

## APPLICATION

## ipsecr: An R package for awkward spatial capture–recapture data

Murray G. Efford 

Department of Mathematics and  
Statistics, University of Otago, Dunedin,  
New Zealand

## Correspondence

Murray G. Efford

Email: [murray.efford@otago.ac.nz](mailto:murray.efford@otago.ac.nz)

Handling Editor: Edward Codling

## Abstract

1. Some capture–recapture models for population estimation cannot easily be fitted by the usual methods (maximum likelihood and Markov-chain Monte Carlo). For example, there is no straightforward probability model for the capture of animals in traps that hold a maximum of one individual ('single-catch traps'), yet such data are commonly collected. It is usual to ignore the limit on individuals per trap and analyse with a competing-risk 'multi-catch' model that gives unbiased estimates of average density. However, that approach breaks down for models with varying density.
2. Simulation and inverse prediction was suggested by Efford (2004) for estimating population density with data from single-catch traps, but the method has been little used, in part because the existing software allows only a narrow range of models. I describe a new R package that refines the method and extends it to include models with varying density, trap interference and other sources of non-independence among detection histories.
3. The method depends on (i) a function of the data that generates a proxy for each parameter of interest and (ii) functions to simulate new datasets given values of the parameters. By simulating many datasets, it is possible to infer the relationship between proxies and parameters and, by inverting that relationship, to estimate the parameters from the observed data.
4. The method is applied to data from a trapping study of brushtail possums *Trichosurus vulpecula* in New Zealand. A feature of these data is the high frequency of non-capture events that disabled traps (interference). Allowing for a time-varying interference process in a model fitted by simulation and inverse prediction increased the steepness of inferred year-on-year population decline. Drawbacks and possible extensions of the method are discussed.

## KEYWORDS

density estimation, interference, non-independence, non-target captures, secr, single-catch traps, spatial capture–recapture, trap saturation

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](https://creativecommons.org/licenses/by-nc-nd/4.0/) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2023 The Author. *Methods in Ecology and Evolution* published by John Wiley & Sons Ltd on behalf of British Ecological Society.

## 1 | INTRODUCTION

Analysis of spatial capture–recapture data to estimate animal population density requires a model for the probabilities of interaction between individuals and sampling devices at known locations ('detectors'). Each probability is assumed to decline with the distance between a detector and an individual's stationary, but unknown, activity centre according to a function such as halfnormal or negative exponential. Devices such as automatic cameras register the presence of an individual without interrupting its movement, and detections may plausibly be modelled as independent. However, if the device is a physical trap then the animal is detained and cannot be caught in nearby traps until released at the end of a trapping interval. If additional individuals may be caught in the same trap then non-independence among traps is readily addressed with a competing risk model (Borchers & Efford, 2008). For traps that can catch at most one individual ('single-catch' traps), the probability depends jointly on competition among individuals and competition among traps, and is not directly computable.

Maximum likelihood and Bayesian methods for spatial capture–recapture have become popular in the last two decades, but neither is strictly applicable to discrete-time data from the single-catch traps used widely in studies of small mammals (Gerber & Parmenter, 2015; Otis et al., 1978; Romairone et al., 2018; Royle & Converse, 2014). A common solution has been to use the model for multi-catch traps, as estimates of average density from the mis-specified model are virtually unbiased (Distiller & Borchers, 2015; Efford et al., 2009). This has two drawbacks. Estimates of the detection function intercept from the mis-specified model become increasingly biased as trap saturation (the proportion of occupied traps) increases; this is a problem if estimates are used in simulations to compare study designs. Further, Distiller and Borchers (2015) showed that high trap saturation could result in biased estimates of spatial trend in density from mis-specified (multi-catch) models. Their solution requires data collected in continuous time and appears to have had little uptake.

I introduce an R (R Core Team, 2022) package *ipsecr* (Efford, 2023a) that implements a closed-population method for data from single-catch traps and other awkward capture–recapture data types. The software relaxes some limitations of the method as originally proposed and implemented (Efford, 2004; Efford et al., 2004). Although motivated by the single-catch-trap problem, the software has application to other scenarios that entail hard-to-model dependence among animals. It readily incorporates a competing process ('interference') that disables traps and increases trap saturation. Data objects and some functions are shared with the R package *secr* whose documentation should be consulted for data input (Efford, 2022).

