Biodiversity Assessment Tools :: **Cheat Sheet**

Using BAT

Biodiversity Assessment Tools (BAT) assesses biodiversity data and provides analysis tools from species, phylogeny and convex-hulls or kernel density hypervolumes depicting the species relationship. **Pedro Cardoso**

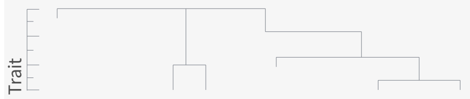
```
#Installation
install.packages("BAT")
library(BAT)
```

Data

Community, *comm*, sites x species
indices or abundance data about the species

Trait, *trait*, species x trait
trait values in each species in the community

Tree, *tree*
hierarchical cluster object based on a trait



Alpha Diversity

alpha(comm, tree, func = "nonparametric")
observe richness of the of the multiple sites

alpha.accum(comm, tree, func = "nonparametric")
estimates accumulated alpha diversity of a single site

alpha.estimate(comm, tree, func = "nonparametric")
estimates accumulated alpha diversity of a single site

optim.alpha(comm, tree, methods, base, …)
optimizes alpha diversity with different methods

optim.alpha.stat(comm, tree, methods, samples, …)
average alpha diversity observed for efficient statistics

Beta Diversity

beta(comm, tree, func = "jaccard")
observe the beta diversity of multiple sites

beta.accum(comm1, comm2, tree, func = "jaccard")
estimates accumulated beta diversity of a single site

beta.evenesss(comm, tree, distance, func = "camargo")
checks differences of evenness between pair of sites

beta.multi(comm, tree, func="jaccard")
observe with average of all pairwise values

optim.beta(comm, tree, methods, base, …)
optimizes beta diversity with different methods

optim.beta.stat(comm, tree, methods, …)
average absolute difference between sample and real beta diversity

Hypervolume

Building and analyzing the n-dimensional hypervolumes

Convex-Hull hypervolume

hull.build(comm, trait, …)
builds convex hull hypervolume for each community and trait data

hull.alpha(comm)
functional richness of one or multiple sites

hull.beta(comm, func = "jaccard")
pairwise decomposition beta diversity

hull.contribution(comm)
contribution of each species or individual to the total volume of a convex hull hypervolume

Kernel hypervolume

kernel.build(comm, trait, method = "gaussian", …)
builds kernel density hypervolume based on given trait data

kernel.alpha(comm)
functional richness of one or multiple sites

kernel.beta(comm, func = "jaccard")
pairwise decomposition beta diversity

kernel.beta.evenness(comm)
simple differences between pair of sites

kernel.evenness(comm, func = "jaccard")
functional eveness of the community

kernel.evenness.contribution(comm, func = "jaccard")
contribution of each species or individual to the total volume of a kernel density hypervolume

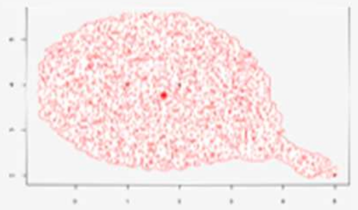
kernel.dispersion(comm, func = "dissimilarity")
calculates average distance to centroid or dissimilarities

kernel.originality(comm,…)
dissimilarity between a species or individuals

kernel.similarity(comm)
similarity indices between species and communities

kernel.contribution(comm, func = "jaccard")
contribution of each species or individual to the total volume of a kernel density hypervolume

kernel.hotspots(comm)
identify hotspots of one or more communities based on minimum volume needed



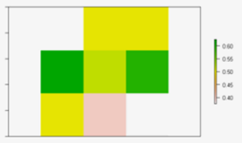
Raster

Observe based on a raster data from diversity samples

Raster

raster.alpha(layers, tree)
maps the alpha diversity using rasters of species distribution

raster.beta(layers, tree, func = "jaccard", …)
maps beta diversity using rasters of species distribution



raster.dispersion(layers, tree, distance, …)
maps averages dissimilarity between any two species or individuals

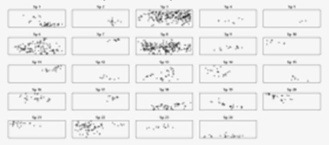
raster.evenness(layers, tree, distance, …)
maps the phylogenic/functional evenness of species or individuals

Simulation

Builds and simulates diversity based on artificial communities

Simulate

sim.plot(comm, …)
plots simulated species spatial distributions

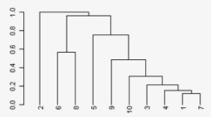


sim.sad(n, s, sad = "lognormal", …)
creates artificial communities following given SADs

sim.sample(comm, …)
simulates a sampling process from artificial communities

sim.spatial(n, s, sad = "lognormal", sd = 1, …)
simulates species spatial distributions

sim.tree(s, m = 100)
simulates a phylogenetic or functional tree



Assessments

gdm(comm, tree, area, time)
compares common supported models for the GDM

iaor(comm)
compares common support models for the IAOR

sar(comm, tree, area)
Compares some of most supported models for the SAR

sad(comm, …)
compares sites with different total abundances with possible rarefactions

hill(comm, …)
obtains hill numbers with possible rarefaction from multiple sites

uniqueness(comm, tree, distance, …)
dissimilarity between each species and the closest

dispersion(comm, tree, distance, …)
average dissimilarity between any two species or individuals randomly chosen

originality(comm, tree, distance, …)
average dissimilarity between a species or individual and all others

evenness(comm, tree, distance, …)
regularity of abundance and distances between species

contribution(comm, tree, …)
contribution of each species or individuals to total phylogenic or functional diversity

coverage(comm, tree)
checks completeness of the data set

fill(trait, method = "regression", …)
estimates the missing trait values NA based on different methods

tree.build(trait, …)
builds a functional tree from trait data

cwd(comm, trait, …)
calculates standard deviation values of each of a series of traits in multiple communities

cwe(comm, trait, func = "camargo", …)
calculates evenness of each of a series of traits in multiple communities

cwm(comm, trait, …)
averages value of each of a series of traits in multiple communities

aic(comm, trait, …)
calculates AIC of any model

gower(comm, trait, …)
Calculates Gower distances

rao(comm, trait, …)
calculates Rao quadratic entropy