**A Brief Guide to the iNEXT.beta3D Package and R code for Graphics in Chao et al. (2023) Manuscript**

This guide introduces the main function in the iNEXT.beta3D package and demonstrates how to make graphics shown in Figures 2 to 6 in Chao et al.’s (2023) manuscript; all data and R code are available in Anne Chao’s Github repository [https://github.com/AnneChao/MS\_iNEXT.beta3D](https://github.com/AnneChao/MS_iNEXT.BetaDiv). To run the code, it is required R 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(readxl)

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.beta3D”** from Anne Chao’s Github. Please make sure to update required packages to their latest version.

library(devtools)

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

library(iNEXT.3D)

install\_github("AnneChao/iNEXT.beta3D")   # Press 'enter' key to skip update options

library(iNEXT.beta3D)

In the following, we briefly introduce the main function in the “**iNEXT.beta3D”** package.

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

Function “iNEXTbeta3D”⸺ 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. 2023 manuscript)

iNEXTbeta3D(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 (“iNEXTbeta3D”) 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 ‘?iNEXTbeta3D’ for help.

Note 1: Examples for abundance data input formats: (1) (Figure 2 for rainforest tree species abundance data) To compare beta diversity between 2 habitats/transects (Edge and Interior) in each of the two fragments (Marim and Rebio 2), we input a list of two matrices/data.frames; each matrix represents species-by-assemblage abundance data (there are 2 habitats/assemblages in a site). (2) (Figure 4 for beetle species abundance data) To compare temporal beta diversity for between two time periods for each of 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: An example for incidence data input formats: (Figure 5 for tree species incidence data) To compare temporal beta diversity between a base year (2005) and any subsequent year (2006-2017) in each of six secondary forests, 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 are 0 (non-detection) or 1 (detection).

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

The source code for graphics (Figures 2 to 6) in Chao et al. (2023) manuscript is provided in Anne Chao’s Github repository. First, import/load the source code:

source("Source R code.txt")

1. **Figure 2: Beta diversity between among 2 subplots for two sites in fragmentation of tropical rain forests in Brazil.**

Figure 2 (a) plots size-based rarefaction and extrapolation curves for gamma and alpha diversity based on tree species abundance data of two transects/habitats in Marim and Rebio 2 Fragments. Figure 2 (b) compares the corresponding sample-coverage-based rarefaction and extrapolation for gamma, alpha, and beta diversity.

**1a. R code for Figure 2a**

The complete data for 12 fragments are stored in an excel file named "Data rainforest trees". The file includes fragment size and tree-species abundance data for 2 habitats/transects (Edge and Interior) in each of the 12 fragments. In the paper, only 16 transects (8 Edge transects and 8 Interior transects) from fragments with area > 100 ha were used for illustration. First, run the following code to read tree species abundance data. We transpose the raw data to conform to the required input format (species-by-habitat) for the function “iNEXTbeta3D”.

raw\_data = read\_xlsx("Data rainforest trees.xlsx", sheet = 1)

data\_for\_beta = lapply(1:12, function(i) {

tmp = raw\_data[c(i, i+12), -(1:4)] %>% t

colnames(tmp) = c('Edge', 'Interior')

tmp

})

names(data\_for\_beta) = raw\_data$Site[1:12]

To plot Figure 2 (a), use the function “iNEXTbeta3D” (with base = "size") to calculate size-based standardized gamma and alpha diversity for orders *q* = 0, 1 and 2 based on 2 transects, separately for ‘Marim’ and ‘Rebio 2’ fragments. Then use function ‘fig\_2a\_or\_4a’ (provided in the source code) to plot Figure 2 (a).

output\_fig\_2a = iNEXTbeta3D(data\_for\_beta[c('Marim', 'Rebio 2')], diversity = 'TD', nboot = 200,

base = 'size')

fig\_2a\_or\_4a(output\_fig\_2a)

**1b. R code for Figure 2b**

To plot Figure 2 (b), use the function “iNEXTbeta3D” (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 ‘Marim’ and ‘Rebio 2’ sites. Then use the function ‘fig\_2b\_or\_4b’ (provided in the source code) to plot Figure 2 (b).

output\_fig\_2b = iNEXTbeta3D(data\_for\_beta[c('Marim', 'Rebio 2')], diversity = 'TD', nboot = 200,

base = 'coverage')

fig\_2b\_or\_4b(output\_fig\_2b)

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

1. **Figure 3:** **Fragment-size beta diversity gradient under six coverage values for 8 fragments, each with 2 habitats.**

To plot Figure 3 based on 8 fragments (filter out 4 sites by fragment size larger than 100-ha), we use the same abundance data used as in Figure 2 and the fragment size information from the original excel file.