For a closed population (one with no gains or losses during sampling), the goal of spatially explicit capture–recapture is to estimate population density  $D$  or a related parameter such as spatial trend or population size. The state model is a spatial point process whose points represent the fixed activity centres of individuals. Observations are made on  $S$  occasions at  $K$  detectors. The observation model is a spatial detection function that in the typical

half-normal case has two parameters—intercept  $\lambda_0$  and spatial scale  $\sigma$ . Detection distances are not observed, but the function may be fitted by integrating over possible activity centres (maximum likelihood, MLE) or by treating centres as latent variables (Markov chain Monte Carlo, MCMC). Further details are given by Efford (2004), Borchers and Efford (2008), Efford et al. (2009) and Royle et al. (2014) and more recent sources. In the basic model, each of the core parameters is constant (expressed as  $D \sim 1$ ,  $\lambda_0 \sim 1$ ,  $\sigma \sim 1$ ). This may be extended by treating each parameter as a function of known covariates that is linear on a user-selected 'link' scale. Thus,  $D \sim x$  may represent an east–west trend in density with coefficients intercept and slope. Coefficients of the linear submodels are then the parameters to be estimated.

## 2 | METHOD

The application uses simulation and inverse prediction to fit spatial capture–recapture models. Datasets are simulated for known levels of the parameters in the model and a vector of proxy statistics is computed from each dataset (a proxy is a measure closely correlated with the parameter of interest). A multivariate multiple linear regression (Johnson & Wichern, 2007) is fitted with parameter vectors as the predictor variable and proxy vectors as the response variable. From the inverted linear model we can infer parameter values from a single proxy vector; specifically, we infer the parameters corresponding to a target vector of proxies from the observed data. The method is mathematically equivalent to controlled multivariate calibration (Brown, 1982; Osborne, 1991). It has been used in capture–recapture to infer the degree of capture heterogeneity in non-spatial open-population data (Carothers, 1979; Pledger & Efford, 1998). A concise mathematical description follows.

Consider a vector of  $m$  unknown parameter values  $\mathbf{x}_p$  and an observed vector  $\mathbf{y}_p$  of proxies of length  $q \geq m$  computed from the data. By simulation, we cheaply generate many vectors  $\mathbf{y}$  corresponding to known parameter vectors  $\mathbf{x}$ , chosen to bracket a plausible value of  $\mathbf{x}_p$ . The simulated data may be described by the multivariate multiple regression:

$$\mathbf{y} = \lambda + \mathbf{B}\mathbf{x} + \mathbf{E}, \quad (1)$$

where  $\lambda$  is a  $q \times 1$  vector of intercepts,  $\mathbf{B}$  is a  $q \times m$  matrix of coefficients and  $\mathbf{E}$  is a  $q \times 1$  vector of errors with  $\text{MVN}(\mathbf{0}, \mathbf{V})$  distribution. The intercepts  $\lambda$  and coefficients  $\mathbf{B}$  are estimated by fitting the regression with standard software such as R function 'lm' (R Core Team, 2022). Point estimates of the parameters are obtained from the vector of observed proxies with

$$\hat{\mathbf{x}}_p = \hat{\mathbf{B}}^{-1}(\mathbf{y}_p - \hat{\lambda}). \quad (2)$$

Additional simulations at  $\hat{\mathbf{x}}_p$  provide a direct estimate of the final proxy variance–covariance matrix  $\mathbf{V}_p$ , and the variance–covariance matrix of the parameter estimates is derived as  $\Gamma = \hat{\mathbf{B}}^{-1} \mathbf{V}_p \hat{\mathbf{B}}^{-1T}$ .

The assumption of approximate multivariate linearity may be met in practice by choosing suitable link scales for the predictor and response variables and by considering a small region in parameter space that brackets the estimates. The region is defined by a hyperbox that is centred initially on arbitrary starting values and iterated until it includes the vector of inferred parameter values. The process is depicted in Figure 1. The function `ipsecr.fit` allows the user to customise most aspects of model fitting, including the computation of proxies, the simulation code and various tuning parameters such as box size, but flexible defaults are provided as I describe further.

## 2.1 | Proxies

The choice of proxy is arbitrary and something of an art. Each parameter requires a distinct proxy, and the pairwise relationships should be monotonic and ideally linear after transformation of the parameter. A proxy should depend closely on the corresponding parameter when other parameters are held constant. These constraints are not absolute—there may be more than one proxy per parameter, and marginal monotonicity is not essential (Brown, 1982)—but they give clarity in proxy selection.

