

selection.MCP
Documentation

packege: nenhum

R

Seleção de dados. Cálculo do índice de área de vida MPC

Description:

A função seleciona os IDs que poderão ser utilizados para o cálculo do índice MPC (mínimo polígono convexo) (em inglês, MCP-minimum convex polygon) de tamanho de Área Vida baseado em alguns fatores de interesse. Depois, ela calcula esse índice. Essa função depende do pacote `gpclib`, que não está disponível para o sistema operacional Windows, dependendo ou de Mac ou de Linux.

Usage:

```
selection.MCP (coords, id, session, minsession, mincapture, calculation = TRUE,
              percent=c(80,95,100), unin = c("m"), unout=c("m2"),
mcpshow=FALSE)
```

Arguments:

`coords` Data.frame contendo pelo menos as coordenadas X e Y (cartesianas) de cada ponto de captura. Caso haja mais informações no data.frame, as coordenadas X e Y devem ocupar a primeira e segunda coluna do data.frame respectivamente.

`id` Vetor com valores ou caracteres que determine a identidade (a qual indivíduo pertence) de cada ponto de captura (a quem pertence aquele ponto). Caso todos os pontos pertençam ao mesmo indivíduo, incluir um vetor com o mesmo valor em todas as posições.

`session` Vetor com as sessões em que foram realizadas cada captura. Esse argumento é importante para garantir independência temporal entre os pontos de captura de cada indivíduo. Caso esse fator não seja relevante, incluir um vetor de número 1 com o comprimento igual ao número de linhas do data.frame "coords" (`nrow(coords)`).

`minsession` Valor numérico: número mínimo de sessões diferentes que os ids devem ter sido capturados para poderem ter suas áreas de vida estimada pelo método MCP. Não aceita conjunto numérico como o argumento 'percent'.

`mincapture` Valor numérico: número mínimo de pontos de captura diferentes que os ids devem ter sido capturados para poderem ter suas áreas de vida estimada pelo método MCP. Não aceita conjunto numérico como o argumento 'percent'.

`calculation` Lógico, indicando se a estimativa MPC da área de vida deve ser realizado ou não. Se `calculation=TRUE`, realiza o cálculo.

`percent` Valor ou conjunto numérico. É a informação de qual porcentagem de pontos (sempre mantendo os mais próximos do centróide da área de vida) que deve ser mantida para realizar a estimativa. Ou então, 100 menos a proporção de outlier que deve ser excluída do cálculo do MCP.

`unin` A unidade dos pontos de captura inseridos. Ou "m", para metros (default), ou "km", para Kilometros.

`unout` A unidade que as áreas devem ser calculadas. Ou "m2" (default), para metros quadrados (default), "km2" para Kilometros ou "ha" para hectares.

`mcpshow` Lógico. Determina se será mostrada uma lista de `data.frames` com os pontos de captura utilizados para "desenhar" o mínimo polígono convexo. Se `"=TRUE"`, exibe um terceiro elemento na lista final (outra lista). Se `"=FALSE"`, não mostra, embora ela seja produzida de qualquer maneira.

Details:

Mínimo polígono convexo é uma propriedade matemática de um conjunto de pontos: é o polígono que, unindo alguns dos pontos desse conjunto, contenha todos os outros pontos dentro de sua área e que, ao mesmo tempo, não possua nenhum ângulo côncavo (ângulo interno maior que 180 graus).

O mínimo polígono convexo foi proposto em 1957, por Mohr, como uma estimativa de área de vida para animais e, em seu próprio trabalho, apontou a importância de fazer a estimativa excluindo os outliers (papel do argumento `"percent"` nessa função).

A função `selection.MCP` deve ser aplicada em um `data.frame` com pelo menos duas colunas e três linhas (1 ID com 3 capturas).

Dentro desta função foram utilizadas duas funções adaptadas do pacote `"adehabitat_1.8.12"`, para que a presente função não dependesse também desse pacote (de difícil instalação). São as funções `mcp()` e `mcp.area()`. As adaptações foram no sentido de tornar essas funções mais flexíveis e adequadas à finalidade da função principal.

Novamente, essa função depende do pacote `gpclib`, que não está disponível para o sistema operacional Windows, dependendo ou de Mac ou de Linux (já foi testada em ambos).

Values:

A função `selection.MCP` retorna, dependendo do argumento `'calculation'` e do argumento `'mcpshow'`, um `data.frame` (se `calculation=FALSE`), uma lista com dois `data.frames` (se `calculation=TRUE` e `mcpshow=FALSE`) e uma lista com os dois `data.frames` e uma lista de `data.frames` (se `calculation=TRUE` e `mcpshow=TRUE`). O primeiro elemento da lista, que estará presente mesmo se `calculation=FALSE`, será um subconjunto do `data.frame` inicial (de acordo com os critérios definidos nos argumentos `'session'`, `'minsession'`, `'mincapture'`), com as mesmas colunas que este, sendo no mínimo 2 colunas (coordenadas X e Y), mas sem limite máximo de colunas.

