APPENDIX A. Bias of population size estimated from model M_0 as function of CV(p)

Cormack (1966) related the bias of a 2-sample closed population estimate to the coefficient of variation of individual capture probability CV(p). Here we show by simulation that CV(p) is also a good predictor of relative bias when the number of samples is more than 2. R code is provided in the Supplement; variation in CV(p) is generated by varying the components of a 2-class finite mixture. The 'closedN' function from **secr** (Efford 2013) is used to compute null model estimates (model M_0 of Otis et al. (1978)). Examples were simulated with 3 and 7 sampling occasions, $N = 200$, and p_i in range 0.05–0.4; results were very similar (Fig. A1).

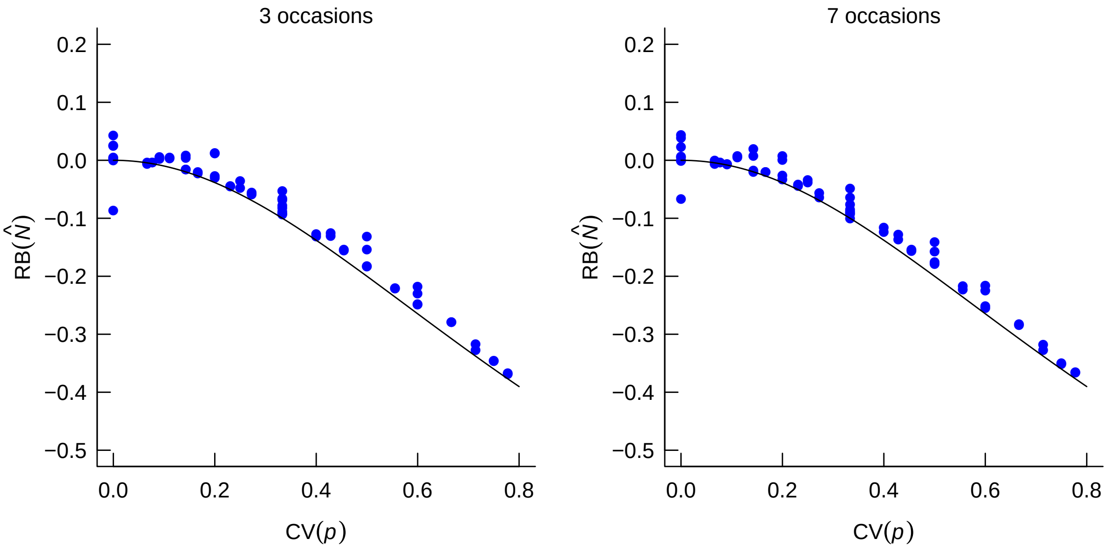

Figure A1. Relative bias of estimated population size $\mathrm{RB}(\hat{N})$ from null closed population model applied to simulated data from populations with varying capture probability p. Averages across 1000 simulations plotted in relation to CV(p). Reference line indicates $1/(1 + \mathrm{CV}(p)^2) - 1$.

Literature cited

Cormack, R. M. 1966. A test for equal catchability. Biometrics 22:330–342.

Efford, M. G. 2013. **secr**: Spatially explicit capture-recapture models. R package version 2.6.1. http://CRAN.R-project.org/package=secr [accessed 20 July 2013]

Otis, D. L., K. P. Burnham, G. C. White, and D. R. Anderson 1978. Statistical inference from capture data on closed animal populations. Wildlife Monographs 62:1–135.

APPENDIX B. Simulations of SECR with covariation of detection parameters

Two-class mixture models are a convenient way to generate heterogeneous populations differing in variance and skewness (Carothers 1973). In our simulations, the classes may be viewed as corresponding to males and females. Fixing the proportion in each class at 0.5 also fixed the skewness for given variance.

R code for the simulations is provided in the Supplement. The code relies on simulation and estimation procedures in the R package 'secr' (Efford 2013).

Scenarios are described in the main text. Four hundred random datasets were generated for each scenario. Datasets were generated with the function 'sim.capthist' (using nominal grid spacing $s = 25m$) and analyzed with function 'secr.fit' in **secr** 2.6.1.

The performance criterion used was relative bias defined as $\mathrm{RB}(\hat{D}) = (\hat{D} - D)/D$ where D was the known density. The relationship between $\mathrm{RB}(\hat{D})$ and various predictors was assessed by fitting a quadratic regression model; fit was expressed as adjusted R^2.

Literature cited

Carothers, A. D. 1973. The effects of unequal catchability on Jolly-Seber estimates. Biometrics 29:79–100.

