APPENDIX to "A full-capture Hierarchical Bayesian model of Pollock's Closed Robust Design and application to dolphins"

Robert W. Rankin[a,*], Krista Nicholson[a], Simon J Allen[b], Michael Krútzen[c], Lars Bejder[a], Kenneth H. Pollock[b]

[a] *Cetacean Research Unit, School of Veterinary and Life Sciences, Murdoch University, Australia*
[b] *Oceans Institute, University of Western Australia, Australia*
[c] *Anthropological Institute and Museum, University of Zurich, Switzerland*
[d] *Department of Applied Ecology, North Carolina State University, USA*

These are the appendices to the peer-reviewed article at `http://journal.frontiersin.org/article/10.3389/fmars.2016.00025`.

5. Appendix

5.1. Full-capture modelling, recruitment ratio, and conditioning on first-capture

In this appendix, we suggest three different ways of modelling the "recruitment ratio" λ, which is necessary for modelling individuals' full-capture histories in the Bayesian PCRD. We also discuss how to condition on first-capture, which is an alternative to full-capture history modelling and is most common in Maximum Likelihood estimation of PCRD models. The recruitment ratio can be interpreted in a couple of ways. For example, in Wen et al. (2011), λ is the proportion of new recruits who are born on the study area, whereas $1-\lambda$ are the proportion of recruits who enter from offsite (permanent immigrants). In our study, we interpret λ as the proportion of newly-marked individuals who start onsite (and therefore available for capture), while $1-\lambda$ is the proportion who recruit offsite. The distinction is that our newly-marked individuals are not necessarily recruits in the biological sense; thus, we can assume they have the same behaviour as already-marked individuals.

*Corresponding author. E-mail: robertw.rankin@gmail.com

One possibility for modelling λ, is to add T new random variables, λ_t^*, one per primary period. Such parameters cannot be reliably estimated for conclusions about the recruitment processes. However, using Bayesian priors and MCMC, we can add a weakly-informative prior based on our understanding of the ecological system, and thereby make the model internally-coherent and workable. For example, we may guess that there is a 50% chance of a recruit being either a migrant or in-situ recruit, and use a weak prior, such as $\lambda_t \sim \text{Beta}(4, 4)$. If one uses JAGS, then the posterior density of λ_t will be almost exactly the same as its prior density, meaning that there is no evidence in the data to resolve a particular value. Fortunately, the process of MCMC and Gibbs sampling means that all our other parameters have posterior densities which have integrated over the entire range of values of λ. As we demonstrated, this doesn't seem to add any extra uncertainty in the posterior intervals of other parameters.

A slight modification, which we call λ_{random}, is to use some of the migration information from the marked population to influence the recruitment process, by assuming that recruits are recruited into the population instantaneously after time $t-1$ and sort immediately into either *onsite* or *offsite* with probability λ_t^*, then temporary migrate out of the study area, or remain inside, according to γ_t' and γ_t'', just like the rest of the marked population. Therefore, the realized probability of a new recruit being inside the study area for time t is:

$$\mathbb{P}(z_t^{(\text{recruit})} = 4) = \lambda_t^*(1 - \gamma_t'') + (1 - \lambda_t^*)(1 - \gamma_t') \tag{1}$$

In other words, there are two paths for a new recruit to be *onsite* for their first primary period: they arrive and stay *onsite*, or they recruit as a migrant and move *onsite* some time before t. This latter modification makes sense when we believe that new recruits should behave similarly to the already marked population.

A second possibility, which we call $\lambda_{\text{one-step-back}}$, does not treat λ as a random variable. Instead, we assume that recruits will sort into *onsite* vs. *offsite* with probability equal to the proportion of marked-animals who are *onsite* vs.

offsite, i.e., $\lambda_t^* = \frac{\#\{\text{individuals onsite}_t\}}{\#\{\text{individuals alive}_t\}}$. The quantities $\#\{\text{individuals onsite}_t\}$ and $\#\{\text{individuals alive}_t\}$ can be directly estimated internally per MCMC step using the imputted values of the latent states of all individuals: we merely count the pertinent entries in the latent state matrix $\mathbf{Z}$. For example, in our specification:

$$\begin{aligned} \#\{\text{individuals onsite}_t\} &= \sum_i^M \mathbb{I}[z_{t,i} = 4] = N_t \\ \#\{\text{individuals alive}_t\} &= \sum_i^M \mathbb{I}[z_{t,i} = 3 \cup z_{t,i} = 4]. \end{aligned} \tag{2}$$

Once again, we assume that new recruits enter immediately after $t-1$ and then move *onsite* or *offsite* with probabilities equal to those of the marked population: $\mathbb{P}(z_t^{(\text{recruit})} = 4) = \lambda_t^*(1-\gamma_t'') + (1-\lambda_t^*)(1-\gamma_t')$.

