Package ‘iSDM’

January 16, 2017

Type Package

Title Invasive Species Distribution Modelling

Version 1.0

Date 2016-04-05

Author Tarek Hattab & Jonathan Lenoir

Maintainer Tarek Hattab <hattab.tarek@gmail.com>

Description iSDM is an open-source R package that implements a few functions useful for modeling the spatial distribution of invasive species

Depends R (>= 3.2.0)

Suggests MASS

Imports ade4, colorRamps, geometry, geosphere, maptools, MASS, pdist, raster, rgdal, rgl, sp, virtualspecies

License GPL-3

RoxygenNote 5.0.1

Vignettes SDM, species distribution modelling, invasive species, RGB plot

R topics documented:

`iSDM-package` *Invasive Species Distribution Modelling*

Description

iSDM is an open-source R package that implements a few functions useful for modeling the spatial distribution of invasive species

Author(s)

Tarek Hattab & Jonathan Lenoir

Maintainer: Tarek Hattab <hattab.tarek@gmail.com>

References

Tarek Hattab, Michael Ewald, Sandra Skowronek, Raf Aerts, Carol X. Garzón-López, Hélène Horen, Boris Brasseur, Emilie Gallet-Moron, Fabien Spicher, Guillaume Decocq, Hannes Feilhauer, Olivier Honnay, Pieter Kempeneers, Sebastian Schmidtlein, Ben Somers, Ruben Van De Kerchove, Duccio Rocchini and Jonathan Lenoir (In prep). An unified framework to model the potential and realized distributions of invasive species within the invaded range

`eSample` *R function to perform environmental systematic sampling*

Description

R function to perform environmental systematic sampling

Usage

```
eSample(envData,nExpect,plot=TRUE,saveShape=TRUE,nf,lowerLim,upperLim)
```

Arguments

`envData`	Either a SpatialPointsDataFrame, a SpatilPixelsDataFrame, or a SpatialGridDataFrame as defined in package sp or a RasterStack or RasterBrick as defined in package raster. Note that this object can contain a mixture of variables type (quantitative, factor and ordered)
`nExpect`	Numeric number of desired sampling points. Note that sometime the exact expected number can not be reached because this function tries to find a grid that best matches the expected number
`plot`	A logic indicating whether or not you would like to have a graphical representation of the environmental systematic sampling (default=TRUE)

`saveShape` A logic indcating whether or not you want to save the geographical coordinates of the pixels corresponding to this environmental systematic sampling as a shapefile format (default=TRUE). The generated shapefile will be saved in your working directory

`nf` A numeric value indicating the number of ordination axes retained in the analysis, this function manages currently only 2 or 3 dimentions

`lowerLim` Numeric value of probability in [0,1] that can be used to produce sample quantiles corresponding to the given lower limit probability. This may be used to reduce the effect of extremes observations

`upperLim` Numeric value of probability in [0,1] that can be used to produce sample quantiles corresponding to the given upper limit probability. This may be used to reduce the effect of extremes observations

Details

The environmental systematic sampling consists in reducing the environmental space into 2 or 3 dimensions using an ordination method (the dudi.mixt method in ade4 package) as a first step. Thereafter convex hull will be created around the reduced environmental space. Then, a multidimensional grid will be created inside the convex hull. The obtained grid represents the perfect configuration required to adequately survey the environmental space. The last step is subsequently to seek the closest pixel to this ideal configuration. This is achieved by searching the nearest neighbour between each grid point and each pixel in the environmental space based on Euclidean distances

Value

Returns a list containing 4 objects: GeoSamples (The geographical coordinates of the pixels corresponding to the environmental systematic sampling); EnvSamples(The coordinates in the reduced environmental space of pixels corresponding to the environmental systematic sampling); EnvGrid (The coordinates of the grid in the reduced environmental space)

