

## **secr - spatially explicit capture–recapture in R**

Version 1.2.11

Murray Efford

22 December 2009

### **Contents**

- The big picture
  - State and observation models in SECR
  - Detector types
  - Software
- Components of SECR models
  - Design matrices
  - Detection submodels
  - Specifying effects on detection parameters
  - Density submodels
- Output
- References
- Appendix 1. Example of `secr` output
- Appendix 2. Comparison of DENSITY 4.4 and `secr` 1.2.
- Appendix 3. Functions in `secr` 1.2 arranged according to use

### **The big picture**

**secr** is an R package for estimating the density of an animal population from capture–recapture data collected with an array of 'detectors'. Spatially explicit capture–recapture (SECR) methods overcome edge effects that are problematic in conventional capture–recapture (Otis et al. 1978). Detectors may be live-capture traps, with animals uniquely marked. Detectors also may be sticky traps or snags that passively sample hair, from which individuals are distinguished by their DNA microsatellites, or cameras that take photographs from which individuals are recognized by their natural marks.

The primary data for SECR are (i) the locations of the detectors, and (ii) detections of known individuals on one or more sampling occasions (i.e. their detection histories). The terms 'detectors' and 'detections' cover the full spectrum of possibilities (see Detector Types

below), but we use them interchangeably with the more familiar ‘traps’ and ‘captures’, respectively. Table 1 gives a concrete example.

Table 1. Example of spatially explicit detection data. Each entry (e.g. A9) records the detector at which a known animal (ID) was observed on the given occasion (sample time). '.' indicates no detection. Each detector has known x-y coordinates.

ID	Occasion				
	1	2	3	4	5
1	A9	.	.	.	.
2	A12	A12	.	.	.
3	.	.	C6	B5	.
4	.	.	G3	.	F3
...					

SECR fits a spatial model of the population and the detection process to the locations where each animal is detected. The resulting estimates of population density are unbiased by edge effects and incomplete detection. Inverse prediction (IP SECR) and maximum likelihood (ML SECR) are alternative methods for fitting the spatial detection model (Efford 2004, Borchers and Efford 2008). Data augmentation and Markov chain Monte Carlo (MCMC) methods have also been used (Royle et al. 2009), but this approach is not yet available in `secr`.

A simple `secr` analysis might look like this (R prompt omitted). Output is in Appendix 1.

```
library(secr)                                # load library
mytraps <- make.grid(nx = 10, ny = 10,       # build detector array
  spacing = 30, originxy = c(365,365))
mycapt <- read.captures (file = 'capt.txt')  # import capture data
myCH <- make.capthist (mycapt, mytraps, fmt='XY') # build 'capthist' object
secr0 <- secr.fit(myCH, model = g0~1)        # fit null model
secrb <- secr.fit(myCH, model = g0~b)       # fit trap response model
AIC (secr0, secrb)                          # compare fit of models
secr0                                       # display estimates
```

## State and observation models in SECR

Like other methods for estimating abundance (Borchers et al. 2002), SECR combines a state model and an observation model. In SECR, the state model describes the distribution of animal home ranges in the landscape, and the observation model relates the probability of detecting an individual at a particular detector to the distance of the detector from a central point in each animal's home range (hence the more common term 'spatial detection model'). The distances are not observed directly (we don't know the range centre), so conventional distance sampling methods do not apply. The detection model uses a parametric function (Fig. 1) modified according to properties of the detector (below). The detection model may be fitted if we simultaneously model the distribution of range centres in the population (Borchers and Efford 2008). This distribution will usually be treated as a homogeneous Poisson point process (Fig 1), but an inhomogeneous distribution may also be fitted. Density is the sole parameter of a Poisson process. By fitting an inhomogeneous Poisson process distribution we can describe the effect of habitat variables on density.

