

Package ‘sdmvspecies’

September 19, 2014

Type Package

Title Create virtual species for species distribution modelling

Version 0.3.1

Date 2014-01-05

Author Xiaoquan Kong, Renyan Duan, Minyi Huang

Maintainer Howl Anderson <u1mail2me@gmail.com>

Description This package includes several methods to create virtual species distribution map. Those maps can be used for species distribution modelling (SDM) study. SDM use environmental data for sites of occurrence of a species to predict all the sites where the environmental conditions are suitable for the species to persist, and may be expected to occur.

License GPL-2

Depends GPArotation

Imports ggplot2, psych, raster, parallel

Suggests testthat, roxygen2

NeedsCompilation no

Repository CRAN

Date/Publication 2014-09-19 10:39:18

R topics documented:

artificialBellResponse	2
autoPCA	3
configStack	4
nicheSynthese	5
pickMean	7
pickMedian	8

rescale	9
rescaleLayer	9
rescaleStack	10
sdmvspecies	10

Index	11
--------------	-----------

artificialBellResponse
artificialBellResponse

Description

artificial bell response method

Usage

```
artificialBellResponse(env.stack, config, stack = FALSE,
  compose = "product", rescale = TRUE)
```

Arguments

env.stack	a rasterStack object that contain the environment variable
config	config is a list or matrix or data.frame that contain config info, details see details part
stack	stack is an option that if you want not compose them together (result return as a rasterStack). Default is FALSE
compose	the method compose the suitability together. Default is product
rescale	if TRUE each environment will rescale before compose together

Details

This method mainly implement artificial bell response method, more detail see references.

Value

rasterLayer or rasterStack if stack is set to TRUE

References

Varela, S., Anderson, R. P., García-Valdés, R., & Fernández-González, F. (2014). Environmental filters reduce the effects of sampling bias and improve predictions of ecological niche models. *Ecography*.

Examples

```
# load the sdmvspecies library
library("sdmvspecies")
library("raster")
# find package's location
package.dir <- system.file(package="sdmvspecies")
# let see where is our sdmvspecies is installed in
package.dir
# find env dir under the package's location
env.dir <- paste(package.dir, "/external/env/", sep="")
# let see env dir
env.dir
# get the environment raster file
file.name <- files <- c("bio1.bil", "bio12.bil", "bio7.bil", "bio5.bil")
files <- paste(env.dir, file.name, sep="")
# make raster stack
env.stack <- stack(files)
# config
config <- list(c("bio1", 150, 50), c("bio12", 2000, 500), c("bio7", 400, 100), c("bio5", 300, 100))
# run pick mean
species.raster <- artificialBellResponse(env.stack, config)
# plot map
plot(species.raster)
# species distribution map
species.distribution.raster <- species.raster > 0.2
# plot map
plot(species.distribution.raster)
```

autoPCA

autoPCA

Description

easily used PCA analysis

Usage

```
autoPCA(env.stack, nfactors)
```

Arguments

env.stack	a rasterStack object that contain the environment variable
nfactors	Number of factors to extract

Details

This method implemented an easily used PCA analysis method

Value

rasterStack object

configStack	<i>configStack</i>
-------------	--------------------

Description

output config layers as rasterStack

Usage

```
configStack(env.stack, config)
```

Arguments

env.stack	a rasterStack object that contain the environment variable
config	config is a list or matrix or data.frame that contain config info, details see details part

Details

This method will extract rasterLayer according to config, then output rasterStack as result

Value

rasterStack object

Examples

```
# load the sdmvspecies library
library("sdmvspecies")
library("raster")
# find package's location
package.dir <- system.file(package="sdmvspecies")
# let see where is our sdmvspecies is installed in
package.dir
# find env dir under the package's location
env.dir <- paste(package.dir, "/external/env/", sep="")
# let see env dir
env.dir
# get the environment raster file
env.files <- list.files(env.dir, pattern="*.bil$", full.names=TRUE)
# see the file list
env.files
# put the environment file in a raster stack,
# which require all the environment should have same resolution and extend
env.stack <- stack(env.files)
```

```
# let see the env.stack var
env.stack
# here let's configure the environment response function and weight
config <- list(c("bio1", "1", 0, 100), c("bio11", "2", -100, NULL))
env.stack <- configStack(env.stack, config)
plot(env.stack)
```

nicheSynthese

nicheSynthese

Description

niche synthese method

Usage

```
nicheSynthese(env.stack, config, stack = FALSE, random.error = FALSE)
```

Arguments

<code>env.stack</code>	a rasterStack object that contain the environment variable
<code>config</code>	config is a list or matrix or data.frame that contain config info, details see details part
<code>stack</code>	stack is an option that if you want not compose them together (result return as a rasterStack). Default is FALSE
<code>random.error</code>	add random error on cell or not. Default is FALSE

Details

This method mainly implement niche synthese method, for more details see references

You can write several paragraphs.

Value

rasterLayer or rasterStack if stack is set to TRUE

References

Hirzel, A. H., Helfer, V., & Metral, F. (2001). Assessing habitat-suitability models with a virtual species. *Ecological modelling*, 145(2), 111-121.

