Package ‘Select’

October 12, 2022

Title  Determines Species Probabilities Based on Functional Traits
Version 1.4

Description The objective of these functions is to derive a species assemblage that satisfies a functional trait profile. Restoring resilient ecosystems requires a flexible framework for selecting assemblages that are based on the functional traits of species. However, current trait-based models have been limited to algorithms that can only select species by optimising specific trait values, and could not elegantly accommodate the common desire among restoration ecologists to produce functionally diverse assemblages. We have solved this problem by applying a non-linear optimisation algorithm that optimises Rao Q, a closed-form functional trait diversity index that incorporates species abundances, subject to other linear constraints. This framework generalises previous models that only optimised the entropy of the community, and can optimise both functional diversity and entropy simultaneously. This package can also be used to generate experimental assemblages to test the effects of community-level traits on community dynamics and ecosystem function. The method is based on theory discussed in Laughlin (2014, Ecology Letters) <doi.org/10.1111/ele.12288>.

Depends  R (>= 3.3.0)
Imports Rsolnp, latticeExtra, grDevices, stats, lattice, graphics, ade4, FD
License GPL (>= 2)
Encoding UTF-8
LazyData true
RoxygenNote 6.0.1
Suggests knitr, rmarkdown, testthat
VignetteBuilder knitr
NeedsCompilation no
Author Daniel Laughlin [cre, aut], Loïc Chalmandrier [aut]
Maintainer Daniel Laughlin <danielclaughlin@gmail.com>
Repository CRAN
Date/Publication 2018-05-11 16:23:03 UTC
pineforest

R topics documented:

pineforest .................................................. 2
plotProbs .................................................... 3
Select ........................................................ 4
selectSpecies ................................................ 5
serpentine .................................................... 7
tropicalforest ............................................... 8

Index

pineforest  ............................................... 9

---

**Description**

A dataset containing two traits for species in a pine forest. The variables are as follows:

**Usage**

```
data(pineforest)
```

**Format**

dataframe of 46 species and 2 traits

**Details**

- species. species name
- flrdate. average flowering date in day of year
- leafcn. average leaf carbon-to-nitrogen ratio

**Source**

plotProbs

Plots results from the selectSpecies function

Description

This function plots results (species probabilities/optimum solutions) from the selectSpecies() function.

Usage

plotProbs(result, traits, colors = c("lightblue", "blue"), xlim = NULL, ylim = NULL, xlab = NULL, ylab = NULL, zlab = "Probability", distance = 0.3, cex.lab = 1.5, box.col = "transparent", xbase = 0.5, ybase = 0.5, ...)

Arguments

- result: A saved object from function selectSpecies()
- traits: A matrix of trait values where traits are columns and rows are species. If one trait is provided, then the function creates a 2D barplot of probabilities for each species. If two traits are provided, then the function creates a 3D barplot that illustrates probabilities of species located within a 2D trait space.
- colors: An optional vector of colors for plotting that must include at least two valid color names. The default color scheme is a ramp palette of lightblue to blue.
- xlim: Vector of two numbers denoting limits on x-axis.
- ylim: Vector of two numbers denoting limits on y-axis.
- xlab: Character string to describe x-axis.
- ylab: Character string to describe y-axis.
- zlab: Character string to describe z-axis. The default axis names is "Probabilities".
- distance: An optional number denoting distance between bars in 3d plot.
- cex.lab: An optional number denoting the size of the labels. The default is set to 1.5.
- box.col: An optional setting for the color of the box. The default setting is transparent.
- xbase: The length of the base of each 3d bar along the x-axis
- ybase: The length of the base of each 3d bar along the y-axis
- ...: Additional arguments to pass to barplot() or lattice::cloud()

Value

2D barplot of probabilities for each species or 3D barplot that illustrates probabilities of species located within a 2D trait space.
Examples

### 1 trait constraint with maximum functional diversity and entropy
Spp <- 5 #S = number of species
trait <- as.matrix(data.frame(trait=c(1:Spp)))
rownames(trait) <- c(letters[1:nrow(trait)])
result1 <- selectSpecies(t2c=trait, constraints=c(3.5), t2d=trait, obj="QH", capd=FALSE)
plotProbs(result1,trait, xlab="Trait")

