

Package: overdispsim (via r-universe)

May 27, 2026

Type Package

Title Simulations of Overdispersion in SECR

Version 1.0.2

Date 2025-05-19

Description Code used to simulate overdispersion for paper by Efford and Fletcher.

Depends R (>= 3.5.0), secr, secrdesign (>= 2.8.2)

Imports parallel

Suggests knitr, rmarkdown, zen4R

VignetteBuilder knitr

Vignettes overdispsim-vignette: overdispsim-vignette.Rmd

License GPL (>= 2)

URL <https://github.com/MurrayEfford/overdispsim>

BugReports <https://github.com/MurrayEfford/overdispsim/issues>

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 22:51:37 UTC

RemoteUrl <https://github.com/MurrayEfford/overdispsim>

RemoteRef HEAD

RemoteSha 7a200a7bf6261d472f8c0620449dbbd54f837b56

Contents

overdispsim-package	2
addCL	2
extend	3
extractfn	4
plotnsum	5

run	6
setparameters	8
sim.cohesion	9
summary	10

Index	13
--------------	-----------

overdispsim-package *Simulations of Overdispersion in SECR*

Description

Code used to simulate overdispersion for paper by Efford and Fletcher.

Details

Package: overdispsim
 Type: Package
 Version: 1.0.2
 Date: 2025-05-19
 License: GNU General Public License Version 2 or later

Type vignette('Overview', 'overdispsim') to see the overview vignette.

See Also

[run_all](#)

addCL *Add Confidence Interval to Summary Plot*

Description

Intended for plot of summary results.

Usage

```
addCL(x, summ, cov = "COV", nrepl = 1000, ...)
```

Arguments

x	numeric x values
summ	dataframe from summary_M etc.
cov	character statistic to plot e.g. RB
nrepl	integer number of replicates for binomial interval for COV statistic
...	other arguments passed to Bucl, Blcl

Details

A binomial interval is computed for COV. Intervals are otherwise +/- 2SE.

Value

None returned

See Also

[summary_M](#)

extend

Extend Argument List

Description

Expand a single list of arguments using all combinations of values.

Usage

```
extend(baseargs, values)
```

Arguments

baseargs	list of arguments for sim.ppn
values	list of value vectors

Details

The vectors in 'values' refer to 'details' arguments of sim.ppn.

Value

List of argument lists for [sim.ppn](#).

See Also

[setparameters](#)

Examples

```

localpar <- setparameters()
basepopargs3 <- list(D = randomDensity,
  core = localpar$mask,
  buffer = 0,
  model2D = "IHP",
  details = list(D = localpar$D, p=0.5, A=0.25, rescale=TRUE))

# vary A, p arguments of randomDensity
popargs3 <- extend(basepopargs3, values = list(A = c(0.25,0.5), p = c(0.25, 0.5)) )

# view details
sapply(popargs3, '[', 'details')

```

 extractfn

Extract Simulation Result

Description

Functions to extract summary statistics from each replicate. Used by [run_all](#) in call to [run.scenarios](#).

Usage

```

extract_n(ch, distribution = "poisson")
extract_M(fit, chatmin = 1)
extract_MCL(fit)

```

Arguments

ch	secr capthist object
distribution	character
fit	fitted secr model
chatmin	numeric threshold of c-hat for adjustment of variance

Details

The value of 'distribution' may be set to "binomial" for fixed-N(A).

Value

For `extract_n`, a numeric vector

n	number of detected individuals
chatnk	Fletcher's c-hat
N	number of simulated individuals

localD detection-weighted local density

For extract_M, a list with components:

N	number of simulated individuals
n	number of detected individuals
chatnk	Fletcher's c-hat
pred	table of estimates from predict()
predF	table of estimates with variance adustment using Fletcher's c-hat
predW	table of estimates with variance adustment using Wedderburn c-hat
trueD	detection-weighted true density

For extract_MCL, a list with components:

N	number of simulated individuals
n	number of detected individuals
chatnk	Fletcher's c-hat
derived	table of estimates from derived()
trueD	detection-weighted true density

See Also

[run_all](#)

plotnsum

Plot Results from summary_n

Description

A wrapper for commands to plot chat vs the empirical variance/mean ratio for n. A log scale is used for both axes.