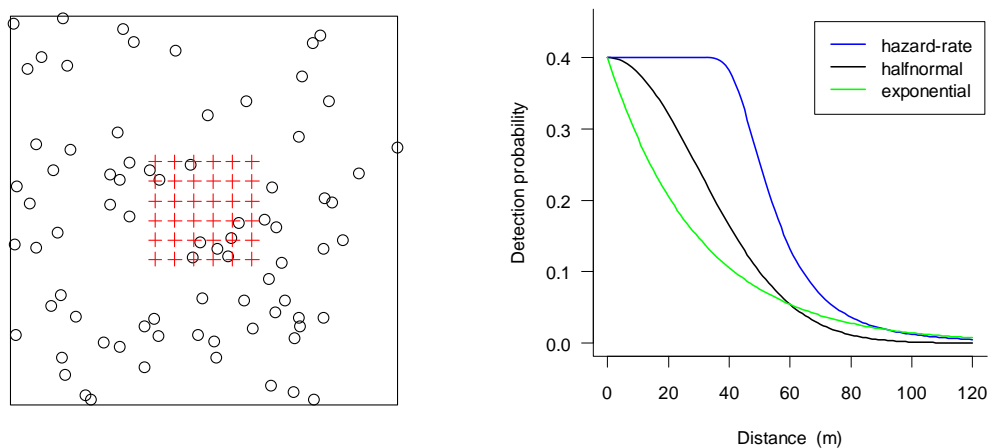


Fig. 1. Hypothetical Poisson distribution of range centres near an array of detectors (left) and alternative shapes for a function relating detection at a point to distance from range centre (right).

## Detector types

Properties of the detectors are an important part of the observation model. Some common detectors (camera ‘traps’ and hair snags for DNA) do not capture animals but provide evidence of their presence, and can be considered to act independently of each other. We call these ‘proximity’ detectors. One consequence of independence is that each animal  $\times$  occasion ‘cell’ of detection history data may contain several positive records; it may also be seen as a binary vector for presence or absence at each detector.

Detectors that are true traps do not act independently because capture of an animal in one trap prevents capture in another (until the animal is released). Traps pose competing risks of capture. The per-trap probability of capture may be adjusted for the competing risk from other traps by using an additive hazard model (Borchers and Efford 2008). However, if the detectors are traps that catch only one animal at a time then there is a further level of competition – between animals for traps. No general adjustment has been found for the per-trap probability of capture in this case. Multi-catch and single-catch traps therefore represent distinct detector types.

## Software

The Windows application DENSITY (Efford et al. 2004, Efford 2009) has been a useful platform for developing and disseminating these methods to biologists, but it lacks portability and transparency. Also, DENSITY fits only homogeneous Poisson models, neglecting the potential of inhomogeneous Poisson models. The package **secr** implements almost all the methods and options described in Borchers and Efford (2008) and Efford et al. (2009). It also has tools for data manipulation. Its relationship to DENSITY 4 is sketched in Appendix 2. Important functions are listed by their principal use in Appendix 3. Computationally intensive operations use external C code for speed.

The package defines a set of R classes<sup>1</sup> and methods for data from detector arrays. The most important classes are:

<code>traps</code> <sup>2</sup>	locations of detectors; detector type ('proximity', 'multi', 'single')
<code>capthist</code>	spatial detection histories; related <code>traps</code> object
<code>mask</code>	points on habitat mask
<code>secr</code>	fitted SECR model.

To perform an SECR analysis you will construct each of these objects in turn, using the functions provided (e.g., `make.grid`, `make.capthist`, `secr.fit`). Fig. 2 summarizes the relationships among the core object classes. The `traps`, `capthist` and `mask` classes also store the values of optional covariates specific to detectors, animals and habitat points respectively. (Each set of covariates is technically a dataframe saved as an attribute of the corresponding object; the 'covariates' method is used to extract or replace covariates).

