Package ‘wildxing’

August 2, 2017

Title An R package for optimal positioning of widlife crossing structures

Version 0.0.0.9000

Description This package provides a series of functions to characterize the importance of linear features for widlife movement and provide an algorithm to select optimal positioning of potential crossing structures.

Depends R (>= 3.3.2), sp, adehabitatLT

License GPL (>=3)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1.9000

Imports raster, rgeos, uuid, Rsymphony

R topics documented:

avg_inds *Averaged intersection between multiple trajectories and segmented line*

Description

Combined a list list of corriIntersects object to obtain population level averaged statistics regarding crossing count

Usage

```
avg_inds(SpLlst)
```

Arguments

`SpLlst`	A list of animal trajectory of class ltraj containing multiple individuals

Value

A SpatialLinesDataFrame object

Examples

```
require(adehabitatLT)
 x <- c(0,0)
y <- c(-6500000,-4500000)
t1<-SpatialLines(list(Lines(Line(cbind(x,y)), ID="a")))
t2<-SegmentSpL(t1, n.parts=20, merge.last=F)
data (albatross) #From package adehabitatLT
 t3<-corriIntersects_All(albatross, t2)
t4<-avg_inds(t3)
```

corriIntersects *Intersection between individual trajectory and segmented line*

Description

Calculate summary statistics regarding frequency of intersections between an animal trajectory and a segmented SpatialLines object

Usage

```
corriIntersects(traj, corri, per_day = TRUE, plot = F)
```

Arguments

`traj`	An animal trajectory of class ltraj
`corri`	A segmented SpatialLines as returned bt function SegmentSpL
`per_day`	Specify if the number of crossing should be standardized over the time (in days) the animal was monitored or over the total number of locations, default=TRUE. Long gap in data should be identified as separate burst
`plot`	Whether a plot should be returned (default=F)

Value

A SpatialLinesDataFrame object

Examples

```
require(adehabitatLT)
 x <- c(0,0)
y <- c(-6500000,-4500000)
t1<-SpatialLines(list(Lines(Line(cbind(x,y)), ID="a")))
t2<-SegmentSpL(t1, n.parts=20, merge.last=F)
data (albatross) #From package adehabitatLT
t3<-corriIntersects(albatross[3], t2, plot=T)
 plot(ltraj2sldf(albatross[3]), add=T)
```

corriIntersects_All *Intersection between multiple trajectories and segmented line*

Description

Wrapper function that apply corriIntersects to all individual in a trajectory object

Usage

```
corriIntersects_All(trajs, corri)
```

Arguments

corri A segmented SpatialLines as returned bt function SegmentSpL

traj An animal trajectory of class ltraj containing multiple individuals

Value

A list of SpatialLinesDataFrame object

Examples

```
require(adehabitatLT)
 x <- c(0,0)
y <- c(-6500000,-4500000)
t1<-SpatialLines(list(Lines(Line(cbind(x,y)), ID="a")))
t2<-SegmentSpL(t1, n.parts=20, merge.last=F)
data (albatross) #From package adehabitatLT
t3<-corriIntersects_All(albatross, t2)
```

`hr_split` *Intersection of a home-range (polygon) with linear features*

Description

Intersects a SpatialPolygons* object with a SpatialLines* object and return a divided polygon (when overlap) with summary metric regarding

Usage

```
hr_split(pol = hr, line = corri)
```

Arguments

`pol` A SpatialPolygons* object representing an animal home-range.

`line` A linear feature of class SpatialLines

Value

A list containing the segmented polygon, the

Examples

```
pol<-SpatialPolygons(list(Polygons(list(Polygon(cbind(c(0,1,1,0,0),c(0,0,1,1,0)))), ID="polygon")))
line<-SpatialLines(list(Lines(list(Line(cbind(c(0,1),c(0.4,0.4)))), ID="line")))
splt<-hr_split(pol, line)
plot(splt[[1]])
splt[2:4]
```

`match_pts` *Associate closest elements between two SpatialPoints features*

Description

Evaluate the distance between a SpatialPoints/Lines* object and another SpatialPoints* object and return which features in first object that are the closest to each feature in second object

Usage

```
match_pts(pts1, pts2)
```

Arguments

`pts1` A SpatialLines* or SpatialPoints object. If a SpatialLines is provided, the object is first converted to point using getSpatialLinesMidPoints

`pts2` A SpatialPoints* object

Examples

```
 x <- c(1,5,4,8)
y <- c(1,3,4,7)
t1<-SpatialLines(list(Lines(Line(cbind(x,y)), ID="a")))
t2<-SegmentSpL(t1, n.parts=10, merge.last=F)
 Pts<-SpatialPoints(matrix(c(1,3,7,1,2,7), nrow=3, ncol=2))
 match_pts(t2, Pts)
```

---

optim_corri *Optimization of wildlife crossing locations over a linear features*

Description

The function use linear programming to optimize the location of wildlife crossing over a linear feature. The function maximise the spatial spread of locations, and the importance of specific location for animal crossing. The user can specify the number of crossing location desired, if some segment should be excluded, or if the location of some crossings are already decided. The user also need to specify the weight given to the spatial argument and the importance of crossing (default is equal importance to each).