References

Tarek Hattab, Michael Ewald, Sandra Skowronek, Raf Aerts, Carol X. Garzón-López, Hélène Horen, Boris Brasseur, Emilie Gallet-Moron, Fabien Spicher, Guillaume Decocq, Hannes Feilhauer, Olivier Honnay, Pieter Kempeneers, Sebastian Schmidtlein, Ben Somers, Ruben Van De Kerchove, Duccio Rocchini and Jonathan Lenoir (In prep). An unified framework to model the potential and realized distributions of invasive species within the invaded range

Examples

```
library(raster)
envData<-getData('worldclim', var='bio', res=10)
envData<-crop(envData,extent(-10,45,20,75))
plot(envData)
par(mfrow=c(1,2))
Mysampling1<-eSample(envData,nExpect=50,plot=TRUE,saveShape=TRUE,nf=2,lowerLim=0.00001,upperLim=0.99999)
plot(envData[[1]])
plot(Mysampling1[[1]],add=TRUE,col=2,pch=19)
par(mfrow=c(1,2))
```

```
Mysampling2<-eSample(envData,nExpect=50,plot=TRUE,saveShape=TRUE,nf=2,lowerLim=0.1,upperLim=0.9)
plot(envData[[1]])
plot(Mysampling2[[1]],add=TRUE,col=2,pch=19)
Mysampling3<-eSample(envData,nExpect=50,plot=TRUE,saveShape=TRUE,nf=3,lowerLim=0.001,upperLim=0.999)
plot(envData[[1]])
plot(Mysampling3[[1]],add=TRUE,col=2)
```

`iForce` *R function to compute a negative exponential dispersal kernel*

Description

R function to compute a negative exponential dispersal kernel

Usage

```
iForce(occData,coords,a,envData,binary=TRUE,longlat=NULL)
```

Arguments

`occData` Either a SpatialPointsDataFrame as defined in package sp, a data.frame or a matrix object containing species data

`coords` Optional 2 columns matrix containing the X and Y coordinates of occData (only consider if occData is a a data.frame or a matrix object)

`a` An integer between 0 and 1 that controls the form of the dispersal kernel

`envData` Either a SpatialPointsDataFrame or a SpatialGridDataFrame as defined in package sp or a RasterStack or RasterBrick as defined in package raster. This object will be used to determine the area for which the negative exponential dispersal kernel will be calculated

`binary` A logic indicating whether the occData object contains presences/absences data or only the geographical coordinates of presences data (default=TRUE)

`longlat` A logic indicating whether point coordinates are projected (longitude-latitude decimal degrees) or not (for plane) (default=TRUE)

Value

Returns a RasterLayer object of the negative exponential dispersal kernel

References

Tarek Hattab, Michael Ewald, Sandra Skowronek, Raf Aerts, Carol X. Garzón-López, Hélène Horen, Boris Brasseur, Emilie Gallet-Moron, Fabien Spicher, Guillaume Decocq, Hannes Feilhauer, Olivier Honnay, Pieter Kempeneers, Sebastian Schmidtlein, Ben Somers, Ruben Van De Kerchove, Duccio Rocchini and Jonathan Lenoir (In prep). An unified framework to model the potential and realized distributions of invasive species within the invaded range

Examples

```
library(raster)
library(sp)
envData<-getData('worldclim', var='bio', res=10)
envData<-crop(envData,extent(-8,15,38,55))
envData<-envData[[1]]

#Generate randomly a SpatilaPointsDataFrame containing occurrences
xy<-coordinates(envData)[sample(which(is.na(values(envData))==FALSE),100),]
occ<-ifelse(xy[,2]>50,0,1)
occData<-SpatialPointsDataFrame(coords=xy,data=as.data.frame(occ), proj4string = CRS(proj4string(envData)))

par(mfrow=c(3,3),mar=c(1.5,1.5,1.5,1.5))
for (a in c(0.01,0.03,0.05,0.06,0.09,0.1,0.3,0.6,0.9)){
propagule<-iForce(occData,envData=envData,a=a,binary=TRUE,longlat=TRUE)
plot(propagule,main=paste("a = ",a))
plot(occData,col=ifelse(occData@data[,1]==1,1,0),add=TRUE,cex=0.3)}
```

pDLA *R function to compute the probability of detecting dispersal-limited absences within a network of presence-absence data*

