**Feb. 20, 2022**

**A Brief Guide to the iNEXT.BetaDiv Package and R code for Graphics in Chao et al. (2022) Manuscript**

This guide introduces the main function in the iNEXT.BetaDiv package and demonstrates how to make graphics shown in Figures 1 to 5 in Chao et al.’s (2022) manuscript; all data and R code are available in Anne Chao’s Github repository <https://github.com/AnneChao/MS_iNEXT.BetaDiv>. The code in this guide was tested in R programme version 4.0.0. Before using the data and code, the following packages in CRAN must be installed and imported:

library(abind)

library(ggpubr)

library(plotly)

library(lmtest)

library(ggplot2)

library(openxlsx)

library(parallel)

library(reshape2)

library(gridExtra)

library(tidyverse)

library(geosphere)

library(patchwork)

library(future.apply)

Next, install and import the package “**iNEXT.3D”** and“**iNEXT.BetaDiv”** from Anne Chao’s Github. Please make sure to update required packages to their latest version (automatically or manually).

library(devtools)

install\_github("AnneChao/iNEXT.3D")   # Press 'enter' key to skip update options

library(iNEXT.3D)

install\_github("AnneChao/iNEXT.BetaDiv")   # Press 'enter' key to skip update options

library(iNEXT.BetaDiv)

In the following, we briefly introduce the main function in the “**iNEXT.BetaDiv”** package.

####################################################################

Function “iNEXTBetaDiv”⸺ computes size-based interpolated and extrapolated gamma and alpha diversity estimates, and also computes the corresponding coverage-based gamma, alpha and beta diversity estimates of orders q = 0, 1 and 2 with the following defaults: (Note that size-based standardized beta diversity is *not* a legitimate differentiation measure and thus omitted; see Chao et al. 2022 manuscript)

iNEXTBetaDiv(data, q = c(0,1,2), datatype = "abundance", base = "coverage", level = NULL, nboot = 20, conf = 0.95)

A description for each argument in the function (“iNEXTBetaDiv”) is given in the following table.

|  |  |
| --- | --- |
| **Argument** | **Description** |
| data | (a) For datatype = "abundance", data can be input as a matrix or data.frame (species by assemblages), or a list of matrices/data.frames, each matrix represents species-by-assemblages abundance matrix; see Note 1 for examples.  (b) For datatype = "incidence\_raw", data can be input as a list of matrices/data.frames, where each matrix represents species-by-sampling units; see Note 2 for an example. |
| q | a numerical vector specifying the diversity orders. Default is c(0, 1, 2). |
| datatype | data type of input data: individual-based abundance data (datatype = "abundance") or species by sampling-units incidence matrix (datatype = "incidence\_raw") with all entries being 0 (non-detection) or 1 (detection). |
| base | Sample-sized-based rarefaction and extrapolation for gamma and alpha diversity (base = "size") or coverage-based rarefaction and extrapolation for gamma, alpha and beta diversity (base = "coverage"). Default is base = "coverage". |
| level | A numerical vector specifying the particular value of sample coverage (between 0 and 1 when base = “coverage”) or sample size (base = “size”).  level = 1 (when base = “coverage”) means complete coverage (the corresponding diversity represents asymptotic diversity).  If base = “size” and level = NULL, then this function computes the gamma and alpha diversity estimates up to double the reference sample size. If base = “coverage” and level = NULL, then this function computes the gamma and alpha diversity estimates up to one (for q = 1, 2) or up to the coverage of double the reference sample size (for q = 0); the corresponding beta diversity is computed up to the same maximum coverage as the alpha diversity. |
| nboot | a positive integer specifying the number of bootstrap replications when assessing sampling uncertainty and constructing confidence intervals. Bootstrap replications are generally time consuming. Enter 0 to skip the bootstrap procedures. Default is 20. If more accurate results are required, set nboot = 100 (or nboot = 200). |
| conf | a positive number < 1 specifying the level of confidence interval. Default is 0.95. |

Use ‘?iNEXTBetaDiv’ for help.

Note 1: For example, to compare spatial beta diversity among 10 transects/assemblages in two plots (Madden and Jejuimi) taken from Gentry’s data, we input a list of two matrices/data.frames; each matrix represents species-by-transect abundance data (there are 10 transects in a plot). For Beetles data, to compare temporal beta diversity between two time periods for two areas (logged and unlogged), we input a list of two matrices/data.frames, each matrix represents species-by-period abundance records (there are 2 periods in an area).

