Package 'secrfunc'

December 1, 2025

Type Package

Title Helper Functions for Package 'secr'

Version 1.0.0

Date 2025-11-26

Description Functions are provided for internal use by the spatial capture-recapture package 'secr' (from version 5.4.0). The idea is to speed up the installation of 'secr', and possibly reduce its size. Initially the functions are those for area and transect search that use numerical integration code from 'RcppNumerical' and 'RcppEigen'. The functions are not intended to be user-friendly and require considerable preprocessing of data.

Depends R (>= 3.5.0)

Imports Rcpp (>= 0.12.14), RcppNumerical, RcppParallel (>= 5.1.1)

Suggests knitr, rmarkdown, testthat

LinkingTo BH, Rcpp, RcppEigen, RcppNumerical, RcppParallel

License GPL (>= 2)

SystemRequirements GNU make

URL https://www.otago.ac.nz/density/

Encoding UTF-8

NeedsCompilation yes

Author Murray Efford [aut, cre] (ORCID: <https://orcid.org/0000-0001-5231-5184>), Philipp Jund [ctb] ((faster transect search and spacing))

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

Repository CRAN

Date/Publication 2025-12-01 14:50:02 UTC

Contents

secrfunc-package *Helper Functions for Package* **secr**

Description

This package provides some functions used internally by the spatial capture-recapture package **secr** (from version 5.4.0). The idea is to speed up the installation of **secr**, and possibly reduce its size. Initially the functions are those for area and transect search that use numerical integration code from RcppNumerical (Qiu et al. 2023) and RcppEigen.

The functions are not intended to be user-friendly and require considerable pre-processing of data. Examples and tests use data pre-formatted and saved in the 'extdata' folder.

Details

Package:	secrfunc
Type:	Package
Version:	1.0.0
Date:	2025-11-26
License:	GNU General Public License Version 2 or later

Author(s)

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

References

Borchers, D. L. and Efford, M. G. (2008) Spatially explicit maximum likelihood methods for capture–recapture studies. *Biometrics* **64**, 377–385.

Efford, M. G. (2011) Estimation of population density by spatially explicit capture–recapture with area searches. *Ecology* **92**, 2202–2207.

Efford, M. G. (2025) The SECR book. A handbook of spatially explicit capture–recapture methods. Version 1.0.1. Zenodo doi:10.5281/zenodo.15109938. Online at https://murrayefford.github.io/SECRbook/.

Qiu, Y., Balan, S., Beall, M., Sauder, M., Okazaki, N., and Hahn, T. (2023). RcppNumerical: 'Rcpp' Integration for Numerical Computing Libraries. R package version 0.6-0. https://CRAN.R-project.org/package=RcppNumerical

See Also

hdotpolycpp
makegkPolygoncpp
polygonhistoriescpp
polygonfxicpp

hdotpolycpp *Net Detection Hazard for Polygon and Transect Detectors*

Description

Compute spatially explicit net probability of detection for individual(s) at given coordinates. Used by `secr::pdot()`.

Usage

```
hdotpolycpp(xy, traps, Tsk, markocc, cumk, detectfn, gsb, convex, dim, grain, ncores)
```

Arguments

`xy`	2-column matrix of coordinates
`traps`	2-column matrix; coordinates of detector polygon vertices
`Tsk`	detector usage (numeric matrix detectors x occasions)
`markocc`	integer vector of marking occasion codes (1 = marking)
`cumk`	integer vector of start indices of vertices for each detector
`detectfn`	integer detector code (see **secr**)
`gsb`	numeric vector of real parameter values (lambda0, sigma)
`convex`	logical; if TRUE all detector polygons are convex (non-convex untested)
`dim`	integer; 1 = transect, 2 = polygon
`grain`	integer grain argument of RcppParallel::parallelFor
`ncores`	integer number of threads

Details

Uses 'integrate' function from RcppNumerical. Sighting occasions are ignored.

Value

Rcpp NumericVector with one hazard for each row in 'xy'.

Examples

```
set.seed(123)

# 50 random points
xy <- matrix(runif(100), ncol = 2) * 2 - 1

# centred unit polygon
traps <- matrix(c(0,0,1,1,0,0,1,1,0,0), ncol = 2) - 0.5
```

```
Tsk <- matrix(1, nrow = 1, ncol = 5)
markocc <- rep(1,5)
cumk <- c(0,5,0)   # zero-terminated
detectfn <- 14   # hazard half-normal
gsb <- c(lambda0 = 0.2, sigma = 0.5)

h <- hdotpolycpp(xy, traps, Tsk, markocc, cumk, detectfn, gsb,
    convex = TRUE, dim = 2, grain = 1, ncores = 2)
```

`makegkPolygoncpp` *Pre-compute Detection Hazard*

Description

Hazards and probabilities are computed for each parameter combination, detector and mask point. Used by **secr** internal functions `generalsecrloglik()` and `secr_makegk()`.