A third option is to exclusively use information about the migration parameters to calculate λ, which we call the *eigenvector decomposition* method or $\lambda_{\text{eigenvector}}$. Like the one-step-back method, it is appropriate when the recruits can be expected to recruit into the *onsite* vs. *offsite* states in a manner consistent with the temporary migration process of the marked population. This method is the fastest and incurs no additional random variables, and merely calculates λ_t directly from the migration parameters γ:

$$\lambda_t = \frac{1-\gamma_t'}{\gamma_t'' - \gamma_t' + 1} \tag{3}$$

We motivate this formula: consider if we thought an unmarked "recruit" had a 50/50 probability of being either onsite or offsite during the previous primary period ($\mathbb{P}(z_{t-1} = 4) = u_{t-1} = 0.5$), then a reasonable expectation for λ_t would be $\lambda_t = u_{t-1}(1-\gamma'') + (1-u_{t-1})(1-\gamma') = 0.5(1-\gamma'') + 0.5(1-\gamma')$. I.e. she was either physically onsite and then stayed onsite with probability $(1-\gamma'')$, or she was physically offsite and migrated into the study area with probability $(1-\gamma')$. We can continue backwards, substituting in $u_{t-1} = u_{t-2}(1-\gamma_t'') + (1-u_{t-2})(1-\gamma_t')$. If we do the recursion infinitely, the probabilities converge to the unconditional probability of being *onsite*. This recursion can be written

in matrix notation:

$$\begin{array}{c} \\ \text{onsite} \\ \text{offsite} \end{array} \begin{pmatrix} \lambda \\ 1-\lambda \end{pmatrix} = \begin{array}{c} \begin{array}{cc} \text{onsite} & \text{offsite} \end{array} \\ \begin{pmatrix} 1-\gamma'' & 1-\gamma' \\ \gamma'' & \gamma' \end{pmatrix} \end{array} \begin{pmatrix} \lambda \\ 1-\lambda \end{pmatrix} \tag{4}$$

This notation makes it clear that the λ vector is just the eigenvector of the two-state transition matrix with the migration parameters γ. While this method is easy, it has the advantage of connecting the ratio of recruits *onsite* and *offsite* with the dynamics implied by the temporary migration parameters of the marked population. To demonstrate the elegance of the idea, consider four scenarios. One scenario is that the probability of going from *onsite* to *offsite* is equal to the probability of going from *offsite* to *onsite* i.e., $\gamma'' = 1 - \gamma'$: in such cases, the individuals' movements are random with no particular preference for either location; here, the calculated ratio λ is 0.5, which matches our intuition. A second scenario is if both γ' and γ'' are low and below 0.5, which means that individuals preferentially enter and stay *onsite*; here, λ will be *high* ($\lambda > 0.5$) which matches our intuition that individuals prefer being *onsite*. A third scenario is the opposite of the second, where both γ' and γ'' are high (> 0.5), such that individuals prefer to migrate outside the study area and stay outside; here, λ will be *low* (< 0.5) to match this intuition. Lastly, consider some other arbitrary scenario, such as $\gamma'' = 0.1$ (individuals are highly unlikely to migrant *offsite* once *onsite*) and $\gamma' = 0.7$ (individuals are moderately likely to stay *offsite* once *offsite*); here, we expect individuals to have a stronger preference to be inside the study area, and $\lambda = 0.75$, which matches our intuition.

Finally, we remind the reader of the option to condition on the first-capture, as is assumed for the PCRD model in Program MARK. In this case, no recruitment or assortment process is necessary: we merely accept as a given the first capture event of each individual and do not try to model it. Such a model may be appropriate for situations when a researcher is not interested in recruitment processes or individual heterogeneity, or cannot accept the assumptions of the above full-capture specifications. Conditioning on the first-capture can mean

slightly different things in the case the PCRD as compared to other capture-recapture models. Here, we interpret "conditioning on first capture" to mean that we condition on an individual having been seen *in at least one secondary period* during the primary period t^* in which they were first encountered. This simplifies the latent-state transitions, because we only include three latent states (*dead*, *offsite* and *onsite*) and no longer need the *not-yet-entered* dummy state that we required for the full-capture models. Also, all individuals are, by definition, initialized in state *onsite* at time t^*, meaning we can drop the parameters ψ and λ altogether.

Conditioning on first-capture (during primary period t^*) changes the likelihood of secondary-period observations during t^*. The first capture history $\omega^{(t^*)}$ must necessarily arise from a set of *conditional* capture-histories, which exclude the possibility of an all-zero capture history during t^*. We denote the all-zero capture history $\omega_\emptyset$, and its compliment set Ω^c which includes all possible capture histories in a secondary period, excluding $\omega_\emptyset$. The conditional probability of a capture history $\omega^{(t^*)}$ arising from Ω^c is

$$\begin{aligned}\mathbb{P}(\omega_i^{(t^*)}|\text{seen at least once during } t^*) &= \frac{\mathbb{P}(\omega_i^{(t^*)})}{\sum_{\omega\in\Omega^c}\mathbb{P}(\omega)} = \frac{\mathbb{P}(\omega_i^{(t^*)})}{1-\mathbb{P}(\omega_\emptyset)} \\ &= \frac{\prod_s^{S_{t^*}}\left(p_{t^*,s}^{(y_{t^*,s,i})}(1-p_{t^*,s})^{(1-y_{t^*,s,i})}\right)}{1-\prod_s^{S_{t^*}}(1-p_{t^*,s})} \end{aligned} \tag{5}$$

This conditional multinomial distribution is not available in JAGS. Therefore, the likelihood must be evaluated explicitly in a peculiar way, using the "zero's trick" which was popular in early releases of WinBUGS. This technical obscurity makes the first-capture model a surprisingly more daunting challenge for the JAGS language than full-capture models, and may dissuade non-expert users. There is a further technical challenge that may confuse JAGS users when conditioning on first-capture: the handling of Markovian state transitions entails slightly different procedures for different primary periods $t_i^* = T$ vs. $t_i^* = (T-1)$ vs. $t_i^* < (T-1)$, which makes the JAGS code much more convoluted than for

full-capture modelling.