The vector of proxies is computed by a function that takes a 'secr' capthist data object as its first argument. The default in `ipsecr.fit` is to use the exported function `proxy.ms` that can provide proxies for different models as specified in the second argument of the function. The suffix 'ms' indicates that the data may span multiple

independent sessions, as understood in 'secr'. For a constant model, `proxy.ms` returns a vector with one proxy for each of the core parameters ( $D$ ,  $\lambda_0$ ,  $\sigma$ ). The corresponding default proxies use

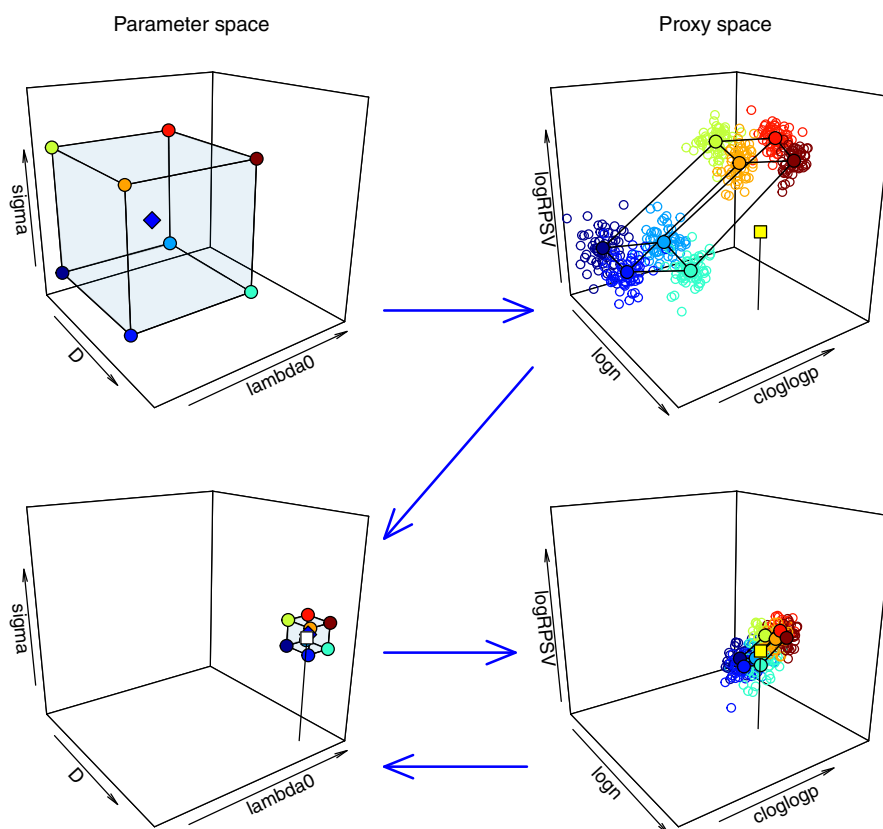
- $n$  the number of distinct individuals detected,
- $p^*$  a naive non-spatial estimate of detection probability, and
- $\sigma^*$  a naive estimate of the scale of movement.

The naive detection probability is  $p^* = \sum c_{is} / (nS)$ , where  $c_{is}$  is an indicator for capture of animal  $i$  on occasion  $s$ , and  $S$  is the number of capture occasions. The naive scale of detection for a circular bivariate normal model is calculated using the formula of Calhoun & Casby (1958, Equation 8a):

$$\sigma^* = \sqrt{\frac{\sum_{i=1}^n \sum_{j=1}^{d_i} [(x_{ij} - \bar{x}_i)^2 + (y_{ij} - \bar{y}_i)^2]}{2 \sum_{i=1}^n (d_i - 1)}}, \quad (3)$$

where  $(x_{ij}, y_{ij})$  is the vector of coordinates for the  $j$ -th capture of the  $i$ -th animal, with animal-specific centroid  $(\bar{x}_i, \bar{y}_i)$  and number of captures  $d_i$ . The estimator is naive because animals can be observed only at the detector locations and home ranges are truncated at the boundary of the study area. In 'secr',  $\sigma^*$  is denoted RPSV, named for the root pooled spatial variance of capture locations (Efford, 2022).

The statistics are transformed in the default function to stabilise the variance and improve linearity. Thus, the actual proxies are  $\log(n)$ ,  $\text{cloglog}(p^*)$  and  $\log(\sigma^*)$ , where 'cloglog' is the complementary log-log transformation  $y = \log(-\log(1-x))$ .



**FIGURE 1** Steps for fitting secr model by simulation and inverse prediction. Simulations are conducted at the vertices of a box in parameter space (top left; link scale) centred on an initial guess (blue diamond). The results in proxy space (top right; frame connects design point means, centre omitted for clarity) support a linear model for proxies as a function of parameters. The model is inverted and applied to the observed proxy vector (yellow square) giving the centre of a new, smaller box in parameter space (bottom left). The model is refined by further simulations (bottom right) from which the final parameter estimates are inferred (white square, bottom left). Function `ipsecr.fit` performs all steps.

