Package 'virtualNicheR'

July 3, 2019

Type Package

Title virtualNicheR Package

Version 1.0

Maintainer Thomas R. Etherington <etheringtont@landcareresearch.co.nz>

Description The virtualNicheR package generates virtual fundamental and realised niches.

License MIT + file LICENSE

Depends R (>= 3.5)

Imports devtools,
diptest (>= 0.75-7)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests testthat,
knitr,
rmarkdown

NeedsCompilation no

R topics documented:

fund.niche 1
niche.grid.coords 3
rainfallMap 4
real.niche 4
temperatureMap 5

Index **7**

`fund.niche` *Fundamental Niche*

Description

Calculates the n-dimensional fundamental niche at `m` coordinates in niche space.

Usage

```
fund.niche(niche.coords, species)
```

Arguments

niche.coords	A m-by-n matrix of niche coordinates.
species	Ordered list of: maximum finite rate of increase, a means vector, and variance-covariance matrix that together define the fundamental niche of the species.

Details

Fundamental niche values are calculated on the basis of a multi-variate normal distribution, and hence are unimodal and convex in shape. The niche.coords can either be coordinates of specific interest, such as for mapping niche suitability for a study area, or for visualisation in niche space a grid of systematic coordinates as can be generated using niche.grid.coords. The species ordered list should contain:

- maximum finite rate of increase, the species' maximum finite rate of increase at the fundamental niche optimum,
- a n-by-1 column vector of means that gives the optimum location of the niche in each dimension, and
- a n-by-n variance-covariance matrix, that gives the size and orientation of the niche in each dimension.

Value

Returns a vector of length m containing the fundamental niche value for each of the input niche space coordinates.

References

Etherington TR, Omondiagbe, OP (2019) virtualNicheR: creating virtual fundamental and realised niches

See Also

Used internally by real.niche

Examples

```
# Create coordinates across niche space
niche.XY = niche.grid.coords(mins=c(15,0), maxs=c(35,200), nCoords=121)
# Define a species as a function of the maximum finite rate of increase,
# a means vector, and covariance matrix
lambdaMax = 2.5
meansVector = matrix(c(25, 100))
covarMatrix = matrix(data=c(9, 60,
                            60, 625), nrow=2, ncol=2, byrow = TRUE)
species = list(lambdaMax, meansVector, covarMatrix)
# Calculate the fundamental niche
fundNiche = fund.niche(niche.XY, species)
# Plot the fundamental niche
fundNicheMatrix = matrix(fundNiche, nrow=length(unique(niche.XY[,1])))
nContour = 10
```

```
filled.contour(unique(niche.XY[,1]), unique(niche.XY[,2]), fundNicheMatrix,
               levels = seq(0, lambdaMax, lambdaMax/nContour),
               col=colorRampPalette(c("gold", "firebrick"))(nContour),
               xlab=expression(paste("Temperature (", degree, "C)")),
               ylab="Rainfall (mm)",
               main ="Fundamental niche",
               key.title = title(main = expression(lambda)))

# Map the potential niche given maps of environmental variables
# Convert matrices of variables into columns
temp1D = matrix(temperatureMap, ncol=1)
rain1D = matrix(rainfallMap, ncol=1)
data.XY = cbind(temp1D, rain1D)
# Calculate the potential niche and form back in a 2D map
poteNiche = fund.niche(data.XY, species)
poteNiche2D = matrix(poteNiche, ncol=100)
filled.contour(z=poteNiche2D,
               levels = seq(0, lambdaMax, lambdaMax/nContour),
               col=colorRampPalette(c("gold", "firebrick"))(nContour),
               asp=1, plot.axes = {}, frame.plot=FALSE,
               main ="Map of the potential niche",
               key.title = title(main = expression(lambda)))
```

`niche.grid.coords` *Niche Grid Coordinates*

Description

Create an n-dimensional grid of coordinates across niche space.

Usage

```
niche.grid.coords(mins, maxs, nCoords)
```

Arguments

`mins`	Vector of length n listing the niche space minimum for each dimension.
`maxs`	Vector of length n listing the niche space maximum for each dimension.
`nCoords`	Number of coordinates across the niche space in all dimensions.