Note 2: For example, to compare temporal beta diversity between a base year (2005) and any subsequent year (2006-2017) for six secondary forests. Here we input for each forest a list of 12 pairs of matrices/data.frames; each pair comprises two matrices: one represents species-by-subplot incidence records (there are 100 subplots) for 2015, and the other represents species-by-subplot incidence records (there are 100 subplots) for any subsequent year. All entries in any matrix being 0 (non-detection) or 1 (detection).

##################################################################################

The source code for graphics (Figures 1 to 5) in Chao et al. (2022) manuscript is provided in Anne Chao’s Github repository. First, import/load the source code:

source("Source R code.txt")

1. **Figure 1: Spatial beta diversity among 10 subplots/transects for two plots in Gentry’s data.**

Figure 1 (a) compares size-based rarefaction and extrapolation curves for gamma and alpha diversity based on species abundance data of two plots (Madden and Jejuimi) taken from Gentry’s data. Figure 1 (b) compares the corresponding sample-coverage-based rarefaction and extrapolation for gamma, alpha, and beta diversity.

**1a. R code for Figure 1a**

The complete set (197 plots) for Gentry’s data is stored as a list of 197 matrices in David Zelený’s webpage. Each matrix represents subplot (row) by species (column) abundance data; there are 10 subplots/transects in each plot. Run the following code to download Gentry’s data and select the data of two plots (Madden and Jejuimi). We first transpose each downloaded matrix to conform to the required input format (species-by-subplot) for the function “iNEXTBetaDiv”.

load (url ('https://raw.githubusercontent.com/zdealveindy/anadat-r/master/data/gentry197.r'))

gentry = gentry197[c('madden', 'jejuimi')]

gentry = lapply(gentry, function(k) t(k))

names(gentry) <- c("Madden", "Jejuimi")

To plot Figure 1 (a), use the function “iNEXTBetaDiv” (with base = "size") to calculate size-based standardized gamma and alpha diversity of orders *q* = 0, 1 and 2 based on 10 subplots, separately for Madden and Jejuimi plots. Then use function ‘fig\_1a\_or\_3a’ (provided in the source code) to plot Figure 1 (a).

output\_fig\_1a = iNEXTBetaDiv(gentry, datatype = 'abundance', base = 'size', nboot = 200)

fig\_1a\_or\_3a(output\_fig\_1a)

**1b. R code for Figure 1b**

To plot Figure 1 (b), use the function “iNEXTBetaDiv” (with base = "coverage") to calculate coverage-based standardized gamma, alpha and beta diversity of orders *q* = 0, 1 and 2 based on 10 subplots, separately for Madden and the Jejuimi plots. Then use the function ‘fig\_1b\_or\_3b’ (provided in the source code) to plot Figure 1 (b).

output\_fig\_1b = iNEXTBetaDiv(gentry, datatype = 'abundance', base = 'coverage', nboot = 200, level = seq(0.5,1,0.05))

fig\_1b\_or\_3b(output\_fig\_1b)

##################################################################################

1. **Figure 2:** **Latitudinal beta diversity gradient under six coverage values for Gentry’s 197 sites, each with 10 subplots/transects.**

To plot Figure 2 based on 197 plots in Gentry’s data, first run the following code to load Gentry’s 197 plots data and the latitude information from David Zelený’s webpage.

load (url ('https://raw.githubusercontent.com/zdealveindy/anadat-r/master/data/gentry197.r'))

gentry.coord <- read.delim ('https://raw.githubusercontent.com/zdealveindy/anadat-r/master/data/gentry.coord.txt', row.names = 1)

Then apply the main function “iNEXTBetaDiv” to calculate coverage-based standardized beta diversity and plot Figure 2 by using the following code:

gentry\_data <- data.frame()

for(i in 1:length(gentry197)) {

x <- t(gentry197[[i]])

out = iNEXTBetaDiv(x, level = seq(0.5,1,0.1), datatype = 'abundance', nboot = 0,conf = 0.95)

df <- out$Region\_1$beta[,c(1:4,6)]

m <- df[df$Method == "Observed\_alpha" & df$Order == 0 ,]$Size %>% round

df\_inter <- subset(df, Method == "Interpolated")

df\_obs <- subset(df, Method == "Observed")

df\_extra <- subset(df, Method == "Extrapolated")

for (j in 1:nrow(df\_extra)){

df\_extra[j,]$Method <- "Extrapolated(short-range)"

if (round(df\_extra[j,]$Size) > 2\*m ) { df\_extra[j,]$Method <- "Extrapolated(long-range)" }