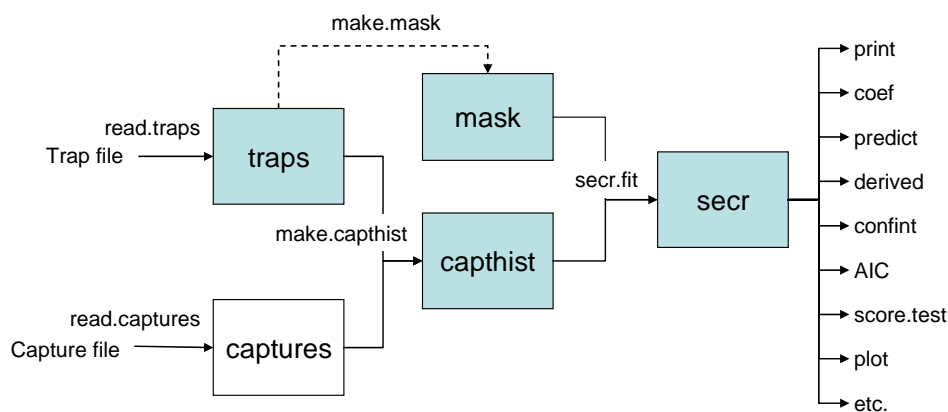


Fig 2. Essentials of the **secr** package. Each object class (shaded boxes) comes with methods to display and manipulate the data it contains (e.g. `print`, `summary`, `plot`, `rbind`, `subset`). Detector coordinates ('traps') are stored with attributes such as detector type and usage. Detection data ('captures') are initially stored in a dataframe with one row per detection. If a habitat mask is not created manually (dashed arrow) it will be generated automatically by `secr.fit`. Any of the objects input to `secr.fit` may include a dataframe of covariates whose names may be used in a model formula. Fitted `secr` models may be further manipulated with the methods shown on the right. Additional functions (not shown) construct a regular detector array (e.g. `make.grid`, `make.circle`) or simulate detection of a known population (`sim.capthist`).

<sup>1</sup> A 'class' specifies a particular type of data object and the functions (methods) by which it is manipulated (computed, printed, plotted etc). See the R documentation for further explanation.

<sup>2</sup> Text in courier font refers to R objects that are documented in online help for the **secr** package, or in base R. A good place to start is the page for `secr.fit`.

## Components of SECR models

A family of capture–recapture models, such as the Cormack-Jolly-Seber models for survival, may include submodels<sup>3</sup> that allow for contingent variation in core parameters, including the effects of covariates. Annual survival, for example, may vary with the severity of winter weather. The MARK software has been particularly successful in packaging submodels for biologists, and **secr** aims for similar flexibility. The language of generalised linear models is convenient for describing submodels (e.g. Huggins 1989, Lebreton et al. 1992). Each parameter is treated as a linear combination of effects on its transformed ('link') scale. This is useful for combining effects because, given a suitable link function, any combination maps to a feasible value of the parameter. The logit scale has this property for probabilities in (0, 1), and the natural log scale works for positive parameters i.e. (0, +Inf). A 'sin' link (strictly  $\arcsin(2x-1)$ ) can also be used for probabilities in [0,1], and including the boundary values is sometimes useful (see MARK help). Only these link functions (logit, log, sin) and the null 'identity' link have so far been implemented in **secr**.

In **secr**, submodels are defined symbolically using the R formula notation. A separate linear predictor is used for each core parameter. Core parameters are 'real' parameters in the terminology of MARK, and **secr** uses that term to reduce confusion. Four real parameters are modelled in **secr** 1.2; these are denoted D (for density), g0, sigma and z. Only the last three real parameters, which jointly define detection probability as a function of location, can be estimated directly when the model is fitted by maximizing the conditional likelihood (CL = TRUE in `secr.fit`). D is then a derived parameter that is computed from an **secr** object with the function 'derived'. 'z' is a shape parameter that is used only when the detection function has the 'hazard-rate' form (Hayes and Buckland 1983).

For each real parameter there is a linear predictor of the form

$$y = X\beta,$$

---

<sup>3</sup> This use of 'submodel' is non-standard – maybe we'll find a better term.

where  $\mathbf{y}$  is a vector of parameter values on the link scale,  $\mathbf{X}$  is a design matrix of predictor values, and  $\boldsymbol{\beta}$  is a vector of coefficients. Each element of  $\mathbf{y}$  and corresponding row of  $\mathbf{X}$  relates to the value of the real parameter in a particular circumstance (e.g. density at a particular point in space, or detection probability of an animal on a particular occasion). The elements of  $\boldsymbol{\beta}$  are coefficients estimated when we fit the model; in the MARK package these are called 'beta parameters' to distinguish them from the transformed 'real' parameter values in  $\mathbf{y}$ .  $\mathbf{X}$  has one column for each element of  $\boldsymbol{\beta}$ . Design matrices are described in more detail in the next section.