## 2.2 | Parameter submodels

Earlier implementations were limited to the constant model. `proxy.ms` expands the scope of the method by providing proxies for the coefficients of linear submodels for each of  $D$ ,  $\lambda_0$  and  $\sigma$ . In each case, we require a quantity for which a proxy submodel provides a close analogy to the submodel for the `secr` parameter, as described next. The approach is to fit a generalised linear model (R function 'glm'; R Core Team, 2022) to observed counts or to a statistic summarising the spatial spread of individual detections. The specifics are parameter-dependent (Table 1). Coefficients of the glm become proxies for the coefficients of the corresponding parameter in the fitted model. Multi-session data are accepted, and any of the following submodels may include 'session' as a factor-level covariate, or use session-specific covariates as in 'secr'.

### 2.2.1 | Varying density $D$

Density in `secr` models is defined across a region that includes the detectors, but extends beyond them. Information on variation in density across the wider region is provided by the distribution of observations across the detector array. In `proxy.ms`, the spatial model for density across the region is replaced by a spatial model for the number of individuals  $n_k$  observed at each detector  $k$ . Habitat covariates in the density model must also be defined as detector covariates.

### 2.2.2 | Varying $\lambda_0$

The glm used as a proxy for  $\lambda_0$  predicts the number of detections of each animal  $i$  on each occasion  $s$  ( $d_{is}$ ) as a function of spatial or individual covariates. The model is binomial when there is a maximum of one detection per animal per occasion (i.e. at physical traps); otherwise a Poisson model is used. Individual covariates are provided in the 'covariates' attribute of the spatial capture histories. Spatial covariates are inferred from the centroid of the locations where each individual is detected.

### 2.2.3 | Varying $\sigma$

Variation in the scale of detection is related to the spatial distribution of the detections of each individual, pooled across occasions.

Data are usually sparse at the individual level, and individuals caught only once provide almost no information for a proxy submodel of  $\sigma$ . Applying Equation 3 separately to each animal  $i$  that was detected at least twice yields data  $\sigma_i^*$  that may be used in a submodel for individual variation in  $\sigma$ .  $\sigma_i^*$  takes non-negative values. `proxy.ms` fits a glm to the non-missing  $\log(\sigma_i^* + 1)$  using the number of detections per animal as weights. Spatial or individual covariates may be applied as for varying  $\lambda_0$ .

## 2.3 | Simulation model

Simulation has two stages—generating a population of activity centres from the state model, and generating a spatial capture–recapture sample ('secr' capthist object) from that using the observation model. Each stage by default uses a built-in function (`simpop`, `simCH`) designed for the range of models for which proxies were described above. The user may substitute other functions.

Observation models that entail competing risk of capture are readily simulated by generating a random latent time-to-event for each possible event (e.g. animal  $i$  in trap  $k$ ). Events are selected one-by-one from the ordered list of times until the sampling interval ends, while eliminating pre-empted events from further consideration (e.g. animal  $i$  cannot be caught elsewhere until released from trap  $k$ , and if trap  $k$  is 'single-catch', it cannot catch another animal; Efford, 2004). This is the approach in 'secr' and 'ipsecr'. Latent time-to-event simulation is not appropriate when the hazards are dependent or time-varying (e.g. Beyersmann et al., 2009), but modelling of capture–recapture data rarely involves such complexity. Royle et al. (2014, p. 267) described an approximate algorithm for simulating from single-catch traps, but approximation is unnecessary.

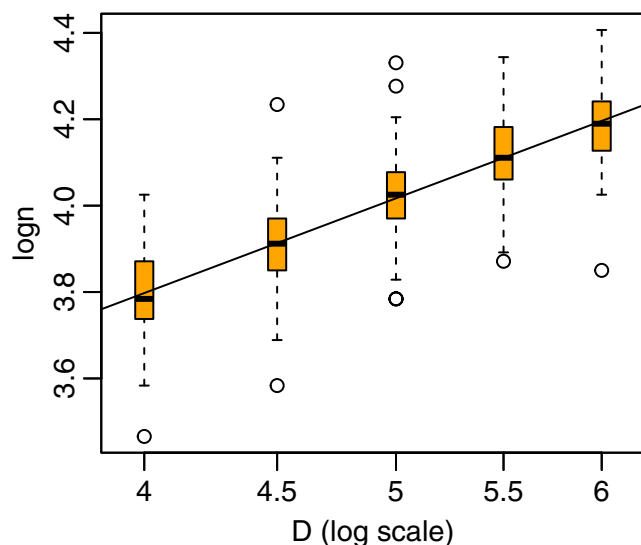
## 2.4 | Proxy validation

Proxies may be validated by plotting simulated values. The package includes function `plotProxy` for this purpose. The code below is an example using the detector array for the dataset 'captdata' from the `secr` package, with the result in Figure 2.

```
library(ipsecr)
trps <- traps(captdata)
msk <- make.mask(trps, buffer = 100)
```