}

Div <- rbind(df\_inter, df\_obs ,df\_extra)

physic <- gentry.coord[which(rownames(gentry.coord) == names(gentry197)[i]),]

df <- cbind(Div,

Latitude = physic$Lat, Longitude = physic$Long,

Elevation = physic$Elev,

Precip =physic$Precip,

Plot = names(gentry197)[i])

df = df %>% filter( Method == "Observed" | level %in% seq(0.5, 1, 0.1) )

gentry\_data <- rbind(gentry\_data,df)

}

fig\_2(gentry\_data)

##################################################################################

1. **Figure 3: Temporal beta diversity for two time periods based on beetle data.**

**3a. R code for Figure 3a**

Beetle data are stored in the file ("Data Beetles.xlsx"). The file includes two sheets (for logged and unlogged areas) each with a species (row) by time-period (column) abundance matrix. First, copy the file "Data Beetles.xlsx " into your working directory and load the data:

beetle = list('Logged' = read.xlsx('Data Beetles.xlsx', rowNames = T, sheet = 1),

'Unlogged' = read.xlsx('Data Beetles.xlsx', rowNames = T, sheet = 2))

To plot Figure 3 (a), use the function “iNEXTBetaDiv” (with base = "size") to calculate size-based standardized gamma and alpha diversity of orders *q* = 0, 1 and 2 for the two time periods, separately in each area (Logged and Unlogged). Then use function ‘fig\_1a\_or\_3a’ (provided in the source code) to plot Figure 3 (a).

output\_fig\_3a = iNEXTBetaDiv(beetle, datatype = 'abundance', base = 'size', nboot = 200)

fig\_1a\_or\_3a(output\_fig\_3a)

**3b. R code for Figure 3b**

To plot Figure 3 (b), use the function “iNEXTBetaDiv” (with base = "coverage") to calculate coverage-based standardized gamma, alpha and beta diversity of orders *q* = 0, 1 and 2 for the two time periods, separately in each area (Logged and Unlogged). Then use the function ‘fig\_1b\_or\_3b’ (provided in the source code) to plot Figure 3 (b).

output\_fig\_3b = iNEXTBetaDiv(beetle, datatype = 'abundance', base = 'coverage', nboot = 200, level= seq(0.8, 1, 0.025))

fig\_1b\_or\_3b(output\_fig\_3b)

##################################################################################

1. **Figure 4: Trajectories of temporal beta diversity over time (2005−2017) for tree species incidence data among 100 subplots in six second-growth rainforests.**

Tree species incidence data of six second-growth forests are stored in the excel file ("Data Second-growth forests.xlsx"). Use the following code to copy the data into your working directory and load the data:

Cuat.raw = read.xlsx("Data Second-growth forests.xlsx", sheet = 1)

LindEl.raw = read.xlsx("Data Second-growth forests.xlsx", sheet = 2)

Tiri.raw = read.xlsx("Data Second-growth forests.xlsx", sheet = 3)

LindSu.raw = read.xlsx("Data Second-growth forests.xlsx", sheet = 4)

FEB.raw = read.xlsx("Data Second-growth forests.xlsx", sheet = 5)

JE.raw = read.xlsx("Data Second-growth forests.xlsx", sheet = 6)

Then, use the function ‘SGF.data.transf’ (provided in the source code) to transform incidence raw data to conform to the data format of the “iNEXTBetaDiv” function. Also input the age of each forest.

inci.raw = list(SGF.data.transf(Cuat.raw %>% filter(Year >= 2005)),

SGF.data.transf(LindEl.raw %>% filter(Year >= 2005)),

SGF.data.transf(Tiri.raw %>% filter(Year >= 2005)),

SGF.data.transf(LindSu.raw %>% filter(Year >= 2005)),

SGF.data.transf(FEB.raw),

SGF.data.transf(JE.raw))

age = data.frame(Assem = c("Cuatro Rios", "Lindero el Peje", "Tirimbina",

"Lindero Sur", "Finca el Bejuco", "Juan Enriquez"),

Age = c(25,20,15,12,2,2))

For each forest, use following code to calculate temporal standardized beta diversity between 2005 (base time) and any subsequent year up to 2017 with orders 0, 1, and 2. Then use function ‘fig\_4’ (stored in the source code) to plot Figure 4.

