




Efficient estimation of large-scale spatial capture–recapture models

DANIEL TUREK ^{1,†}, CYRIL MILLERET,² TORBJØRN ERGON,³ HENRIK BRØSETH ⁴, PIERRE DUPONT ²,
RICHARD BISCHOF,² AND PERRY DE VALPINE⁵

¹*Department of Mathematics and Statistics, Williams College, Williamstown, Massachusetts, USA*

²*Faculty of Environmental Sciences and Natural Resource Management, Norwegian University of Life Sciences, Ås, Norway*

³*Department of Biosciences, Centre for Ecological and Evolutionary Synthesis, University of Oslo, Oslo, Norway*

⁴*Department of Terrestrial Ecology, Norwegian Institute for Nature Research, Trondheim, Norway*

⁵*Department of Environmental Science, Policy & Management, University of California Berkeley, Berkeley, California, USA*

Citation: Turek, D., C. Milleret, T. Ergon, H. Brøseth, P. Dupont, R. Bischof, and P. de Valpine. 2021. Efficient estimation of large-scale spatial capture–recapture models. *Ecosphere* 12(2):e03385. 10.1002/ecs2.3385

Abstract. Capture–recapture methods are a common tool in ecological statistics, which have been extended to spatial capture–recapture models for data accompanied by location information. However, standard formulations of these models can be unwieldy and computationally intractable for large spatial scales, many individuals, and/or activity center movement. We provide a cumulative series of methods that yield dramatic improvements in Markov chain Monte Carlo (MCMC) estimation for two examples. These include removing unnecessary computations, integrating out latent states, vectorizing declarations, and restricting calculations to the locality of individuals. Our approaches leverage the flexibility provided by the nimble R package. In our first example, we demonstrate an improvement in MCMC efficiency (the rate of generating effectively independent posterior samples) by a factor of 100. In our second example, we reduce the computing time required to generate 10,000 posterior samples from 4.5 h down to five minutes, and realize an increase in MCMC efficiency by a factor of 25. These approaches can also be applied generally to other spatially indexed hierarchical models. We provide R code for all examples, an executable web-appendix, and generalized versions of these techniques are made available in the nimbleSCR R package.

Key words: Markov chain Monte Carlo; Mark–recapture; nimble; sampling efficiency; spatial capture–recapture.

Received 25 February 2020; **revised** 20 August 2020; **accepted** 23 October 2020. Corresponding Editor: Cory Merow.

Copyright: © 2021 The Authors. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

† **E-mail:** dbt1@williams.edu

INTRODUCTION

Capture–recapture methods are the primary tools for estimating abundance and demographic parameters in populations of wild animals (Williams et al. 2002). These methods rely on statistical modeling of longitudinal encounter histories of individuals in a population, where repeated observations (individuals seen or not seen) within short (closed) periods provide information about population density and structure, and repeated observations over longer (open) periods provide information about demographic rates such as mortality, recruitment, and maturation.

Spatial capture–recapture (SCR) models accommodate heterogeneity in capture probabilities by modeling individual and trap-specific capture probabilities depending on the individuals' latent centers of activity and space use in relation to the explicit location of traps or other detectors (Efford 2004, Borchers and Efford 2008). Closed SCR models have proved to provide more precise and robust estimates of population densities than non-spatial models (Royle et al. 2014), and also enable estimation of the distribution of individuals within study areas and parameters relating to individuals' spacing behavior (Reich and Gardner 2014, Sutherland et al. 2015, Royle et al. 2016).

Further, by using spatially referenced capture–recapture data obtained from Pollock’s robust sampling design with open SCR models (Gardner et al. 2010), one may model dispersal between primary sampling occasions (Ergon and Gardner 2014, Royle et al. 2018). When dispersal distance distributions are accurately identified in such models, the process of emigration can be disentangled from that of mortality (Ergon and Gardner 2014). By estimating true survival, as opposed to “apparent survival” which confounds mortality and emigration, these SCR models can estimate life-history traits and other population processes in a more mechanistic way than non-spatial models.

Despite their popularity (Royle et al. 2018), SCR models encounter numerous computational challenges which pose serious obstacles for their practical use (Gardner et al. 2018). For large study areas with many detectors, determining the probability of capture history records involves calculations for all detectors, even those extremely distant from any given individual activity, which becomes very costly for large-scale studies (Milleret et al. 2018b). Markov chain Monte Carlo (MCMC) updating of activity center (AC) locations may be done separately for x - and y -coordinates when they could be done jointly. Modeling the movement of ACs often induces inefficient MCMC updating, as do methods for constraining ACs to the valid study area. And data augmentation of never-observed individuals inherently leads to some unnecessary calculations of observation probabilities when individuals are deemed as not being a true member of the population.

Bayesian hierarchical models, such as SCR models, are often formulated using the BUGS modeling language (Lunn et al. 2009) and estimated using MCMC (Brooks et al. 2011). Mainstream MCMC software includes WinBUGS, JAGS (Plummer 2003), and Stan (Stan Development Team 2014). Recently, the nimble R package has been developed, offering new degrees of customization for MCMC (de Valpine et al. 2017). The use of custom-written distributions, and the flexibility of nimble’s MCMC engine, has provided huge gains in areas of ecological statistics (Turek et al. 2016) and the study of MCMC algorithms (Turek et al. 2017). Here, for example, the flexibility introduced by nimble allows us to

work around the issue of calculating likelihood of capture for all traps, providing significant computational savings for fitting SCR models.

In this paper, we use nimble to demonstrate several generally applicable techniques for improving the MCMC sampling efficiency of (1) a simple but computationally intensive SCR model covering a large spatial extent (Milleret et al. 2019), and (2) an open robust-design SCR model with nested sampling occasions of varying length (Ergon and Gardner 2014). Focusing on these examples, we demonstrate techniques to decrease the overall algorithm runtime, while increasing MCMC mixing to improve the accuracy of posterior inferences. These techniques include vectorizing model calculations to reduce model size; implementing model-specific likelihood functions to remove latent states thus reducing the dimensionality of MCMC sampling; disabling unnecessary model calculations for never-detected individuals when using data augmentation; applying joint MCMC sampling to correlated model parameters to improve MCMC mixing; and restricting trap calculations to the locality of each individual. Using these techniques, we realize efficiency gains by a factor of 100 in our first example, and a factor of 25 in our second example, the robust-design SCR model. Similar techniques as those demonstrated would also apply to other spatially indexed hierarchical models.

MATERIALS AND METHODS

We consider two example SCR models which both present computational challenges. The first (“Wolverine”) considers a simple closed SCR model for data from noninvasive genetic sampling of wolverines on a large spatial scale in Norway (*Gulo gulo*, Milleret et al. 2019). The second (“Vole”) is a more complex SCR model on a smaller spatial scale, modeling an open population of field voles with activity center movements (*Microtus agrestis*, Ergon and Gardner 2014). Functions and distributions implementing the techniques described in this section are available in generally applicable forms in the accompanying nimbleSCR R package (Bischof et al. 2020). We first describe each model, followed by the strategies used to improve MCMC efficiency. Finally, we describe the metric used to measure MCMC efficiency.