Other possibilities exist for λ_t, and the decision about which specification to use should be rooted in ecology and the particulars of the focal population. For example, in the case of bottlenose dolphins, we have two considerations: first, coastal populations are generally faithful to one geographic location throughout their adult lifetimes; second, the recruitment process is mostly due to the *mark-accumulation* process, meaning that recruits are actually onsite adults who just recently became photo-identifiable. Therefore, we expect that "recruits" will have the same migration dynamics and the same ratio of being *onsite*/*offsite* as the marked population; we say approximately because immigrants and newly-marked juveniles may not exactly have the same dynamics as the onsite marked population. This means that the *one-step-back* or *eigenvector decomposition* formulation for λ_t is most appropriate. For other situations, this may not make sense, especially for highly migratory taxa, such as migratory birds or baleen whales, or other situations where the captureability of an animal is uniform for all animals (like placing a band at first-capture), in which case recruits may literally be recruits in the biological sense. New births and immigrants are likely to have different preferences for *onsite* vs *offsite* than other adults who constitute the marked population. In such situations, it is better to default to the λ_{random} specification which makes fewer assumptions; or use external information, if available (Wen et al., 2011).

We suggest that for long capture series with many primary periods, the specification of the initial states will be less influential on parameter estimation. We explore these different specifications in the reanalysis of the western gulf Shark Bay bottlenose dolphins, which has $T = 5$ primary periods (Section 2.7).

5.2. Posterior predictive checks

Our Posterior Predictive Check (Gelman et al., 1996) is based on two discrepancy statistics, χ^2_{within} and χ^2_{between}. A discrepancy statistic compares an observed quantity vs. its expectation conditional on model parameters. For the within- and between-period discrepancy statistics, we compare the discrep-

ancies evaluated on real data (χ^2) vs. the same discrepancy (χ'^2) evaluated on idealized simulated data, conditional on model parameters and the original data. Over the entire joint posterior distribution of model parameters, the simulated data approximate the posterior predictive distribution of the response variable $\pi(\mathbf{Y}'|\theta, \mathbf{Y})$. A useful statistic is the posterior p-value p_{ppc}, which is $p(\chi^2 > \chi'^2)$, the probability that the observed data differs more from model expectations than the posterior predictive distribution. The posterior p-value is evaluated $p_{\text{ppc}} = \int_{\theta^*} \mathbb{I}[\chi^2_{\theta^*} > \chi'^2_{\theta^*}]\pi(\theta^*|\mathbf{Y})d\theta$, where $\pi(\theta|\mathbf{Y})$ is the joint posterior of model parameters. If the data conforms to the model expectations, then we expect that χ^2 is not systematically different from χ'^2, and values of $p_{\text{ppc}} \approx 0.5$. Values of p_{ppc} close to 0 or 1 suggest poor model fit. Unfortunately, posterior p-values are generally not uniform on 0 to 1, and one cannot meaningfully compare p_{ppc} to a predetermined rejection region like $(p_{\text{ppc}} < 0.025 | p_{\text{ppc}} > 0.975)$. There are other unfortunate deficiencies in interpreting posterior p-values (Gelman, 2013), especially when imputing large amounts of missing data or latent variables, which is why we restrict the check only to the processes that can be conditioned on the first capture, rather than the full generative model.

We make use of MCMC to calculate p_{ppc} as follows. For each k^{th} posterior sample of model parameters $\theta^{(k)}$: i) we calculate the $\chi^{2(k)}_{within}$, $\chi^{2(k)}_{between}$ and $\chi^{2(k)}_{total}$ using the observed data $\mathbf{Y}$; ii) we predict new data $\mathbf{Y}'^{(k)}$ conditional on $\theta^{(k)}$ and first-capture; iii) we calculate the $\chi'^{2(k)}_{within}$, $\chi'^{2(k)}_{between}$ and $\chi'^{2(k)}_{total}$ using the $\mathbf{Y}'^{(k)}$; iv) we score whether or not the discrepancy evaluated on the observed data is greater than the discrepancy evaluated on the simulated data, $I_{\theta^{(k)}} = \mathbb{I}[\chi^2_{\theta^{(k)}} > \chi'^2_{\theta^{(k)}}]$; v) we calculate p_{ppc} as the mean of I over all draws from the posterior of θ; $p_{\text{ppc}} = \int_{\theta^*} \mathbb{I}[\chi^2_{\theta^*} > \chi'^2_{\theta^*}]\pi(\theta^*|\mathbf{Y})d\theta \approx \frac{1}{N_{\text{MCMC}}} \sum_{k=1}^{N_{\text{MCMC}}} I_{\theta^{(k)}}$.

The within-period discrepancy statistic is based on the sufficiency statistics for closed-population capture (Darroch, 1958): $r_{t,s}$ the vector of number of encounters per secondary period, and R_t the total number of *uniquely* captured individuals over all secondary periods within a focal primary period t. These quantities are evaluated once for the observed data in the case of the observed χ^2_{within}, and once per k posterior draw of $\mathbf{Y}'^{(k)}$ in the case of χ'^2_{within}. The

expected number of encounters per secondary period, conditional on $\theta^{(k)}$, is $r_{t,s}^{(k)} = p_{t,s}^{(k)} \times N_t^{(k)}$, where $N_t^{(k)}$ is the total number of animals *alive* and *onsite* at primary period t, and $p_{t,s}^{(k)}$ is the detection probability per secondary period. There are a number of ways one can estimate N_t. Fortunately, the Bayesian HMM directly imputes the latent states $z_{t,i}^{(k)} \in \{1,2,3,4\}$ and to calculate $N_t^{(k)}$ we simply count the number of individuals (both observed and augmented) whose latent state is $z_{t,i}^{(k)} = 4$ for *alive and onsite* at time t.

