OTU-fungi'
# Bacterial data set is 'OTU-bacteria'

# If analyzing fungi:
Bacteria <- F # This would be changed to 'Bacteria <- T' for the bacteria data set

# An idiosyncracy of our data sets is that bacteria and fungi have different plot numbers
# If working with the bacterial data set
if(Bacteria){DataOTU$Plot <- DataOTU$Plot - 100}

# Comments about the above command:
# read.table reads a file in table format and creates a data frame from it
# file.choose() opens a window to pick the file for reading
# header=T tells the program to look for a header
# sep="," tells it to use commas as separators
# fill=T tells it to fill up remaining columns with NAs if there are too few columns in a line

# Checking the column names of the table
```

```
names(DataOTU)

# Checking the top of the table
head(DataOTU)

# Checking the tail of the table
tail(DataOTU)

# Checking the dimensions of the table
dim(DataOTU)

## Create file with number of OTUs per sample with no global singletons
# That is, no sequences that occur only once in the entire data set

# In the command below, you can access the column ‘Cluster’ in the object DataOTU as: #
DataOTU$Cluster

# DataOTU$Cluster contains OTU (cluster) names
# DataOTU$Count contains the number of times the OTU occurs within a plot

### find the unique OTU and plot names from the whole dataset
cnames <- unique(DataOTU$Cluster)  # create a vector with all the unique OTU (cluster) names
pnames<- unique(DataOTU$Plot)    # create a vector with all the unique plot names

# Create cnum which has the same length as cnames,
# starts out with zeros, and in the end will include
# the number of times that each OTU occurs in the whole data set
cnum <- 0*(1:length(cnames))

# Create a new version of the data which will not include global singletons
DataOTUns <- DataOTU

# for each OTU j, do the following
for (j in 1:(length(cnames))){
   # create an index indicating whether each row corresponds to OTU j or not
   tname <- cnames[j]
   temp <- DataOTU$Cluster == tname
   # keep only those rows that correspond to OTU j
   tempdat <- DataOTU[temp,]
   # enter the number of times the OTU occurs in cnum
   tcount <- sum(tempdat$Count)
   cnum[j] <- tcount
   # if singleton, remove rows corresponding to tnamex1
   if(tcount == 1) {DataOTUns <- DataOTUns[DataOTUns$Cluster != tname, ]}
}

# to check DataOTUns
summary(DataOTUns)
# (Note that the count within a sample can still be 1, but global singletons are removed)
# comparing this data with the one that still has singletons
summary(DataOTU)
```

```
# Read in file with the experimental design information for each sample
exp.design <- read.csv(file.choose(), header=T)
# data set is 'Altiplano_design'
summary(exp.design)

# Read in file with the soil physical and chemical analyses for each sample

## Associated soil measures
# Sand
# Clay
# Silt
# pH-water
# pH-KCL
# K
# OM – organic matter
# NTot – total N
# P

phys <- read.csv(file.choose(), header=T)
# data set is 'Analysis Phy_chem'
summary(phys)

### 2. Diversity measures ########################################

# The vegan package includes tools for community analysis.
# Diversity indices such as Simpson, Shannon, and Pielou’s evenness
# can be evaluated using vegan.
# http://cran.r-project.org/web/packages/vegan/vignettes/diversity-vegan.pdf

# Install vegan package (if not already installed)
install.packages("vegan")
# Load vegan package
library(vegan)

# Reformat the data for use in diversity index calculations in vegan package
# Keep only the first three columns: Plot, Cluster, and Count

# The new data set can be constructed with or without singletons
# depending on whether DataOTU or DataOTUns are used in the next command

x1 <- DataOTUns[,1:3]   # This command uses the data set without singletons
# x1<-DataOTU[,1:3] # This command would use the data set with singletons

# Many OTUs are not present in every plot - we want to include zero frequency rows for vegan
unames <- unique(x1$Cluster)
length(unames)
uplots <- unique(x1$Plot)
uplots
length(unique(x1$Plot))
```