Wolverine model

This example has a spatial extent over 200,000 km². The data, collected throughout Norway, consist of 453 detections from 196 individually identified female wolverines using non-invasive genetic sampling and search encounter methods (Milleret et al. 2019). The search area was discretized to a detector grid with a 2 km resolution, and only searched grid cells were included in the analysis. This resulted in 17,266 unique detectors, with binary-valued detections of individuals within grid cells. Data and additional details are available at the dryad repository (Milleret et al. 2018a).

The Wolverine model combines a spatial point process model of individual ACs, data augmentation to model the true population size, and an observation model for detection probabilities and capture histories. Define the AC of individual i as $\mathbf{s}_i = (s_i^x, s_i^y)$, where s_i^x and s_i^y follow independent uniform prior distributions spanning the study area. As some regions are unsuitable habitat (i.e., water), AC locations must be constrained. We use a habitat mask by defining a binary matrix H over the study area, where $H_{x,y} = 1$ indicates that cell (x, y) is suitable habitat. Activity center locations are then constrained as $w_i \sim \text{Bernoulli}(H_{s_i^x, s_i^y})$, where each w_i is a data value specified as being equal to one.

For data augmentation (Royle 2009), we add N_{aug} virtual individuals. The augmented matrix \mathbf{y} has dimension $(N_{\text{obs}} + N_{\text{aug}}) \times R$, with $R = 17,266$ detectors and $N_{\text{obs}} = 196$ unique individuals. Define binary variables z_i with independent $z_i \sim \text{Bernoulli}(\phi)$ prior distributions, representing inclusion in the population. For the N_{obs} sighted individuals, $z_i = 1$ is observed data, while the remaining z_i is unobserved. Total population size N is estimated as $N = \sum_{i=1}^{N_{\text{obs}} + N_{\text{aug}}} z_i$ using the prior distribution $\phi \sim \text{Uniform}(0, 1)$ to induce a flat prior on N (Royle et al. 2007).

The probability of detecting individual i at detector r is $p_{i,r} = p_0 \exp(-\|\mathbf{s}_i - \mathbf{x}_r\|^2 / (2\sigma^2))$, where \mathbf{x}_r is the location of detector r , and p_0 and σ are the maximal value and scale of decay for detection probability. Detections are modeled as $y_{i,r} \sim \text{Bernoulli}(p_{i,r} z_i)$. The complete Wolverine model definition is given in Eq. 1, where indices r take the range $1, \dots, R$.

$$\begin{aligned}
 \phi &\sim \text{Uniform}(0,1) \\
 p_0 &\sim \text{Uniform}(0,1) \\
 \sigma &\sim \text{Uniform}(0,50) \\
 N &= \sum_{i=1}^{N_{\text{obs}} + N_{\text{aug}}} z_i \\
 i &= 1, \dots, (N_{\text{obs}} + N_{\text{aug}}): \\
 s_i^x &\sim \text{Uniform}(x_{\min}, x_{\max}) \\
 s_i^y &\sim \text{Uniform}(y_{\min}, y_{\max}) \\
 w_i &\sim \text{Bernoulli}(H_{s_i^x, s_i^y}) \\
 z_i &\sim \text{Bernoulli}(\phi) \\
 \mathbf{s}_i &= (s_i^x, s_i^y) \\
 p_{i,r} &= p_0 \cdot \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{s}_i - \mathbf{x}_r\|^2\right) \\
 y_{i,r} &\sim \text{Bernoulli}(p_{i,r} z_i)
 \end{aligned} \tag{1}$$

We use four refinements of the model and MCMC sampling, with the goal to improve MCMC efficiency: (1) Vectorize computations and put the habitat mask into a custom distribution, (2) jointly sample AC location coordinates, (3) restrict calculations to local detectors and sparse representation of data, and (4) skip unnecessary calculations when $z_i = 0$. We next describe each of these techniques, and nimble code corresponding to each cumulative refinement appears in Appendix S1.

Vectorized computations.—Vectorization refers to carrying out a set of matching model computations more efficiently, as is possible in nimble but neither WinBUGS or JAGS. nimble supports vectorized model declarations, reducing the total nodes in the model and potentially improving MCMC efficiency. We vectorized both detection probabilities and data likelihoods for each individual across the R detectors. For the vector of detection probabilities $\mathbf{p}_{i,1:R}$ we used a vectorized model declaration. For the vectorized data likelihood of $\mathbf{y}_{i,1:R}$ we used a custom likelihood function for the entire (length- R) observation history of one individual.

This technique is only beneficial when the *entire* joint likelihood of $\mathbf{y}_{i,1:R}$ is always calculated simultaneously, as is the case here for updates of p_0 , σ , or z_i . In a different model, this technique could result in inefficiencies if any MCMC updates require likelihood calculation for only a subset of $\mathbf{y}_{i,1:R}$.

Joint sampling of AC locations.—We apply joint (block) sampling of the s_i^x and s_i^y coordinates of each AC. nimble allows the assignment of block samplers to arbitrary variables, applying multi-dimensional Metropolis–Hastings sampling. This results in computational savings since an MCMC update of s_i requires only one calculation of all (length- R) relevant detection probabilities and data likelihoods. In contrast, independent updates of the s_i^x and s_i^y components will require two likelihood evaluations, one for each component.

Local detector evaluations and sparse observation matrix.—We move detection probability calculations inside the vectorized likelihood and additionally restrict these calculations to detectors within a maximum realistic radius (d_{\max}) of the AC s_i . In advance, we identify the set of detectors located within d_{\max} from each cell of the habitat matrix. The modified distribution identifies the grid cell containing s_{i_r} and the set of detectors within d_{\max} from it. Calculations of p_{i_r} are then restricted to this set of detectors.

We also convert to a sparse representation of the detection matrix y . In this representation, each row contains the detector identification numbers (values of r) that detected the corresponding individual. The number of columns is therefore equal to the maximum number of detections of any particular individual. This sparse representation allows for a smaller model and equivalent, but more efficient, likelihood calculations.

Skip unnecessary calculations.—Calculations can be avoided when any $z_i = 0$, that is, an augmented virtual individual is not currently included in the population. In that case, neither the distances to each detector nor the detection probabilities need be calculated. We modify the custom likelihood again, to accept z_i as an argument. When $z_i = 1$, the calculations take place as before. When $z_i = 0$, the likelihood is one if the individual was never observed—always the case for augmented individuals—which can be calculated without any distances or detection probabilities. This modification can save substantial computation, especially when N_{aug} is large, that being the conservative approach.