##### 2 traits: Constrain trait X to value 2.5, diversify trait Y
traitX <- matrix(c(rep(1,3),rep(2,3),rep(3,3)))
traitY <- matrix(c(rep(c(1,2,3),3)))
rownames(traitX) <- c(letters[1:9]); colnames(traitX) <- c("traitX")
rownames(traitY) <- c(letters[1:9]); colnames(traitY) <- c("traitY")
result2 <- selectSpecies(t2c=traitX,constraints=c(traitX=2.5), t2d=traitY,capd=TRUE, obj="QH")
plotProbs(result2,traits = cbind(traitX, traitY))

Select

Select: Determines Species Probabilities Based on Functional Traits

Description

This package determines species probabilities (i.e., relative abundances) that satisfy a given functional trait profile. Restoring resilient ecosystems requires a flexible framework for selecting assemblages that are based on the functional traits of species. However, current trait-based models have been limited to algorithms that can only select species by optimising specific trait values, and could not elegantly accommodate the common desire among restoration ecologists to produce functionally diverse assemblages. We have solved this problem by applying a non-linear optimisation algorithm that optimises Rao Q, a closed-form functional trait diversity index that incorporates species abundances, subject to other linear constraints. This framework generalises previous models that only optimised the entropy of the community, and can optimise both functional diversity and entropy simultaneously.

Details

The Select package has two functions:

Select functions

The selectSpecies() function is used to generate species assemblages (relative abundances/probabilities) based on a functional trait profile. The function can constrain the abundances to conform to certain average trait values, and it can simultaneously optimize functional diversity.

The plotProbs() function plots the resulting species abundances (probabilities) using the resulting object from the selectSpecies() function.
**selectSpecies**

*Select species based on traits*

**Description**

This function returns a probability distribution for a species pool based on their traits and a desired trait profile (Laughlin 2014). It can simultaneously constrain specific trait value(s) and optimise functional diversity.

**Usage**

```r
selectSpecies(t2c = NULL, constraints = NULL, t2d, obj = c("QH", "Q", "H"), phi = 0.5, capd = FALSE, euclid = TRUE)
```

**Arguments**

- `t2c` "Traits to constrain": A matrix of species trait values, where species are organized as rows and traits as columns. These should be numeric arrays, and no missing values are tolerated. Ordinal data will yield meaningful results, but nominal data entered as numeric vectors are invalid. Row and column names are optional. This matrix can be any dimension, but the number of traits should be less than the number of species.

- `constraints` Trait constraints: A vector of trait values that serve as constants in the constraint equations. If not specified, the function will not constrain solutions to trait means. These trait contraints must be listed in the same order as columns in `t2c`. These constraints are community-weighted mean trait values, i.e., average traits weighted by the relative abundance of each species.

- `t2d` "Traits to diversify": Can be 1) a matrix of species trait values to diversify where species are organized as rows and traits as columns. NAs are tolerated as long as each pair of species can be compared by at least one trait. In this case, dissimilarities among species are computed using Euclidean distance. The number of species in `t2d` must match those in `t2c`. Or 2) a distance matrix of class ‘dist’ that contains dissimilarities among species, no NAs are tolerated in the distance matrix.

- `obj` Objective function: The objective function to optimise, one of three possibilities = c("QH", "Q", "H"). QH = Quadratic entropy (Q) plus entropy (H'); Q = Quadratic entropy; H = entropy.

- `phi` A parameter bounded between 0 and 1 that weights the importance of either quadratic entropy or entropy (default = 0.5). Phi of 1 corresponds to 100 percent Q, phi of 0.5 corresponds to 50 percent Q and 50 percent H', phi of 0 corresponds to 100 percent H'.

- `capd` A logical stating whether the distance matrix should be capped at the mean distance among species (default = FALSE). Mean distance is calculated as the average of all upper triangular entries in the distance matrix calculated from `t2d`.