Then apply the main function “iNEXTbeta3D” to calculate coverage-based standardized beta diversity and plot Figure 3 by using the following code:

output\_fig\_3 = iNEXTbeta3D(data\_for\_beta, "TD", q = c(0,1,2), nboot = 0,

level = c(0.6, 0.7, 0.75, 0.8, 0.9, 0.95, 1))

fig\_3(output\_fig\_3)

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

1. **Figure 4: Temporal beta diversity for two time periods based on beetle data.**

**3a. R code for Figure 4a**

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 4 (a), use the function “iNEXTbeta3D” (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\_2a\_or\_4a’ (provided in the source code) to plot Figure 4 (a).

output\_fig\_4a = iNEXTbeta3D(beetle, datatype = 'abundance', base = 'size', nboot = 200)

fig\_2a\_or\_4a(output\_fig\_4a)

**3b. R code for Figure 4b**

To plot Figure 4 (b), use the function “iNEXTbeta3D” (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\_2b\_or\_4b’ (provided in the source code) to plot Figure 4 (b).

output\_fig\_4b = iNEXTbeta3D(beetle, datatype = 'abundance', base = 'coverage', nboot = 200,

level= seq(0.8, 1, 0.025))

fig\_2b\_or\_4b(output\_fig\_4b)

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

1. **Figure 5: Trajectories of temporal beta diversity over time (2005−2017) for tree species incidence data among 100 subplots in six second-growth rainforests.**

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

Cuat.raw = read.xlsx("Data second-growth trees.xlsx", sheet = 1)

LindEl.raw = read.xlsx("Data second-growth trees.xlsx", sheet = 2)

Tiri.raw = read.xlsx("Data second-growth trees.xlsx", sheet = 3)

LindSu.raw = read.xlsx("Data second-growth trees.xlsx", sheet = 4)

FEB.raw = read.xlsx("Data second-growth trees.xlsx", sheet = 5)

JE.raw = read.xlsx("Data second-growth trees.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 “iNEXTbeta3D” 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\_5’ (stored in the source code) to plot Figure 5.

cpu.cores <- detectCores() - 1

cl <- makeCluster(cpu.cores)

clusterExport(cl, varlist = c("inci.raw", "for\_fig\_5"), envir = environment())

clusterEvalQ(cl, c(library(tidyverse), library(iNEXT.beta3D), library(reshape2)))

forests.output = parLapply(cl, inci.raw, function(x) for\_fig\_5(x, nboot = 200))

stopCluster(cl)

fig\_5(forests.output)

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

1. **Figure 6: 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 “iNEXTbeta3D” 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.beta3D), 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 = iNEXTbeta3D(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(SC %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', SC %in% cov) %>%

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

output.temp[output.temp$Method == 'Observed', 'SC'] = '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 = iNEXTbeta3D(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(SC %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', SC %in% cov) %>%

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

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

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

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

})

stopCluster(cl)

output\_fig\_6 = 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\_6$temporal = full\_join(output\_fig\_6$temporal,

simu\_output[[i]]$temporal,

by = c('Order.q', 'SC', 'Region', 'div\_type', 'Latitude'))

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

output\_fig\_6$spatial = full\_join(output\_fig\_6$spatial,

simu\_output[[i]]$spatial,

by = c('Order.q', 'SC', 'Region', 'div\_type', 'Latitude'))

}

output\_fig\_6$temporal = cbind(output\_fig\_6$temporal[,1:5],

'Estimate' = apply(output\_fig\_6$temporal[,-(1:5)], 1, mean))

output\_fig\_6$spatial = cbind(output\_fig\_6$spatial[,1:5],

'Estimate' = apply(output\_fig\_6$spatial[,-(1:5)], 1, mean))

Based on the output, the following code leads to Figure 6(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 6(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\_6a(output\_fig\_6)

fig\_6b(output\_fig\_6, goalC = 0.999)