Description

R function to compute the probability of detecting dispersal-limited absences

Usage

```
pDLA(occData,envData, longlat=TRUE,nf=5,occNative=NULL,envNative=NULL)
```

Arguments

`occData` A SpatialPointsDataFrame as defined in package sp containing species data (a single species) in binary format (ones for presences, zeros for absences)

`envData` An abject containing your explanatory variables. Either a SpatialPointsDataFrame or SpatialGridDataFrame as defined in package sp or a RasterStack or RasterBrick as defined in package raster. Note that this object can contain a mixture of variables type (quantitative, factor and ordered).

`longlat` A logic indicating whether point coordinates are projected (longitude-latitude decimal degrees) or not (for plane) (default=TRUE)

`nf` If envData contains any factor and ordered variables, the Hill and Smith method will be used to perform an ordination of explanatory variables. nf corresponds in this case to the number of kept axes for the caluluation of mahalanobis distances.

`occNative` (optional) a SpatialPoint object containing the occurrences from the native range

`envNative` (optional) an object of the same class as envData containing the explanatory variables from the native range

Details

This function campute the probability of detecting dispersal-limited absences within a network of presence-absence data. It's based on the idea that absences data that are environmentally close but geographically distant to presences data are more likely to be dispersal-limited absences. This function allows combining presence data from both the native and the invaded range, note that in this case the set of presence from the native range will only be added to the set of presences from the invaded range when calculating distances between absences and presences in the environnental space as it does not make sens to add presence data from the native range when calculating these distances within the geographical space.

Value

Returns a SpatialPointsDataFrame containing the probability values

References

Tarek Hattab, Michael Ewald, Sandra Skowronek, Raf Aerts, Carol X. Garzón-López, Hélène Horen, Boris Brasseur, Emilie Gallet-Moron, Fabien Spicher, Guillaume Decocq, Hannes Feilhauer, Olivier Honnay, Pieter Kempeneers, Sebastian Schmidtlein, Ben Somers, Ruben Van De Kerchove, Duccio Rocchini and Jonathan Lenoir (In prep). An unified framework to model the potential and realized distributions of invasive species within the invaded range