```
# Create new matrix with the zero frequency rows included - to be filled in
# rows are plots
# columns are OTUs

data.vegan <- matrix(0,nrow=length(unique(x1$Plot)),ncol=length(unique(x1$Cluster)))
for(j1 in 1:length(unique(x1$Cluster))){
  for(j2 in 1:length(unique(x1$Plot))){
    if(dim(x1[x1$Plot == uplots[j2] & x1$Cluster == unames[j1],])[1] > 0) {
      data.vegan[j2,j1] <- (x1[x1$Plot == uplots[j2] & x1$Cluster == unames[j1],]$Count)
    }
  }
}

# Include the names for the columns (which correspond to OTUs (clusters))
# Include the names for the rows (which correspond to 'plot')

colnames(data.vegan) <- unames
rownames(data.vegan) <- uplots

# Get the diversity measure for each sample
# Shannon's
Shannon <- diversity(data.vegan)
Shannon

# Pielou's evenness, where log() gives the natural log (log10() gives log base 10)
Pielou <- Shannon/log(specnumber(data.vegan))
Pielou

# Simpson
Simp <- diversity(data.vegan, index="simpson")
Simp

# Inverse Simpson
InvSimp <- diversity(data.vegan, index="invsimpson")
InvSimp

#Species richness or OTU richness
data.vegan.bin <- data.vegan > 0
richness <- apply(data.vegan.bin,MARGIN=1,sum)
richness

# Fisher's alpha
Falpha <- fisher.alpha(data.vegan)
Falpha

# merge diversity measures with the experimental design information
div <- data.frame(Shannon, Pielou, Simp, InvSimp, richness, Falpha)
div$Plot <- as.numeric(row.names(div))

div.design <- merge(div,exp.design, by = "Plot")
```

```
# remove individual diversity indices to avoid potential confusion when attaching data
rm(Shannon,Pielou,Simp,InvSimp,richness,Falpha)
# div.design is attached because it is used frequently in the following commands
attach(div.design)

# Analyze environmental effect (fallow period) on diversity measures for two sites
# where one site has multiple samples with and without the presence of the
# plant type ‘thola’
# Examples are given for the Shannon index

# Regression analysis for the site Ancoraimes in absence of thola
# create index for Ancoraimes without Thola (ST)
indexAst <- Site=='Ancoraimes' & Thola=='ST'
summary(lm(Shannon[indexAst ] ~ Fallow.years[indexAst ]))

# Regression analysis for the site Umala in absence of thola
# create index for Umala without Thola (ST)
indexUst <- Site=='Umala' & Thola=='ST'
summary(lm(Shannon[indexUst] ~ Fallow.years[indexUst]))

# Regression analysis for the site Umala in presence of thola
# create index for Umala with Thola (CT)
indexUct <- Site=='Umala' & Thola=='CT'
summary(lm(Shannon[indexUct] ~ Fallow.years[indexUct]))

# Analyze biotic effect (presence or absence of thola) on diversity measures
# where only a subset of fields (1,4,7,19,21, and 23) have both samples in the
# presence of thola and samples in the absence of thola

UST <- div.design[Fieldnum == 1 | Fieldnum == 4 | Fieldnum == 7 | Fieldnum == 19 | Fieldnum
==21 | Fieldnum == 23, ]
UCT <- div.design[Fieldnum == 2 | Fieldnum == 3 | Fieldnum == 6 | Fieldnum == 18 | Fieldnum
==20 | Fieldnum == 22, ]

# Using paired t-test for the Shannon's index example
t.test(UST$Shannon,UCT$Shannon,paired=T)

## Diversity graphs
# Examples of fitted models are for fungi
# Graph diversity as a function of fallow period
# using indexes created above (indexAst, ...)

# Using the golden rectangle for the graph dimensions
par(pin=c(4*1.618,4))

# If you want to save the following plot as a PDF file and you are using the R terminal
# pdf ('Shannon’s diver_Umala_Ancor.pdf')
# the plot would be saved in the working directory
# the name inside the quotation marks could be changed
# the dev.off command must be used to close creation of the file, as pointed out below
# Using Shannon’s diversity index as example
plot(Fallow.years[indexAst],Shannon[indexAst],ylim=c(1.4,4),xlab="Fallow
years",ylab="Shannon’s diversity (H')",xlim=c(0,30),pch=7, col="red")
points(Fallow.years[indexUst],Shannon[indexUst],pch=6, col="blue")