Efford, M. G. 2013. **secr**: Spatially explicit capture-recapture models. R package version 2.6.1. http://CRAN.R-project.org/package=secr [accessed 20 July 2013]

APPENDIX C. Compensatory detection inferred from SECR 2-class finite mixture models applied to various datasets

Here we provide further documentation for the analysis of compensatory heterogeneity in the example datasets provided in version 2.6.1 of the R package ‘secr’ (Efford 2013). The taxa in these examples (Table C1) span a range that complements the large-carnivore focus of the main text. See Efford (2013) and the original sources for descriptions of these studies. We omit the stoat DNA dataset in Efford (2013) because it is too small to fit a mixture model.

We fitted two models to each dataset: a null model ($\lambda_0 \sim 1, \sigma \sim 1$) and a 2-class finite mixture model with variation in both detection parameters ($\lambda_0 \sim \text{h2}, \sigma \sim \text{h2}$). Individuals in a particular latent class shared the same combination of values for λ_0 and σ. Models were fitted by maximizing the full likelihood, with other settings as appropriate (see R code at end).

The mixture model enables us to quantify variation in the components of detection and determine the direction of any correlation. The single-detector sampling area ($a_0 = 2\pi\lambda_0\sigma^2$) was computed from the estimated detection parameters of each latent class (the class-specific a_0 may also be estimated directly by fitting a mixture model $a_0 \sim \text{h2}, \sigma \sim \text{h2}$). The CV of each detection parameter in the finite mixture model was computed using Eq. 2 of the main text. The AIC weight of the mixture model relative to the null model was computed following Burnham and Anderson (2002); weights exceeding 0.5 indicate a preference for the mixture model, and weights approaching 1.0 indicate strong relative support for that model. The mixture model was strongly supported for all datasets except ovenbird and possibly spotted skink (Table C2).

Literature cited

Burnham, K. P., and D. R. Anderson. 2002. Model selection and multimodel inference: a practical information-theoretic approach, 2nd edn. Springer, New York, USA.

Dawson, D. K., and M. G. Efford, M. G. 2009. Bird population density estimated from acoustic signals. Journal of Applied Ecology 46:1201–1209.

Efford, M. G. 2013. **secr**: Spatially explicit capture-recapture models. R package version 2.6.1. http://CRAN.R-project.org/package=secr

Otis, D. L., K. P. Burnham, G. C. White, and D. R. Anderson. 1978. Statistical inference from capture data on closed animal populations. Wildlife Monographs No. 62:1–135.

Royle, J. A., and K. V. Young. 2008. A hierarchical model for spatial capture–recapture data. Ecology 89:2281–2289.

Table C1. Example datasets from R package ‘secr’.

Dataset	Species	Field method	Note	Source
Flat-tailed horned lizard	*Phrynosoma macalli*	Area search		Royle and Young (2008)
Deer mouse ESG	*Peromyscus maniculatus*	Live trapping	East Stuart Gulch	Otis et al. (1978)
Deer mouse WSG	*Peromyscus maniculatus*	Live trapping	Wet Swizer Gulch	Otis et al. (1978)
House mouse	*Mus musculus*	Live trapping	mornings only	Otis et al. (1978)
Speckled skink	*Oligosoma infrapunctatum*	Pitfall trapping	two sessions	Efford et al. unpubl.
Spotted skink	*Oligosoma lineoocellatum*	Pitfall trapping	two sessions	Efford et al. unpubl.
Brushtail possum	*Trichosurus vulpecula*	Live trapping	includes retag errors	Efford et al. (2005)
Ovenbird	*Seiurus aurocapilla*	Mist netting	2005–2009	Dawson and Efford (2009)

Table C2. Heterogeneity in detection parameters estimated by fitting a 2-class finite mixture. Variation is compensatory (‘Comp’) if the class with larger $\hat{\lambda}_0$ has the smaller $\hat{\sigma}$. $\hat{a}_0$ is computed for each latent class from $a_0 = 2\pi\hat{\lambda}_0\hat{\sigma}^2$. ‘AIC wt’ is the weight associated with the heterogeneity model when compared to the null model. ‘PredictedRB’ refers to the relative bias of the null density estimator predicted from $\widehat{\mathrm{CV}}(a_0)$ using the curve fitted to simulated data in the main text (Fig. 2d).