### Design matrices

Design matrices are specific to a 'real' parameter. Each design matrix  $\mathbf{X}$  contains a column of '1's (for the constant or intercept term) and additional columns as needed to describe the effects in the submodel for the parameter. Depending on the model, these may be continuous predictors (e.g. air temperature to predict occasion-to-occasion variation in  $g_0$ ), indicator variables (e.g. 1 if animal  $i$  was caught before occasion  $s$ , 0 otherwise), or coded factor levels.

Within **seer**, a design matrix is constructed automatically from the input data (`capthist`) and the model formula (e.g. `model$g0`) in a 2-stage process. First, a data frame is built containing 'design data' with one column for each variable in the formula. Second, the R function `model.matrix` is used to construct the design matrix. This process is hidden from the user. The design matrix will have at least one more column than the design data; there may be more if the formula includes interactions or factors with more than two levels. For a good description of this general approach see the documentation for RMark (Laake and Rexstad 2008). The necessary design *data* are either extracted from the inputs (`capthist` and `mask`) or generated automatically (as is the indicator of previous capture, mentioned in the previous paragraph).

'Real' parameters fall into two groups: density (D) and detection ( $g_0$ ,  $\sigma$  and  $z$ ). Density and detection parameters are subject to different effects, so they use different design matrices as described in the next three sections.

## Detection submodels

For SECR, we want to model the detection of each individual  $i$  on occasion  $s$  at detector  $k$ . Given  $n$  observed individuals on  $S$  occasions at  $K$  detectors, there are therefore  $n.S.K$  detection probabilities of interest. We treat these as elements in a 3-dimensional array. Strictly, we are also interested in the detection probabilities of unobserved individuals, but these are estimated only by extrapolation from those observed so we do not include them in the array. Analyses may combine data from several independent samples, dubbed ‘sessions’, so there is potentially a fourth dimension for the  $R$  sessions. Here we refer to sessions only in passing.

In a null model, all  $n.S.K$  detection probabilities are assumed to be the same. The conventional sources of variation in capture probability (Otis et al. 1978) appear as variation either in the  $n$  dimension (‘individual heterogeneity’  $h$ ), or in the  $S$  dimension (‘time variation’  $t$ ), or as a particular interaction in these two dimensions (‘behavioural response to capture’  $b$ ). Combined effects are possible.

SECR introduces two sorts of additional complexity. Firstly, detection probability is no longer a scalar (even for a particular animal-occasion-detector combination); it is described by a ‘detection function’. The detection function may have two parameters (e.g.  $g_0$ ,  $\sigma$  for a half-normal function), three parameters (e.g.  $g_0$ ,  $\sigma$ ,  $z$  for the Hayes and Buckland hazard-rate function), or potentially more. Any of the parameters of the detection function may vary with respect to individual (subscript  $i$ ), occasion (subscript  $s$ ) or detector (subscript  $k$ ).

Secondly, many more types of variation are possible. For example, a behavioural response may be specific to the trap in which an individual was caught (e.g. future capture at that particular site is more or less likely because of a learned response).

The full design matrix for each detection submodel has one row for each combination of  $i$ ,  $s$  and  $k$ . Allowing a distinct probability for each animal (the ‘ $n$ ’ dimension) may seem excessive, and truly individual-specific covariates are feasible only when a model is fitted by maximizing the conditional likelihood (cf Huggins 1989). However, the full  $n.S.K$  array is convenient for coding both group membership (Lebreton et al. 1992, Cooch and White 2008) and experience of capture, even when pure individual-level heterogeneity cannot be modelled.