# Adding regression line for Ancoraimes without thola
X <- 0:20
lines(X, 3.042461 +0.006355*X, col="red")

# Adding regression line for Umala without thola
X <- c(0,30)
lines(X,2.602674 -0.020365*X, col="blue")

# Labeling each site and including the corresponding model
text(16,3.9,'Ancoraimes', col="red")
text(16,3.7, 'y=3.042+0.006x')
text(16,3.5, expression(paste(R^2, '=0')))
text(16,3.3, 'p=0.54')

text(25,3.0,'Umala', col="blue")
text(25,2.8, 'y=2.602-0.020x')
text(25,2.6, expression(paste(R^2, '=0.18')))
text(25,2.4, 'p=0.05')

# if saving the plot using PDF format, then do not forget to add the following command
# dev.off()
# to complete creation of PDF

## Exporting the plot
# if using RStudio, a PDF file can be saved as follows
# click in Export, select ‘Save plot as PDF’ then click in the directory and select the PDF size,
orientation and place (Directory) where you want to keep the graph, choose the file name and
save it!
# for large figures choose PDF sizes that have 8 x 11 inches or more.

#if using R terminal window
# You have the option to ‘Save as metafile (.emf)’
# Or you can save the plot as PDF using the following code:

pdf ('Shannon’s diver_Umala_Ancor.pdf')  # plot would be saved in the working directory

plot(Fallow.years[indexAst],Shannon[indexAst],ylim=c(1.4,4),xlab="Fallow
years",ylab="Shannon’s diversity (H')",xlim=c(0,30),pch=7, col="red")
points(Fallow.years[indexUst],Shannon[indexUst],pch=6, col="blue")

# Adding regression line for Ancoraimes without thola
X <- 0:20
lines(X, 3.042461 +0.006355*X, col="red")

# Adding regression line for Umala without thola
X <- c(0,30)
```

```
lines(X,2.602674 -0.020365*X, col="blue")

# Labeling each site and including the corresponding model
text(16,3.9,'Ancoraimes', col="red")
text(16,3.7, 'y=3.042+0.006x')
text(16,3.5, expression(paste(R^2, '=0')))
text(16,3.3, 'p=0.54')

text(25,3.0,'Umala', col="blue")
text(25,2.8, 'y=2.602-0.020x')
text(25,2.6, expression(paste(R^2, '=0.18')))
text(25,2.4, 'p=0.05')

dev.off()   # to complete creation of PDF

# Graph diversity as a function of fallow period indicating presence or absence of thola

plot(Fallow.years[indexUst],Shannon[indexUst],ylim=c(1.4,3.5),xlab=' Fallow years',ylab="
Shannon’s diversity (H')",xlim=c(0,30),col='blue',pch=7)
points(Fallow.years[indexUct], Shannon[indexUct],col='darkgreen',pch=2)

# Adding regression line for Umala with (CT) thola
X <- c(10,30)
lines(X,2.484002 -0.001928*X, col='darkgreen')

# Adding regression line for Umala without Thola
X <- c(0,30)
lines(X,2.602674 -0.020365*X, col="blue")

# Labeling each line
text(18,2.0,'Umala: away from thola', col="blue")
text(18,1.85, 'y=2.602-0.020x')
text(18,1.7, expression(paste(R^2, '=0.18')))
text(18,1.55, 'p=0.05')

text(24,3.1,'Umala: under thola', col="darkgreen")
text(24,2.9, 'y=2.236-0.004x')
text(24,2.75, expression(paste(R^2, '=0')))
text(24,2.6, 'p=0.83')

# Detach data set div.design so other data sets can readily be used
detach(div.design)


### 3. Analysis of individual OTU frequencies ##################################

# Determine whether to include singletons by choosing DataOTU or DataOTUns
Data1 <- DataOTUns

# Select analysis of bacteria (TRUE) or fungi (FALSE)
Bacteria <- FALSE
```

```
# For bacteria, select confidence level cut-off for taxonomic assignment from RDP
# for bacteria, retain those OTUs for which the relevant confidence > 89
if (Bacteria){Data1 <- Data1[Data1$Confidence.7 > 89, ]}  # Confidence.7 corresponds to genus
# if (Bacteria){Data1 <- Data1[Data1$Confidence.3 > 89, ]}  # Confidence.3 corresponds to
phylum