$$
\begin{aligned}
r_{s_t}^{(k)} &= p_{s_t}^{(k)} \sum_{i}^{N} \mathbb{I}[z_{t,i}^{(k)} = 4] \\
R_t^{(k)} &= \big(1 - \prod_{s_t}^{S_t} (1 - p_{s_t}^{(k)})\big) \big(\sum_{i}^{N} \mathbb{I}[z_{t,i}^{(k)} = 4]\big) \\
\chi_{within}^{2(k)} &= \sum_{t}^{T} \Big(\sum_{s_t}^{S_t} \frac{(r_{t,s} - r_{t,s}^{(k)})^2}{r_{t,s}^{(k)}} + \frac{(R_t - R_t^{(k)})^2}{R_t^{(k)}} \Big)
\end{aligned}
\tag{6}
$$

We remind readers that $\{r_{s_t}^{(k)}\}_{s_t=1}^{S_t}$, $\{R_t^{(k)}\}_{t=1}^{T}$, $\chi_{within}^{2(k)}$ and $\chi_{within}^{\prime 2(k)}$ are evaluated *per* k^{th} *draw* from the joint posterior $\pi(N_t, p_{t,1}, ..., p_{t,S_t}|\mathbf{Y})$. The between-period discrepancy statistic is based on the observed and expected m-array of primary periods: $R_{t_1:t_2}$, the total number of recaptures first seen at primary period t_1 and not seen again until t_2, conditional on R_{t_1} the total number of encounters at time t_1. The m-array also includes $R_{t_1:T+}$ the number of encounters seen at time t_1 and never seen again.

The m-array constitutes a sufficient statistic for the simple Cormack-Jolly-Seber (CJS) open population model. The expected values of $R_{t_1:t_2}^{(k)}$ and $R_{t_1:T+}^{(k)}$ need to be calculated over all permissible pathways among hidden states, and is more complicated to calculate than the m-arrays of the CJS model. Therefore, we used the matrix-multiplication algorithm of Fujiwara & Caswell (2002) to simplify such calculations, but we illustrate the calculations of individual m-array elements below. p_t refers to the effective primary period detection

probability:

$$\chi^{2(k)}_{between} = \Big(\sum_{t_1=1}^{T-1}\sum_{t_2>t_1}^{T}\frac{(R_{t_1:t_2}-R^{(k)}_{t_1:t_2})^2}{R^{(k)}_{t_1:t_2}}\Big)+\Big(\sum_{t=1}^{T-1}\frac{(R_{t:T+}-R^{(k)}_{t:T+})^2}{R^{(k)}_{t:T+}}\Big) \tag{7}$$

$$\begin{aligned}
R^{(k)}_{1:2} &= R_1\phi^{(k)}_2(1-\gamma''^{(k)}_2)p^{(k)}_2\\
R^{(k)}_{1:3} &= R_1\Big(\phi^{(k)}_2(1-\gamma''^{(k)}_2)(1-p^{(k)}_2)\phi^{(k)}_3(1-\gamma''^{(k)}_3)p^{(k)}_3+\phi^{(k)}_2\gamma''^{(k)}_2\phi^{(k)}_3(1-\gamma'^{(k)}_3)p^{(k)}_3\Big)\\
R^{(k)}_{1:4} &= R_1\Big(\phi^{(k)}_2(1-\gamma''^{(k)}_2)(1-p^{(k)}_2)\phi^{(k)}_3(1-\gamma''^{(k)}_3)(1-p^{(k)}_3)\phi^{(k)}_4(1-\gamma''^{(k)}_4)p^{(k)}_4\\
&\quad+\phi^{(k)}_2(1-\gamma''^{(k)}_2)(1-p^{(k)}_2)\phi^{(k)}_3\gamma''^{(k)}_3\phi^{(k)}_4(1-\gamma'^{(k)}_4)p^{(k)}_4\\
&\quad+\phi^{(k)}_2\gamma''^{(k)}_2\phi^{(k)}_3\gamma'^{(k)}_3\phi^{(k)}_4(1-\gamma'^{(k)}_4)p^{(k)}_4\\
&\quad+\phi^{(k)}_2\gamma''^{(k)}_2\phi^{(k)}_3(1-\gamma'^{(k)}_3)\phi^{(k)}_4(1-\gamma''^{(k)}_4)p^{(k)}_4\Big)\\
R^{(k)}_{1:5} &= ...
\end{aligned} \tag{8}$$

Likewise, the calculations of the expected number of animals never seen again involves explicit inclusion of all permissible pathways (dead, outside study area, within study area but not detected) that results in a sequence of no detections; it is likewise best calculated using matrix multiplication.

Evidence for lack of fit in p_{ppc}-values, or high residuals, can help diagnose violation of model assumptions. Problematic p_{within} values may result from violations of population closure during primary periods (e.g., secondary periods are too spread out, or non-random temporary immigration) or strongly heterogeneous detection probabilities. Problematic p_{between} could arise from individual heterogeneity in transition parameters (survival, temporary migration), or non-Markovian migration processes, for example, if an animal has a greater probability of leaving the study area after two years of being inside vs. after just one year.

5.3. Hyperpriors for Hierarchical Bayesian PCRD

We now describe the hierarchical model for the western gulf Shark Bay bottlenose dolphins, followed by the details of our choice of hyperparameters.