Dataset	$\widehat{\mathrm{CV}}(\lambda_0)$	$\widehat{\mathrm{CV}}(\sigma^2)$	Comp	$\widehat{\mathrm{CV}}(a_0)$	AIC wt	Predicted RB
Flat-tailed horned lizard	0.76	0.41	yes	0.13	0.96	+0.00
Deer mouse ESG	0.54	0.78	yes	0.10	0.99	+0.00
Deer mouse WSG	0.59	1.24	yes	0.07	1.00	+0.01
House mouse	0.43	1.32	yes	0.38	1.00	–0.05
Speckled skink	0.19	1.33	yes	0.62	1.00	–0.16
Spotted skink	0.52	1.04	yes	0.23	0.82	–0.01
Brushtail possum	0.46	0.96	yes	0.02	1.00	+0.01
Ovenbird	0.14	0.50	no	0.73	0.37	–0.23

```
library(secr)
## detectfn = 14 selects a halfnormal function for lambda0
FTHL.fit.0 <- secr.fit(hornedlizardCH, model=list(lambda0~1, sigma~1), detectfn = 14)
FTHL.fit.h2h2 <- secr.fit(hornedlizardCH, model=list(lambda0~h2, sigma~h2), detectfn = 14)
FTHL.fit.a0h2 <- secr.fit(hornedlizardCH, model=list(a0~1, sigma~h2), detectfn = 14)

morning <- subset(housemouse, occ = c(1,3,5,7,9))  ## mornings only, following Otis et al. 1978
morning.fit.0 <- secr.fit(morning, buffer = 20, model=list(lambda0~1, sigma~1), detectfn = 14)
morning.fit.h2h2 <- secr.fit(morning, buffer = 20, model=list(lambda0~h2, sigma~h2), detectfn = 14)
morning.fit.a0h2 <- secr.fit(morning, buffer = 20, model=list(a0~1, sigma~h2), detectfn = 14)

oven.fit.0 <- secr.fit(ovenCH, buffer = 300, model=list(lambda0~1, sigma~1), detectfn = 14)
oven.fit.h2h2 <- secr.fit(ovenCH, buffer = 300, model=list(lambda0~h2, sigma~h2), detectfn = 14)
oven.fit.a0h2 <- secr.fit(ovenCH, buffer = 300, model=list(a0~1, sigma~h2), detectfn = 14)

possum.fit.0 <- secr.fit(possumCH, buffer = 200, model=list(lambda0~1, sigma~1), detectfn = 14)
possum.fit.h2h2 <- secr.fit(possumCH, buffer = 200, model=list(lambda0~h2, sigma~h2), detectfn = 14)
possum.fit.a0h2 <- secr.fit(possumCH, buffer = 200, model=list(a0~1, sigma~h2), detectfn = 14)

ESG.fit.0 <- secr.fit(deermouse.ESG, model=list(lambda0~1, sigma~1), detectfn = 14, trace = F)
ESG.fit.h2h2 <- secr.fit(deermouse.ESG, model=list(lambda0~h2, sigma~h2), detectfn = 14, trace = F)
ESG.fit.a0h2 <- secr.fit(deermouse.ESG, model=list(a0~1, sigma~h2), detectfn = 14, trace = F)

WSG.fit.0 <- secr.fit(deermouse.WSG, model=list(lambda0~1, sigma~1), detectfn = 14, trace = F)
WSG.fit.h2h2 <- secr.fit(deermouse.WSG, model=list(lambda0~h2, sigma~h2), detectfn = 14, trace = F)
WSG.fit.a0h2 <- secr.fit(deermouse.WSG, model=list(a0~1, sigma~h2), detectfn = 14, trace = F)

## problems with multi-session run ... bad starting valuescentring or mask too coarse?
LSmask <- make.mask(LStraps, type='trapbuffer', buffer = 20, spacing = 2)
infra.fit.0 <- secr.fit(infraCH, buffer = 20, model=list(lambda0~1, sigma~1), detectfn = 14,
                         start = c(2,-1.5, log(5)))
infra.fit.h2h2 <- secr.fit(infraCH, buffer = 20, model=list(lambda0~h2, sigma~h2), detectfn = 14,
```

```
                         start = c(2,-1.5, 0, log(5), 0, 0))
infra.fit.a0h2 <- secr.fit(infraCH, buffer = 20, model=list(a0~1, sigma~h2), detectfn = 14,
                         start = c(2,-1.5, log(5), 0, 0))

lineo.fit.0 <- secr.fit(lineoCH, buffer = 20, model=list(lambda0~1, sigma~1), detectfn = 14,
                         start = c(2,-1.5, log(5)))
lineo.fit.h2h2 <- secr.fit(lineoCH, buffer = 20, model=list(lambda0~h2, sigma~h2), detectfn = 14,
                         start = c(2,-1.5, 0, log(5), 0, 0))
lineo.fit.a0h2 <- secr.fit(lineoCH, buffer = 20, model=list(a0~1, sigma~h2), detectfn = 14,
                         start = c(2,-1.5, log(5), 0, 0))
```