Parameter	Proxy response variable	Detector type	Family	Link
$D$	Individuals at detector $k$ ( $n_k$ )	All	poisson	log
$\lambda_0$	Detections of individual $i$ on	single, multi	binomial	cloglog
	Occasion $s$ ( $d_{is}$ )	proximity, count	poisson	log
$\sigma$	Individual $\sigma^*(\log(\sigma_i^* + 1))$	All	gaussian	identity
$\lambda_{NT}$	State of detector $k$ , occasion $s$ ( $q_{sk}$ )	single, multi	binomial	cloglog

**TABLE 1** Generalised linear models used in function `proxy.ms` for proxies of spatially explicit capture–recapture parameters. The interference parameter  $\lambda_{NT}$  is introduced in a later section.



**FIGURE 2** Relationship between the proxy variable  $\log(\text{number of individuals detected})$  ('logn') and the density parameter  $D$  while holding detection parameters constant. Distribution of values for 100 simulated datasets at each level. Boxes span 25th to 75th percentiles. The relationship is nearly linear for a narrow range of densities.

```
base <- list(D = 5, lambda0 = 0.2, sigma = 25)
plotProxy (parameter = "D", proxyfn = proxy.ms, traps = trps,
mask = msk, basepar = base, boxplotargs = list(col = "orange"), nrepl = 100)
```

### 3 | PACKAGE STRUCTURE AND VALIDATION

The interface for model fitting follows the parent package *secr* (Efford, 2022). The fitting function *ipsecr.fit* returns an R object of class 'ipsecr' for which there are the usual S3 methods (*predict*, *print*, *summary*, *coef*, *vcov*, *plot*). Some functions of 'secr' also work with 'ipsecr' objects (e.g. *collate* for tabulating estimates from several models). C++ code is used via the R package *Rcpp* (Eddelbuettel & Francois, 2011) for fast simulation in the default functions *simpop* and *simCH*. Simulations are distributed among multiple cores, as specified with *setNumThreads*, for a considerable reduction in the total execution time.

Simulations were performed to check the performance of the software. Some of these essentially confirmed results from Efford (2004) and Efford et al. (2009) and are reported only in the Supplements (Appendix 1). Others are described below.

### 4 | TUNING THE ALGORITHM

The simulation-based method is inevitably slower than likelihood maximisation—in the preceding example by a factor of 12. This directs attention to software settings that might reduce fitting

**TABLE 2** Tuning parameters for fitting model with *ipsecr.fit*.

Parameter	Default	Description
factorial	'full'	Full vs fractional factorial design
boxsize1	0.2	Dimension(s) of first hyperbox
boxsize2	0.05	Dimension(s) of second and later hyperboxes
boxtype	'absolute'	Absolute vs relative scaling of box size
centre	3	Number of centre points
min.nsim	20	Minimum number of simulations per vertex
max.nsim	200	Maximum number of simulations per vertex
dev.max	0.002	Precision criterion for stopping rule
min.nbox	2	Minimum number of hyperboxes
max.nbox	5	Minimum number of hyperboxes
var.nsim	2000	Number of bootstrap simulations for variance

time without intolerable loss of accuracy. Tuning parameters are specified in the 'details' argument of *ipsecr.fit* (Table 2). The user specifies the minimum and maximum number of simulations at each design point (hyperbox vertex or centre) and the desired precision of proxy means. Simulations are added in blocks of the minimum number until the target precision is achieved for all proxies or the maximum number is reached. The default settings have been found to work well with some datasets, such as those used in the examples, but no claim has been made for their efficiency in general.

The default is to simulate at the vertices of a hyperbox in parameter space, that is to use a full factorial experimental design with two levels of each of the  $np$  parameters (plus some central points). The required number of simulations increases with the number of vertices  $2^{np}$ , and may become prohibitive for large  $np$ . A fractional factorial design reduces the burden; fractional designs may be specified in *ipsecr.fit* using the framework of Groemping (2014).