Vole robust-design model

Our second example considers a robust-design SCR model of field voles in the Kielder Forest of northern England (*M. agrestis*, Ergon and Gardner 2014), with four primary sampling occasions and nested secondary trapping sessions. A total of 158 unique individuals are considered to have static ACs within primary occasions, but to disperse between primary occasions. See Ergon and Gardner (2014, Appendix S2) for further details, (Ergon and Lambin 2013) for the data, and Appendix S2 for the original JAGS code.

The Vole model contains individual survival between primary sampling occasions, dispersal of ACs between primary occasions, and spatial capture–recapture histories. Define the AC of individual i on primary occasion k as $s_{i,k} = (s_{i,k}^x, s_{i,k}^y)$. On first capture, the components s_{i,F_i}^x and s_{i,F_i}^y are given uniform prior distributions spanning the mean location of captures for individual i during that occasion. The dispersal between primary occasions k and $k + 1$ uses a uniformly distributed dispersal angle θ_{ik} and an exponentially distributed dispersal distance d_{ik} with rate parameter λ_{G_i} , where G_i is the sex of individual i (1: female; 2: male), and λ_1 and λ_2 are sex-specific parameters. Thus, the AC components are related across primary occasions as $s_{i,k+1}^x = s_{i,k}^x + d_{ik} \cos(\theta_{ik})$ and $s_{i,k+1}^y = s_{i,k}^y + d_{ik} \sin(\theta_{ik})$.

The survival model uses binary indicator variables, where $z_{i,k} = 1$ indicates individual i is alive on occasion k . We condition on the first observation in primary occasion F_i , as $z_{i,F_i} = 1$. The survival process follows as $z_{i,k+1} \sim \text{Bernoulli}((\phi_{G_i})^{T_k} z_{i,k})$, where survival probability depends on sex and temporal duration. G_i gives the sex of individual i , T_k is the time (in months) between occasions k and $k + 1$, and ϕ_1 and ϕ_2 are sex-specific survival rates. When ϕ_{G_i} is a function of a continuous covariate, the model is only invariant to the choice of time unit of T_k when using a loglog (log-hazard) link (Ergon et al. 2018).

The observation model uses hazard rates to calculate trap capture probabilities specific for each individual, trap, and secondary trapping session. Individual voles can be captured in at most one trap during any trapping session, and we condition on the primary session of the first capture of each individual. For individual i , on

secondary trapping session j of primary occasion k , the capture hazard rate $h_{ijk} = b_{ijk} \cdot \exp(-(\|s_{i,k} - x_r\|/\sigma_{G_i})^{\kappa_{G_i}})$, where the location of trap r is x_r , and each κ_j and σ_j are sex-specific observation parameters. Baseline hazard is $b_{ijk} = \lambda_0(\beta_1)^{I(\text{TOD}_{jk}=2)}(\beta_2)^{I(G_i=2)}$, using indicator function $I(\cdot)$, time of day TOD_{jk} (1: evening; 2: morning), and baseline hazard rate λ_0 . β_1 is the effect of morning trapping sessions, and β_2 is that of males.

Total capture hazard rate is $h_{ijk*} = \sum_{r'=1}^R h_{ijk'r'}$. Probability of “no capture” is $\pi_{ijk0} = \exp(-h_{ijk*}z_{i,k})$, which is unity when $z_{i,k} = 0$. Probability of capture is $\pi_{ijk} = (1 - \pi_{ijk0})h_{ijk}/h_{ijk*}$ in trap r , accounting for competing risks among traps and satisfying $\sum_{r=0}^R \pi_{ijk} = 1$.

We again use data values $w_{ijk} = 1$ along with a Bernoulli distribution to induce likelihood calculations. Observation data y is a 3-dimensional array, where $y_{ijk} = 0$ indicates that individual i was not captured in trapping session j of primary occasion k , and $y_{ijk} = r$ indicates a capture in trap r . The complete Vole model definition is given in Eq. 2, where all indices j take the range of the number of secondary trapping sessions in the relevant primary occasion k , and all indices r assume the range $1, \dots, R$.

$$\begin{aligned}
 &\beta_1, \beta_2 \sim \text{Uniform}(0.1, 10) \\
 &p \sim \text{Uniform}(0.01, 0.99) \\
 &\lambda_0 = -\log(1 - p) \\
 &g = 1, 2: \\
 &\quad \kappa_g \sim \text{Uniform}(0, 50) \\
 &\quad \sigma_g \sim \text{Uniform}(0.1, 20) \\
 &\quad \lambda_g \sim \text{Uniform}(0, 100) \\
 &\quad \phi_g \sim \text{Uniform}(0, 1) \\
 &i = 1, \dots, N_{\text{obs}}: \\
 &\quad s_{i,F_i}^x \sim \text{Uniform}(x_{\min}^i, x_{\max}^i) \\
 &\quad s_{i,F_i}^y \sim \text{Uniform}(y_{\min}^i, y_{\max}^i) \\
 &\quad z_{i,F_i} = 1 \\
 &k = F_i, \dots, L - 1: \\
 &\quad \theta_{ik} \sim \text{Uniform}(0, 2\pi) \\
 &\quad d_{ik} \sim \text{Exponential}(\lambda_{G_i}) \\
 &\quad s_{i,k+1}^x = s_{i,k}^x + d_{ik} \cos(\theta_{ik}) \\
 &\quad s_{i,k+1}^y = s_{i,k}^y + d_{ik} \sin(\theta_{ik}) \\
 &\quad z_{i,k+1} \sim \text{Bernoulli}\left(\left(\phi_{G_i}\right)^{T_k} z_{i,k}\right)
 \end{aligned} \tag{2}$$

$k = F_i, \dots, L:$

$$\begin{aligned}
 &s_{ik} = \left(s_{i,k}^x, s_{i,k}^y\right) \\
 &b_{ijk} = \lambda_0(\beta_1)^{I(\text{TOD}_{jk}=2)}(\beta_2)^{I(G_i=2)} \\
 &h_{ijk} = b_{ijk} \cdot \exp\left(-\left(\frac{\|s_{i,k} - x_r\|}{\sigma_{G_i}}\right)^{\kappa_{G_i}}\right) \\
 &h_{ijk*} = \sum_{r'=1}^R h_{ijk'r'} \\
 &\pi_{ijk0} = \exp(-h_{ijk*}z_{i,k}) \\
 &\pi_{ijk} = (1 - \pi_{ijk0}) \frac{h_{ijk}}{h_{ijk*}} \\
 &w_{ijk} \sim \text{Bernoulli}\left(\pi_{ijk} y_{ijk}\right)
 \end{aligned}$$

We apply three cumulative refinements to the model and MCMC sampling: (1) Jointly sample correlated dimensions and marginalize over z_i indicator variables, (2) use a custom bivariate dispersal distribution, and (3) restrict trap calculation to the vicinity of each AC. Next we describe these techniques, and nimble code corresponding to each appears in Appendix S2.