Examples

```
library(raster)
library(sp)
library(virtualspecies)
library(colorRamps)

envData<-getData('worldclim', var='bio', res=10)
envData<-crop(envData,extent(-8,15,38,55))

#Generate virtual species responses with formatfunctions
my.parameters <- formatFunctions(bio1 = c(fun = "dnorm", mean = 140, sd = 40), bio5 = c(fun = "dnorm",
mean = 230, sd = 70),bio6 = c(fun = "dnorm",mean = 10, sd = 40))

#Generate a virtual species distributions with responses to environmental variables
potential.dist <- generateSpFromFun(envData[[c(1,5,6)]], my.parameters)

#Limit a virtual species distribution to a defined area.
#It will thus generate a species which is not at the equilibrium with its environment
realized.dist<-limitDistribution(x=potential.dist$suitab.raster, area=extent(-8,15,38,48))

#Generate a random presence absence dataset from the realized distribution
# using a probability threshold of 0.5
Presence<-coordinates(realized.dist$occupied.area
)[sample(which(values(realized.dist$occupied.area)>0.5),300),]
Absence<-coordinates(realized.dist$occupied.area
)[sample(which(values(realized.dist$occupied.area)<0.5),300),]
occData<-as.data.frame(rbind(cbind(Presence,SP=rep(1,300)),cbind(Absence,SP=rep(0,300))))
coordinates(occData)<-~x+y
```

```
proj4string(occData)<-proj4string(envData)

# Calculat  the probability of detecting dispersal-limited absences
probability<-pDLA(occData=occData,envData=envData[[c(1,5,6)]],longlat=TRUE)

# Display the results
par(mfrow=c(1,2),mar=c(2,2.5,2,2.5))
plot(realized.dist$occupied.area,main="Realized distribution")
plot(occData,col=ifelse(occData$SP==1,2,1),add=TRUE,pch=19,cex=0.8)
plot(potential.dist$suitab.raster,main="Potential distribution")

scatterCol<-function(x){
x<-(x-min(x))/(max(x)-min(x))
colorFunction <- colorRamp(matlab.like(100))
zMatrix <- colorFunction(x)
zColors <- rgb(zMatrix[,1], zMatrix[,2], zMatrix[,3], maxColorValue=255)
return(zColors)}

points(probability,pch=21, col=1,bg=scatterCol(probability@data[,"PDLA"]),cex=1)

# Example based on occurrences from both the invaded and the native range
envData<-getData('worldclim', var='bio', res=10)
envNative<-crop(envData,extent(-90,-70,20,40))
envData<-crop(envData,extent(-8,15,38,55))
native.dist <- generateSpFromFun(envNative[[c(1,5,6)]], my.parameters)
occNative<-as.data.frame(coordinates(native.dist$suitab.raster)
[sample(which(values(native.dist$suitab.raster)>0.5),100),])
coordinates(occNative)<-~x+y
proj4string(occNative)<-proj4string(envData)
plot(native.dist,main=" Native range distribution")
plot(occNative,add=TRUE,pch=19,cex=0.8)
probability<-pDLA(occData=occData,envData=envData[[c(1,5,6)]],longlat=TRUE,
occNative=occNative,envNative=envNative[[c(1,5,6)]])
```

`plotRPD` *R function to plot the realized versus the potential distribution using a RGB colour space*

Description

R function to plot the realized versus the potential distribution using a RGB colour space

Usage

```
plotRPD(x=x,y=y,cex,xlab=xlab,ylab=ylab)
```

Arguments

`x` Either a SpatialPixelsDataFrame, or a SpatialGridDataFrame as defined in package sp or a RasterLayer as defined in package raster y	An object belonging to the same class as x

cex	A numerical value giving the amount by which plotting symbols should be magnified relative to the default

xlab	A title for the x axis

ylab	A title for the y axis

References

Tarek Hattab, Michael Ewald, Sandra Skowronek, Raf Aerts, Carol X. Garzón-López, Hélène Horen, Boris Brasseur, Emilie Gallet-Moron, Fabien Spicher, Guillaume Decocq, Hannes Feilhauer, Olivier Honnay, Pieter Kempeneers, Sebastian Schmidtlein, Ben Somers, Ruben Van De Kerchove, Duccio Rocchini and Jonathan Lenoir (In prep). An unified framework to model the potential and realized distributions of invasive species within the invaded range

Examples

```
library(raster)
library(sp)
library(virtualspecies)
library(colorRamps)

envData<-getData('worldclim', var='bio', res=10)
envData<-crop(envData,extent(-8,15,38,55))

# Generate virtual species responses with formatfunctions
my.parameters <- formatFunctions(bio1 = c(fun = "dnorm", mean = 140, sd = 40), bio5 = c(fun = "dnorm",
mean = 230, sd = 70),bio6 = c(fun = "dnorm",mean = 10, sd = 40))
# Generate a virtual species potential distributions with responses to environmental variables
potential.dist <- generateSpFromFun(envData[[c(1,5,6)]], my.parameters)$suitab.raster

#Limit the distribution
realized.dist<-potential.dist
cell.id<-which(coordinates(realized.dist)[,2]>48)
dis.lim<-sample(seq(0,1,by=0.01),length(cell.id),replace=TRUE)
values(realized.dist)[cell.id]<-ifelse(values(realized.dist)[cell.id]>dis.lim,
values(realized.dist)[cell.id]-dis.lim,0)
plotRPD(x=potential.dist,y=realized.dist,cex=0.6,xlab="Potential distribution"
,ylab="Realized distribution")
```

Index