- Scaled half Student-t hyperpriors $T(\sigma_\theta; s, \nu)\mathbb{I}(\sigma_\theta > 0)$ on random-effects' dispersion parameters σ_θ, with hyperparameter scale s and degrees-of-freedom ν,

$$\begin{aligned}
\pi(\sigma_{\gamma'}) &\propto T(\sigma_{\gamma'}; 0.175, 4)\mathbb{I}(\sigma_{\gamma'} > 0) \\
\pi(\sigma_{\gamma''}) &\propto T(\sigma_{\gamma''}; 0.3, 2)\mathbb{I}(\sigma_{\gamma''} > 0) \\
\pi(\sigma_{\phi}) &\propto T(\sigma_{\phi}; 0.2, 13)\mathbb{I}(\sigma_{\phi} > 0) \\
\pi(\sigma_{p(t)}) &\propto T(\sigma_{p(t)}; 0.3, 2)\mathbb{I}(\sigma_{p(t)} > 0) \\
\pi(\sigma_{p(s)}) &\propto T(\sigma_{p(s)}; 0.11, 2)\mathbb{I}(\sigma_{p(s)} > 0) \\
\pi(\sigma_{p(i)}) &\propto T(\sigma_{p(i)}; 0.11, 2)\mathbb{I}(\sigma_{p(i)} > 0)
\end{aligned} \tag{9}$$

- Normal hyperpriors on the mean location of state parameters (μ_θ):

$$\begin{aligned}
\pi(\mu_{\gamma'}) &= \mathcal{N}(\mu_{\gamma'}; 0, 1.1^2) \\
\pi(\mu_{\gamma''}) &= \mathcal{N}(\mu_{\gamma''}; 0, 1.55^2) \\
\pi(\mu_{\phi}) &= \text{Unif}(\mu_{\phi}; 0.8, 1)
\end{aligned} \tag{10}$$

  We used a different prior for μ_ϕ because small deviations in ϕ can result in large differences in expected lifespan. We used a Uniform hyperprior from 0.8 to 1, which is "informative" in a naive sense, but actually results in a highly diffused prior in a biological sense. I.e., it allows dolphins' expected lifespan to vary from from 4.98 to > 99 years, the lower bound being approximately half the age-of-first-parturition.

- time-varying state parameters with logit-Normal priors, conditional on μ_θ and σ_θ

$$\begin{aligned}
\text{logit}(\gamma'_t) &\sim \mathcal{N}(\mu_{\gamma'}, \sigma^2_{\gamma'}) \text{ for } t = 1, ..., T \\
\text{logit}(\gamma''_t) &\sim \mathcal{N}(\mu_{\gamma''}, \sigma^2_{\gamma''}) \text{ for } t = 1, ..., T \\
\text{logit}(\phi_t) &\sim \mathcal{N}(\text{logit}^{-1}(\mu_\phi), \sigma^2_\phi) \text{ for } t = 1, ..., T-1
\end{aligned} \tag{11}$$

- multilevel hierarchical relationship of detection probabilities, across pri-

mary periods, within primary periods, and among individuals:

$$\begin{aligned}
\pi(\mu_p) &= \mathcal{N}(\mu_p; 0, 1.55^2) \\
\mu_{p(t)} &\sim \mathcal{N}(\mu_p, \sigma^2_{p(t)}) \text{ for } t = 1, ..., T \\
\mu_{p(s)} &\sim \mathcal{N}(\mu_{p(t)}, \sigma^2_{p(s)}) \text{ for } s_t = 1, ..., S_t;\ t = 1, ..., T \\
\epsilon_i &\sim \mathcal{N}(0, \sigma^2_{p(i)}) \text{ for } i = 1, ..., M \\
\text{logit}(p_{t,s,i}) &= \mu_{p(s)} + \epsilon_i \text{ for } i = 1, ..., M;\ s = 1, ..., S_t;\ t = 1, ..., T
\end{aligned} \tag{12}$$

- recruitment parameters

$$\begin{aligned}
\psi_t &\sim \text{Unif}(0, 1) \\
\lambda_t &= f(\gamma'_t, \gamma''_t)
\end{aligned} \tag{13}$$

- latent state transitions and the likelihood (see matrix 15).

$$\begin{aligned}
\text{initialize: } z_{0,i} &= 1 \text{ for } i = 1, ..., m \\
p(z_{t,i}|z_{t-1,i}, \mathbf{A}_t) &= \text{Cat}(\mathbf{A}_t[\cdot, z_{t-1,i}]) \text{ for } i = 1, ..., m;\ t = 1, ..., T \\
p(y_{t,s,i}|z_{t,i}, p_{t,s,i}) &= \text{Bern}(p_{t,s,i}^{\mathbb{I}[z_{t,i}=4]}) \text{ for } i = 1, ..., m; s_t = 1, ..., S_t;\ t = 1, ..., T
\end{aligned} \tag{14}$$

The transition matrix $\mathbf{A}_t$ is parametrized with the demographic parameters from the PCRD:

$$\mathbf{A}_t = \begin{array}{c} \\ \text{not yet entered} \\ \text{dead} \\ \text{offsite} \\ \text{onsite} \end{array} \begin{array}{c} \begin{array}{cccc} \text{not yet entered} & \text{dead} & \text{offsite} & \text{onsite} \end{array} \\ \begin{pmatrix} 1-\psi_t & 0 & 0 & 0 \\ 0 & 1 & 1-\phi_t & 1-\phi_t \\ \psi_t(1-\lambda_t) & 0 & \phi_t\gamma'_t & \phi_t\gamma''_t \\ \psi_t\lambda_t & 0 & \phi_t(1-\gamma'_t) & \phi_t(1-\gamma''_t) \end{pmatrix} \end{array} \tag{15}$$

