Package 'expp'

August 29, 2016

Type Package

Title Spatial analysis of extra-pair paternity

Version 1.1

Date 2014-Aug-01

Author Mihai Valcu, Lotte Schlicht

Maintainer Mihai Valcu <valcu@orn.mpg.de>

Depends R (>= 3.0.0), methods, sp, spdep, rgeos, deldir

Suggests knitr, spatstat, lme4

VignetteBuilder knitr

Description Tools and data to accompany Schlicht, Valcu and Kempenaers ``Spatial patterns of extra-pair paternity: beyond paternity gains and losses''

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2014-08-05 09:12:20

R topics documented:

`expp-package` *Tools and data to accompany Schlicht, Valcu and Kempenaers "Spatial patterns of extra-pair paternity: beyond paternity gains and losses"*

Description

The expp package provides classes and functions for the investigation of the probability of having extra-pair young within local networks of breeding pairs including both realized and potential extra-pairings.

Details

Package:	expp
Type:	Package
Version:	1.0
Date:	2014-Aug-04
License:	GPL-3

See `help(epp)` and `vignette('expp')`

Functions

epp	Final data-transformation to male-female combinations and their extra-pair levels
eppSimDat	"Toy"-dataset creation to investigate potential Type I error rate inflations for models where
DirichletPolygons	Territory calculation via Dirichlet tesselation
eppMatrix	`data.frame` to `eppMatrix` object
neighborsDataFrame	nb object to `data.frame`
SpatialPointsBreeding	`data.frame` to `SpatialPointsBreeding object`

Author(s)

Mihai Valcu and Lotte Schlicht
Maintainer: Mihai Valcu <valcu@orn.mpg.de>

References

Schlicht, Valcu and Kempenaers "Spatial patterns of extra-pair paternity: beyond paternity gains and losses" (in prep.)

`bluetit_boundary` *Study area boundary.*

Description

`SpatialPolygonsDataFrame` Study area boundary of two Blue Tit populations: Kolbeterberg, Vienna, Austria (1998 through 2004) and Westerholz, Bavaria, Germany (2007 through 2011) .

Usage

```
data(bluetit_boundary)
```

Format

A SpatialPolygonsDataFrame with 12 SpatialPolygons.

`year_` numeric. The year of the observation.

Examples

```
data(bluetit_boundary)
summary(bluetit_boundary)
```

`bluetit_breeding` *Blue Tit breeding data.*

Description

Breeding data recorded for two Blue Tit populations in Kolbeterberg, Vienna, Austria (1998 through 2004) and Westerholz, Bavaria, Germany (2007 through 2011) . The data set contains breeding attempts locations, the respective social pair, and several individual and nest parameters.

Usage

```
data(bluetit_breeding)
```

Format

A data frame with 1025 observations on the following 10 variables.

`year_` numeric. The year of the observation.

`id` numeric. The identity of the nest box in which the breeding attempt took place.

`x` numeric. The east-west location of the nest box.

`y` numeric. The north-south location of the nest box.

`female` character. The identity of the female.

`male` character. The identity of the male.

`layingDate` numeric. The day of the year when the first egg was produced.

`male_age` character. The age class of the male ('juv' = 1st year breeder; 'adult' = older)

`male_tarsus` numeric. tarsus length (mm))

`study_area` character. The study area name.

Examples

```
data(bluetit_breeding)
head(bluetit_breeding)
```

`bluetit_epp` *Blue tit extra-pair paternity data.*

Description

`data.frame` Extra-pair paternity data recorded for two Blue Tit populations in Kolbeterberg, Vienna, Austria (1998 through 2004) and Westerholz, Bavaria, Germany (2007 through 2011) .

Usage

```
data(bluetit_epp)
```

Format

A data frame with 425 observations on the following 3 variables.

`year_` numeric. The year of the observation.

`female` character. The female involved in the respective EPP event.

`male` character. The male involved in the respective EPP event.

Examples

```
data(bluetit_epp)
head(bluetit_epp)
```

DirichletPolygons — *Dirichlet Polygons*

Description

Computes the Dirichlet polygons using a `SpatialPointsBreeding` object and optionally a boundary `SpatialPolygons` or a vector containing id-s located at the boundary.

Methods

`signature(x = "SpatialPointsBreeding", boundary = "SpatialPolygons")` a boundary polygon

`signature(x = "SpatialPointsBreeding", boundary = "missing")` boundary is inferred using `ripras` in `spatstat`

`signature(x = "SpatialPointsBreeding", boundary = "numeric")` a vector of integers containing the id-s located at the boundary. The boundary is defined using a 'Follow-The-Dots' strategy. The `width` argument passed to `gBuffer` defines the distance between boundary boxes and the boundary polygon; it is set by default to half of the average distance between boundary boxes.