### 5 | INTERFERENCE

Detectors for spatial capture–recapture may become inactive during the sampling period due to processes broadly termed 'interference'. For single-catch traps, these include capture of non-target individuals, and triggering or removal of bait without capture of the target species. Interference may be modelled as a Poisson process that competes with the capture process of the target species. We assume traps that have yet to catch an animal are subject to hazard  $\lambda_{NT}$  of entering the 'disturbed' state (parameter 'NT' for 'non-target' in package *ipsecr*). In simulations, a latent random 'time of disturbance' is drawn for the interference process at each trap, and a trap

becomes inactive if that precedes potential capture events. The hazard is non-spatial in that it is not attributed to localised individuals of non-target species, although it may vary between sampling occasions and among detectors. The default proxy for  $\lambda_{NT}$  in 'ipsecr' is the complementary log-log of the observed proportion of disturbed traps, excluding any that capture the target species.

## 6 | EXAMPLES

### 6.1 | Single-catch traps

I first analyse data from one session of a long-term live-trapping study of brushtail possums *Trichosurus vulpecula* (Efford & Cowan, 2004). The data are from 5 nights of sampling with 167 single-catch traps at 30-m spacing in February, 1996, when population density was high and 47%–61% of traps were occupied nightly. The model considered possums with activity centres within 120m of a trap, excluding an adjoining shingle river bed. Full code is in [Appendix 2](#). I start here with data prepared as the object 'Feb96'.

For reference, first fit the mis-specified multi-catch model:

```
fitML <- secr.fit(Feb96, mask = msk, detectfn = 'HHN')
predict(fitML)[-c(1,3)]
```

#	estimate	lcl	ucl
# D	14.3802	12.5447	16.4842
# lambda0	0.1015	0.0854	0.1206
# sigma	27.3765	25.5349	29.3508

Next, fit the single-catch model by simulation and inverse prediction ([Figure 1](#)):

**TABLE 3** Frequency of trap states over 5 nights of trapping for brushtail possums in February of consecutive years. 'Disturbed' includes traps that were triggered or from which the bait was removed.

	1996		1997	
	n	%	n	%
Possum captures	450	53.9	383	45.9
Disturbed	349	41.8	372	32.6
Undisturbed	36	4.3	180	21.6
Total	835	100.0	835	100.0

Model of interference	Density (possums/ha)		Interference $\lambda_{NT}$	
	1996	1997	1996	1997
None	12.9 (11.1,14.9)	10.4 (9.0,11.9)	—	—
Constant	13.1 (11.3,15.2)	10.2 (8.9,11.7)	0.96 (0.88,1.04)	
Year-specific	15.0 (13.0,17.3)	8.7 (7.8, 9.7)	1.50 (1.31,1.71)	0.71 (0.61,0.83)

```
fitIP <- ipsecr.fit(Feb96, mask = msk, detectfn = 'HHN')
predict(fitIP)[-c(1,3)]
```

#	estimate	lcl	ucl
# D	14.1773	12.5152	16.060
# lambda0	0.1555	0.1279	0.189
# sigma	27.4818	25.7748	29.302

As expected, the estimates of density ( $D$ , possums per hectare) and detection scale ( $\sigma$ , m) are very similar, but the estimate of the baseline detection parameter ( $\lambda_0$ ) is considerably larger when the model is correctly specified. Inverse prediction was also used to fit the mis-specified multi-catch model, with results that were very similar to the MLE ([Appendix 2](#)).

Note also that the averages of the proxy values simulated 2000 times at the final parameter estimates are a close match to the target values from proxy.ms(Feb96):

```
fitIP$variance.bootstrap[-c(1,3)]
```

#	target	simulated
# logn	5.4072	5.399
# cloglogp	-0.6601	-0.657
# logRPSV	3.2605	3.260

### 6.2 | Single-catch traps with interference

The brushtail possum study experienced a high and variable rate of trap interference largely due to ship rats *Rattus rattus* that removed baits and triggered traps, but escaped through the mesh. The analysis in the previous example accounted for traps becoming inactive when they caught a possum, but it did not account for the high frequency of inactivation from other causes ([Table 3](#)).

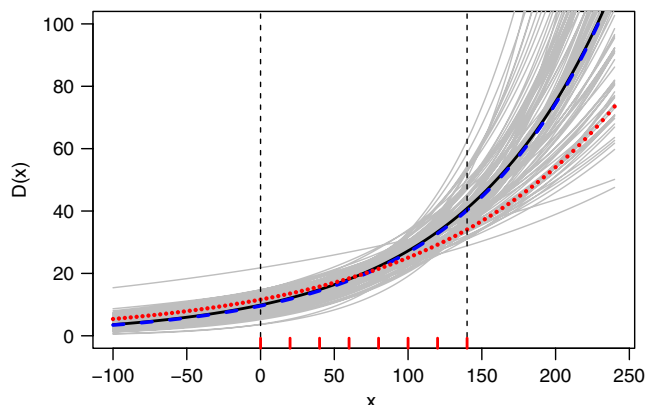
I extend the possum example to contrast density in February 1996 and February 1997, with and without allowance for interference (see [Appendix 2](#) for details). A model with year-specific interference suggested a substantially steeper decline in density between 1996 and 1997 than models that ignored this effect on trap saturation or treated it as constant ([Table 4](#)). This is consistent with the bias reported by Distiller and Borchers (2015).