Usage

```
optim_corri(corri, var = 4, n = 5, pct_keep = 1, nb_ind = 1, ln = F,
  rm = NULL, add = NULL, weight = 0.5, plot = T, time_limit = -1,
  gap_limit = -1, first_feasible = F, ...)
```

Arguments

corri	A segmented SpatialLines object returned by avg_inds
var	Variable used to represent importance of segment for animal. Number refer to columns of dataframe produced by avg_inds (default = 4, average percent crossing)
n	Number of crossing to place (default =5) in addition to fixed points (i.e. if a SpatialPoints* object is provided to the add argument)
pct_keep	Percentage of segment to consider in optimization. Removal is based on percentile of values of importance of segment for animal crossing
ln	Whether the natural logarithm of the variable value should be taken. Default=F
rm	A SpatialPoints object indicating the location that most be excluded from the optimization (defaul is NULL)
add	A SpatialPoints object indicating the location where a crossing is already present, or must be place to this location. Number of points included here will be added to n to give the total number of crossing selected.
weight	Argument setting the weight given to the spatial spread relative to the crossing importance (default = 0.5 meaning equal importance)
plot	Whether a plot showing the crossing location should be returned (default=T)
...	additional arguments that can be specify to Rsymphony_solve_lp (time_limit, gap_limit, first_feasible)
cost	A vector of same length that the number of segment in corri giving the cost for each segment

Value

A list containing the segmented polygon, the

Examples

```
require(adehabitatLT)
 x <- c(0,0)
y <- c(-6500000,-4500000)
t1<-SpatialLines(list(Lines(Line(cbind(x,y)), ID="a")))
t2<-SegmentSpL(t1, n.parts=20, merge.last=F)
data (albatross) #From package adehabitatLT
 t3<-corriIntersects_All(albatross, t2)
t4<-avg_inds(t3)
 opti1<-optim_corri(t4, var=4, n=3, nb_ind=1, weight=0.5, plot=T) #Equal weight
 opti2<-optim_corri(t4, var=4, n=3, nb_ind=1, weight=0.95, plot=T) #More importance to spatial spread
```

---

optim_mclp *Optimization of wildlife crossing locations over a linear features using the maximum coverage location problem*

Description

The function use linear programming to optimize the location of wildlife crossing over a linear feature. The function maximise the number of individuals the selected features will assist. The user can specify the number of crossing location desired, a coverage distance, if some segment should be excluded, or if the location of some crossings are already decided.

Usage

```
optim_mclp(corri_ls, n = 5, dist = 10 * 1000, rm = NULL, add = NULL,
  plot = T, time_limit = -1, gap_limit = -1, first_feasible = F, ...)
```

Arguments

`corri_ls` A segmented SpatialLines list returned by corriIntersects_All

`n` Number of crossing to place (default =5) in addition to fixed points (i.e. if a SpatialPoints* object is provided to the add argument)

`dist` Distance used to considered a given segment as covered (ie radius)

`rm` A SpatialPoints object indicating the location that most be excluded from the optimization (defaul is NULL)

`add` A SpatialPoints object indicating the location where a crossing is already present, or must be place to this location. Number of points included here will be added to n to give the total number of crossing selected.

`plot` Whether a plot showing the crossing location should be returned (default=T)

`...` additional arguments that can be specify to Rsymphony_solve_lp (time_limit, gap_limit, first_feasible)

Value

A list containing the segmented polygon, the

Examples

```
require(adehabitatLT)
x <- c(0,0)
y <- c(-6500000,-4500000)
t1<-SpatialLines(list(Lines(Line(cbind(x,y)), ID="a")))
t2<-SegmentSpL(t1, n.parts=20, merge.last=F)
data (albatross) #From package adehabitatLT
t3<-corriIntersects_All(albatross, t2)
test<-optim_mclp(t3, n=2,dist=10*1000, plot=T)
```

plotcorri_grp *Plot of density of crossing of multiple individuals with linear features*

