Package 'Select'

February 23, 2026

Title Determines Species Probabilities Based on Functional Traits

Version 1.5

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:10.1111/ele.12288>.

Depends R (>= 3.3.0)

Imports Rsolnp, latticeExtra, grDevices, stats, lattice, graphics,
ade4

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, testthat, FD

VignetteBuilder knitr

NeedsCompilation no

Author Daniel Laughlin [cre, aut],
Loic Chalmandrier [aut]

Maintainer Daniel Laughlin <danielclaughlin@gmail.com>

Repository CRAN

Date/Publication 2026-02-23 19:20:08 UTC

Contents

Index **9**

---

pineforest	*Functional trait data for understory species in ponderosa pine forest*

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

Laughlin, D.C., J.J. Leppert, M.M. Moore, and C.H. Sieg. 2010. A multi-trait test of the leaf-height-seed plant strategy scheme with 133 species from a pine forest flora. Functional Ecology 24:493-501.

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))
```

---

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

```
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 perfect 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

Laughlin, D.C. 2014. Applying trait-based models to achieve functional targets for theory-driven ecological restoration. Ecology Letters, 17, 771-784.

Examples

```
### 1 trait constraint with maximum 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(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)))
traitY <- matrix(c(rep(c(1,2,3,4),4)))
rownames(traitX) <- c(letters[1:16]); colnames(traitX) <- c("traitX")
rownames(traitY) <- c(letters[1:16]); colnames(traitY) <- c("traitY")

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(1,3,2,2,3,1,2,3,1,2,1,3,2,3,2,2)))
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

Funk, J.L. & Wolf, A.A. 2016. Testing the trait based community framework: Do functional traits predict competitive outcomes? Ecology 97:2206-2211.

---

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

McCarthy, James K. 2018. Predicting the diversity and functional composition of woody plant communities under climate change PhD Thesis, School of Biological Sciences, The University of Queensland. Advisor: Dr. John Dwyer. doi:10.14264/uql.2018.134

Index