Package ‘PseuAbs’

January 31, 2022

Type Package

Title Flexibly sample pseudoabsences with diversrse strategies

Version 0.1.0

Depends R (>= 3.1.0), raster, dismo, spatstat

Maintainer Philipp Brun <philipp.brun@wsl.ch>

Description This package allows conveniently sampling pseudoabsences according to the methods introduced by Descombes et al.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

R topics documented:

`copy_PseuAbs` *Copy pseudoabsences from an existing wsl.pseudoabsences object*

Description

Copy pseudoabsences from existing wsl.pseudoabsences object to a wsl.pseudoabsences object generated for new species and its presence observations.

Usage

```
copy_PseuAbs(env.stack, pres = SpatialPoints(), taxon = character(), x)
```

Arguments

`env.stack` RasterStack/RasterBrick with environmental layers for sampling and extraction. Needs to contain the same layers as in the wsl.pseudoabsences object.

`pres` SpatialPoints object with presence observations.

`taxon` Character; name of taxon of interest to store in meta information.

`x` A wsl.pseudoabsences object

Details

if the desired pseudoabsence sampling strategy is not species-specific it may be more efficient to copy on the sampled pseudoabsences from another object. Note that this function does not allow for any thinning of presence observations.

Value

an object of class 'wsl.pseudoabsences'

Author(s)

Philipp Brun

Examples

```
### =====================================================================
### Data preparation
### =====================================================================

# Predictors
bio=raster::getData('worldclim',var='bio',lon=16, lat=48,res=.5)
bio=bio[[c(1,4,12)]]

# install.packages("rgbif")
library(rgbif)
# extract species
spn1='Boletus aestivalis'
xt=as.vector(extent(bio))
baest <- occ_search(scientificName=spn1,
                  hasCoordinate=TRUE,
                  decimalLongitude=paste0(xt[1],",",xt[3]),
                  decimalLatitude=paste0(xt[2],",",xt[4]))

pbaest=baest$data[,c('decimalLongitude','decimalLatitude')]
baest_spp=SpatialPoints(pbaest,proj4string = crs(bio))

spn2='Boletus edulis'
bedu <- occ_search(scientificName=spn2,
                  hasCoordinate=TRUE,
                  decimalLongitude=paste0(xt[1],",",xt[3]),
                  decimalLatitude=paste0(xt[2],",",xt[4]))
```

```
pbedu=bedu$data[,c('decimalLongitude','decimalLatitude')]
bedu_spp=SpatialPoints(pbedu,proj4string = crs(bio))

### ========================================================================
### Sample pseudoabsences
### ========================================================================

# Geo.start
pseu.abs1=wsl.samplePseuAbs(type="geo.strat",
                            n=5000,
                            env.stack=bio,
                            pres=baest_spp,
                            geores_fact=3,
                            add.strat=0,
                            taxon=spn1)

plot(pseu.abs1)

pseu.abs2=copy_PseuAbs(env.stack=bio,
                       pres=bedu_spp,
                       taxon=spn2,
                       x=pseu.abs1)

plot(pseu.abs2)
```

plot.wsl.pseudoabsences

A simple plotting function for objects of class 'wsl.pseudoabsences'

Description

This function allos quickly visualizing the different thinning and sampling strategies determine the spatial patterns of presences and pseudoabsences.

Usage

```
plot.wsl.pseudoabsences(object)
```

Arguments

`object`	An object of class wsl.pseudoabsences

Value

Presences (red crosses) and pseudoabsences (grey transparent points) are plotted in geographic space. If the wsl.pseudoabsences object contains information on a template raster file (see documentation of the wsl.samplePseuAbs function), this template will be plotted in the background.

wsl.pseudoabsences-class *An S4 class to store pseudoabsence data*

Description

Information on coordinates, presence/pseudoabsence, and environmental conditions is organized in seperate slots. Moreover, the slot meta keeps track of meta information.

Slots

- `meta` a list with meta information
- `pa` a vector with 1 encoding presences and 0 encoding pseudoabsences
- `env_vars` a data.frame containing extractions of the env.stack at the points xy
- `xy` two-column matrix with coordinates of the presence points and the sampled pseudoabsence points. The projection is the same as the one of the env.stack supplied
- `call` the call made to the wsl.samplePseuAbs function

Author(s)

Philipp Brun

wsl.samplePseuAbs *Sample pseudoabsences using various strategies*

Description

Flexlible function to sample pseudoabsences with various strategies and and thin presences and pseudoabsences with flexible distance constraints. This is the core function of the PseuAbs package.