Joint sampling and marginalization.—We apply joint samplers for updating two pairs of parameters: $\{\kappa_1, \sigma_1\}$ and $\{\kappa_2, \sigma_2\}$, as these pairs each determine the trap hazard rates for one sex. Trial runs confirm that these pairs exhibit high posterior correlation, so we expect block samplers will improve mixing.

Next, we integrate (marginalize) over the latent $z_{i,k}$ indicator variables to directly calculate the unconditional likelihood of capture histories. This reduces the model size and the dimension of sampling, and can improve MCMC mixing since parameter updates are no longer conditional on the “current” values of each $z_{i,k}$. This is done in nimble using a custom likelihood. This calculation is a finite summation over the possible $z_{i,k}$ states, similar to the filtering employed in Turek et al. (2016: Section 2.3.2). When individuals are known to be alive (up to the final capture), the likelihood is survival multiplied by the probability of the observed capture history. Subsequent to the final capture, forward filtering is used to calculate the likelihood of the remaining non-capture events, accounting for uncertainty in survival.

Custom dispersal distribution.—We originally modeled dispersal distances and angles as random variables subject to MCMC sampling, a

standard approach for movement models. This results in high computational cost because any proposed update to dispersal distance or angle (especially for *early* primary occasions) results in a large chain of calculations to determine the updated ACs, detection probabilities, and detection likelihoods for all subsequent occasions. Specifically, say we make an MCMC proposal modifying d_{11} , the dispersal distance for the first ($i = 1$) individual, between the first and second primary occasions. This MCMC update will require evaluating the entire chain of deterministic AC location calculations, for the second ($k = 2$) occasion ($s_{1,2}^x = s_{1,1}^x + d_{11}\cos\theta_{11}$ and $s_{1,2}^y = s_{1,1}^y + d_{11}\sin\theta_{11}$), then to update the AC locations in the third ($k = 3$) primary occasion ($s_{1,3}^x = s_{1,2}^x + d_{12}\cos\theta_{12}$ and $s_{1,3}^y = s_{1,2}^y + d_{12}\sin\theta_{12}$), then to update the AC locations in the fourth ($k = 4$) occasion, and so on. In total, parameter updates under this formulation will require re-evaluating each $s_{1,2}, s_{1,3}, \dots, s_{1,L}$, up through the AC of the final primary occasion. Furthermore, the deterministic calculations of all trap hazard rates (h_{1jkr}) and detection probabilities (π_{1jkr}) corresponding to the first individual must all be recalculated, as well as the corresponding data likelihoods (w_{1jk}). For large values of L , this will result in significant required computation.

We reparameterize this model using a custom distribution of AC locations that is induced by the distributions of turning angle and distance, as $s_{i,k+1} \sim \text{Dispersal}(s_{ik}, \lambda_{G_i})$. This distribution is centered around the current AC and is mathematically equivalent to the original $\{d, \theta\}$ parameterization. Now, updates of $s_{i,k}$ do *not* induce a large chain of ensuing calculations, but rather, only the likelihoods corresponding to $s_{i,k}$ and $s_{i,k+1}$ must be calculated. The custom distribution is given by $p(s_{i,k+1}|s_{ik}, \lambda_{G_i}) \propto (1/d) \cdot \lambda_{G_i} e^{-\lambda_{G_i} d}$, where $d = \|s_{i,k+1} - s_{ik}\|$, and omitting constants of proportionality which are not necessary for sampling. We recognize $\lambda_{G_i} e^{-\lambda_{G_i} d}$ as the exponential density for the dispersal distance d . The factor of $(1/d)$ results from the Jacobian term in the change-of-variables between polar and Cartesian coordinates. From an implementation standpoint, these density calculations take place on a logarithmic scale. This technique can be similarly applied when using other distributions for dispersal distance, by substituting in the density of the alternate dispersal distribution.

Local trap calculations.—During MCMC sampling, the capture hazard rate h_{ijk} and associated likelihood terms are calculated for all traps, regardless of an individual's current AC. This is inefficient, since when $s_{i,k}$ is “far” from a trap r , then h_{ijk} will be extremely low, and a capture in trap r is exceedingly unlikely. Its contribution to $h_{ijk*} = \sum_{r=1}^R h_{ijk*r}$ is negligible, as is the probability of capture in trap r . The original BUGS modeling language lacks the ability to conditionally disable calculations, and hence all the capture hazard rates must always be computed.

We introduce logic such that h_{ijk} is only calculated for traps within a distance d_{\max} of the individual's AC. For traps located further, we assign h_{ijk} a small positive value. This will not affect the sum h_{ijk*} , but still allows for a nonzero probability of capture. Here, we let $h_{ijk} = 10^{-14}$ for traps outside a radius $d_{\max} = 40$ from each individual's AC.

We introduce a discretized grid over the study area, and pre-compute indices of traps within a radius d_{\max} from each grid cell. Using this, a custom nimble function returns the indices of “local traps” nearest to any $s_{i,k}$ and subsequently calculates hazard rates only for the “local” traps. This is similar to the local trap calculations used in the previous example, but implementations are different on account of the discretized habitat mask used there.

MCMC efficiency

We define MCMC efficiency as the number of effectively independent posterior samples produced per second of MCMC runtime (excluding upfront time of model building and compilation). Distinct model parameters will typically mix at different rates, thus having distinct posterior effective sample sizes (ESS), and therefore a distinct measure of efficiency. In addition to presenting the MCMC runtimes and MCMC efficiency of all model parameters, we also summarize performance using the minimum and mean efficiencies among all model parameters. This definition of efficiency captures the tradeoff between quality of mixing and computational speed. Some algorithms may mix slowly (producing a low ESS) but execute sufficiently fast that they achieve high efficiency. Other algorithms may mix quickly (producing a high ESS)

but require significantly longer execution time and thus achieve low efficiency.

RESULTS

Here we describe the performance resulting from each formulation or sampling strategy of the Wolverine and Vole example models. All algorithm runtimes, ESS estimates, and MCMC efficiencies reflect independent chains of 10,000 posterior samples. We do not present the posterior inferences (e.g., posterior mean and median), as they are qualitatively identical to the original published analyses.

Wolverine model

We assess performance of the Wolverine model using total population size (N), probability of detection (p_0), and scale factor (σ). Results for the four stages of iterative improvement described in *Materials and Methods: Wolverine model* will be denoted as Nimble1 (vectorization), Nimble2 (joint sampling), Nimble3 (evaluating local detectors), and Nimble4 (skipping unnecessary calculations).

As in Milleret et al. (2019), the JAGS model was unable to complete, crashing after 30 d. Transitioning to nimble considerably reduced memory usage and runtime, as we fit the Nimble1 model in 26 h. Effective sample size values were in the range of 100–200 for all parameters, indicating high posterior autocorrelation. In combination with the long runtime, this produced MCMC efficiencies on the order of 10^{-3} for all parameters. The addition of joint sampling in the Nimble2 version decreased runtime to 20 h. Parameter ESS values were similar to the Nimble1 model, giving a small improvement in efficiency.