cpu.cores <- detectCores() - 1

cl <- makeCluster(cpu.cores)

clusterExport(cl, varlist = c("inci.raw", "for\_fig\_4"), envir = environment())

clusterEvalQ(cl, c(library(tidyverse), library(iNEXT.BetaDiv), library(reshape2)))

forests.output = parLapply(cl, inci.raw, function(x) for\_fig\_4(x, nboot = 200))

stopCluster(cl)

fig\_4(forests.output)

##################################################################################

1. **Figure 5: Temporal and spatial gamma, alpha and beta diversity for fish species based on SWC-IBTS data from 1985 to 2010.**

Fish species abundance (in terms of catch per unit effort, CPUE, i.e., the number of individuals per species caught during a 1-hour trawl) data from 1985-2010 are stored in the file " Data Fish\_Lat55-60.csv". The data file consists of five columns: ‘LatBand’, ‘Year’, ‘SampleID’, ‘Species’, ‘Abundance’. Data in four latitudinal bands (55.5°, 56°, 56.5°, and 57°N) are pooled to form the South area, and data in another four latitudinal bands (58°, 58.5°, 59°, and 59.5°N) are pooled to form the North area. We also pooled annual data into two-year periods, i.e., we considered a total of 13 time periods/intervals: 1985-86, 1987-88, …, 2009-2010. Run the following code to load data:

fish <- read.csv("Data Fish\_Lat55-60.csv")

fish = fish %>% mutate(region = ifelse(LatBand %in% c(55.5, 56, 56.5, 57), 'South',

ifelse(!(LatBand %in% c(57.5, 60)), 'North', 'Other'))) %>% filter(region != 'Other')

groupyear = matrix(1985:2010, nrow = 2)

colnames(groupyear) = paste(groupyear[1,], groupyear[2,], sep = '~')

Sampling effort was standardized by randomly selecting 28 samples in each time period within any given area; the “iNEXTBetaDiv” function was applied to the pooled abundance (in terms of CPUE) data over these randomly selected samples. Under three standardized coverage values (0.99, 0.999, 1), temporal gamma, alpha, and beta diversity were computed between the first time period (1985-86, base time period) and each subsequent period; spatial gamma, alpha, and beta diversity were computed between the South and North areas within each time period. The random selection procedure of 28 samples was replicated 200 times to obtain the average values. Run the following code to obtain temporal and spatial gamma, alpha and beta diversity based on four estimation methods⸺ observed, standardized (at two coverage values, 99% and 99.9%) and asymptotic beta values. Note that the code is computationally demanding and may take up to several hours to complete (depending on the speed of your processor and the number of cores).

cpu.cores <- detectCores()-1

cl <- makeCluster(cpu.cores)

clusterExport(cl, varlist = c("rarefysamples", "fish", "groupyear"), envir = environment())

clusterEvalQ(cl, c(library(tidyverse), library(iNEXT.BetaDiv), library(reshape2)))

simu\_output = parLapply(cl, 1:200, function(k) {

region <- unique(fish$region)

TSrf <- list()

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

data2 <- fish[fish$region == region[i],]

TSrf[[i]] <- rarefysamples(data2)

}

names(TSrf) <- region

rf <- do.call(rbind, TSrf)

rf <- data.frame(rf, LatBand = rep(names(TSrf), times = unlist(lapply(TSrf, nrow))))

rf <- rf[!is.na(rf$Year),-1]

rownames(rf) <- NULL

data = rf

cov = c(0.99, 0.999, 1)

## ================== Temporal ================== ##

beta.temp = lapply(region, function(i) {

tmp = data %>% filter(LatBand %in% i)

tmp2 = lapply(2:length(unique(tmp$Year)), function(j) {

g1 = dcast(tmp %>% filter(Year == sort(unique(tmp$Year))[1]), Species ~ LatBand,

value.var = 'Abundance')

g2 = dcast(tmp %>% filter(Year == sort(unique(tmp$Year))[j]), Species ~ LatBand,

value.var = 'Abundance')

out = full\_join(g1, g2, by = 'Species')[,-1]

out[is.na(out)] = 0

out

})

names(tmp2) = sort(unique(tmp$Year))[-1]

return(tmp2)