Usage

```
wsl.samplePseuAbs(
  n = 10000,
  env.stack,
  type = "geographic",
  add.strat = 0,
  pres = numeric(),
  taxon = character(),
  geodist_fact = 1,
  geores_fact = 1,
  template_dir = tempdir(),
  geo_nrep = 7,
  target.group_dir = NA,
  env.strat_path = NA,
  rAll = TRUE,
  force_spat_thin = "no",
  limdist = NA,
  set_max_npres_to_nabs = TRUE
)
```

Arguments

n Positive integer; number of pseudoabsence points desired. Default is 10000.

env.stack RasterStack/RasterBrick with environmental layers for sampling and extraction

type Character; desired sampling strategy. Options are 'geographic', 'density', 'random', 'target.group', 'geo.strat', 'env.strat' and 'env.semi.strat' (see details). Default is 'geographic'.

add.strat Fraction between 0 and 1; should strategy be complemented by a fraction of environmental strata? (Does not apply when type env.strat or env.semi.strat is chosen)

pres SpatialPoints object; location of presence points. Necessary for 'geographic' and 'density' strategies, otherwise optional. Presence points can be thinned according to adhere to flexible distance constraints.

taxon Character (optional); name of taxon of interest to keep track of in meta information.

geodist_fact Positive floating point number to adjust spatial autocorrelation lengths: for 'geographic' pseudoabsence point patterns, values below 1 increase autocorrelation length; values above 1 decrease it; for 'density' sampling it is the other way around.

geores_fact Positive integer; aggregation factor for template raster from which pseudoabsences are sampled (for 'geographic', 'density', 'random', and 'geo.strat' strategies). Larger values save computation time, but derease resolution of sampling points.

template_dir Character; directory where template raster should be saved in/loaded from. Default is tempdir(). The template will be saved in/loaded from the directory depending on whether a template has already been created by a previous function call.

geo_nrep Positive Integer; number of replicates of models fitted for the 'geographic' strategy. More will create a smoother pattern but increase computation time. Default is seven.

target.group_dir
Character; directory where xy files of traget group taxa are stored. Must be supplied if sampling stragegy is 'target.group', must contain a column names 'x' and 'y' with coordinates in the same projection as the RasterStack/RasterBrick provided to the env.stack argument.

env.strat_path Character; directory where sample of environmental strata for 'env.strat' or 'env.semi.strat' sampling should be saved in/loaded from. If NA, nothing will be saved; if provided, environmental strata will be saved in/loaded from directory depending on whether a file has been created by a previous function call.

`rAll` Boolean; should all raster data be read into memory for computation of environmental strata? This is faster but you may run into memory issues for large rasters. Default is TRUE.

`force_spat_thin`
Character; should minimum distance be enforced between points? Options are 'no', 'presences', 'absences', and 'both'. By default thinning is defined for pseudoabsences from 'geographic', 'density', 'random', and 'geo.strat' methods with minimum distance according to the resoultion of the template raster (see argument geores_fact). 'presences' takes the minimum distance criterion from the template raster over to the 'presence' points; 'absences' takes the criterion over to 'env.strat', 'env.semi.strat', and 'target.group'; 'both' does it for both.

`limdist` Positive float; the minimum distance accepted for spatial thinning. Units should be km if the spatial data is projected, otherwise the units of the coordinate reference system are used. If no value is supplied, the maximum distance between two cell centres of the template raster will be taken (see above).

`set_max_npres_to_nabs`
Boolean; should the maximum number of presences be equal to the number of pseudoabsences defined. Default is TRUE.

Details

'geographic' samples pseudoabsences with a sampling probability. inversely proportional to the geographic distance to presence observations. density' samples pseudoabsences proportional to the density of presence observations. 'random' samples pseudoabsences randomly with a sampling probability proportional the area of the cells. 'target.group' samples pseudoabsences from the presences of the taxa of the target group, attempting to correct for sampling bias. It depends on a directory with taxa defined by the user as target group. 'geo.strat' samples pseudoabsences geographically stratified either on a plane, or on a sphere depending on the projection of the supplied env.stack. 'env.strat' samples pseudoabsences environmentally stratified. Points are sampled form all realized combinations of environmental conditions occuring in the environmental stack that have a minimal occurrence frequency. Environmental strata are calculated based on all raster layers supplied. If a directory is supplied as 'env.strat_path', a large sample of stratified points will be saved to speed up computations for follow-up species. If environmental strata based on different predictors than supplied are preferred 'env.strat_path' can be an .RData file from a previous sampling of strata form different environmental predictors. 'env.semi.strat' is similar to 'env.strat' but samples environmental strata proportional to the logarithm of the area they cover. See Descombes et al. for more details.

Value

a S4 object of class 'wsl.pseudoabsences'. It contains the follwing slots: meta, a list with meta information; pa, a vector with 1 encoding presences and 0 encoding pseudoabsences; xy, two-column matrix with coordinates for the (thinned) presence points and the sampled pseudoabsences. The projection is the same as the one of the env.stack supplied; env_vars, a data.frame containing extractions of the env.stack at the points xy; and call, which is the call made to the wsl.samplePseuAbs function.