We observed major improvement in the Nimble3 version, using the local trap evaluations and a sparse representation of the observation matrix. Markov chain Monte Carlo runtime reduced to 30 min, by a factor of 40 relative to the Nimble2 model. As we expected, ESS values were unchanged, and the resulting MCMC efficiencies were in the range of 0.1–0.3 (Fig. 1).

The Nimble4 version, disabling unnecessary model calculations, reduced MCMC runtime by an additional factor of 2, down to 16 min. Accordingly, MCMC efficiencies increased by nearly a factor of 2. Relative to the initial Nimble1

formulation, we have achieved increases in both the minimum and mean efficiencies of 100-fold. Concretely, while it was not even possible to fit the original version of this model using JAGS, the initial Nimble1 formulation would require 3.5 d to generate 1000 ESS for all parameters, and the final Nimble4 model can accomplish the same in 51 min.

Fig. 2 presents the minimum and mean efficiencies across all model parameters for each formulation of the Wolverine model, and all results for the Wolverine example appear in Table 1. An executable version of the Nimble4 Wolverine model is available at the web-appendix http://nimble-dev.github.io/nimbleSCR/wolverine_example.html.

Vole robust-design model

The Vole model contains a total of 11 hyperparameters, which we use to assess MCMC efficiency. Results for the three stages of iterative improvement described in *Materials and Methods: Vole robust-design model* will be denoted as Nimble1 (marginalization), Nimble2 (customizing dispersal distribution), and Nimble3 (evaluating local detectors). The original formulation of the model, running in JAGS, required over 4.5 h to generate 10,000 posterior samples, and resulted in a minimum MCMC efficiency of 0.002, and a mean efficiency of 0.04.

The Nimble1 version introduced joint sampling of correlated parameters, and a custom likelihood to remove the z_{ij} latent states. This reduced the total model size from 4460 nodes down to 3562, while the number of unobserved nodes undergoing MCMC sampling was reduced from 1437 down to 1067. This model yielded an MCMC runtime of 15 min. Effective sample size values were higher than those of JAGS, particularly for the jointly sampled σ_i and κ_i parameters. Markov chain Monte Carlo efficiency was therefore higher for all parameters (Fig. 3), while the average efficiency increased by a factor of 7.5 relative to JAGS.

The Nimble2 model introduced a custom bivariate dispersal distribution for individual ACs. This reduced the total model size from 3562 nodes to 2452, and the number of nodes for MCMC sampling from 1067 to 697. Runtime decreased by a factor of 2, to seven minutes, and all parameter MCMC efficiencies increased.

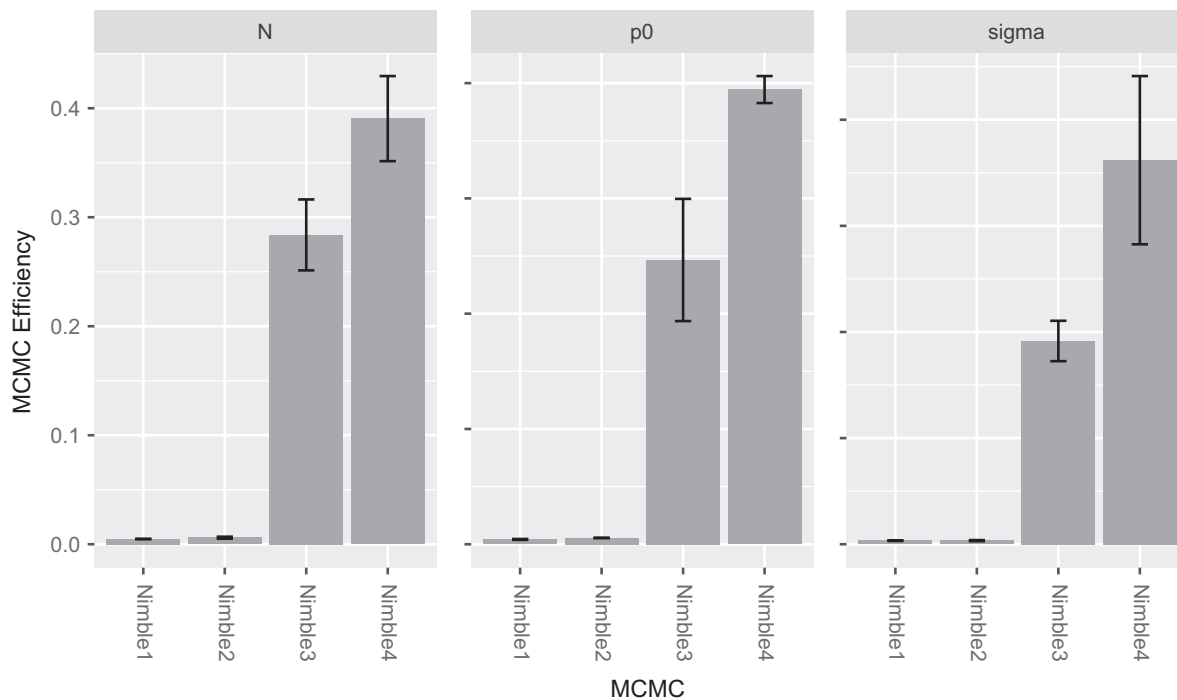


Fig. 1. Markov chain Monte Carlo (MCMC) efficiency for the Wolverine example, for all parameters and model formulations. Efficiency values are averaged over three independent chains, error bars showing standard deviation.

Using local trap calculations in the Nimble3 model reduced MCMC runtime further, to five minutes. Overall, relative to the initial analysis appearing in Ergon and Gardner (2014), these strategies reduced MCMC runtime by more than a factor of 50 and increased both minimum and mean MCMC efficiencies 25-fold. Concretely, the original model fitted in JAGS would require over seven days to produce 1000 ESS for all parameters, whereas the Nimble3 formulation requires less than 6 h to accomplish the same.

Fig. 4 presents the minimum and mean efficiencies across all model parameters for each formulation of the Vole model, and all results for the Vole example appear in Table 2.

DISCUSSION

Spatial capture–recapture models are now commonplace given the abundance of geolocated ecological data, but remain challenging from the perspective of model fitting. Indeed, large numbers of individuals or expansive study areas can

render some problems intractable, without employing custom approaches. No less, incorporating more complex study designs (e.g., individual heterogeneity, nested sampling occasions, non-uniform spatial structure, or environmental covariates; Dorazio and Andrew Royle 2003; Karanth et al. 2006; Gardner et al. 2009) only exacerbates an already difficult problem.