# Select similarity cut-off for results from BLAST
# for fungi, retain those OTUs for which the similarity > 94
if(!Bacteria){Data1 <- Data1[Data1$Similarity > 94, ]}

# Some code below could be simplified, if modified after using the command: attach(Data1)

# Select taxonomic level and put taxonomic data in object RelCol
# RelCol <- Data1$Phylum
# RelCol <- Data1$Family
# RelCol <- paste(Data1$Genus,Data1$Species)
# RelCol<- Data1$Order
RelCol <- Data1$Genus

## Prepare a loop to evaluate generalized linear models for individual OTU frequencies

# number of samples being considered
Nplot <- 37 # to be modified, or calculated from dimensions of experimental design object
# create the list of plot names
pnames<- unique(Data1$Plot)

# For outer loop, find the number of unique names in the taxonomic column
Tnames <- unique(sort(RelCol)) # a vector of the unique taxa at the chosen taxonomic level
Tnum <- length(Tnames) # the number of unique taxa at the chosen taxonomic level

# Calculate the number of sequences per sample (Sseqnum)
# (An alternative would be to add this information, generated from programs like Pyrotagger or
# mothur, to the experimental design object, and read it in from there)

# Note that the order of plots is the same in both exp.design and Data1
# If that were not the case, would need to modify to match correctly

Taxnum <- 0*(1: Nplot) # this will contain the number of sequences for a given taxon for each
plot
# create a set of the essential treatment information that will be available in ‘one row per plot’
form
Sthola <- exp.design$Thola # Biotic treatment levels
Ssite <- exp.design$Site
Splot <- exp.design$Plot
Syears <- exp.design$Fallow.years
Sseqnum <- Taxnum # Sseqnum will contain the total number of sequences per sample

for (j in 1:Nplot){
   # make an index to find the rows that correspond to plot j
   pindex <- Data1$Plot == pnames[j]
   Sseqnum[j] <- sum(Data1$Count[pindex]) # the total number of sequences for plot j
 }

# create matrix for entering information for each taxon
# a row per a taxon
# placeholder columns labeled phold# are not used in this version,
# and could be replaced with another output value

Taxout <- as.data.frame(matrix(nrow=Tnum,ncol=22))
colnames(Taxout) <- c('taxon', 'test.p', 'STmean', 'CTmean', 'UmST.slope.p', 'UmST.slope',
'phold7', 'UmST.intercept',
'AnST.slope.p', 'AnST.slope', 'phold11', 'AnST.intercept', 'phold13', 'phold14', 'phold15',
'phold16', 'phold17', 'phold18', 'phold19', 'phold20', 'AnSTmean', 'UmSTmean')

# Col 1 – taxonomic label
# Cols 2-4 correspond to test of biotic effect (thola) in glm
# Col 2 – p-value from test
# Col 3 – average frequency away from thola (ST) in Umala samples
# Col 4 – average frequency under thola (CT) in Umala samples
# Cols 5-8  correspond to analysis of  environmental effect (fallow period)
#               on OTU frequency at site 1 (Umala)
# Col 5 – p-value for slope
# Col 6 – slope parameter
# Col 7 – placeholder – no data generated in this version
# Col 8 – intercept estimate
# Cols 9-12 – correspond to analysis of environmental effect (fallow period)
#               on OTU frequency at site 2 (Ancoraimes)
# Col 9 – p-value for slope
# Col 10 – slope parameter
# Col 11 – place holder – no data generated in this version
# Col 12 – intercept estimate
# Col 13-20 – placeholders – no data generated in this version
# Col 21 – average OTU frequency away from thola in Ancoraimes
# Col 22 – average OTU frequency away from thola in Umala

### Begin loop through each taxon in turn

# Determine whether (Do.thola <- T) or not (Do.thola <- F) to perform analyses of biotic impact,
# in this case analyses of the thola effect

Do.thola <- T

for (z in 1:Tnum){ # start of big loop

Taxon <- Tnames[z] # Select the taxon to be evaluated during this time through the loop
Taxout[z,1] <- as.character(Taxon) # start filling in the output matrix

