

Package ‘Digiroo2’

February 19, 2015

Type Package

Title An application programming interface for generating null models of social contacts based on individuals' space use

Version 0.6

Date 2013-02-21

Author Ross Dwyer, Emily Best and Anne Goldizen

Maintainer Ross Dwyer <ross.dwyer@uq.edu.au>

Description Digiroo2 is an R package developed by researchers at the University of Queensland to investigate association patterns and social structure in wild animal populations. Proximity between individuals is generally considered to be an appropriate proxy for associations and pairwise association indices are the most widely used technique for analysing animal social structure. However, little attention is given to identifying how patterns of spatial overlap affect these association patterns. For example, do individuals associate randomly with others with whom they share home ranges, or do some individuals go out of their way to associate with or avoid particular individuals? This program builds a null model of random associations based on an individual's space use determined using home range methodologies. Random points may be generated within a specified home range contour or according to the Utilization Distribution (UD). Expected associations of individuals are extracted based on probability of occurrence and the proximity between home range weighted random points. Association matrices can be generated from multiple permutations for analysis using SOCPROG 2.4 (Whitehead 2009) to create 'expected' pairwise half-weight association indices (HWIs). These may be compared with the 'observed' HWIs from field observations to reveal whether pairs of animals associate more (= attraction) or less (= avoidance) than expected by chance.

Depends R (>= 2.14.0), maptools, spatstat, spdep

Suggests adehabitatHR, raster

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2013-02-21 07:52:44

R topics documented:

Digiroo2-package	2
fAssocmatrix	3
fAssocplot	5
fAssoctable	7
fDrawfigure	8
fRanXY	9
Roos	10
StudyArea	11
Index	13

Digiroo2-package	<i>An application programming interface for generating null models of social contacts based on individuals' space use</i>
------------------	---

Description

Digiroo2 is an R package developed by researchers at the University of Queensland to investigate association patterns and social structure in wild animal populations. Proximity between individuals is generally considered to be an appropriate proxy for associations and pairwise association indices are the most widely used technique for analysing animal social structure. However, little attention is given to identify how patterns of spatial overlap affect these association patterns (Carter et al. 2009). For example, do individuals associate randomly with others with whom they share home ranges, or do some individuals go out of their way to associate with or avoid particular individuals? This program uses home range methodologies to build a null model of space use with individuals exhibiting a 'random' social structure. Random points may be generated within a specified home range contour or according to the Utilization Distribution (UD). Expected associations of individuals are extracted based on probability of occurrence and the proximity between home range weighted random points. Association matrices can be generated from multiple permutations for analysis using SOCPROG 2.4 (Whitehead 2009) to create 'expected' pairwise half-weight association indices (HWIs). These may be compared with the 'observed' HWIs from field observations to reveal whether pairs of animals associate more (= attraction) or less (= avoidance) than expected by chance.

Details

Package: Digiroo2
 Type: Package
 Version: 0.6

Date: 2013-02-21
License: GPL (>= 2)

Author(s)

Ross Dwyer, Emily Best and Anne Goldizen, *Behavioural Ecology Research Group (BERG), School of Biological Sciences, University of Queensland, Queensland, Australia.*

Maintainer: Ross Dwyer <ross.dwyer@uq.edu.au>

References

Carter A.J., MacDonald, S.L., Thompson, V.A. & Goldizen, A.W. (2009) Structured association patterns and their energetic benefits in female eastern grey kangaroos, *Macropus giganteus*. *Animal Behaviour* 77: 839-846.

Whitehead, H. (2009) SOCPROG programs: analysing animal social structures. *Behavioral Ecology and Sociobiology* 63: 765-778.

fAssocmatrix

Extract expected associations over multiple permutations

Description

This function combines the functions fRanXY, dnearest and fAssocTable, to enable the user to extract repeated expected associations over multiple permutations. Unlike fAssocTable, this function operates in a single step where the user is only required to input the home range of interest, the association distance threshold Gprox and those individuals to be used in the analysis. The user is offered the flexibility to include all individuals in the analysis, or a series of possible combinations of individuals based on mean group size and probability of occurrence. The output table generated by this function creates a series of 'expected' associations according to what would be expected if individuals within a population distributed themselves at random within their home range. The output table is a data frame with a permutation number, a group identifier (i.e. those individuals occurring within a defined distance threshold) and the individual id. Each line in a single permutation corresponds to a unique individual. Note, the running time of this function is positively related to the number of associations in the system. Running time can be decreased by reducing the number of permutations iPerm, limiting numbers of individuals contained in the study iID or by decreasing the maximum association distance threshold Gprox.