The techniques demonstrated here produce posterior results identical (within Monte Carlo error) to the original versions, with the exception of the local trap evaluations. This technique attributed a small trap hazard rate for traps (or probability of detection, for detectors) situated outside a radius d_{\max} from each individual AC. The choice of d_{\max} is important: Large values will produce identical inferences but offer no computational gain, while small values offer a large computational gain, but may introduce bias. In the Voles example, we selected $d_{\max} = 40$ as a compromise which did not noticeably affect the resulting inferences, while also providing useful gains in computational efficiency. The choice of

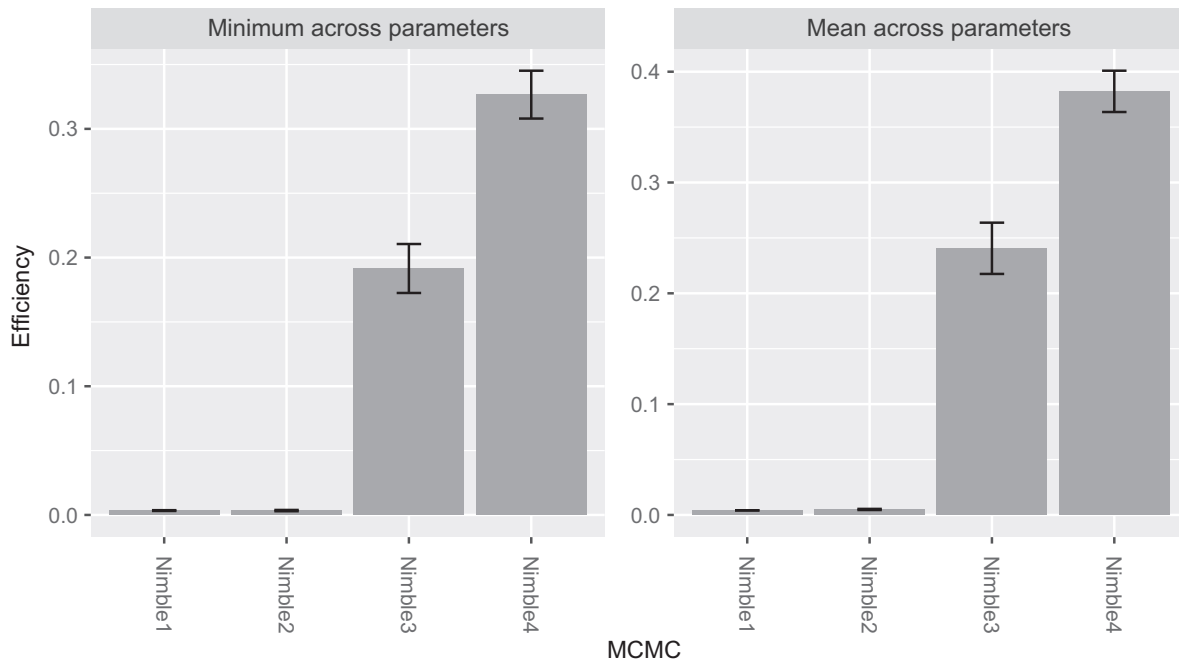


Fig. 2. Minimum and mean Markov chain Monte Carlo (MCMC) efficiency among the three model parameters for the Wolverine example. Values are averaged over three independent chains, error bars showing standard deviation.

Table 1. Markov chain Monte Carlo efficiency values for the Wolverine example, for all parameters and model formulations.

| Model | Parameter | | |
|---------|-----------|-------|----------|
| | N | p_0 | σ |
| Nimble1 | 0.005 | 0.004 | 0.003 |
| Nimble2 | 0.006 | 0.005 | 0.003 |
| Nimble3 | 0.284 | 0.247 | 0.192 |
| Nimble4 | 0.390 | 0.394 | 0.362 |

Note: Results are averaged over three independent chains.

d_{max} is subjective, and in general would involve consideration of detector spacing, individual home range characteristics, and model scale parameters governing probability of detection or trap hazard rates. The process of selecting d_{max} should involve expert opinion (or trial runs) to determine an appropriate value. In practice, smaller values of d_{max} may be used for exploratory analyses, but a conservative higher value should be used to minimize any biases in the final inference.

We are aware that conditioning on the primary occasion of first capture, as in the Vole example, has the potential to induce bias into parameter estimates (Efford and Schofield 2019, Appendix E). Simulations presented in Ergon and Gardner (2014) suggest minimal bias in mortality estimates, although the scale parameter in the observation model may be inflated. Thus, care should be taken when applying this model to data from other systems. Alternatively, a full spatial robust-design model (Bischof et al. 2016, Jiménez et al. 2018, Gardner et al. 2018), including recruitment and occasion-specific population size, may be fitted, in which case no such bias is expected (Efford and Schofield 2019). That said, our purpose has been to investigate efficiency of estimation methods rather than statistical properties (such as bias or goodness of fit) of particular models. Indeed, the ability to perform inference more efficiently will support a deeper exploration of alternative models and model validation steps.

Many software packages are available for fitting SCR models, making these analyses faster and more accessible to practitioners (e.g., secr or