# Start analyses specific to Taxon

# make an index for whether or not the row corresponds to Taxon
aindex  <- RelCol  == Taxon
```

```
# do for each plot j, to find the number of sequences in Taxon in that plot
for (j in 1:Nplot){
   # make an index to find the rows that correspond to plot j
   pindex <- Data1$Plot == pnames[j]
   # keep only those rows corresponding to plot j that are contigs grouped in Taxon
   tempdat <- Data1[pindex & aindex ,]
   # enter the number of members of Taxon for plot j in Taxnum
   Taxnum[j] <- sum(tempdat$Count)
}

# Divide by the total number of sequences per plot (Sseqnum) to get proportion in Taxon
# (a vector, one entry per plot)

Ptaxon <- Taxnum/Sseqnum

## For taxon z, analyze the biotic effect (presence or absence of thola) on frequency

if(Do.thola){
#Analysis of thola effect (site Umala only)
# output (1) p-value from GLM
# (2) estimated frequency in presence of thola
# (3) estimated frequency in absence of thola
# Criterion 1: taxon must appear in a minimum of three plots relevant to the analysis
# setting up the plots for inclusion in analysis in order by pairs, by plot number
UmalaSTP <- c(101,104,107,119,121,123) # members of pair in absence of thola
UmalaCTP <- c(102,103,106,118,120,122) # members of pair in presence of thola

# prepare to calculate mean frequency of taxon in presence and absence of thola
UmalaSTTax <- UmalaSTP * 0
UmalaCTTax <- UmalaSTP * 0
# create vectors with frequencies for the 6 paired samples in presence and absence of thola
for (x in 1:6){
   UmalaSTTax[x] <- Ptaxon[pnames==UmalaSTP[x]]
   UmalaCTTax[x] <- Ptaxon[pnames==UmalaCTP[x]]
}
STmean <- mean(UmalaSTTax) # mean in absence of thola
CTmean <-mean(UmalaCTTax) # mean in presence of thola
Taxout[z,3] <- STmean
Taxout[z,4] <- CTmean
T1 <- sum(UmalaSTTax >0) # num of samples in absence of thola where taxon freq is nonzero
T2 <- sum(UmalaCTTax>0) # num of samples in presence of thola where taxon freq is nonzero

# prepare input for GLM
# create index for those plots that are part of Thola pairs in Umala
Tindex <- (Splot < 105 | Splot == 106 | Splot == 107 | (Splot > 117 & Splot < 124))
# take the counts for the current taxon from each of these plots
TTaxnum <- Taxnum[Tindex]
# take the corresponding total number of sequences from each of these plots
TSseqnum <- Sseqnum[Tindex]
# create corresponding vector indicating whether thola was present (CT = thola, ST = no thola)
TSthola <- as.factor(Sthola[Tindex])
# create a vector indicating pairs of samples from the same field
Tttpair <- as.factor(c(1,1,2,2,3,3,4,4,5,5,6,6))

# Generalized linear model (GLM) with binomial family
# In R, the function glm fits GLMs, and the parameter ‘family’ specifies a
# choice of error distribution and link function, in this case binomial.

if(T1 + T2 > 3){
  glmthola <- glm(cbind(TTaxnum,(TSseqnum-TTaxnum)) ~ TSthola + Tttpair, family=binomial)
  glmthola.sum <- summary.glm(glmthola)
  ptout <- glmthola.sum$coefficients[2,4]
  # enter p-value in output matrix
   Taxout[z,2] <- ptout
}

# additional analyses could test for the importance of other variables, including inclusion of
# the pair label

} # end thola analysis

## For taxon z, analyze the environmental (fallow period) effect on frequency

## GLM for effect of environmental factor (fallow years) for site=Umala

# create index for Umala
IUmala <- Ssite == 'Umala'
#Create index for under thola (CT)
IThola <- Sthola == 'CT'
# combined index for site Umala in absence of thola
Tindex <- IUmala & !IThola

# the frequency taxon in each relevant plot
Tpro <- Ptaxon[Tindex]

# how many sequences correspond to taxon z in each relevant plot
TTaxnum <- Taxnum[Tindex]
# how many sequences total in each relevant plot
TSseqnum <- Sseqnum[Tindex]