We now explain our choice of hyperpriors which govern the distributions $\text{logit}(\theta) \sim \mathcal{N}(\mu_\theta, \sigma^2_\theta)$, beginning with the hyperprior on the location parameters μ_θ, the "default" and least-informative which is $\pi(\mu_\theta) = \mathcal{N}(\mu_\theta; 0, 1.55^2)$. The dispersion hyperparameter 1.55 ensures that our prior on $\pi(\mu_\theta)$ is approximately uniform on the probability scale, conveying no prior knowledge. We use this hyperparameter on the $\pi(\mu_{\gamma''})$ (mean out-migration) and for $\pi(\mu_{p(t)})$

(mean detection probability). A tighter dispersion of 1.1^2 for $\pi(\mu_{\gamma'})$ serves to remove some density from the boundaries 0 and 1 on the probability scale but remains largely uninformative. For ϕ, we use our ecological intuition about the long-lived, high-survival life-history of marine mammals (Silva et al., 2009; Nicholson et al., 2012), creating a moderately informative distribution uniform on 0.8 to 1.

For the dispersion parameters σ_θ, we use the scaled half Student-t distribution, parametrized with scale s and degrees-of-freedom ν. The half-t distribution is recommended by Gelman (2006), in situations when: i) shrinkage to zero is preferred, and ii) there are few (< 5) groups. In such situations, Uniform or Inverse-Gamma hyperpriors can lead to inflated variance of the random variables. The decisions to parametrized s and ν can be challenging, especially because it is the σ_θ's which are instrumental in promoting shrinkage, or, if mis-specified, can inflate the variance of the time-varying state parameters θ_t. Generally, s is (approximately) our prior expectation for σ_θ, such that low values $s << 0.5$ can help "fix" the state parameters to be constant over all primary periods. Recognizing the logit-to-probability transformation, values of $\sigma_\theta > 1.55$ are undesirable, in that they allow the state-parameters to have a bimodal distribution peaked at 0 and 1 on the probability space, which is nonsensical for our purposes. Rather, we desire a high probability density at zero, and very small density for values of σ_θ beyond 1.55. ν governs the tail of the distribution of σ_θ, and generally controls the degree to which strong evidence and a high value of σ_{MLE} is permissible under the prior. Particular cases are: $\nu = 1$, the standard half-Cauchy with infinite variance, which makes the MLE estimates much more likely to drive the posterior and deflates the influence of our prior expectation s; and $\nu > 5$ which makes the T distribution look increasingly half-Normal, thereby driving the posterior expectation of σ_θ towards the hyperparameter s.

Our hyperparameters of $s = 0.3$ and $\nu = 2$ merely encode the belief that σ_θ has a mode at 0 and low probability beyond $\sigma_\theta > 1.55$. We used these hyperparameters for $\sigma_{p(t)}$ and $\sigma_{\gamma''}$, which, based on our simulations, should be easily estimated by the data. The value of $\nu = 2$ yields a long-tailed distribution

which allows potentially higher σ_{MLE} values to overwhelm the prior and drive the posterior distribution. For other parameters which suffer greater correlations in parameter estimates (γ', ϕ) and are more difficult to estimate, we employ stronger hyperparameters (lower s and larger ν) to shrink $\sigma_{\gamma'}$ and σ_{ϕ} towards zero; or, equivalently, shrink the time-varying values towards their means, $\mu_{\gamma'}$ and μ_{ϕ}.

For our hyperprior on $\sigma_{\gamma'}$, we used distributions which best conformed to the following beliefs, stated as specific probability statements about the likely *range* of values of γ'_t (expressed as the inter-quantile range between 68% and 95% intervals): $\mathbb{P}[\text{range}_{68\%\text{CI}}(\gamma'_t) < 0.1] = 0.76$; $\mathbb{P}[\text{range}_{68\%\text{CI}}(\gamma'_t) < 0.3] = 0.98$; $\mathbb{P}[\text{range}_{95\%\text{CI}}(\gamma'_t) < 0.2] = 0.82$; $\mathbb{P}[\text{range}_{95\%\text{CI}}(\gamma'_t) > 0.5] = 0.014$. In plain speak, the first statement merely states that the majority of values of γ'_t should be within 0.1 of each other; whereas the final statement expresses the belief that there is a tiny probability that any two values of γ'_t are more than 0.5 probability units apart. A value $\nu = 4$ ensures that even in the face of strong evidence in the data, the posterior of $\sigma_{\gamma'}$ will still not be able to take on high values.

A similar exercise was performed for ϕ_t, with even more extreme fixing of the variance to promote a near constant ϕ. We used the following beliefs: $\mathbb{P}[\text{range}_{68\%\text{CI}}(\phi_t) < 0.02] = 0.64$; $\mathbb{P}[\text{range}_{68\%\text{CI}}(\phi_t) < 0.1] = 0.997$; $\mathbb{P}[\text{range}_{95\%\text{CI}}(\phi_t) < 0.02] = 0.42$; $\mathbb{P}[\text{range}_{95\%\text{CI}}(\phi_t) < 0.04] = 0.71$; $\mathbb{P}[\text{range}_{95\%\text{CI}}(\phi_t) > 0.1] = 0.035$. In other words, there is a $2/3$ chance that the majority of ϕ_t values are within 0.02 units of each other, and the chance that any two values are greater than 0.1 probability units from each other is very small. A value of $\nu = 13$ makes the posterior of σ_{ϕ} insensitive to potentially higher values of σ_{MLE}.