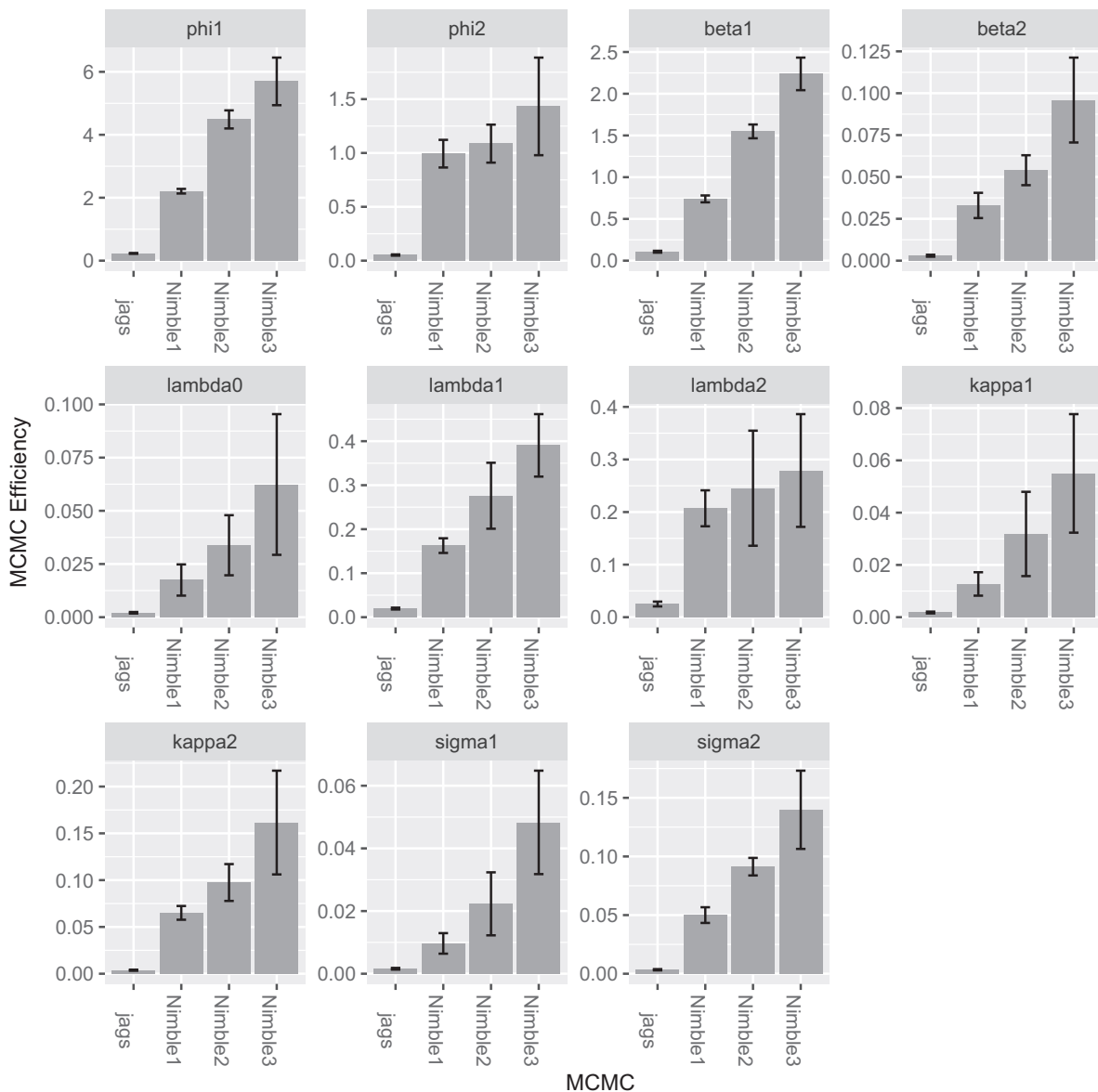


Fig. 3. Markov chain Monte Carlo (MCMC) efficiency for the Vole example, for all parameters and model formulations. Efficiency values are averaged over five independent chains, error bars showing standard deviation.

oSCR). The prevalence of specialized software underscores the complex nature of SCR problems, and furthermore that no single software package could be general enough to approach all SCR problems. nimble does not attempt to provide “canned” algorithms for SCR models, or any other particular application, but rather a flexible programming environment suitable for customized (and highly efficient) analysis of complex data.

We have made use of the nimble software package for R, to demonstrate techniques for improving the performance of SCR model fitting using MCMC. The techniques demonstrated are not exhaustive, but rather suggest the potential performance gains made possible using nimble, where we observed between one and two orders of magnitude improvement. These approaches can provide significant computational gain, permitting large-scale spatial and temporal analyses

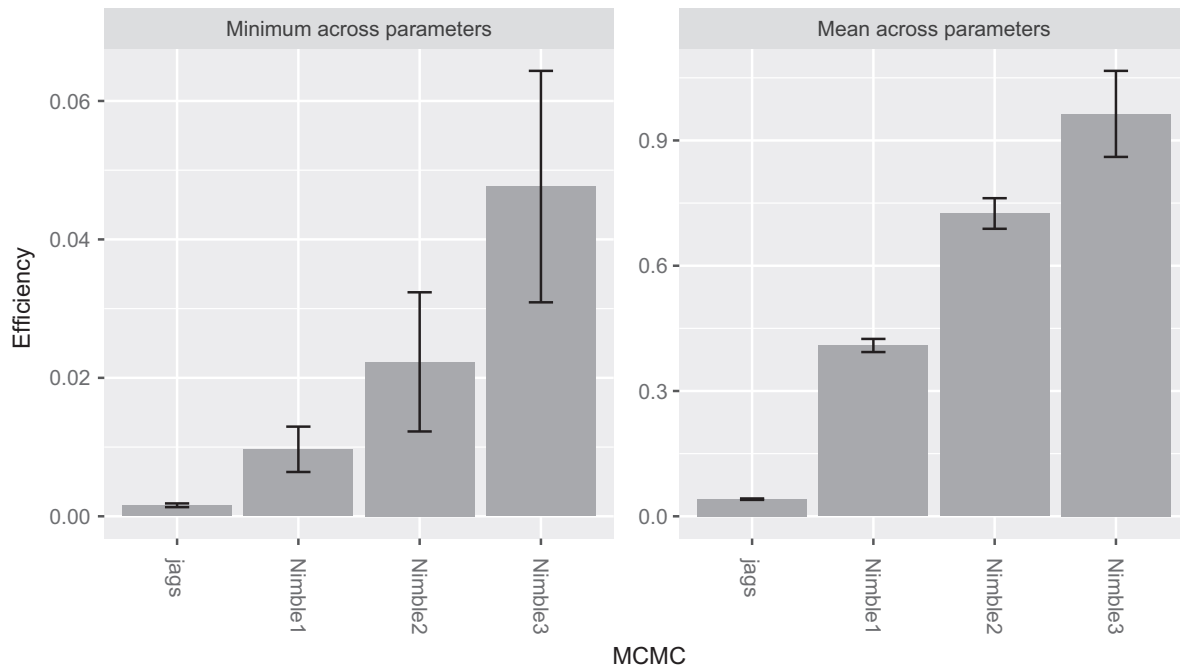


Fig. 4. Minimum and mean Markov chain Monte Carlo (MCMC) efficiency among the eleven model parameters for the Vole example. Values are averaged over five independent chains, error bars showing standard deviation.

Table 2. Markov chain Monte Carlo efficiency values for the Vole example, for all parameters and model formulations.

| Model | Parameter | | | | | | | | | | |
|---------|-----------|----------|-----------|-----------|-------------|-------------|-------------|------------|------------|------------|------------|
| | ϕ_1 | ϕ_2 | β_1 | β_2 | λ_0 | λ_1 | λ_2 | κ_1 | κ_2 | σ_1 | σ_2 |
| JAGS | 0.23 | 0.05 | 0.11 | 0.003 | 0.002 | 0.02 | 0.03 | 0.002 | 0.004 | 0.002 | 0.003 |
| Nimble1 | 2.21 | 0.99 | 0.74 | 0.033 | 0.017 | 0.16 | 0.21 | 0.013 | 0.065 | 0.010 | 0.050 |
| Nimble2 | 4.49 | 1.09 | 1.55 | 0.054 | 0.034 | 0.28 | 0.25 | 0.032 | 0.097 | 0.022 | 0.091 |
| Nimble3 | 5.70 | 1.43 | 2.24 | 0.096 | 0.062 | 0.39 | 0.28 | 0.055 | 0.162 | 0.048 | 0.140 |

Note: Results are averaged over five independent chains.

to support major conservation and management decisions, and the ability to fit increasingly complex models to large datasets. More broadly, similar techniques are also applicable to the analysis of general spatially indexed hierarchical model structures.