Examples

```
d = data.frame(
  x = c(4, 17, 16, 41, 41, 43, 86, 62, 71, 92, 95,53, 34, 27, 53),
  y = c(3, 18, 36, 6, 18, 50, 3, 21, 40, 43, 57, 62, 62, 45, 37),
  id = 1:15,male = paste0('m', 1:15), female = paste0('f', 1:15),
    stringsAsFactors=FALSE)

b = SpatialPointsBreeding(d, id = 'id', breeding = ~ male + female)

# boundary is inferred based on the Ripley-Rasson estimate of the spatial domain
dp1 = DirichletPolygons(b)
plot(dp1)

# boundary is given
brdy2 = readWKT("POLYGON((28 71,67 68,70 49,84 49,90 74,111 65,107
                      36,78 28,98 15,98 -4,74 -7,-2 -8,0 31,28 71) )")
dp2 = DirichletPolygons(b, boundary = brdy2)
plot(dp2)

# boundary is inferred based on the boundary id-s.
# define boundary id-s using a 'Follow-The-Dots' strategy.
brdy3 = as.integer(c(1, 2, 4, 7, 9, 10, 11, 12, 13, 14, 3))

dp3 = DirichletPolygons(b, boundary = brdy3)
plot(dp3)

# setting width manually
dp4 = DirichletPolygons(b, boundary = brdy3, width = 2)
```

```
plot(dp4)


plot(dp1)
plot(dp2, add = TRUE, border = 2)
plot(dp3, add = TRUE, border = 3)
plot(dp4, add = TRUE, border = 4)
plot(b, add = TRUE)
```

---

epp *Building data-set for realized and unrealized EPP-pairs*

Description

epp combines a SpatialPointsBreeding, a SpatialPolygons* as obtained from DirichletPolygons and a eppMatrix to create the spatial context for every potential and realized extra-pair male-female combination.

Usage

```
epp(breedingDat, polygonsDat, eppDat, maxlag = 3)
as.data.frame(x)
## S4 method for signature 'epp,missing'
plot(x, zoom, maxlag = 3, zoom.col = 'grey', ...)
## S4 method for signature 'epp'
barplot(height, relativeValues = FALSE, ...)
```

Arguments

breedingDat	A SpatialPointsBreeding object, created by the SpatialPointsBreeding function
polygonsDat	A SpatialPolygons* object as obtained by calling DirichletPolygons function
eppDat	An object of class eppMatrix
maxlag	A numeric value indicating the maximum breeding distance for which male-female combinations should be calculated. When plotting it defines the outermost row of neighbors plotted around a focal id set by zoom
x, height	an epp object
zoom	a SpatialPointsBreeding id which is used for subsetting prior to plot
zoom.col	background color of the id (and hence the polygon) set by zoom
relativeValues	Defines the unit of the y-axis. TRUE plots proportions, FALSE absolute numbers.
...	further arguments to pass to plot.SpatialPointsBreeding and plot.SpatialPolygons*

Details

'plot' plots the territories, the identity of males ('m...') and females ('f...') at breeding sites (numbers), and the extra-pair events (dashed red arrows). Individuals that had extra-pair offspring are marked red. The parameter 'zoom' can be used to make a detailed visual check of a specific location (nestbox) and its surroundings.

'barplot' displays the distribution of extra-pair events over different breeding distances between the partners (in the number of territories) as vertical bars. Note that the distribution of all potential extra-pair partners is displayed as a dashed line only if the argument 'relativeValues' is TRUE.

Value

Returns an S4-class epp-object with 5 slots:

- `breedingDat` Input breeding data-set.
- `polygonsDat` Either polygons are estimated automatically using Thiessen Polygons, or input breeding polygons.
- `eppDat` Input data.frame with all male-female combinations that had EPP together.
- `maxlag` Input rank. Defaults to 3.
- `EPP` `data.frame` containing columns for the focal male and female ("male", "female"), their breeding distance ("rank"), and the parameters associated either with the male (column with prefix "_MALE") or the female (column with prefix "_FEMALE") territory.