Author(s)

Philipp Brun

Examples

```
### =========================================================================
### Data preparation
### =========================================================================

# Predictors
bio=raster::getData('worldclim',var='bio',lon=16, lat=48,res=.5)
bio=bio[[c(1,4,12)]]

# install.packages(”rgbif”)
library(rgbif)
# extract species
spn='Boletus aestivalis'
xt=as.vector(extent(bio))
baest <- occ_search(scientificName=spn,
                  hasCoordinate=TRUE,
                  decimalLongitude=paste0(xt[1],”,”,xt[3]),
                  decimalLatitude=paste0(xt[2],”,”,xt[4]))

pbaest=baest$data[,c('decimalLongitude','decimalLatitude')]
baest_spp=SpatialPoints(pbaest,proj4string = crs(bio))

# extract target group
targr <- occ_search(familyKey = 8789,
                   hasCoordinate=TRUE,
                   limit = 10000,
                   decimalLongitude=paste0(xt[1],”,”,xt[3]),
                   decimalLatitude=paste0(xt[2],”,”,xt[4]))

ptargr=as.matrix(targr$data[,c('decimalLongitude','decimalLatitude')])
colnames(ptargr)=c(”x”,”y”)

# create temporary directory for target.group info
tdir=paste0(tempdir(),”/trgr”)
dir.create(tdir)
write.table(ptargr,file=paste0(tdir,”/targetxy.txt”),row.names = F)

# create temporary directory for template raster and env strata
strdir=paste0(tempdir(),”/str”)
dir.create(strdir)

# Note that for these should not be temporary files for a real analysis.

### =========================================================================
### Sample pseudoabsences
### =========================================================================

# Geograhpic method with 20% env strata
```

```
pseu.abs1=wsl.samplePseuAbs(type=”geographic”,
                          n=5000,
                          env.stack=bio,
                          pres=baest_spp,
                          add.strat=0.2,
                          template_dir=strdir,
                          env.strat_path=strdir,
                          geodist_fact=1,
                          geores_fact=3,
                          geo_nrep=7,
                          taxon=spn)

plot(pseu.abs1)

# Only geographic with longer autocorrelation length
pseu.abs2=wsl.samplePseuAbs(type=”geographic”,
                           n=5000,
                           env.stack=bio,
                           pres=baest_spp,
                           add.strat=0,
                           template_dir=strdir,
                           env.strat_path=strdir,
                           geodist_fact=.5,
                           geores_fact=3,
                           geo_nrep=7,
                           taxon=spn)

plot(pseu.abs2)

# Random and thin presences by default resolution (determined by
# template raster)
pseu.abs3=wsl.samplePseuAbs(type=”random”,
                           n=5000,
                           env.stack=bio,
                           template_dir=strdir,
                           pres=baest_spp,
                           geores_fact=3,
                           add.strat=0,
                           taxon=spn,
                           force_spat_thin=”presences”)

plot(pseu.abs3)

# Geo.start and thin presences by min 10 km distance
pseu.abs4=wsl.samplePseuAbs(type=”geo.strat”,
                           n=5000,
                           env.stack=bio,
                           template_dir=strdir,
                           pres=baest_spp,
                           geores_fact=3,
                           add.strat=0,
                           taxon=spn,
                           force_spat_thin=”presences”,
                           limdist=10)

plot(pseu.abs4)

# Target group with 20% env.strat & thining presences & absences
pseu.abs5=wsl.samplePseuAbs(type="target.group",
                         n=5000,
                         env.stack=bio,
                         template_dir=strdir,
                         target.group_dir=tdir,
                         env.strat_path=strdir,
                         geores_fact=3,
                         pres=baest_spp,
                         add.strat=0.2,
                         taxon=spn,
                         force_spat_thin="both")

plot(pseu.abs5)

# Environmental semi-stratified

plot(pseu.abs6)

# Environmental semi-stratified with a non-feasible threshold of
# 30km limdist. CAREFUL, this one takes a few minutes!
pseu.abs7=wsl.samplePseuAbs(n = 5000,
                           env.stack=bio,
                           type = "env.semi.strat",
                           add.strat = 0,
                           pres = baest_spp,
                           taxon = spn,
                           template_dir=strdir,
                           env.strat_path=strdir,
                           force_spat_thin="both",
                           limdist=30)

plot(pseu_abs7)

# Density dependent
pseu.abs8=wsl.samplePseuAbs(n = 5000,
                           env.stack=bio,
                           type = "density",
                           add.strat = 0,
                           pres = baest_spp,
                           taxon = spn,
                           geores_fact=3,
                           template_dir=strdir,
                           env.strat_path=strdir)

plot(pseu.abs8)
```

Index