Examples

```

# load the sdmvspecies library
library("sdmvspecies")
library("raster")
# find package's location
package.dir <- system.file(package="sdmvspecies")
# let see where is our sdmvspecies is installed in
package.dir
# find env dir under the package's location
env.dir <- paste(package.dir, "/external/env/", sep="")
# let see env dir
env.dir
# get the environment raster file
env.files <- list.files(env.dir, pattern="*.bil$", full.names=TRUE)
# see the file list
env.files
# put the environment file in a raster stack,
# which require all the environment should have same resolution and extend
env.stack <- stack(env.files)
# let see the env.stack var
env.stack
# here let's configure the environment response function and weight
config <- list(
  c("bio1", "1", 2),
  c("bio14", "2", 2),
  c("bio5", "3", 1),
  c("bio11", "4", 2),
  c("bio16", "5", 1)
)
# call the niche synthesis method
species.raster <- nicheSynthese(env.stack, config)
# let see the result raster,
# you should noticed that it's continue value map not distributin map
species.raster

# write the map to file, so you can use it latter in GIS software
# or further analysis.
#
#writeRaster(species.raster, "synthese.img", "HFA", overwrite=TRUE)

# to make binary distribution map, you should chosee a threshold to make map
# see the map then to decide the threshold to binary
plot(species.raster)
# choice threshold, here we choice 4
threshold <- 14
# make binary map
distribution.map <- species.raster > threshold
# plot the map out
plot(distribution.map)

```

pickMean	<i>pickMean</i>
----------	-----------------

Description

pick mean method

Usage

```
pickMean(env.stack, subset = NULL, stack = FALSE)
```

Arguments

env.stack	a rasterStack object that contain the environment variable
subset	subset is a string vector that contain environment variables names which into calculate, if NULL that all var in env.stack will calculate.
stack	stack is an option that if you want not compose them together (result return as a rasterStack). Default is FALSE

Details

This method mainly implement pick mean method

Value

rasterLayer or rasterStack if stack is set to TRUE

References

Jiménez-Valverde, A., & Lobo, J. M. (2007). Threshold criteria for conversion of probability of species presence to either-or presence-absence. *Acta oecologica*, 31(3), 361-369.

Examples

```
# load the sdmvspecies library
library("sdmvspecies")
library("raster")
# find package's location
package.dir <- system.file(package="sdmvspecies")
# let see where is our sdmvspecies is installed in
package.dir
# find env dir under the package's location
env.dir <- paste(package.dir, "/external/env/", sep="")
# let see env dir
env.dir
# get the environment raster file
files <- list.files(path=env.dir, pattern="*.bil$", full.names=TRUE)
# make raster stack
```

```
env.stack <- stack(files)
# run pick mean
species.raster <- pickMean(env.stack)
# plot map
plot(species.raster)
```

pickMedian

pickMedian

Description

pick median method

Usage

```
pickMedian(env.stack, subset = NULL, stack = FALSE)
```

Arguments

<code>env.stack</code>	a rasterStack object that contain the environment variable
<code>subset</code>	subset is a string vector that contain environment variables names which into calculate, if NULL that all var in env.stack will calculate.
<code>stack</code>	stack is an option that if you want not compose them together (result return as a rasterStack). Default is FALSE

Details

This method mainly implement pick median method

Value

rasterLayer or rasterStack if stack is set to TRUE

References

Lobo, J. M., & Tognelli, M. F. (2011). Exploring the effects of quantity and location of pseudo-absences and sampling biases on the performance of distribution models with limited point occurrence data. *Journal for Nature Conservation*, 19(1), 1-7.

Examples

```
# load the sdmvspecies library
library("sdmvspecies")
library("raster")
# find package's location
package.dir <- system.file(package="sdmvspecies")
# let see where is our sdmvspecies is installed in
package.dir
```



```

# find env dir under the package's location
env.dir <- paste(package.dir, "/external/env/", sep="")
# let see env dir
env.dir
# get the environment raster file
files <- list.files(path=env.dir, pattern="*.bil$", full.names=TRUE)
# make raster stack
env.stack <- stack(files)
# run pick mean
species.raster <- pickMedian(env.stack)
# plot map
plot(species.raster)

```

rescale

rescale

Description

rescale the RasterStack or RasterLayer values to min:0 max:1

Usage

```
rescale(raster.object)
```

Arguments

`raster.object` an object of RasterStack or RasterLayer class

Value

an object of RasterStack or RasterLayer that rescaled.

rescaleLayer

rescaleLayer

Description

rescale the RasterLayer values to min:0 max:1

Usage

```
rescaleLayer(raster.layer)
```

Arguments

`raster.layer` an object of RasterLayer class

Value

an object of RasterLayer that rescaled.

rescaleStack *rescaleStack*

Description

rescale the RasterStack values to min:0 max:1

Usage

```
rescaleStack(raster.stack)
```

Arguments

raster.stack an object of RasterStack class

Value

an object of RasterStack that rescaled.

sdmvspecies *SDMvspecies*

Description

Species Distribution Modelling (SDM) tools for Virtual Species (vspecies)

Details

This package contain some useful tools for create and study virtual species in SDM

SDMvspecies is a tools package for creating virtual species in Species Distribution Modelling (SDM) It contains several algorithms (methods) that already report and used in current virtual species study or application. Also many useful tools are include to help user development new algorithms (methods) and study virtual species.

Author(s)

Xiaoquan Kong

Renyan Duan

Minyi Huang

Index

[artificialBellResponse](#), [2](#)
[autoPCA](#), [3](#)

[configStack](#), [4](#)

[nicheSynthese](#), [5](#)

[pickMean](#), [7](#)
[pickMedian](#), [8](#)

[rescale](#), [9](#)
[rescaleLayer](#), [9](#)
[rescaleStack](#), [10](#)

[sdmvspecies](#), [10](#)
[sdmvspecies-package \(sdmvspecies\)](#), [10](#)