})

names(beta.temp) = region

output.temp = lapply(1:length(beta.temp), function(i) {

result = iNEXTBetaDiv(beta.temp[[i]], q = c(0, 1, 2), datatype = 'abundance', level = cov, nboot = 0)

cbind(lapply(result, function(y) lapply(1:3, function(i) cbind(y[[i]], div\_type = names(y)[i])) %>% do.call(rbind,.)) %>% do.call(rbind,.) %>% filter(level %in% cov | Method == 'Observed'),

Latitude = names(beta.temp)[i])

}) %>% do.call(rbind,.)

output.temp$Region = as.numeric(output.temp$Region)

output.temp = rbind(output.temp,

output.temp %>% filter(Method == 'Observed', level %in% cov) %>%

mutate('Method' = paste('Observed\_', div\_type, sep = '')))

output.temp[output.temp$Method == 'Observed', 'level'] = 'Observed'

## ================== Spatial ================== ##

beta.spat = lapply( list( c('South', 'North') ), function(i) {

tmp1 = data %>% filter(LatBand %in% i[[1]])

tmp2 = data %>% filter(LatBand %in% i[[2]])

year = unique(data$Year) %>% sort

tmp = lapply(year, function(j) {

g1 = dcast(tmp1 %>% filter(Year == j), Species ~ LatBand, value.var = 'Abundance')

g2 = dcast(tmp2 %>% filter(Year == j), Species ~ LatBand, value.var = 'Abundance')

out = full\_join(g1, g2, by = 'Species')[,-1]

out[is.na(out)] = 0

out

})

names(tmp) = year

return(tmp)

})

names(beta.spat) = 'South vs. North'

output.spat = lapply(1:length(beta.spat), function(i) {

result = iNEXTBetaDiv(beta.spat[[i]], q = c(0, 1, 2), datatype = 'abundance', level = cov, nboot = 0)

cbind(lapply(result, function(y) lapply(1:3, function(i) cbind(y[[i]], div\_type = names(y)[i])) %>%

do.call(rbind,.)) %>% do.call(rbind,.) %>% filter(level %in% cov | Method == 'Observed'),

Latitude = names(beta.spat)[i])

}) %>% do.call(rbind,.)

output.spat$Region = as.numeric(output.spat$Region)

output.spat = rbind(output.spat,

output.spat %>% filter(Method == 'Observed', level %in% cov) %>%

mutate('Method' = paste('Observed\_', div\_type, sep = '')))

output.spat[output.spat$Method == 'Observed', 'level'] = 'Observed'

list("temporal" = output.temp[,c('Order', 'level', 'Region', 'div\_type', 'Latitude', 'Estimate')],

"spatial" = output.spat[,c('Order', 'level', 'Region', 'div\_type', 'Latitude', 'Estimate')])

})

stopCluster(cl)

output\_fig\_5 = list('temporal' = simu\_output[[1]]$temporal,

'spatial' = simu\_output[[1]]$spatial)

for (i in 2:length(simu\_output)) {

if (sum(simu\_output[[i]]$temporal == 'Inf') == 0)

output\_fig\_5$temporal = full\_join(output\_fig\_5$temporal, simu\_output[[i]]$temporal,

by = c('Order', 'level', 'Region', 'div\_type', 'Latitude'))

if (sum(simu\_output[[i]]$spatial == 'Inf') == 0)

output\_fig\_5$spatial = full\_join(output\_fig\_5$spatial, simu\_output[[i]]$spatial,

by = c('Order', 'level', 'Region', 'div\_type', 'Latitude'))

}

output\_fig\_5$temporal = cbind(output\_fig\_5$temporal[,1:5],

'Estimate' = apply(output\_fig\_5$temporal[,-(1:5)], 1, mean))

output\_fig\_5$spatial = cbind(output\_fig\_5$spatial[,1:5],

'Estimate' = apply(output\_fig\_5$spatial[,-(1:5)], 1, mean))

Based on the output, the following code leads to Figure 5(a) which depicts temporal and spatial beta diversity for orders *q* = 0, 1 and 2, based on four estimation methods⸺ observed, standardized (at two coverage values, 99% and 99.9%) and asymptotic beta values. By setting a goal coverage = 99.9%, we then obtain Figure 5(b) which depicts temporal and spatial gamma, alpha, and beta diversity for orders *q* = 0, 1 and 2 specifically for a standardized coverage value 99.9%.

fig\_5a(output\_fig\_5)

fig\_5b(output\_fig\_5, goalC = 0.999)