Usage

```
plotnsum(nsum, ny = 5, by.y = FALSE, xlim = c(0.75, 120), ylim = xlim,
label = "", labelx = 0.15, type = c("Fletcher", "Wedderburn"), ...)
```

Arguments

nsum	matrix output by summary_n
ny	integer number of levels
by.y	logical; see Details
xlim	numeric vector of limits on x-axis
ylim	numeric vector of limits on y-axis
label	character optional label
labelx	numeric label offset
type	character type of c-hat
...	other arguments passed to points()

Details

Used for text and appendix figures in Efford & Fletcher (2024).

Each column of nsum is of length ny*nx (nx inferred). Internally the columns for 'varration' and 'chatF' (or chatW) are formed into ny x nx matrices. Then lines are plotted either for each column (by.y = TRUE) or each row of these matrices.

Value

none

See Also

[summary_n](#)

run

Run Simulations

Description

A wrapper for **secrdesign** function `run.scenarios()`.

Usage

```
run_all(nrepl, popargs, CH.function = "sim.caphist", detargs = list(savepopn = TRUE),
  fit = FALSE, distribution = c("poisson", "binomial"), CL = FALSE,
  start = NULL, byscenario = FALSE, extractfn = NULL, seed = 12345, ...)
```

Arguments

nrepl	integer number of replicates
popargs	list of named arguments to sim.popn
CH.function	character name of function to simulate capthist
detargs	list of named arguments to sim.capthist
fit	logical; model fitted if TRUE
distribution	character distribution of n in fitted model ("poisson" or "binomial")
CL	logical; if TRUE fitting maximises the conditional likelihood
start	'start' argument for <code>secr.fit</code>
byscenario	logical (see <code>secrdesign</code>)
extractfn	function to extract statistics from each replicate
seed	integer random seed
...	additional arguments for <code>extractfn</code>

Details

Arguments generally follow [run.scenarios](#).

The number of threads is determined by `.local$maxncores` (see code).

By default, 'extractfn' is chosen depending on fit and CL:

fit	CL	extractfn
FALSE	——	extract_n
TRUE	FALSE	extractfn_M
TRUE	TRUE	extractfn_MCL

Value

See [run.scenarios](#). Depends on `extractfn`.

See Also

[extractfn](#)

setparameters *Set Parameters*

Description

Several variables are used by multiple functions and it is tidier to store them in an environment that is internal to the package but shared among functions. This function assigns values to the variables.

Usage

```
setparameters(lambda0 = 0.5, sigma = 1, detectfn = 'HHN', noccasions = 5,
              traps = make.grid(12, 12, detector = "proximity", spacing = 2),
              maskspacing = 0.5, maskbuffer = 4, N = 256, maxncores = 18)
```

Arguments

lambda0	numeric detection parameter
sigma	numeric detection parameter
detectfn	secr detectfn code
noccasions	integer
traps	secr traps object
maskspacing	numeric spacing of generated mask
maskbuffer	numeric buffer width of generated mask
N	integer expected population size in masked area
maxncores	integer available cores for parallel processing

Details

A mask is generated. The derived variables 'detectpar', 'D', 'pd' and 'enk' are also computed (see vignette).

setparameters is called by 'on.attach()' and may be called again to replace values.

'maxncores' is capped at the number available, as given by parallel::detectCores().

Value

Invisibly returns the environment as a list with these components (order varies) -

lambda0	intercept of HHN detection function
sigma	spatial scale of intercept of HHN detection function
traps	secr 'traps' object
detectpar	list with components lambda0 and sigma*
N	population size in mask area
D	density*

mask	secr 'mask' object*
maxcores	maximum number of threads to use
noccasions	number of sampling occasions
detectfn	secr detectfn code
pd	numeric overall probability of detection of AC at each mask point*
enk	numeric vector: expected number of individuals detected at each detector*
En	numeric expected number of individuals detected*
Evvarn	numeric expected variance of binomial n*

* computed by setparameters(); other values are as input

sim.cohesion	<i>Simulate Cohesion</i>
--------------	--------------------------

Description

Simulates detection of a clustered population with cohesion of detection sensu Bischof et al. (2020). The main use is to replace [sim.caphist](#) in the **secrdesign** function [run.scenarios](#), as called by [run_all](#).