# glm-  frequency of taxon z predicted by fallow years for Umala away from thola (ST)
# note that Syears can be replaced to evaluate other variables as predictors
if (sum(Tpro > 0) > 3) { # taxon present in at least three samples
   GLMUST.out <- glm(cbind(TTaxnum,(TSseqnum-TTaxnum)) ~ Syears[Tindex],
family=binomial)
  GLMUST.sum <- summary.glm(GLMUST.out)
   Taxout[z,9] <- GLMUST.sum$coefficients[2,4]
   Taxout[z,10] <- GLMUST.sum$coefficients[2,1]
   Taxout[z,12] <- GLMUST.sum$coefficients[1,1]
}

## GLM for effect of environmental factor (fallow years) for site=Ancoraimes
## This analysis repeats most of the analyses from site=Umala

# create index for Ancoraimes
IAncoraimes <- Ssite == 'Ancoraimes'
#Create index for under thola (CT)
IThola <- Sthola == 'CT'
# combined index for site Umala in absence of thola
Tindex <- IAncoraimes & !IThola

# the proportion taxon in each relevant plot
Tpro <- Ptaxon[Tindex]

# how many sequences correspond to taxon z in each relevant plot
TTaxnum <- Taxnum[Tindex]
# how many sequences total in each relevant plot
TSseqnum <- Sseqnum[Tindex]

# glm-  frequency of taxon z on fallow years for Ancoraimes away from thola (ST)
# note that Syears can be replaced to evaluate other variables as predictors
if (sum(Tpro > 0) > 3) { # taxon present in at least three samples
    GLMUST.out <- glm(cbind(TTaxnum,(TSseqnum-TTaxnum)) ~ Syears[Tindex],
family=binomial)
   GLMUST.sum <- summary.glm(GLMUST.out)
    Taxout[z,5] <- GLMUST.sum$coefficients[2,4]
    Taxout[z,6] <- GLMUST.sum$coefficients[2,1]
    Taxout[z,8] <- GLMUST.sum$coefficients[1,1]
}


# adding the overall non-thola means for each site
# for each of Ancoraimes and Umala, find all the frequencies in non-thola samples
# that correspond to this taxon
Taxout[z,21] <- mean(Ptaxon[Ssite=='Ancoraimes' & Sthola == 'ST'])
Taxout[z,22] <- mean(Ptaxon[Ssite=='Umala' & Sthola == 'ST'])

} # end of big loop


# For focal taxa, consider performing analysis of residuals and leverage

# To find rows with Ancoraimes slope p-value < 0.05 or Umala slope p-value < 0.05
Taxout.p <- Taxout[((!is.na(Taxout$UmST.slope.p) & Taxout$UmST.slope.p < 0.05 ) |
(!is.na(Taxout$AnST.slope.p) & Taxout$AnST.slope.p < 0.05)),]

# to find rows with thola GLM  p < 0.05
Taxout.t <- Taxout[(!is.na(Taxout$test.p) & Taxout$test.p < 0.05 ),]
# and just the p-values for thola that are not missing
Taxout.tpres <- Taxout[(!is.na(Taxout$test.p)),]
hist(Taxout.tpres$test.p)

# plotting mean frequency under thola vs. away from thola
```

```
plot(Taxout$STmean,Taxout$CTmean)
plot(log10(Taxout$STmean),log10(Taxout$CTmean))
lines(c(-3.5,-.5),c(-3.5,-.5))


## pie charts of most abundant taxa from each location

# To find most frequent taxa in Ancoraimes or Umala
Taxout[rev(order(Taxout$AnSTmean)),c(1,2,3,21,22)]
Taxout[rev(order(Taxout$UmSTmean)),c(1,2,3,21,22)]

# away from thola
pie.index <- Taxout$STmean > 0.01 # could replace 0.01 with other criteria
pie.Taxout <- Taxout[pie.index, ]
pie.col <- pie.Taxout$STmean
pie.x <- c(pie.col, 1-sum(pie.col))
pie.labels <- c(pie.Taxout$taxon, 'Other')
pie(x=pie.x,labels=pie.labels,main='Away from thola')
# under thola
pie.index <- Taxout$STmean > 0.01
pie.Taxout <- Taxout[pie.index, ]
pie.col <- pie.Taxout$CTmean
pie.x <- c(pie.col, 1-sum(pie.col))
pie.labels <- c(pie.Taxout$taxon, 'Other')
pie(x=pie.x,labels=pie.labels,main='Under thola')