Usage

```
makegkPolygoncpp(detectfn, dim, convex, grain, ncores, gsbval, cumk, traps, mask)
```

Arguments

`detectfn`	integer detector code (see **secr**)
`dim`	integer; 1 = transect, 2 = polygon
`convex`	logical; if TRUE all detector polygons are convex
`grain`	integer grain argument of RcppParallel::parallelFor
`ncores`	integer number of threads
`gsbval`	matrix of real parameter values, one row per combination
`cumk`	integer vector of start indices of vertices for each detector
`traps`	2-column matrix; coordinates of detector polygon vertices
`mask`	2-column matrix; coordinates of mask points

Value

List with components

`H`	unbounded integrated hazard from radial function
`gk`	probability vector for elements of cc x detector x mask array
`hk`	hazard vector for elements of cc x detector x mask array

Examples

```
detectfn <- 14   # hazard half-normal
dim <- 2
convex <- TRUE
grain <- 1
ncores <- 2
# for a single parameter combination
gsbval <- matrix(c(0.2,1), nrow = 1)
cumk <- c(0,5)
# centred unit polygon
traps <- matrix(c(0,0,1,1,0,0,1,1,0,0), ncol = 2) - 0.5
mask <- as.matrix(expand.grid(x=-2:2, y=-2:2))

makegkPolygoncpp(detectfn, dim, convex, grain, ncores, gsbval, cumk, traps, mask)
```

---

polygonhistoriescpp *Probability of Polygon Capture Histories*

Description

polygonhistoriescpp is the core function for the contribution of individual histories to likelihood. It is called by internal **secr** function generalsecrloglik().

polygonfxicpp is the core code for 'fxi' calculations in **secr** when the detector type is polygon or transect based. It is called by the internal **secr** function allhistpolygonfxi.

Usage

```
polygonhistoriescpp(nc, detectfn, grain, ncores, minp, binomN, w, xy, start,
    group, hk, H, gsbval, pID, mask, density, PIA, Tsk, h, hindex, debug)

polygonfxicpp(nc, detectfn, grain, ncores, minp, binomN, w, xy, start,
    group, hk, H, gsbval, pID, mask, density, PIA, Tsk, h, hindex)
```

Arguments

nc	integer number of individuals
detectfn	integer detector code (see secr)
grain	integer grain argument of RcppParallel::parallelFor
ncores	integer number of threads
minp	double minimum probability
binomN	integer vector binomial code
w	3-D integer array of capture histories (individual x occasion x detector)
xy	2-column matrix of detection location coordinates

start	integer vector of index of row in xy for each first detection
group	integer vector; code for group of each individual (zero-based)
hk	numeric array of detection hazards from makegkPolygoncpp
H	numeric unbounded integrated hazard from radial function from makegkPolygoncpp
gsbval	matrix of real parameter values, one row per combination
pID	numeric matrix occasions x nmix. NOT USED?
mask	2-column matrix of mask coordinates
density	matrix relative density at each mask point (row) for each group? (column)
PIA	integer vector for the slice of the parameter index array for a given mixture component and session
Tsk	detector usage (numeric matrix detectors x occasions)
h	for exclusive detectors, pre-computed hazard; otherwise -1
hindex	for exclusive detectors, index to precomputed hazard; otherwise -1
debug	logical; if TRUE...

Details

The capture history 'w' merely records detection/nondetection on a particular occasion. 'xy' records the location of each positive detection.

Value

For polygonhistoriescpp – Numeric vector of probabilities, one per individual.

For polygonfxicpp – Numeric matrix with probability density for each individual (rows) at each mask point (columns).

See Also

makegkPolygoncpp

Examples

```
datafilename <- system.file("extdata/testdata.RData", package = "secrfunc")
load(datafilename)  # loads w, dimw, xy, start, traps, mask, gsbval, cumk, binomN

detectfn <- 14   # hazard half-normal
dim <- 2
convex <- TRUE
grain <- 1
ncores <- 2

gkhk <- makegkPolygoncpp(detectfn, dim, convex, grain, ncores, gsbval, cumk, traps, mask)
# see tests for expected result

nc      <- dimw[1]  # individuals
```

```
S       <- dimw[2]  # occasions
K       <- dimw[3]  # single polygon
minp    <- 1e-200
group   <- rep(0,nc)
pID     <- matrix(1, nrow = S, ncol = 1)
density <- matrix(1/nrow(mask), nrow(mask), 1)
PIA     <- as.integer(array (1, dim = c(nc, S, K)))
Tsk     <- matrix(1, nrow = K, ncol = S)
h       <- matrix(-1)
hindex  <- matrix(-1)
debug   <- FALSE

prw <- polygonhistoriescpp(nc, detectfn, grain, ncores, minp, binomN, w, xy, start,
    group, gkhk$hk, gkhk$H, gsbval, pID, mask, density, PIA, Tsk, h, hindex, debug)
# see tests for expected result

fxi <- polygonfxicpp(nc, detectfn, grain, ncores, minp, binomN, w, xy, start,
    group, gkhk$hk, gkhk$H, gsbval, pID, mask, density, PIA, Tsk, h, hindex)
# see tests for expected result
```

Index