Description

Produce a color-coded plot of density of crossing of multiple individuals with a segmented linear features. Used to display result of function avg_inds

Usage

```
plotcorri_grp(SpL, nb_breaks = 5, var = 4, main = "Default")
```

Arguments

SpL	A SpatialLinesDataFrame object returned by avg_inds
nb_breaks	The number of breaks to use in display (default =5), must be <=10
var	Variable to display (default =4) 1= Average count, 2=Total count, 3=Average count standardized by length of tracking, 4=Average count standardized by length of tracking and spatial sampling bias, 5= Number of different individuals

Value

A plot

Examples

```
require(adehabitatLT)
 x <- c(0,0)
y <- c(-6500000,-4500000)
t1<-SpatialLines(list(Lines(Line(cbind(x,y)), ID="a")))
t2<-SegmentSpL(t1, n.parts=20, merge.last=F)
data (albatross) #From package adehabitatLT
t3<-corriIntersects_All(albatross, t2)
t4<-avg_inds(t3)
 plotcorri_grp(t4, nb_breaks=5, var=4, main="Albatross")
```

plotcorri_ind *Plot of density of crossing of an individual with linear features*

Description

Produce a color-coded plot of density of crossing of an individual with a segmented linear features. Used to display result of function corriIntersects

Usage

```
plotcorri_ind(SpL, nb_breaks = 5, extent = NULL)
```

Arguments

SpL	A SpatialLinesDataFrame object returned by corriIntersects
nb_breaks	The number of breaks to use in display (default =5), must be <=10
extent	If not NULL, an extent object (from package raster) that specify boundaries of the display.

Value

A plot

Examples

```
require(adehabitatLT)
 x <- c(0,0)
y <- c(-6500000,-4500000)
t1<-SpatialLines(list(Lines(Line(cbind(x,y)), ID="a")))
t2<-SegmentSpL(t1, n.parts=20, merge.last=F)
data (albatross) #From package adehabitatLT
t3<-corriIntersects(albatross[3], t2, plot=F)
plotcorri_ind(t3, nb_breaks=4, extent=raster::extent(ltraj2spdf(albatross)))
```

plot_optim *Plot of selected crossing structures following optimization using ILP*

Description

Produce a color-coded plot of density of crossing of multiple individualsand optimal crossing structure. Used to display result of function optim_corri

Usage

```
plot_optim(corri, var = 4, optim, main = "Default", ...)
```

Arguments

`corri`	A SpatialLinesDataFrame object returned by avg_inds
`var`	Variable to display (default =4) 1= Average count, 2=Total count, 3=Average count standardized by length of tracking, 4=Average count standardized by length of tracking and spatial sampling bias, 5= Number of different individuals
`optim`	Output of the optimization function optim_corri

Value

A plot

Examples

```
require(adehabitatLT)
 x <- c(0,0)
y <- c(-6500000,-4500000)
t1<-SpatialLines(list(Lines(Line(cbind(x,y)), ID="a")))
t2<-SegmentSpL(t1, n.parts=20, merge.last=F)
data (albatross) #From package adehabitatLT
t3<-corriIntersects_All(albatross, t2)
t4<-avg_inds(t3)
 opti1<-optim_corri(t4, var=4, n=3, nb_ind=1, weight=0.5, plot=T) #Equal weight
 plot_optim(t4,  var=4, opti1, main="Crossings")
```

`range01` *Range standardisation (0,1)*

Description

This function standardises a vector between 0 and 1

Usage

```
range01(x)
```

Arguments

`x`	A vector

Examples

```
v<-c(1,2,2,3,4,4,5,6)
range01(v)
```

SegmentSpL — *Segmentation of SpatialLines object into muliple segments*

Description

Segment a SpatialLines object in segment of equal length (taken from: http://rstudio-pubs-static.s3.amazonaws.com/1068

Usage

```
SegmentSpL(sl, length = 0, n.parts = 0, merge.last = FALSE)
```

Arguments

`sl` SpatialLines object

`length` Length of individual segment (in units of sl)

`n.parts` Alternatively, the number of segments to create

`merge.last` Whether the last segment (of a different length) should be merged with the previous segment

Value

A SpatialLines object

Examples

```
 x <- c(1,5,4,8)
y <- c(1,3,4,7)
t1<-SpatialLines(list(Lines(Line(cbind(x,y)), ID="a")))
t2<-SegmentSpL(t1, n.parts=10, merge.last=F)
plot(t2, col = rep(c(1, 2), length.out = length(t2)), axes = T)
```

Index