## Specifying effects on detection parameters

Effects on parameters of detection probability are specified with R formulae. The variable names used in formulae are either names for standard effects (Table 2) or the names of user-supplied covariates. The formula for any detection parameter ( $g_0$ ,  $\sigma$ ,  $z$ ) may be constant ( $\sim 1$ , the default) or some combination of terms in standard R formula notation (see `help(formula)`).

Table 2. Predictor variables used in detection models.

Variable	Description	Data source	Array dim
$g$	Group	interaction of the <code>capthist</code> individual covariates listed in argument <code>groups</code> of <code>secr.fit</code>	$n$
[user]	individual covariate	columns of <code>covariates</code> ( <code>capthist</code> ) as named in model formula	$n$
$t$	time factor (one level for each occasion)	[automatic]	$S$
$tcov^*$	time-specific covariate	argument <code>timecov</code> of function <code>secr.fit</code>	$S$
$kcov^*$	Detector-specific covariate	covariate from <code>traps</code> attribute of <code>capthist</code> :	$K$
$b$	learned response (step change in parameter after first detection of animal)	[automatic]	$n.S$
$B$	Transient (Markovian) response (parameter depends on detection at previous occasion)	[automatic]	$n.S$
[user]	session covariate	Argument <code>sessioncov</code> of function <code>secr.fit</code>	$R$
session	session factor (one level per session)	[automatic]	$R$

\*Other names may be used for time-specific and detector-specific covariates if they appear as columns in the respective covariate dataframes; these defaults use the first column.

For other effects, the design matrix for detection parameters may also be provided manually in the argument `dframe` of `secr.fit`. This feature is untested.

## Density submodels

The SECR log likelihood is evaluated by summing values at points on a 'habitat mask'. Each point in a habitat mask represents a grid cell of potentially occupied habitat (their combined area may be almost any shape). The full design matrix for density (D) has one row for each point in the mask. As for the detection submodels, the design matrix has one column for the intercept (constant) term and one for each predictor. Predictors may be based on Cartesian coordinates (e.g. 'x' for an east-west trend), a continuous habitat variable (e.g. vegetation cover) or a categorical (factor) habitat variable. Predictors should be measured at all points in the mask (non-habitat excluded). The variables 'x' and 'y' are the coordinates of the habitat mask; other covariates should be named columns in the 'covariates' attribute of the habitat mask.

## Output

The output from the function `secr.fit` is an object of class `secr`. Assigning the output to a named object (such as `secr0` or `secrb` in the example) saves both the fit and the data for further manipulation. Typing the name invokes `print.secr` with results like those in Appendix 1. Functions are provided for further computations on `secr` objects (e.g., density as a derived parameter, profile-likelihood confidence intervals for beta parameters, AIC model selection, likelihood ratio and score tests).

In **secr**, all input and output assumes distances are measured in metres and areas in hectares (1 hectare (ha) = 10000 m<sup>2</sup>). A density of 1 animal per ha corresponds to 100 animals per km<sup>2</sup>.

Help is obtained in the usual way by typing a question mark at the R prompt, followed by a function or dataset name. Open the manual from the R prompt by typing

```
RShowDoc ('secr-manual', package='secr')
```

