Package: secrRFS (via r-universe)

May 27, 2026

Type Package

Title Spatially Explicit Capture-Recapture Random Field Simulator

Version 1.0.1

Date 2025-05-19

Description SECR random field simulator.

Depends R (>= 3.5.0), secr (>= 4.6.5)

Imports parallel

Suggests igraph, knitr, rmarkdown, spatstat (>= 3.0-2), spatstat.geom, spatstat.random, testthat

Additional_repositories https://spatstat.r-universe.dev

VignetteBuilder knitr, rmarkdown

License GPL (>= 2)

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev make libicu-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository https://murrayefford.r-universe.dev

Date/Publication 2025-05-18 22:49:06 UTC

RemoteUrl https://github.com/MurrayEfford/secrRFS

RemoteRef HEAD

RemoteSha 64ee926e8003305d278b6ba96714b9f39d7d57ee

Contents

secrRFS-package *SECR random field simulator*

Description

Functions to estimate the overdispersion of number detected n by simulating intensity surfaces (Efford and Fletcher 2024).

Details

Package:	secrRFS
Type:	Package
Version:	1.0.1
Date:	2025-05-19
License:	GNU General Public License Version 2 or later

See the vignette for an extended description (type 'vignette('secrRFS')' after loading package).

Author(s)

Murray Efford <murray.efford@otago.ac.nz>

References

Efford, M. G. (2025) secr: Spatially explicit capture–recapture models. R package version 5.2.1. https://CRAN.R-project.org/package=secr/

Efford, M. G. and Fletcher, D. (2024) The effect of spatial overdispersion on confidence intervals for population density estimated by spatially explicit capture–recapture. *bioRxiv* https://doi.org/10.1101/2024.03.12.584742.

See Also

RFS, carray

carray *Overdispersion Array*

Description

Computes overdispersion for array of parameter values.

Usage

```
carray(parmlevels, RFSargs, drop = TRUE)
```

Arguments

parmlevels	list of levels for each parameter of randomfn in 'parm'
RFSargs	other arguments passed to RFS
drop	logical

Details

RFS is executed for every combination of levels in 'parmlevels'. 'parmlevels should include all parameters for which the generating function (randomfn) has no default.

Other arguments of RFS (i.e. excluding 'parm') are provided in 'RFSargs'.

Value

Array of values of c. Dimensions correspond to components of parmlevels. If drop = TRUE then dimensions are deleted that have only one level.

Author(s)

Murray Efford

See Also

RFS, randomfn

Examples

```
detectpar <- list(lambda0 = 0.5, sigma = 1)
grid144 <- make.grid(12,12, detector='proximity', spacing = 2.0)
grid144mask <- make.mask(grid144, spacing = 0.5, buffer = 4)
D <- 256/maskarea(grid144mask)
parmlevels <- list(D = D, mu = 2^(0:5), scale = c(1e-04, 1, 2, 4, 8))
RFSargs <- list (
    randomfn   = randomParents,
    nrepl      = 1000,
 traps      = grid144,
 mask       = grid144mask,
    detectfn   = 'HHN',
 detectpar  = detectpar,
    noccasions = 5)
ca <- carray (parmlevels, RFSargs)
round(ca,1)
#     scale
# mu   1e-04    1    2    4    8
#   1    1.9  1.9  1.8  1.6  1.4
#   2    2.8  2.9  2.5  2.4  1.8
#   4    4.5  4.4  4.1  3.9  2.6
#   8    8.2  8.4  7.2  6.5  4.5
#   16  15.6 15.7 13.5 12.0  8.0
#   32  29.9 30.5 26.0 23.1 13.4
```

randomfn *Intensity Generators*

Description

Functions to generate the intensity surface for one realisation of a Cox process.