See Also

vignette(expp)

Examples

```
### Simple example with three breeding pairs

# create raw data
set.seed(1310)
b = data.frame(id = as.integer(10:12), x = rnorm(3), y = rnorm(3),
male = paste0("m",1:3), female =  paste0("f",1:3), xx = rnorm(3), stringsAsFactors=FALSE  )
eppPairs = data.frame(male = c("m1", "m2", "m1"), female=c("f3", "f1", "f2") )

# prepare data
breedingDat = SpatialPointsBreeding(b, id = 'id', coords = ~ x + y, breeding = ~ male + female)
polygonsDat = DirichletPolygons(breedingDat)
eppDat   = eppMatrix(eppPairs, pairs = ~ male + female)

plot(breedingDat, eppDat)

# convert to epp class
x = epp(breedingDat, polygonsDat, eppDat, maxlag = 3)
as.data.frame(x)
```

```
#plot
plot(x)

### Example on a random data set with n breeding pairs and  n/2 extra-pair paternity rate
# create raw data
set.seed(123)
n = 20
b = data.frame(id = 1:n, x = rnorm(n), y = rnorm(n),
male = paste0("m",1:n), female =  paste0("f",1:n), xx = rnorm(n), stringsAsFactors=FALSE  )
eppPairs = data.frame(male = sample(b$male, round(n/2) ), female = sample(b$female, round(n/2) ) )

# prepare data
breedingDat = SpatialPointsBreeding(b, id = 'id', coords = ~ x + y, breeding = ~ male + female)
polygonsDat = DirichletPolygons(breedingDat)
eppDat   = eppMatrix(eppPairs, pairs = ~ male + female)

# convert to epp class
x = epp(breedingDat, polygonsDat, eppDat, maxlag = 10)

# plot
plot(x)
barplot(x)
barplot(x, relativeValues = TRUE)


## Not run:
### Real data example
# Raw datasets
data(bluetit_breeding)
data(bluetit_epp)
# select one year
year = 2010
b = bluetit_breeding[bluetit_breeding$year_ == year, ]
eppPairs = bluetit_epp[bluetit_epp$year_ == year, ]

# prepare data
breedingDat  = SpatialPointsBreeding(b, id = 'id', coords = ~ x + y, breeding = ~ male + female)
polygonsDat = DirichletPolygons(breedingDat)
eppDat = eppMatrix(eppPairs, pairs = ~ male + female)

# convert to epp class
x = epp(breedingDat, polygonsDat, eppDat, maxlag = 20)

# plot
plot(x)
barplot(x)
# plot zoom
plot(x, zoom = 120, maxlag = 2)

# run model on epp probability
dat = as.data.frame(x)
nrow(dat[dat$epp == 1, c('male', 'female')] )
```

```
  nrow(unique(eppPairs))

  if(require(lme4))
   (summary(glmer(epp ~ rank + male_age_MALE + (1|male) + (1|female),
    data = dat, family = binomial)))


## End(Not run)
```

---

eppMatrix — *Convert a* `data.frame` *to an eppMatrix object.*

Description

Converts a `data.frame` to a eppMatrix object using a `~male+female` formula.

Usage

```
eppMatrix(data, pairs = ~ male + female)
```

Arguments

`data`	a `data.frame`
`pairs`	a formula indicating the extra-pair male and the extra-pair female in that order.

Value

An object of class `eppMatrix` with two slots.

Slots

`male`**:** Object of class `"character"`: extra-pair male ID

`female`**:** Object of class `"character"`:extra-pair female ID

See Also

`epp`

Examples

```
eppPairs = data.frame(male = c("m1", "m2", "m1"), female=c("f3", "f1", "f2") )
e = eppMatrix(eppPairs,  pairs = ~ male + female)
class(e)
showClass("eppMatrix")

data(bluetit_breeding)
data(bluetit_epp)
b = bluetit_breeding[bluetit_breeding$year_ == 2010, ]
```

```
eppPairs = bluetit_epp[bluetit_epp$year_ == 2010, ]

breedingDat  = SpatialPointsBreeding(b, id = 'id', coords = ~ x + y, breeding = ~ male + female)
eppDat = eppMatrix(eppPairs, pairs = ~ male + female)

plot(breedingDat, eppDat)
```

eppSimDat *Type I error rate simulations*

Description

A helper function to perform Type I error rate simulations.

Usage

```
eppSimDat(N, meanClutch, eppRate, eppMax, eppMales, nLags)
```

Arguments

N	Number of breeding pairs; default value is 10
meanClutch	Mean clutch size (integer); clutch size it is assumed to be Poisson distributed; default is 10
eppRate	Proportion of extra-pair young in population; default is 0.10
eppMax	Maximum number of extra-pair young by male; default is 12
eppMales	Proportion of extra-pair males in population; default is 0.35
nLags	`maxlag` parameter to pass to DirichletPolygons

Details

All default values match the values found in one of our study populations ('Westerholz').

Value

An object of class epp The `data.frame` of the EPP slot contains two variable (`trait_MALE trait_FEMALE`) simulated independent from the epp variable.