### 6.3 | Spatial trend in density

This example demonstrates the use of ipsecr to address bias in estimates of trend in density, the problem identified by Distiller and

**TABLE 4** Population density of brushtail possums in February of consecutive years estimated from models with varying allowance for interference. All models assumed single-catch traps and constant detection parameters ( $\lambda_0$ ,  $\sigma$ ) between years. (95% CI in parentheses).





**FIGURE 3** Simulations of a spatial trend in density fitted using inverse prediction ( $N = 100$ , grey curves; mean dashed blue line) or with mis-specified likelihood for multi-catch traps (mean dotted red line). True trend (solid black curve) was a log-linear function of distance in the east–west direction ( $x$ ). Red ticks and vertical dashed lines mark extent of trapping grid.

Borchers (2015). Data were from simulated sampling of a population with a strong log-linear east–west trend in density. Sampling used an  $8 \times 8$  grid of single-catch traps over 5 occasions. See Appendix 3 for further details.

The average trend fitted by simulation and inverse prediction closely matched the true trend (Figure 3).

## 7 | DISCUSSION

Simulation and inverse prediction is a general method for fitting awkward models involving dependence among detection histories. Models not currently packaged in *ipsecr* may be tackled by providing custom simulation and proxy functions. The present package extends the method to include trap interference and spatial trend in density. The interference parameter  $\lambda_{NT}$  cannot be interpreted simply when there are multiple causes of disturbance, but when causes can be isolated, as with the capture of a single non-target species, the parameter corresponds to the trap-catch index estimated by Linn and Downton (1975).

The method has distinct drawbacks that will limit its use. Foremost is the lack of a criterion for choosing among competing models and for model averaging. This may ultimately be addressed by cross-validation. A secondary limitation is that extension to new models may require coding of both a proxy function and a simulation function. It was straightforward to find suitable proxy variables for the models considered here, but that may not be so easy for other models. I speculate that if there is no accessible summary of the data that correlates to a parameter in the model, then the model itself may be unidentifiable, or nearly so.

My motivation and examples relate to non-independence of detection histories induced by changing availability of detectors. Other mechanisms that lead to detection-level

non-independence, and that may be addressed with simulation and inverse prediction, include post-collection subsampling of hairs for DNA individuation (e.g. Jiménez et al., 2021) and the coordinated behaviour of group-living animals (Bischof et al., 2020). Simulations of post-collection subsampling may be found in Appendix 4.

Another form of non-independence in spatial capture–recapture results when activity centres are clumped or overdispersed, but detection histories are independent when conditioned on each animal's activity centre. Reich and Gardner (2014) developed an algorithm for the case of territorial animals whose activity centres follow a Strauss process. Simulation and inverse prediction might be applied to this scenario given a suitable proxy for spacing.

## ACKNOWLEDGEMENTS

I thank Shirley Pledger, Greg Distiller and two anonymous reviewers for suggestions that significantly improved the paper. I acknowledge the use of New Zealand eScience Infrastructure (NeSI) high performance computing facilities URL <https://www.nesi.org.nz>.

## CONFLICT OF INTEREST STATEMENT

I declare no conflict of interest.

## PEER REVIEW

The peer review history for this article is available at <https://www.webofscience.com/api/gateway/wos/peer-review/10.1111/2041-210X.14088>.

## DATA AVAILABILITY STATEMENT

The *ipsecr* package is available on CRAN (Efford, 2023a) and on GitHub (<https://github.com/MurrayEfford/ipsecr>). It includes the brushtail possum dataset. The current version (*ipsecr* 1.4.0) is archived on Zenodo along with simulation results from Appendix 1 and additional data on brushtail possum trap interference (Efford, 2023b).

## ORCID

Murray G. Efford  <https://orcid.org/0000-0001-5231-5184>

## REFERENCES

- Beyersmann, J., Latouche, A., Buchholz, A., & Schumacher, M. (2009). Simulating competing risks data in survival analysis. *Statistics in Medicine*, 28, 956–971.
- Bischof, R., Dupont, P., Milleret, C., Chipperfield, J., & Royle, J. A. (2020). Consequences of ignoring group association in spatial capture–recapture analysis. *Wildlife Biology*, 2020, 1–10.
- Borchers, D. L., & Efford, M. G. (2008). Spatially explicit maximum likelihood methods for capture–recapture studies. *Biometrics*, 64, 377–385.
- Brown, P. J. (1982). Multivariate calibration. *Journal of the Royal Statistical Society, Series B*, 44, 287–321.
- Calhoun, J. B., & Casby, J. U. (1958). *Calculation of home range and density of small mammals*. Public health monograph. No. 55. U.S. Government Printing Office.