5.4. JAGS code: PCRD full-capture "fixed-effect" model

The following is JAGS code for the simple "fixed-effect" version of the Bayesian HMM PCRD. For the variant that conditions on the first-capture, see Appendix section 5.5; for the Hierarchical Bayesian PCRD, see the Appendix section 5.6. For an online demonstration with real data, see the R code

at: https://github.com/faraway1nspace/PCRD_JAGS_demo.

Users should tweak which parameters they want to be time-variant or time-invariant (in the following example gamma1 and phi survival are time-invariant, whereas gamma2 varies per primary period, and detection probabilities pd vary per secondary periods). The necessary input data (in R or Matlab) is: T is the number of primary periods; T2 is a vector of the number of secondary periods per primary period, of length T; M is the total number of individuals, including both observed individuals and pseudo individuals; Y is a 3D array where rows are M individuals, columns are the secondary periods, and the 3rd dimension represents T primary periods, where every cell is either 1 for observed, 0 for unobserved, and NA for missing. Note, if there are a different number of secondary periods per primary period, then the number of columns of Y will be the maximum number of secondary periods, and the extra entries for those primary periods with fewer secondary periods should be padded with NA, keeping in mind that the vector T2 (see above) sets the upper-bound for the number of secondary periods per primary period for JAGS to evaluate. Users must augment their observed capture history matrix with "all-zero" capture histories, for a total of M observed and pseudo-individuals.

A special challenge for the JAGS version of the PCRD is initializing the latent states' matrix z in a manner consistent with the observed data. See section 5.7 for example R-code.

```
model{
  # priors
  phi ~ dbeta(1,1) # apparent survival probability (adjust for ecological
  gamma1 ~ dbeta(1.2,1,2) #  temp migration: probabilty stays out conditional on being out
  # loop through primary periods: parameters for detection probability pd_t, and gamma2_t
  for(t_ in 1:T){ # T primary periods...
     pd_a[t_] ~ dgamma(3,2) # hyperprior on detect probs
     pd_b[t_] ~ dgamma(3,2) # hyperprior on  detect probs
     for(tt_ in 1:T2[t_]){ # loop through secondary periods...
        pd[t_,tt_] ~ dbeta(pd_a[t_],pd_b[t_]) # secondard-period-level detection probs
     }
     gamma2[t_] ~ dbeta(1,1) # temp migration: prob migrates outside conditional on being inside
     # recruitment process from 'eigenvector decomposition'
     lambda[1,t_] <- (1-gamma1)/(gamma2[t_]-gamma1+1) # recruitment ratio, or long-term prob of being
     lambda[2,t_] <- 1-lambda[1,t_] #
```

```
   psi[t_] ~ dunif(0,1) # inclusion probability
   # trmat: transition matrix for Markovian latent-states
   # 1 =not yet in population;2=dead;3=offsite;4=onsite (only observable state)
   # transition are from the column --> rows
   # trmat[row,column,time] = [state at time=t_; state at time t_-1; time=t_]
   trmat[1,1,t_] <- 1-psi[t_] # excluded from pop
   trmat[2,1,t_] <- 0 # dead
   trmat[3,1,t_] <- psi[t_]*lambda[2,t_] # inclusion into pop, outside study are
   trmat[4,1,t_] <- psi[t_]*lambda[1,t_] # inclusion into pop, inside study area
   trmat[1,2,t_]<- 0
   trmat[2,2,t_]<- 1 # stay dead
   trmat[3,2,t_]<- 0
   trmat[4,2,t_]<- 0
   trmat[1,3,t_]<- 0
   trmat[2,3,t_]<- 1-phi # dies outside
   trmat[3,3,t_]<- gamma1*phi # stays outside | outside
   trmat[4,3,t_]<- (1-gamma1)*phi # reenters study area | outside
   trmat[1,4,t_]<- 0 #
   trmat[2,4,t_]<- 1-phi # dies inside
   trmat[3,4,t_]<- gamma2[t_]*phi # leaves study area | inside
   trmat[4,4,t_]<- (1-gamma2[t_])*phi # stays inside | inside
} # t_
# likelihood: loop through M individuals, both real and pseudo-individuals
for (i in 1:M){
  #draw latent state at primary period 1:
  # ... by definition, everyone starts in z=1 (not-in-population) at time=0
  z[i,1]~ dcat(trmat[1:4,1,1]) # first z strictly excluded from pop
  # likelihood for first primary period
  for(tt_ in 1:T2[1]){ # loop through secondary periods
       # Bernouilli process, conditional on z=4, otherwise no observation
       y[i,tt_,1] ~ dbern(pd[1,tt_]*equals(z[i,1],4))
  }
  alive_i[i,1] <- step(z[i,1]-3) # count if i is alive or not
  Nin_i[i,1] <- equals(z[i,1],4) # count if i is within study area
  # loop through primary periods after 1st primary periods
  for(t_ in 2:T){
    # state process: draw z(t_) conditional on z(t_-1)
    z[i,t_] ~ dcat(trmat[1:4, z[i,t_-1] , t_])
    # likelihood: loop through secondary period observations
    for(tt_ in 1:T2[t_]){
       # Bernouilli observation error, conditional on z=4
       y[i,tt_,t_] ~ dbern(pd[t_,tt_]*equals(z[i,t_],4))
    }
    # tally population size
    alive_i[i,t_] <- step(z[i,t_]-3) # check alive or not
    Nin_i[i,t_] <- equals(z[i,t_],4) # count if i is within study area
  } # t_
} # i
# estimate population size per primary periods
for(t_ in 1:T){
```

```
    alive[t_] <- sum(alive_i[,t_]) # number alive
    Nin[t_] <- sum(Nin_i[,t_]) # number in study area
  } # t_
}
```