Usage

```
randomDensity(mask, parm, plt = FALSE, ...)
randomGaussian(mask, parm, plt = FALSE, ...)
randomParents(mask, parm, plt = FALSE, parentcex = 0, ...)
```

Arguments

mask	secr mask object
parm	list of parameter values (see Details)
plt	logical
...	other arguments passed to plot.mask when plt = TRUE
parentcex	numeric

Details

randomDensity	secr	randomHabitat	D, A, p
randomGaussian	spatstat.random	rLGCP	D, var, scale
randomParents	spatstat.random	rThomas	D, mu, scale

All functions generate a surface with expected global density 'D'.

randomDensity generates a binary habitat mosaic using the algorithm of Saura and Martinez-Millan (2000) as in randomHabitat.

randomGaussian generates a log-gaussian intensity surface with point-specific variance 'var' and exponential covariance function with scale 'scale'.

randomParents generates an intensity surface from a Thomas process with 'mu' expected offspring per parent and circular bivariate normal dispersion 'scale' about each parent. Parent points are plotted if parentcex > 0.

The 'parm' argument can include maskscale = TRUE to adjust the expected number within the mask (area A) to equal DA (default maskscale = FALSE).

Value

Vector of intensity values, one for each point in mask.

randomDensity is modified from the **secr** function of the same name.

Author(s)

Murray Efford

See Also

RFS, randomHabitat

Examples

```
grid144 <- make.grid(12,12, detector='proximity', spacing = 2.0)
grid144mask <- make.mask(grid144, spacing = 0.5, buffer = 4)
D <- 256/maskarea(grid144mask)

# random habitat
parm <- list(D = D, A = 0.5, p = 0.5, rescale = TRUE)
Di <- randomDensity(grid144mask, parm)
# non-habitat cells have density = 0
table(Di)

# Gaussian
parm <- list(D = D, var = 0.5, scale = 5)
randomGaussian(grid144mask, parm, plt = TRUE, border = 1)

# Thomas process
parm <- list(D = D, mu = 10, scale = 3)
randomParents(grid144mask, parm, plt = TRUE, border = 1, parentcex = 0.5)
```

RFS *Overdispersion Simulator*

Description

Compute overdispersion of n by simulating CV(Da)

Usage

```
RFS(randomfn = randomDensity, parm = list(), nrepl = 2, traps, mask,
detectfn = "HHN", detectpar, noccasions, verbose = FALSE,
seed = NULL)
```

Arguments

randomfn	function to generate random intensity surface
parm	list of parameter values for randomfn
nrepl	integer number of replicates >= 2
traps	secr traps object
mask	secr mask object
detectfn	character or integer code for spatial detection function (see detectfn)
detectpar	list of detection parameter values
noccasions	integer number of sampling occasions
verbose	logical
seed	optional seed

Details

'randomfn' should be one randomDensity, randomGaussian, randomParents. The required parameters (list 'parm') are specific to each 'randomfn'.

See the vignette for an extended description (type 'vignette('secrRFS')' after loading package).

Value

Either a scalar estimate of overdispersion (c) or a list including

localD	vector of simulated local densities
c	estimated overdispersion c

Author(s)

Murray Efford

References

Efford, M. G. and Fletcher, D. J. unpubl. The effect of spatial overdispersion on confidence intervals for population density estimated by spatially explicit capture–recapture.

See Also

carray, parallel-package

Examples

```
detectpar <- list(lambda0 = 0.5, sigma = 1)
grid144 <- make.grid(12,12, detector='proximity', spacing = 2.0)
grid144mask <- make.mask(grid144, spacing = 0.5, buffer = 4)
D <- 256/maskarea(grid144mask)
```

```
nrepl <- 100  # increase to 1000 for serious use

parm1 <- list(D = D, A = 0.5, p = 0.5)
RFS(randomDensity, parm1, nrepl, grid144, grid144mask, 'HHN',
    detectpar, 5)

parm2 <- list(D = D, var = 1, scale = 5)
RFS(randomGaussian, parm2, nrepl, grid144, grid144mask, 'HHN',
    detectpar, 5)

parm3 <- list(D = D, mu = 8, scale = 2)
RFS(randomParents, parm3, nrepl, grid144, grid144mask, 'HHN',
    detectpar, 5)
```

Index