# pies for intermediate frequency taxa
# away from thola
pie.index <- Taxout$STmean < 0.01 & Taxout$STmean > 0.002
pie.Taxout <- Taxout[pie.index, ]
pie.col <- pie.Taxout$STmean
pie.x <- c(pie.col, sum(Taxout$STmean[Taxout$STmean < 0.002]))
pie.labels <- c(pie.Taxout$taxon, 'Other')
pie(x=pie.x,labels=pie.labels,main='Away from thola – less common taxa')
# under thola
pie.index <- Taxout$CTmean < 0.01 & Taxout$CTmean > 0.002
pie.Taxout <- Taxout[pie.index, ]
pie.col <- pie.Taxout$CTmean
pie.x <- c(pie.col, sum(Taxout$CTmean[Taxout$CTmean < 0.002]))
pie.labels <- c(pie.Taxout$taxon, 'Other')
pie(x=pie.x,labels=pie.labels,main='Under thola – less common taxa')

## q values

# The q-value for a test addresses the proportion of false positives incurred (called the false
discovery rate) when that particular test is called significant. Controlling the false discovery rate
is important when there are many simultaneous tests
# In R, the qvalue package can be used to estimate the q-values for a given set of p-values.

source("http://bioconductor.org/biocLite.R")
biocLite("qvalue")
```

```
library (qvalue)

# get a p-value vector from Taxout
# starting with GLM p-value for thola
# create index for those values that are not missing
pthola.notNA <- !is.na(Taxout[,2])
pthola  <- Taxout[pthola.notNA,2]
j <- qvalue(p=pthola)
cbind(Taxout[pthola.notNA,1], j$pvalue, j$qvalue)

#exploring relationship between p-values and q-values
plot(j$pvalues,j$qvalues)
title('thola')
lines(c(0,1),c(0,1))
hist(j$pvalue)

# q-values from fallow years analysis in Umala
pUmST.notNA <- !is.na(Taxout[,5])
pUmST  <- Taxout[pUmST.notNA,5]
j2 <- qvalue(p=pUmST)
cbind(Taxout[pUmST.notNA,1], j2$pvalue, j2$qvalue)
plot(j2$pvalues,j2$qvalues)
title('Umala glm')
lines(c(0,1),c(0,1))
hist(j2$pvalue)

# q-values from fallow years analysis in Ancoraimes
pAnST.notNA <- !is.na(Taxout[,9])
pAnST  <- Taxout[pAnST.notNA,9]
j3 <- qvalue(p=pAnST)
cbind(Taxout[pAnST.notNA,1], j3$pvalue, j3$qvalue)
plot(j3$pvalues,j3$qvalues)
title('Ancoraimes glm')
lines(c(0,1),c(0,1))
hist(j3$pvalue)
# NOTE: qvalues do not appear to be a useful approach for this data set


### 4. Soil chemical and physical analyses ####################################

# the data set 'phys' was read in above

head(phys)

# merge physical/chemical data and the experimental design data

physdesign <- merge(phys,exp.design, by="Plot")

attach(physdesign)

# using the same indices created above
# create index for Ancoraimes without Thola (ST)
indexAst <- Site=='Ancoraimes' & Thola=='ST'
# create index for Umala without Thola (ST)
indexUst <- Site=='Umala' & Thola=='ST'
# create index for Umala with Thola (CT)
indexUct <- Site=='Umala' & Thola=='CT'

# For each response variable, evaluate the environmental effect (fallow period)

# Using organic matter (OM) as an example
# Using the response variable Organic Matter (OM) for Umala without (ST) thola
lmout <- lm(OM[indexUst] ~ Fallow.years[indexUst])
summary(lmout)

# Using the response variable Organic Matter (OM) for Umala with (CT) thola
lmout <- lm(OM[indexUct] ~ Fallow.years[indexUct])
summary(lmout)

# Using the response variable Organic Matter (OM) for Ancoraimes without (ST) thola
lmout <- lm(OM[indexAst] ~ Fallow.years[indexAst])
summary(lmout)

# T-test for comparison between Umala and Ancoraimes for organic matter without thola
US <- OM[indexUst]
AS<- OM[indexAst]
t.test(US, AS)