Usage

```
fAssocmatrix(sPerm, Gprox, iextract, iID)
```

Arguments

sPerm	a sequence of numbers representing the number of permutations to be run. If 100 permutations are to be run, iPerm=1:100
Gprox	a numeric field representing the distance threshold (max) for an association to occur
iextract	either a SpatialPolygonsDataFrame or a SpatialGridDataFrame object. This determines whether points are sampled from within a bounding area or according to a probability surface
iID	either a sequence of numbers (=IDs) or a matrix

Value

Permutation	a numeric vector containing the permutation number
Group	a numeric vector representing the group id. Individuals which occur within the distance threshold (i.e. between 0 and Gprox) have the same group number
IDs	a character vector containing the identity of the animal within each group. IDs will be unique within each permutation

Author(s)

Ross Dwyer <ross.dwyer@uq.edu.au>

See Also

[fRanXY](#), [fAssoctable](#), [fAssocmatrix](#), [fAssocplot](#), [dnearest](#)

Examples

```
## Load required packages
library(adehabitatHR)

## Load data
data(StudyArea)
data(Roos)

## Convert df into a spatial points df object
coordinates(Roos) <- ~x+y

## Estimation of UD and KUD 95% for the 10 animals
ud <- kernelUD(Roos,h="href",same4all=TRUE)
ver95 <- getverticeshr(ud,95,unin = c("m"),unout=c("ha"))
udsgdf <- as(estUDm2spixdf(ud),"SpatialGridDataFrame")

## Run 1 permutation using only the first 5 individuals using random points taken
## from within each individuals 95% KUD.
## Associations are determined if individuals are within 50 m from one another
fAssocmatrix(sPerm=1,
             Gprox=50,
```

```

        iextract=ver95,
        iID=1:5)

## Run 20 permutations using individuals according to sighting probability (ave no. roos seen on survey = 6)
## Probability-weighted random points are extracted according to the UD
## Associations are determined if individuals are within 50 m from one another

## Extract sighting probability

ID_counts <- tapply(rep(1,length(Roos[[1]])),Roos[[1]],sum)
No_SurveyDays <- 400

pObs <- as.vector(ID_counts)/No_SurveyDays # proportion of time individuals found on survey
nameObs <- names(ID_counts)
TotalAnimals <- 10 # Total number of animals in study
samplesize <- 6 # Number of IDs to include in simulations (with replacement)
bootlength <- 20 # Number of permutations

ExpProb <- data.frame(Kangaroo=nameObs,Probability=pObs) # Table of Expected Probabilities

## Generate ID combinations based on observation probability (with replacements)
pID <- sapply(1:bootlength,function(i) sample(x=TotalAnimals,
                                             size=samplesize,
                                             replace=FALSE,
                                             prob=pObs))

## What individuals are contained in the first permutation?
nameObs[pID[,1]]
## What individuals are contained in the last permutation?
nameObs[pID[,bootlength]]

## Run association matrix function to generate multiple permutations of 'expected' associations
AMud_50m <- fAssocmatrix(sPerm=1:bootlength,
                        Gprox=50,
                        iextract=udsgdf,
                        iID=pID)

```

fAssocplot

Plot the random points and the corresponding association matrix

Description

This function plots the output data spatially, allowing the user to visualise the random locations generated for individuals and the resulting associations defined by group memberships. This map can be plotted with or without individual ID labels.

Usage

```
fAssocplot(dnndata, idataxy, iplotnames = FALSE)
```

Arguments

dnndata	a nb type object
idataxy	A SpatialPointsDataframe object containing the random locations of the animals
iploenames	logical. Whether or not the user wants the animal IDs printed on the association plot. Default = FALSE

Author(s)

Ross Dwyer <ross.dwyer@uq.edu.au>

See Also

[fAssoctable](#), [fAssocmatrix](#), [dnearneigh](#)