Details

This function creates a grid of coordinates systematically located throughout the specified niche space to enable visualisation of niche patterns. The extent of the grid is given by the `mins` and `maxs`, and the number of coordinates for each dimension is given by `nCoords`.

Value

A matrix with n columns.

Examples

```
# Niche space grid coordinates usage
niche.XY = niche.grid.coords(mins=c(15,0), maxs=c(35,200), nCoords=5)
```

rainfallMap *A randomly generated field representing a continuous rainfall gradient*

Description

A randomly generated field representing a continuous rainfall gradient

Usage

```
rainfallMap
```

Format

A two-dimensional matrix with 100 rows and 100 columns

...

real.niche *Realised Niche*

Description

Calculates the n-dimensional realised niche for a community of s species at m coordinates in niche space.

Usage

```
real.niche(niche.coords, community, interactions)
```

Arguments

niche.coords	A m-by-n matrix of niche coordinates.
community	List of length s containing definitions of the fundamental niches of species in the community.
interactions	A s-by-s matrix that defines the species interspecific interactions.

Details

The first step in calculating the realised niches is to calculate a fundamental niche for each species in the community list. This is done using the fund.niche function, so see there for a full description of how to define a the species fundamental niches.

The interactions matrix defines the form of the ecological interaction between each of the species in the community. These interactions can be either positive (+) or negative (-) in effect, and as the interactions can be asymmetrical there are a range of possible interspecific species interactions: competition (–), predator-prey or parasite-host (+-), commensalism (+0), amensalism (-0), or mutualism (++). When building the interactions matrix row 1 column 2 is the effects of species 2 on species 1, and conversely row 2 column 1 is the effects of species 1 on species 2, etc.

Please note that this function assumes that the species are consistently ordered in the community list and interactions matrix, and that no species effects itself so the diagonal of the interactions matrix has values equal to zero.

Value

Returns a m-by-s matrix realised niche value for each species at each of the input niche space coordinates.

References

Etherington TR, Omondiagbe, OP (2019) virtualNicheR: creating virtual fundamental and realised niches

See Also

Uses fund.niche

Examples

```
# Define the community
species1 = list(2.5, matrix(c(25, 100)), matrix(data=c(9, 60,
                                                  60, 625), nrow=2, ncol=2, byrow=TRUE))
species2 = list(5.0, matrix(c(28, 110)), matrix(data=c(4, -20,
                                                 -20, 500), nrow=2, ncol=2, byrow=TRUE))
species3 = list(3.0, matrix(c(25, 80)), matrix(data=c(4, 0,
                                                   0, 150), nrow=2, ncol=2, byrow=TRUE))
community = list(species1, species2, species3)

# Define the interactions
interactions = matrix(data=c(0.0, 0.2,-0.4,
                             -1.0, 0.0,-1.0,
                             -0.6, 0.3, 0.0), nrow=3, ncol=3, byrow = TRUE)

# Calculate the realised niche values
niche.XY = niche.grid.coords(mins=c(15,0), maxs=c(35,200), nCoords=121)
realNiche = real.niche(niche.XY, community, interactions)

# Plot the realised niche for species 1
realNiche1 = matrix(realNiche[,1], nrow=length(unique(niche.XY[,1])))
nContour = 10
filled.contour(unique(niche.XY[,1]), unique(niche.XY[,2]), realNiche1,
               levels = seq(0, 5, 5/nContour),
               col=colorRampPalette(c("gold", "firebrick"))(nContour),
               xlab=expression(paste("Temperature (", degree, "C)")),
               ylab="Rainfall (mm)",
               main ="Realised niche",
               key.title = title(main = expression(lambda)))
```

---

temperatureMap *A randomly generated field representing a continuous temperature gradient*

Description

A randomly generated field representing a continuous temperature gradient

Usage

```
temperatureMap
```

Format

A two-dimensional matrix with 100 rows and 100 columns

...

Index

∗Topic **datasets**
rainfallMap, 4
temperatureMap, 5

fund.niche, 1, *4*, *5*

niche.grid.coords, *2*, 3

rainfallMap, 4
real.niche, *2*, 4

temperatureMap, 5