# For each response variable, evaluate the biotic effect (presence or absence of thola)
# T-test for comparison between ST and CT
UST <- physdesign[Fieldnum == 1 | Fieldnum == 4 | Fieldnum == 7 | Fieldnum == 19 | Fieldnum
==21 | Fieldnum == 23, ]
UCT <- physdesign[Fieldnum == 2 | Fieldnum == 3 | Fieldnum == 6 | Fieldnum == 18 | Fieldnum
==20 | Fieldnum == 22, ]

t.test(UST$OM,UCT$OM,paired=T)


## Graphs of chemical and physical responses

# Using the golden rectangle for the graph dimensions
par(pin=c(4*1.618,4))

range(OM) # check the range to determine good limits for the y axis

plot(Fallow.years[indexAst],OM[indexAst],ylim=c(0,5.5),xlab="Fallow years",ylab="Organic
Matter (%)",xlim=c(0,30),col="red",pch=7)
points(Fallow.years[indexUst], OM[indexUst],col='blue',pch=6)

# Adding regression line for Ancoraimes without thola  (ST)
X <- 0:20
lines(X, 2.29402 +0.07715*X, col="red")
```

```
# Adding regression line for Umala without thola (ST)
X <- c(0,30)
lines(X,0.570930 +0.010057*X, col='blue')

# Labeling each line for the graph
text(8,4.8,'Ancoraimes', col="red")
text(8,4.4, 'y=2.294+0.077x')
text(8,4.1, expression(paste(R^2, '=0.33')))
text(8,3.7, 'p=0.03')

text(23,2.8,'Umala', col="blue")
text(23,2.4, 'y=0.571+0.010x')
text(23,2.1, expression(paste(R^2, '=0.31')))
text(23,1.7, 'p=0.01')

detach(physdesign)

### 5. NMDS community analysis  ########################################
# (where the examples of format details refer to Gomez-Montano et al. 2013 fungal analyses)

# Non-metric multidimensional scaling (NMDS) is a method that takes an ordination of objects in full-dimensional space and represents them in a reduced number of dimensions, so that they can be visualized and interpreted, without assumptions about normality.
# http://cc.oulu.fi/~jarioksa/opetus/metodi/vegantutor.pdf

# The following commands use the object 'data.vegan' created above
# also using the object 'physdesign'
# confirm that the order by row is the same in both objects by checking Plot numbers

cbind(rownames(data.vegan),physdesign$Plot)

# Using the golden rectangle for the graph dimensions
par(pin=c(4*1.618,4))

# Perform nonmetric multidimensional scaling
vare.mds <- metaMDS(data.vegan, trace=2)

# Create a plot with points labeled by Site
barefig <- ordiplot(vare.mds, type='none',xlim=c(-1,1.2),ylim=c(-1,1.5))
points(barefig, "sites", pch=(1:2)[as.factor(physdesign$Site)])
lines(c(0.3,0.3),c(-1,1.5),lty=2)
text(1.5,1.3,'Ancoraimes')
text(-1.3,1.3,'Umala')

# Make another version of physdesign for better presentation in figure
physdesign2 <- physdesign
names(physdesign2)[c(5,8,14,16,17)] <- c('pH','Al','ECEC','SOM','N')
# Limit the numeric variables considered in 'physdesign2'
physdesign2 <-physdesign2[c(2:5,7:12,14,16:18,20)]
```

```
# Test the range of numeric env variables and plot those significant
ef <- envfit(vare.mds, physdesign2, permu = 999)
ef
plot(ef, p.max = 0.05,col='black') # labeling those with p <= 0.05

dev.off()

# Some other variations
plot(vare.mds, type = "t") # plots individual OTUs
plot(vare.mds, display = "sites") # plots sites

# Comparisons of biotic effect groups (thola and non-thola)

Tholalab <- as.character(physdesign$Thola)
Tholalab[Tholalab == 'CT'] <- 'thola'
Tholalab[Tholalab == 'ST'] <- 'non-thola'

plot(vare.mds, display = "sites", type = "p")
ordispider(vare.mds,Tholalab, label=T)

plot(vare.mds, display = "sites", type = "p")
ordiellipse(vare.mds,Tholalab, kind='se', conf=0.95, label=T)

plot(vare.mds, display = "sites", type = "p")
ordihull(vare.mds,Tholalab, lty=2, label=T)
```