Examples

```
d = eppSimDat()
plot(d)


## Not run:
# Type I error rate simulation
```

```
require(lme4)
pval_glmer = vector(mode = "numeric", length = 0)
pval_glm = vector(mode = "numeric", length = 0)

for(i in 1:500) {
  x = as.data.frame(eppSimDat(N = 120, meanClutch = 10, eppRate = 0.10, eppMax = 12,
      eppMales = 0.35, nLags = 3))

  fm1glmer = glmer(epp ~ rank + trait_MALE + trait_FEMALE + (1 | male) + (1 | female) ,
  data = x, family = binomial, nAGQ =  0)
  fm0glmer = update(fm1glmer, epp ~ 1 + (1 | male) + (1 | female) )
  pval_glmer[i] = anova(fm0glmer, fm1glmer)$"Pr(>Chisq)"[2]

  fm1glm = glm(epp ~ rank + trait_MALE + trait_FEMALE  , data = x, family = binomial)
  fm0glm = update(fm1glm, epp ~ 1 )
  pval_glm[i] = anova(fm0glm, fm1glm, test = "Chisq")$"Pr(>Chi)"[2]

  print(i)
 }

# Type I error rate of glmer models
table(pval_glmer<0.05)[2]/length(pval_glmer)

# TRUE
# 0.038

# Type I error rate of the equivalent glm models
table(pval_glm<0.05)[2]/length(pval_glm)

# TRUE
# 0.078


## End(Not run)
```

`neighborsDataFrame` *Convert a* `nb` *object to* `data.frame`.

Description

Convert an object of class nb in package spdep or a `list` of nb objects to `data.frame`.

Usage

```
neighborsDataFrame(nb)
higherNeighborsDataFrame(nb, maxlag)
```

Arguments

nb	an object of class nb
maxlag	maximum lag, see nblag

Value

data.frame

SpatialPointsBreeding *Converts a* data.frame *to an object of class* SpatialPointsBreeding

---

Description

Converts a data.frame to a SpatialPointsBreeding object. The SpatialPointsBreeding class extends SpatialPointsDataFrame with three extra slots defining the id (i.e. nest or breeding box) and the pair identity (i.e. male and female), respectively.

Usage

```
SpatialPointsBreeding(data, proj4string, coords = ~x + y, breeding = ~male + female, id)

## S4 method for signature 'SpatialPointsBreeding,missing'
plot(x, pch = 20, axes = FALSE, add = FALSE,
                  xlim = NULL, ylim = NULL, ..., cex = 1, col = "grey",
                  lwd = 1, bg = "grey90")

## S4 method for signature 'SpatialPointsBreeding,eppMatrix'
plot(x, y, pch = 20, axes = FALSE, add = FALSE,
                   xlim = NULL, ylim = NULL, ..., cex = 1, col = "grey",
                   col.epp = "red",
                   lwd = 1, lty = 2, bg = "grey90")
```

Arguments

data	a data.frame containing the coordinates (e.g. "x", "y"), the location id, and the pair identity (e.g. "male", "female") together with any other optional variables (e.g. individuals or nest traits).
proj4string	A CRS object containing a valid proj4 string. See CRS proj4string for details.
coords	Formula specifying which columns in object are the spatial coordinates. Argument passed to coordinates
id	Integer specifying the location id (e.g. nest box number, den ID).
breeding	Formula defining the male and female ID in that order (in the form of e.g. ~male+female).

`x`	a `SpatialPointsBreeding` object
`y`	an `eppMatrix` object
`pch`	see `plot.default`
`axes`	see `plot.default`
`add`	see `plot.default`
`xlim`	see `plot.default`
`ylim`	see `plot.default`
`...`	further arguments to pass to plot(as(x, "Spatial")
`cex`	see `plot.default`
`col`	see `plot.default`
`col.epp`	extra-pair partners color
`lwd`	see `plot.default`
`lty`	see `plot.default`
`bg`	see `plot.default`

Value

Item of the class 'SpatialPointsBreeding' with 8 slots.

`id`	Location ID (see input)
`male`	(Male IDs.)
`female`	Female IDs.
`data`	A `data.frame` containing all the optional variables (except location and individual ID-s)
`coords.nrs`	see `SpatialPoints`
`coords`	coordinates
`bbox`	see `Spatial`
`proj4string`	see `CRS`

See Also

`epp`

Examples

```
d = data.frame(
  x = c(4, 17, 16, 41, 41, 43, 86, 62, 71, 92, 95,53, 34, 27, 53),
  y = c(3, 18, 36, 6, 18, 50, 3, 21, 40, 43, 57, 62, 62, 45, 37),
  id = 1:15,male = paste0('m', 1:15), female = paste0('f', 1:15),
    stringsAsFactors = FALSE)

b = SpatialPointsBreeding(d, id = 'id', breeding = ~ male+female)

plot(b)
```

Index