

Package ‘Select’

February 16, 2017

Title Determines Species Probabilities Based on Functional Traits

Version 1.1

Maintainer Daniel Laughlin <d.laughlin@waikato.ac.nz>

Description For determining species probabilities 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 biased toward 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’s Q, a closed-form functional diversity index that incorporates species abundances, subject to other linear constraints. This framework generalises previous models that could only optimise the entropy of the community, and can optimise both functional diversity and entropy simultaneously.

Depends R (>= 3.3.2)

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

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 5.0.1

NeedsCompilation no

Author Daniel Laughlin [aut, cre]

Repository CRAN

Date/Publication 2017-02-16 07:55:35

R topics documented:

plotprobs	2
select	3

Index	5
--------------	----------

plotprobs

Plots results from the select function

Description

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

Usage

```
plotprobs(result, traits)
```

Arguments

result	A saved object from function select()
traits	A matrix of trait values where traits are columns and rows are species. Maximum of 2 traits.

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 = select(t2c=trait, constraints=c(3.5), t2d=trait, obj="QH", capd=FALSE)
plotprobs(result1,trait)

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

select	<i>Select species based on traits</i>
--------	---------------------------------------

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

```
select(t2c, constraints, t2d, obj = "QH", phi = 0.5,
      traitConstraint = TRUE, capd = FALSE)
```

Arguments

t2c	Traits to constrain: A matrix of species trait values. Organize species as rows and traits as columns. 5 traits maximum.
constraints	Trait constraints: A vector of trait values that serve as constants in the constraint equations. 5 constraints maximum. Must be listed in same order as columns in t2c.
t2d	Traits to diversify: A matrix of species trait values to diversify. Organize species as rows, traits as columns. Can be any dimension (there is no upper limit to the number of traits to diversify).
obj	Objective function: The objective function to optimise, one of three possibilities = c("QH", "Q", "H"). QH = Quadratic entropy (Q) plus entropy (H ²); 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 ² , phi of 0 corresponds to 100 percent H ² .
traitConstraint	A logical stating whether solutions should be constrained to a trait mean (default = TRUE), when TRUE, a vector of constraints must be provided as an argument.
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.

Value

A list with the elements:

prob	Probabilities, i.e. optimal solutions
cwm	Final moment of constraint computed as prob x t2c using matrix multiplication.

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 = select(t2c=trait, constraints=c(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 = select(t2c=trait, constraints=c(3.5), t2d=trait, obj="Q", capd=FALSE)
### 1 trait constraint with maximum functional diversity and entropy
result3 = select(t2c=trait, constraints=c(3.5), t2d=trait, obj="QH", capd=FALSE)

### Plot results
plotprobs(result1,trait)
plotprobs(result2,trait)
plotprobs(result3,trait)

### 1 trait and no trait constraint
result4 = select(t2c=trait, t2d=trait, obj="QH", traitConstraint=FALSE, capd=FALSE)
plotprobs(result4,trait)

##### 2 traits: Constrain trait X at X=3, diversify trait Y
trait.matrix <- as.matrix(cbind(traitX=c(rep(1,4),rep(2,4),rep(3,4),rep(4,4)),
  traitY=c(rep(c(1,2,3,4),4))))
rownames(trait.matrix)=c(letters[1:16])
result5 = select(t2c=as.matrix(trait.matrix[,1]),constraints=c(3),
  t2d=as.matrix(trait.matrix[,2]),obj="Q",capd=FALSE)
result6 = select(t2c=as.matrix(trait.matrix[,1]),constraints=c(3),
  t2d=as.matrix(trait.matrix[,2]),obj="QH",capd=TRUE)
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)))
result7 = select(t2c=traitZ,constraints=c(2.5),t2d=trait.matrix, capd=TRUE, obj="QH")
plotprobs(result7,trait.matrix)
```

Index

plotprobs, 2

select, 3