- Carothers, A. D. (1979). The effects of unequal catchability on Jolly-Seber estimates. *Biometrics*, 29, 79–100.
- Distiller, G., & Borchers, D. L. (2015). A spatially explicit capture-recapture estimator for single-catch traps. *Ecology and Evolution*, 5, 5075–5087.
- Eddelbuettel, D., & Francois, R. (2011). Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8), 1–18.
- Efford, M. G. (2004). Density estimation in live-trapping studies. *Oikos*, 106, 598–610.
- Efford, M. G. (2022). secr: Spatially explicit capture-recapture models. R package version 4.5.8. <https://CRAN.R-project.org/package=secur>
- Efford, M. G. (2023a). ipsecur: Spatially explicit capture-recapture using inverse prediction. R package version 1.4.0. <https://CRAN.R-project.org/package=ipsecur>
- Efford, M. G. (2023b). Spatially explicit capture-recapture by inverse prediction. *Zenodo*. <https://doi.org/10.5281/zenodo.7668378>
- Efford, M. G., Borchers, D. L., & 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* (pp. 255–269). Springer.
- Efford, M. G., & Cowan, P. E. (2004). Long-term population trend of *Trichosurus vulpecula* in the Orongorongo Valley, New Zealand. In R. L. Goldingay & S. M. Jackson (Eds.), *The biology of Australian possums and gliders* (pp. 471–483). Surrey Beatty & Sons.
- Efford, M. G., Dawson, D. K., & Robbins, C. S. (2004). DENSITY: Software for analysing capture-recapture data from passive detector arrays. *Animal Biodiversity and Conservation*, 27, 217–228.
- Gerber, B. D., & Parmenter, R. R. (2015). Spatial capture-recapture model performance with known small-mammal densities. *Ecological Applications*, 25, 695–705.
- Groemping, U. (2014). R package FrF2 for creating and analyzing fractional factorial 2-level designs. *Journal of Statistical Software*, 56, 1–56.
- Jiménez, J., Augustine, B. C., Linden, D. W., Chandler, R. B., & Royle, J. A. (2021). Spatial capture-recapture with random thinning for unidentified encounters. *Ecology and Evolution*, 11, 1187–1198.
- Johnson, R., & Wichern, D. (2007). *Applied multivariate statistical analysis* (6th ed.). Prentice-Hall.
- Linn, I. J., & Downton, F. (1975). The analysis of data obtained from small mammal index trappings. *Acta Theriologica*, 20, 319–331.
- Osborne, C. (1991). Statistical calibration: A review. *International Statistical Review*, 59, 309–336.
- Otis, D. L., Burnham, K. P., White, G. C., & Anderson, D. R. (1978). Statistical inference from capture data on closed animal populations. *Wildlife Monographs*, 62, 1–135.
- Pledger, S., & Efford, M. G. (1998). Correction of bias due to heterogeneous capture probability in capture-recapture studies of open populations. *Biometrics*, 54, 888–898.
- R Core Team. (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing. <https://www.R-project.org/>
- Reich, B. J., & Gardner, B. (2014). A spatial capture-recapture model for territorial species. *Environmetrics*, 25, 630–637.
- Romairone, J., Jiménez, J., Luque-Larena, J. J., & Mougeot, F. (2018). Spatial capture-recapture design and modelling for the study of small mammals. *PLoS ONE*, 13, e0198766.
- Royle, J. A., Chandler, R. B., Sollmann, R., & Gardner, B. (2014). *Spatial capture-recapture*. Academic Press.
- Royle, J. A., & Converse, S. J. (2014). Hierarchical spatial capture-recapture models: Modelling population density in stratified populations. *Methods in Ecology and Evolution*, 5, 37–43.

## SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

**Appendix 1.** Simulations with constant density and single-catch traps

**Appendix 2.** Brushtail possum examples

**Appendix 3.** Simulations with density trend and single-catch traps

**Appendix 4.** Simulations of post-collection subsampling

**How to cite this article:** Efford, M. G. (2023). ipsecur: An R package for awkward spatial capture-recapture data. *Methods in Ecology and Evolution*, 00, 1–8. <https://doi.org/10.1111/2041-210X.14088>