Examples

```
## Load required packages
library(adehabitatHR)

## Load the data
data(StudyArea)
data(Roos)

## Convert into a spatial object
coordinates(Roos) <- ~x+y

## Estimation of UD and KUD 95% for the 10 animals
ud <- kernelUD(Roos,h="href",same4all=TRUE)
ver95 <- getverticeshr(ud,95,unin = c("m"),unout=c("ha"))
udsgdf <- as(estUDm2spixdf(ud),"SpatialGridDataFrame")

## For all 10 individuals, extract probability weighted random points
## falling within their respective utilisation distributions
ranXY <- fRanXY(1:10,udsgdf)
coordinates(ranXY) <- ~x+y

## Run Nearest neighbour function with those interactions < 50 m to generate the nb object
## See ?dnearneigh in the spdep package for more information
Gprox <- 50
dnn_digi <- dnearneigh(ranXY,0,Gprox,row.names=as.character(ranXY$ID))

## Plot and visualise these groups with and without animal IDs
fAssocplot(dnn_digi,ranXY,iploenames=TRUE)
fAssocplot(dnn_digi,ranXY,iploenames=FALSE)
```

fAssoctable	<i>Convert nearest neighbour object into a dataset of expected groups which is compatible with SOCPROG 2.4</i>
-------------	--

Description

This function converts a nearest neighbour (codenb) object into a format which can be accepted by the SOCPROG 2.4 program (Whitehead 2009). The output table generated by this function creates a series of 'expected' associations according to what would be expected if individuals within a population distributed themselves at random within their home range. Random points may be determined within a home range polygon or according to a probability surface within an individual's utilisation distribution. The output table is a dataframe with a group identifier (i.e. those individuals occurring within a defined distance threshold) and each line corresponding to a different individual. Note, the running time of this function is positively related to the number of associations in the system. Running time can be decreased by reducing the numbers of individuals contained in the study or by decreasing the association distance range in variables d1 and d2.

Usage

```
fAssoctable(dnndata)
```

Arguments

dnndata	a nb type object
---------	------------------

Value

Group	a numeric vector containing the group identifier
IDs	a character vector containing the individuals in the study

Author(s)

Ross Dwyer <ross.dwyer@uq.edu.au>

See Also

[fAssocmatrix](#), [fAssocplot](#)

Examples

```
## Load required packages
library(adehabitatHR)

## Load study data
data(StudyArea)
data(Roos)
```

```

## Convert into a spatial object
coordinates(Roos) <- ~x+y

## Estimation of UD and KUD 95% for the 10 animals
ud <- kernelUD(Roos,h="href",same4all=TRUE)
ver95 <- getverticeshr(ud,95,unin = c("m"),unout=c("ha"))
udsgdf <- as(estUDm2spixdf(ud),"SpatialGridDataFrame")

## For all 10 individuals, extract probability weighted random points
## falling within their respective utilisation distributions
ranXY <- fRanXY(1:10,udsgdf)
coordinates(ranXY) <- ~x+y

## Run Nearest neighbour function with those interactions < 50 m to generate the nb object
## See ?dnearest in the spdep package for more information
Gprox <- 50
dnn_digi <- dnearest(ranXY,d1=0,d2=Gprox, row.names=as.character(ranXY$ID))

## Convert random points into a dataset of expected groups by detecting associations
## by proximity and converting them into a format accepted by SOCPROG 2.4
fAssoctable(dnn_digi)

```

fDrawfigure

Plot the location of the random point falling within an individual's home range

Description

fDrawfigure extracts either a random point or a probability weighted random point falling within a specified home range. The random point, the observed locations, the specified ud vertice or the utilisation distribution can be plotted on a user-defined background.

Usage

```
fDrawfigure(x, iextract, idataxy = NULL, istudy = NULL)
```

Arguments

x	a numeric vector representing the individual to be plotted. This number relates to order in which animals are listed within the iextract object
iextract	either a SpatialPolygonsDataFrame object containing the vertice of interest or a SpatialGridDataFrame object containing the utilisation distribution of interest
idataxy	an optional dataframe containing the animal locations
istudy	optional background layer for the points to be plotted. This can be a spatial polygon object, a raster or blank = NULL

Author(s)