## References

- Borchers, D. L., Buckland, S. T. and Zucchini, W. (2002) *Estimating animal abundance: closed populations*. Springer, London.
- Borchers, D. L. and Efford, M. G. (2008) Spatially explicit maximum likelihood methods for capture–recapture studies. *Biometrics* **64**, 377–385.
- Cooch, E. and White, G. (eds) (2008) *Program MARK: A Gentle Introduction*. 6th edition. Available online at <http://www.phidot.org>.
- Efford, M. G. (2004) Density estimation in live-trapping studies. *Oikos* **106**, 598–610.
- Efford, M. G. (2009) *DENSITY 4.4: software for spatially explicit capture–recapture*. Department of Zoology, University of Otago, Dunedin, New Zealand  
<http://www.otago.ac.nz/density>.
- Efford, M. G., Borchers D. L. and Byrom, A. E. (2009) Density estimation by spatially explicit capture–recapture: likelihood-based methods. In: D. L. Thomson, E. G. Cooch, M. J. Conroy (eds) *Modeling Demographic Processes in Marked Populations*. Springer. Pp 255–269.
- Hayes, R. J. and Buckland, S. T. (1983) Radial-distance models for the line-transect method. *Biometrics* **39**, 29–42.
- Huggins, R. M. (1989) On the statistical analysis of capture experiments. *Biometrika* **76**, 133–140.
- Laake, J. and Rexstad E. (2008) Appendix C. RMark - an alternative approach to building linear models in MARK. In: Cooch, E. and White, G. (eds) *Program MARK: A Gentle Introduction*. 6th edition. Available online at <http://www.phidot.org>.
- Lebreton, J.-D., Burnham, K. P., Clobert, J., and Anderson, D. R. (1992) Modeling survival and testing biological hypotheses using marked animals: a unified approach with case studies. *Ecological Monographs* **62**, 67–118.
- Otis, D. L., Burnham, K. P., White, G. C. and Anderson, D. R. (1978) Statistical inference from capture data on closed animal populations. *Wildlife Monographs* **62**.
- Royle, J. A., Nichols, J. D., Karanth, K. U. and Gopalaswamy, A. M. (2009). A hierarchical model for estimating density in camera-trap studies. *Journal of Applied Ecology* **46**, 118–127.

Appendix 1. Example of secr output (see box on p. 2 for R code).

```
> AIC (secr0, secrb)                                ## compare fit of models
      model detectfn npar   logLik      AIC      AICc dAICc  AICwt
secr0 D~1 g0~1 sigma~1 halfnormal    3 -759.0198 1524.040 1524.373 0.000 0.7513
secrb D~1 g0~b sigma~1 halfnormal    4 -759.0106 1526.021 1526.584 2.211 0.2487
```

A model with learned trap response (g0~b) showed no improvement in fit over a null model. In this instance the estimates of density from the two models were also very close (not shown) and we rely on the null model for estimation.

```
> secr0                                              ## display estimates
secr.fit(capthist = myCH, model = g0 ~ 1)
19 Aug 2009 18:09:09

N animals      : 76
N captures     : 235
N occasions    : 5
N detectors     : 100

Detector type  : multi

Model          : D~1 g0~1 sigma~1
Fixed          : none
Detection fn   : halfnormal
Distribution    : poisson
N parameters   : 3
Log likelihood : -759.0198
AIC            : 1524.040
AICc           : 1524.373

Beta parameters (coefficients)
      beta      SE.beta      lcl      ucl
D      1.7009396 0.11763559 1.470378 1.931501
g0     -0.9783632 0.13623934 -1.245387 -0.711339
sigma  3.3798711 0.04442682 3.292796 3.466946

Unconditional variance-covariance matrix of beta parameters
      [,1]      [,2]      [,3]
[1,] 0.0138381320 0.0001592604 -0.0009937316
[2,] 0.0001592604 0.0185611579 -0.0033439722
[3,] -0.0009937316 -0.0033439722 0.0019737426

Fitted (real) parameters evaluated at base levels of covariates
      link estimate SE.estimate      lcl      ucl
D      log 5.4790929 0.64677256 4.3508797 6.8998597
g0     logit 0.2732167 0.02705295 0.2234996 0.3293030
sigma  log 29.3669858 1.30532592 26.9180251 32.0387493
```

The density estimate is  $5.48 \text{ ha}^{-1}$  (95% confidence interval 4.35–6.90  $\text{ha}^{-1}$ ). The calculation used a default habitat mask with a buffer of 100 m around the detectors; this is reasonable in the light of the estimate of sigma (29.4 m).

Appendix 2. Comparison of DENSITY 4.4 and **secr** 1.2.