- `euclid` A logical stating whether the distance matrix should be transformed to an Euclidean matrix if necessary (default = TRUE).
Value

A list with the elements:

- **prob** Probabilities, i.e. optimal solutions of species relative abundance
- **cwm_t2d** Community-weighted mean trait values of resulting community for traits that were diversified, computed as probabilities x t2d using matrix multiplication
- **cwm_t2c** Community-weighted mean trait values of resulting community for traits that were constrained, computed as probabilities x t2d using matrix multiplication
- **H** Final entropy of community
- **Q** Final Rao Q of community
- **objval** Values of the objective function being maximized. The last value is the maximum.
- **lagrange** Lagrange multipliers.
- **hessian** The Hessian at the optimal solution.

References


Examples

```r
### 1 trait constraint with maximum entropy
Spp <- 5  # S = number of species
trait <- as.matrix(data.frame(trait=c(1:Spp)))ownames(trait) <- c(letters[1:nrow(trait)])
result1 <- selectSpecies(t2c=trait, constraints=c(trait=3.5), t2d=trait, obj="H", capd=FALSE)
### compare result1 with virtually identical maxent output from FD package
#FD::maxent(constr=c(3.5),states=trait)$prob
### 1 trait constraint with maximum functional diversity
result2 <- selectSpecies(t2c=trait, constraints=c(trait=3.5), t2d=trait, obj="Q", capd=FALSE)
### 1 trait constraint with maximum functional diversity and entropy
result3 <- selectSpecies(t2c=trait, constraints=c(trait=3.5), t2d=trait, obj="QH", capd=FALSE)

### Plot results
plotProbs(result1,trait, xlab="Trait")
plotProbs(result2,trait, xlab="Trait")
plotProbs(result3,trait, xlab="Trait")

### 1 trait and no trait constraint
result4 <- selectSpecies(t2d=trait, obj="QH", capd=FALSE)
plotProbs(result4,trait, xlab="Trait")

##### 2 traits: Constrain trait X at X=3, diversify trait Y
traitX <- matrix(c(rep(1,4),rep(2,4),rep(3,4),rep(4,4)))
rownames(traitX) <- c(letters[1:16]); colnames(traitX) <- c("traitX")
rownames(traitY) <- c(letters[1:16]); colnames(traitY) <- c("traitY")
```
serpentine

result5 <- selectSpecies(t2c=traitX, constraints=c(traitX=3), t2d=traitY, obj="Q", capd=FALSE)
result6 <- selectSpecies(t2c=traitX, constraints=c(traitX=3), t2d=traitY, obj="QH", capd=TRUE)

trait.matrix <- cbind(traitX, traitY)
plotProbs(result5, trait.matrix)
plotProbs(result6, trait.matrix)

#### 3 traits: Constrain trait Z to value 2.5, diversify trait X and Y
traitZ <- as.matrix(data.frame(c(1,3,2,2,3,1,2,3,1,2,3,2,2)))
rownames(traitZ) <- c(letters[1:16]); colnames(traitZ) <- c("traitZ")
result7 <- selectSpecies(t2c=traitZ, constraints=c(traitZ=2.5), t2d=trait.matrix, capd=TRUE, obj="QH")
plotProbs(result7, trait.matrix)

---

serpentine

*Functional trait data for species in serpentine grassland*

**Description**

A dataset containing two traits for species in a serpentine grassland. The variables are as follows:

**Usage**

data(serpentine)

**Format**

dataframe of 48 species and 2 traits

**Details**

- species. species name
- wue. average water use efficiency, rate of carbon assimilation per unit of water used
- rootdepth. average rooting depth in soil

**Source**

tropicalforest  

Functional trait data for tree species in tropical forest

Description
A dataset containing two traits for species in a tropical forest. The variables are as follows:

Usage
data(tropicalforest)

Format
dataframe of 41 species and 2 traits

Details
• species. species name
• sla. specific leaf area
• wd. wood density

Source
Index

* datasets
  - pineforest, 2
  - serpentine, 7
  - tropicalforest, 8

pineforest, 2
plotProbs, 3
Select, 4
Select-package (Select), 4
selectSpecies, 5
serpentine, 7

tropicalforest, 8