ACKNOWLEDGMENTS

This work was partly funded by the Norwegian Environment Agency (Miljødirektoratet), the Swedish Environmental Protection Agency (Naturvårdsverket), and the Research Council of Norway (NFR 286886).

LITERATURE CITED

Bischof, R., H. Brøseth, and O. Gimenez. 2016. Wildlife in a politically divided world: Insularism inflates estimates of brown bear abundance. *Conservation Letters* 9:122–130.

Bischof, R., D. Turek, C. Milleret, T. Ergon, P. Dupont, and P. de Valpine. 2020. Spatial capture-recapture (SCR) methods using nimble. R package version 0.1.0. <https://cran.r-project.org/web/packages/nimbleSCR/index.html>

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

- Brooks, S., A. Gelman, G. Jones, and X.-L. Meng. 2011. Handbook of Markov Chain Monte Carlo. CRC Press, Boca Raton, FL.
- de Valpine, P., D. Turek, C. J. Paciorek, C. Anderson-Bergman, D.-C. Lang, and R. Bodik. 2017. Programming with models: writing statistical algorithms for general model structures with NIMBLE. *Journal of Computational and Graphical Statistics* 26:403–413.
- Dorazio, R. M., and J. Andrew Royle. 2003. Mixture models for estimating the size of a closed population when capture rates vary among individuals. *Biometrics* 59:351–364.
- Efford, M. 2004. Density estimation in live-trapping studies. *Oikos* 106:598–610.
- Efford, M. G., and M. R. Schofield. 2019. A spatial open-population capture-recapture model. *Biometrics* 76:392–402.
- Ergon, T., Ø. Borgan, C. R. Nater, and Y. Vindenes. 2018. The utility of mortality hazard rates in population analyses. *Methods in Ecology and Evolution* 9:2046–2056.
- Ergon, T., and B. Gardner. 2014. Separating mortality and emigration: modelling space use, dispersal and survival with robust-design spatial capture-recapture data. *Methods in Ecology and Evolution* 5:1327–1336.
- Ergon, T., and X. Lambin. 2013. Data from: separating mortality and emigration: modelling space use, dispersal and survival with robust-design spatial-capture-recapture data. Tech. Rep. Dryad Digital Repository. <https://doi.org/10.5061/dryad.r17n5>
- Gardner, B., J. Andrew Royle, and M. T. Wegan. 2009. Hierarchical models for estimating density from DNA mark-recapture studies. *Ecology* 90:1106–1115.
- Gardner, B., N. Rahel Sollmann, S. Kumar, D. Jathanna, and K. Ullas Karanth. 2018. State space and movement specification in open population spatial capture-recapture models. *Ecology and Evolution* 8:10336–10344.
- Gardner, B., J. Reppucci, M. Lucherini, and J. A. Royle. 2010. Spatially explicit inference for open populations: estimating demographic parameters from camera-trap studies. *Ecology* 91:3376–3383.
- Jiménez, J., J. M. Hernández, J. Feliú, M. Carrasco, and R. Moreno-Opo. 2018. Breeding in a dry wetland. Demographic response to drought in the common reed-warbler *Acrocephalus scirpaceus*. *Ardeola* 65:247–259.
- Karanth, K. U., J. D. Nichols, N. Samba Kumar, and J. E. Hines. 2006. Assessing tiger population dynamics using photographic capture-recapture sampling. *Ecology* 87:2925–2937.
- Lunn, D., D. Spiegelhalter, A. Thomas, and N. Best. 2009. The BUGS project: evolution, critique and future directions. *Statistics in Medicine* 28:3049–3067.
- Milleret, C., P. Dupont, C. Bonenfant, H. Brøseth, Ø. Flagstad, C. Sutherland, and R. Bischof. 2019. A local evaluation of the individual state-space to scale up Bayesian spatial capture-recapture. *Ecology and Evolution* 9:352–363.
- Milleret, C., P. Dupont, C. Bonenfant, H. Brøseth, Ø. Flagstad, C. Sutherland, and R. Bischof. 2018a. Data from: a local evaluation of the individual state-space to scale up Bayesian spatial capture-recapture. <https://doi.org/10.5061/dryad.42m96c8>
- Milleret, C., P. Dupont, H. Brøseth, J. Kindberg, J. A. Royle, and R. Bischof. 2018b. Using partial aggregation in spatial capture-recapture. *Methods in Ecology and Evolution* 9:1896–1907.
- Plummer, M. 2003. JAGS: a program for analysis of Bayesian graphical models using Gibbs sampling. Pages 1–10 in *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*. Volume 124. Vienna, Austria.
- Reich, B. J., and B. Gardner. 2014. A spatial capture-recapture model for territorial species: a spatial capture-recapture model for territorial species. *Environmetrics* 25:630–637.
- Royle, J. A. 2009. Analysis of capture-recapture models with individual covariates using data augmentation. *Biometrics* 65:267–274.
- Royle, J. A., R. B. Chandler, R. Sollmann, and B. Gardner. 2014. Spatial capture-recapture. Academic Press.
- Royle, J. A., R. M. Dorazio, and W. A. Link. 2007. Analysis of multinomial models with unknown index using data augmentation. *Journal of Computational and Graphical Statistics* 16:67–85.
- Royle, J. A., A. K. Fuller, and C. Sutherland. 2016. Spatial capture-recapture models allowing Markovian transience or dispersal. *Population Ecology* 58:53–62.
- Royle, J. A., A. K. Fuller, and C. Sutherland. 2018. Unifying population and landscape ecology with spatial capture-recapture. *Ecography* 41:444–456.
- Stan Development Team. 2014. Stan: a C++ library for probability and sampling, version 2.5.0. <https://mc-stan.org>
- Sutherland, C., A. K. Fuller, J. Andrew Royle, and O. Gimenez. 2015. Modelling non-Euclidean movement and landscape connectivity in highly structured ecological networks. *Methods in Ecology and Evolution* 6:169–177.
- Turek, D., P. de Valpine, and C. J. Paciorek. 2016. Efficient Markov chain Monte Carlo sampling for

- hierarchical hidden Markov models. *Environmental and Ecological Statistics* 23:549–564.
- Turek, D., P. de Valpine, C. J. Paciorek, and C. Anderson-Bergman. 2017. Automated parameter blocking for efficient Markov chain Monte Carlo sampling. *Bayesian Analysis* 12:465–490.
- Williams, B. K., J. D. Nichols, and M. J. Conroy. 2002. *Analysis and management of animal populations*. Academic Press, San Diego, CA.

SUPPORTING INFORMATION

Additional Supporting Information may be found online at: <http://onlinelibrary.wiley.com/doi/10.1002/ecs2.3385/full>