Feature	DENSITY 4.4	<b>secr</b> 1.2
Graphical interface	Yes	No
Inverse prediction (IP) [single-catch traps]	Yes	Yes (1.2.7)
Maximum likelihood estimation (MLE) [proximity detectors and multi-catch traps]	Yes	Yes
Non-spatial open-population models (CJS etc.)	Yes	No
Simulation of spatial sampling	Yes (many options)	Yes
Build detector arrays	Yes (many options)	Yes
Finite mixture models for individual heterogeneity	Yes	No
Fixed parameters (MLE)	Yes (rough)	Yes
‘Session’-level models (trend etc.) (MLE)	Yes	Yes
Groups	No	Yes
Profile likelihood confidence intervals (MLE)	Yes	Yes
Jackknife confidence intervals (MLE)	Yes	No
Set of detectors used may vary with occasion	Yes	Yes
Formula-based model notation	No	Yes
Density models (inhomogeneous Poisson distribution; MLE)	No	Yes
Model within-session variation in shape parameter $z$ of hazard-rate detection (MLE)	No	Yes
Terminology consistent with MARK (groups, ‘real’ parameters, ‘beta’ parameters)	No	Yes
Score tests	No	Experimental
Model averaging	No	Yes
Parametric bootstrap	Incomplete	Yes
Structural relationships between real parameters	No	Yes
‘pdot’ criterion for region of integration	No	Yes
Random number generators	Borland Pascal	R (full control)

Appendix 3. Functions in **secr** 1.2.11 arranged according to use (many functions for data manipulation and plotting are omitted). S3 methods are marked with an asterisk \*

*Manipulate core objects*

<code>make.grid</code>	construct detector array
<code>read.traps</code>	input detector locations from text file
<code>read.captures</code>	input detection (capture) data in Density format
<code>make.caphist</code>	form caphist object from traps object and detection data
<code>make.mask</code>	construct habitat mask (mesh)
<code>sim.caphist</code>	simulate capture histories
<code>verify*</code>	check caphist, traps or mask object for internal consistency

*Extract or replace attributes of traps object*

<code>covariates*</code>	detector-level covariates
<code>detector*</code>	detector type ('multi', 'proximity' etc.)
<code>usage*</code>	disable detectors (occasion- and detector-specific )

*Extract or replace attributes of caphist object*

<code>covariates*</code>	individual-level covariates, including grouping factors
<code>session*</code>	session identifier(s)
<code>traps*</code>	embedded 'traps' object(s)

*Fit SECR model*

<code>secr.fit</code>	maximum likelihood fit; result is a fitted secr object
<code>ip.secr</code>	fit simple SECR model by simulation & inverse prediction

*Operate on fitted secr object(s)*

<code>AIC*</code>	model selection, model weights
<code>coef*</code>	'beta' parameters
<code>collate</code>	tabulate estimates from several models
<code>confint*</code>	profile likelihood confidence intervals
<code>derived</code>	density from conditional likelihood models
<code>deviance*</code>	model deviance
<code>df.residual*</code>	degrees of freedom for deviance
<code>LR.test</code>	likelihood-ratio test of two models
<code>model.average</code>	combine estimates using AICc weights
<code>plot*</code>	plot detection functions with confidence bands
<code>predict*</code>	'real' parameters for arbitrary levels of predictor variables
<code>score.test</code>	model selection with score statistic using observed information
<code>simulate*</code>	generate realisations of fitted model
<code>sim.secr</code>	parametric bootstrap
<code>vcov*</code>	variance-covariance matrix of 'beta' or 'real' parameters

*Miscellaneous*

<code>autoini</code>	generate starting values of D, g0 and sigma for <code>secr.fit</code>
----------------------	---

counts	summary data from capthist object
dbar	a simple (and unreliable) home-range measure
distancetotrap	from an arbitrary set of points
nearesttrap	from an arbitrary set of points
pdot	location-specific net probability of detection
RPSV	another simple and unreliable home-range measure

*Test data* [restore with `data(xxx)`]

rawdata	dataframes of raw data
captdata	rawdata as a capthist object
secrdemo	secr.fit applied to some simulated data
ovenbird	multi-year mist-netting study of ovenbirds at a site in Maryland, USA. Dawson & Efford <i>Journal of Applied Ecology</i> in press.
possum	brushtail possum <i>Trichosurus vulpecula</i> live trapping at Waitarere, North Island, New Zealand April 2002 (Efford et al. 2005 <i>Wildlife Society Bulletin</i> 33: 731–738.