Usage

```
sim.cohesion(traps, popn, detectfn = 0, detectpar = list(), noccasions = 5,
             savepopn = FALSE, gamma = 1, ...)
```

Arguments

traps	secr traps object
popn	popn object output by sim.popn
detectfn	integer or character value for shape of detectfn
detectpar	list of detection parameter values
noccasions	integer number of occasions
savepopn	logical passed to sim.caphist
gamma	numeric within-cluster cohesion 0-1 (Bischof et al. 2020)
...	other arguments passed to sim.caphist

Details

The argument gamma controls the degree of cohesion.

The popn argument should be a clustered population with saved attributes 'parents' and 'parentid'.

Value

A caphist object.

References

Bischof, R., Dupont, P., Milleret, C., Chipperfield, J., and Royle, J. A. (2020) Consequences of ignoring group association in spatial capture–recapture analysis. *Wildlife Biology* wlb.00649. doi:10.2981/wlb.00649

See Also

[run_all](#)

summary

Summarise Simulations

Description

Custom summary functions for output from [run_all](#).

Usage

```
summary_n(sims, binomial = FALSE)
summary_M(sims, true = NULL)
summary_MCL(sims, true = NULL, truea = 0.0635256)
```

Arguments

sims	output from run
binomial	logical; if TRUE then expected variance is .local\$Evarn
true	true density
truea	true value of effective sampling area a

Value

Table with one row per scenario. Columns are defined as follows -

summary_n

[var, scale etc.]	Parameters specific to the scenarios
varration	Ratio var(n) / mean(n)
chatF	Mean 'Fletcher-chat' for count n_k (individuals per detector)
chatW	Mean 'Wedderburn-chat' for count n_k (individuals per detector)
Nsim	number of datasets simulated
N	mean number of individuals in simulated population
varN	variance of number of individuals in simulated population

n	mean number of individuals detected
varn	variance of number of individuals detected
localD	mean of detection-weighted density
varlocalD	variance of detection-weighted density
VRlocalD	variance ratio from localD : $(\text{varlocalD}/\text{localD}^2 + 1/\text{En}) / (1/\text{En})$

summary_M

[var, scale etc.]	Parameters specific to the scenarios
n	mean number of individuals detected
N	mean number of individuals in simulated population
nvalid	number of successful simulations
estimate	mean estimated density
SE.estimate	mean SE of estimate
RSE	ratio of preceding
trueD	true density; either detection-weighted (default) or global as specified by the 'true' argument
RB	estimated relative bias relative to trueD
seRB	SE of RB
COV	unadjusted coverage of 95% interval relative to trueD
COVF	adjusted coverage
chatF	mean 'Fletcher-chat' for count n_k (individuals per detector)
varration	Ratio $\text{var}(n) / \text{mean}(n)$

summary_MCL

[var, scale etc.]	Parameters specific to the scenarios
n	mean number of individuals detected
nvalid	number of successful simulations
estimate	mean estimated density
SE.estimate	mean SE of estimate
RSE	ratio of preceding
trueD	true density; either detection-weighted (default) or global as specified by the 'true' argument
RB	estimated relative bias relative to trueD
COV	unadjusted coverage of 95% interval relative to trueD
chatF	mean 'Fletcher-chat' for count n_k (individuals per detector)
varration	Ratio $\text{var}(n) / \text{mean}(n)$
a	mean a-hat effective sampling area
SEa	SE of a-hat
RBa	RB(a-hat)
RSEa	RSE(a-hat)
COVa	coverage of 95% CI for a-hat
pCVn	fraction of $\text{var}(\text{D-hat})$ attributable to $\text{var}(n)$

See Also

[run_all](#)

plotnsum

Index

`addCL`, 2

`detectfn`, 8, 9

`extend`, 3

`extract_M` (`extractfn`), 4

`extract_MCL` (`extractfn`), 4

`extract_n` (`extractfn`), 4

`extractfn`, 4, 7

`overdispsim` (`overdispsim-package`), 2

`overdispsim-package`, 2

`plotnsum`, 5, 12

`run`, 6

`run.scenarios`, 4, 7, 9

`run_all`, 2, 4, 5, 9–11

`run_all` (`run`), 6

`setparameters`, 3, 8

`sim.caphist`, 7, 9

`sim.cohesion`, 9

`sim.popn`, 3, 7, 9

`summary`, 10

`summary_M`, 3

`summary_M` (`summary`), 10

`summary_MCL` (`summary`), 10

`summary_n`, 6

`summary_n` (`summary`), 10