Ross Dwyer <ross.dwyer@uq.edu.au>

Examples

```
## Load required packages
library(adehabitatHR)

## Load study data
data(StudyArea)
data(Roos)

# Subset the data for only 1 individual
RooSubid <- 'Stripes'
RooSub <- subset(Roos, Roos$ID==RooSubid)
RooSub$ID <- as.character(RooSub$ID)

coordinates(RooSub) <- ~x+y

## Estimation of UD and KUD 95% for 1 animal
ud <- kernelUD(RooSub, h="href", same4all=TRUE)
ver95 <- getverticeshr(ud, 95, unin = c("m"), unout=c("ha"))
udsgdf <- as(estUDm2spixdf(ud), "SpatialGridDataFrame")

## Extract random points falling within animal #1's KUD 95%
fDrawfigure(1, ver95, RooSub, StudyArea)

## Extract probability-weighted random points falling within
## animal #1's UD and draw on map of the study area.
fDrawfigure(1, udsgdf, RooSub, StudyArea)
```

fRanXY

*Extract random points from within an individual's home range***Description**

Extracts either either a random point from a home range vertice (i.e. a `SpatialPolygonsDataFrame` object) or a probability-weighted random point from a Utilisation Distribution (i.e. a `SpatialGridDataFrame` object)

Usage

```
fRanXY(x, iextract)
```

Arguments

x	a numeric vector representing the individuals of interest. This number relates to order in which animals are listed within the <code>iextract</code> object
iextract	either a <code>SpatialPolygonsDataFrame</code> object containing the vertice of interest or a <code>SpatialGridDataFrame</code> object containing the utilisation distribution of interest

Author(s)

Ross Dwyer <ross.dwyer@uq.edu.au>

Examples

```
## Load required packages
library(adehabitatHR)
library(raster)

## Load study data
data(StudyArea)
data(Roos)

coordinates(Roos) <- ~x+y

## Estimation of UD and KUD 95% for the 10 animals
ud <- kernelUD(Roos,h="href",same4all=TRUE)
ver95 <- getverticeshr(ud,95,unin = c("m"),unout=c("ha"))

## Conversion from a adehabitat Udm object into a sgdf object
udsgdf <- as(estUDm2spixdf(ud),"SpatialGridDataFrame")

## For 3 animals, extract a random point falling within their respective KUD 95%
fRanXY(1:3,ver95)

## For 3 animals, extract a probability weighted random point falling within
## their respective utilisation distributions
fRanXY(1:3,udsgdf)
```

Roos

Social network study of Eastern grey kangaroos

Description

This GPS dataset contains the locations of 10 Eastern grey kangaroos *Macropus giganteus* monitored between 05/01/2010 to 21/12/2011 in South East Queensland, Australia. Data supplied E.C. Best from the School of Biological Sciences, University of Queensland, Australia.

Usage

```
data(Roos)
```

Format

A data frame containing GPS location data of individual sightings of *M. giganteus* with the following 3 variables.

ID a character vector giving the identity of each kangaroo

x a numeric vector containing the x coordinate of each individual
y a numeric vector containing the y coordinate of each individual

Details

The coordinates are given in meters GDA94/MGA zone 55

Source

www.berglabs.org

Examples

```
## Load required packages
library(adehabitatHR)

## Load study data
data(Roos)

## Convert df into a spatial points df object
coordinates(Roos) <- ~x+y

## Plot observed location data with individual kangaroos by colour
plot(Roos,col='red')

## Estimation of UD for the 10 kangaroos
ud <- kernelUD(Roos, same4all=TRUE)

## Plot the UD for the 10 kangaroos
image(ud)
```

StudyArea

The study site at Sundown National Park, Queensland, Australia.

Description

This dataset contains a spatial polygon of the study area at Sundown National Park

Usage

```
data(StudyArea)
```

Details

This S4 class dataset contains a SpatialPolygonsDataFrame object of the study area at Sundown National Park.

The coordinates are given in meters GDA94/MGA zone 55.

Source

www.berglabs.org

Examples

```
## Load required packages
library(maptools)

## Load study data
data(StudyArea)

## Plot study area polygon
plot(StudyArea,border=2)
```

Index

*Topic **datasets**

Roos, [10](#)

StudyArea, [11](#)

*Topic **package**

Digiroo2-package, [2](#)

Digiroo2 (Digiroo2-package), [2](#)

Digiroo2-package, [2](#)

dnearneigh, [4](#), [6](#)

fAssocmatrix, [3](#), [4](#), [6](#), [7](#)

fAssocplot, [4](#), [5](#), [7](#)

fAssoctable, [4](#), [6](#), [7](#)

fDrawfigure, [8](#)

fRanXY, [4](#), [9](#)

Roos, [10](#)

StudyArea, [11](#)