O segundo elemento da lista será outro `data.frame` com os valores das áreas de vida estimadas pelo método MCP para cada nível de `id` (indivíduo) nas colunas, e para cada valor de porcentagem de pontos (argumento `'percent'`) que devem ser considerados no cálculo do MPC nas linhas.

O terceiro elemento da lista será outra lista de `data.frames` com pontos de captura de fato utilizados para estimativa de MPC (dependendo do argumento `'percent'`). Para cada valor de `'percent'` é adicionado um `data.frame` nessa segunda lista com os `'percent'` pontos mais próximos do centróide e que foram incluídos na análise. Se o cálculo não foi realizado, esse resultado não aparecerá mesmo que `"mcpshow=TRUE"`.

É importante notar que mesmo após a seleção de dados realizada, ainda podem aparecer áreas de vida com estimativa `MCP=0`. Isso acontece quando todas as coordenadas X ou todas as Y têm um mesmo valor, formando portanto, uma linha. Ou ainda, quando as coordenadas formam uma diagonal em relação ao plano cartesiano.

Author:

Gabriela de Lima Marin
gabi_lm88@hotmail.com

References:

Calenge, C. (2011). "Home Range Estimation in R: the adehabitatHR Package." Office national de la classe et de la faune sauvage Saint Benoist - 78610 Auffargis - France.

Weisstein, Eric W. "Convex Polygon." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/ConvexPolygon.html>

Morh, C. O. Table of equivalent populations of North American small mammals. American Midland Naturalist, v. 37, p. 223-249, 1947.

See Also:

packedge adehabitat, para análises de seleção de habitat
packedge sp, para análises de dados espaciais

Exemples:

```
#1
library(gpclib) #chamando o pacote
ako<-read.table("tab0akodoncamila.csv",sep=";",head=TRUE, as.is=TRUE) #lendo
arquivo
ako<-cbind(ako[,14:15],ako[,1:13],ako[,16:21])#é exigencia da funcao que as col 1
e 2 sejam X e Y!
options(max.print=30000) #definindo o máximo de linhas q será mostrado dos
objetos...

#Opção sem o cálculo do MPC
ako.mcpA<-selection.MCP(coords = ako, id = ako$Tag , session = ako$Session ,
minsession = 2 , mincapture = 5, calculation = FALSE)

#Opção com o cálculo do MPC e com os pontos do MPC
#capturas em no mínimo 2 sessões diferentes e pelo menos 5 pontos diferentes.
ako.mcpB<-selection.MCP(coords = ako, id = ako$Tag , session = ako$Session ,
minsession = 2 , mincapture = 5, calculation = TRUE, percent=c(90,100),
unin=c("m"), unout=c("m2"), mcpshow=TRUE)

#2
#criando um data.frame de acordo com exigências da função:
id1<-rep(1,10)
id2<-rep(2,7)
id3<-rep(3,8)
id4<-rep(4,3)
x1<-seq(1:10)
y1<-c(seq(2:8),rep(9,3))
x2<-c(rep(2,5),rep(3,2))
y2<-seq(10:3)
x3<-rep(2,8)
y3<-c(seq(4:9),1)
x4<-c(1,4,5)
y4<-c(3,7,8)
indiv<-as.data.frame(c(id1,id2,id3,id4))
coordx<-as.data.frame(c(x1,x2,x3,x4))
```

```
coordy<-as.data.frame(c(y1,y2,y3,y4))
teste<-cbind(coordx,coordy,indiv)
library(gpclib)
#Opção com o cálculo do MPC e com os pontos do MPC #capturas em no mínimo 1
sessão e pelo menos 4 #pontos diferentes.
testeresulta<-selection.MCP(coords = teste, id = teste[,3] , session = rep(2,28),
minsession = 1 , mincapture = 4,calculation =TRUE, percent=c(50,75,100),
unin=c("m"), unout=c("m2"),mcpshow=TRUE)

#3
library(gpclib)
del<-read.table("tab0delomyscamila.csv",sep=";",head=TRUE, as.is=TRUE)
del<-cbind(del[,14:15],del[,1:13],del[,16:21])#é exigencia da funcao que as col 1
e 2 sejam X e Y!
delomys<-selection.MCP(coords = del, id = del$Tag , session = del$Session ,
minsession = 2 , mincapture = 3,calculation =TRUE, percent=c(80,95,100),
unin=c("m"), unout=c("m2"))
```