5.5. JAGS code: PCRD first-capture "fixed-effect" model

The following is JAGS code for the "fixed-effects" version of the PCRD which conditions on an animal's first-capture. For the fixed-effect model that uses the full-capture histories, see Appendix section 5.4; for the Hierarchical Bayesian PCRD, see Appendix section 5.6. For an online demonstration with real data, see the R code at: `https://github.com/faraway1nspace/PCRD_JAGS_demo`.

Users should tweak which parameters they want to be time-variant or time-invariant (in the following example `gamma1` and `phi` survival are time-invariant, whereas `gamma2` varies per primary period, and detection probabilities `pd` vary per secondary periods). The necessary input data (in R or Matlab) are: `T` is the number of primary periods; `T2` is a vector of the number of secondary periods per primary period, of length `T`; `N` is a integer on the number of observed individuals; `Y` is a 3D array where rows are `N` individuals, columns are the secondary periods, and the 3rd dimension represents `T` primary periods, where every cell is either 1 for observed, 0 for unobserved, and `NA` for missing. `zeros` is a vector of length `N` filled with zero, and is used for the WinBUGS "zeros trick" to manually calculate likelihood of observations during secondary period in which we know there was at least one observation during the entire primary period; `first` is a vector of length `N` which lists which primary period each i^{th} individual was first encountered; `N.ix2` is a vector of indices pointing to those individuals who were potentially available for 2 or more primary periods, i.e., their first primary period was at least $T-1$ or earlier. `N.ix3` is a vector of indices pointing to those individuals who were potentially available for 3 or more primary periods, i.e., their first primary period was at least $T-2$ or earlier, which may include some of the same individuals in `N.ix2`. Note, if there are a different number of secondary periods per primary period, then the number of columns of `Y` will be the maximum number of secondary periods, and the extra entries for

those primary periods with fewer secondary periods should be padded with NA, keeping in mind that the vector T2 (see above) sets the upper-bound for the number of secondary periods per primary period for JAGS to evaluate.

A special challenge for the JAGS version of the PCRD is initializing the latent states' matrix z in a manner consistent with the observed data. See Appendix section 5.7 for example R-code.

```
model{
  # priors
  phi ~ dbeta(1,1) #apparent survival probability
  gamma1 ~ dbeta(1.2,1,2) #temporary migration: probabilty stays out conditional on being out
  for(t_ in 1:T){ #T primary periods...
     pd_a[t_] ~ dgamma(3,2) #hyperprior on detect probs
     pd_b[t_] ~ dgamma(3,2) #hyperprior on  detect probs
     for(tt_ in 1:T2[t_]){ #loop through secondary periods...
        pd[t_,tt_] ~ dbeta(pd_a[t_],pd_b[t_]) #secondard-period-level detection probs
     }
     p.eff[t_] <- 1-prod(1-pd[t_,1:T2[t_]]) #effective detection prob per primary period
  }
  #loop through (T-1) primary periods
  #NOTE: trmat's are offset -1 in time, eg. t_=1 implies a transition between period 1 to period 2.
  for(t_ in 1:(T-1)){
     gamma2[t_] ~ dbeta(1,1) #temp migration: prob migrate out conditional on being inside
     #trmat: transition matrix for Markovian latent-states
     #1=dead;2=offsite;3=onsite
     #transition are from the column --> rows
     #trmat[row,column,time] = [state at time=t_; state at time t_-1; time=t_]
     trmat[1,1,t_]<- 1 #stay dead
     trmat[2,1,t_]<- 0
     trmat[3,1,t_]<- 0
     trmat[1,2,t_]<- 1-phi #dies outside
     trmat[2,2,t_]<- gamma1*phi #stays outside | outside
     trmat[3,2,t_]<- (1-gamma1)*phi #reenters study area | outside
     trmat[1,3,t_]<- 1-phi #dies inside
     trmat[2,3,t_]<- gamma2[t_]*phi #leaves study area | inside
     trmat[3,3,t_]<- (1-gamma2[t_])*phi #stays inside | inside
  } #t_ state process
  #PART1: likelihood for first-capture (all individuals)
  for(i in 1:N){
    #Observation error during 1st capture: condition on at least one capture:
    #the following formula is the (conditional) multinomial log-likelihood of
    #...the sequence of observations in a primary period, conditional on that we
    #...know they were seen at least once
        LL[i]<-sum(y[i,1:T2[first[i]],first[i]]*log(pd[first[i],1:T2[first[i]]])+
                 (1-y[i,1:T2[first[i]],first[i]])*log(1-pd[first[i],1:T2[first[i]]])) -
                 log(p.eff[first[i]]) #multinomial log-likelihood
zeros[i] ~ dpois(-1*LL[i]+C) #Winbugs zeros trick, likelihood passed to JAGS as ZIP
```

```
 } #i
 mintrick <- max(LL[1:N]) #strictly for monitoring the first-capture likelihood trick
 #PART2: loop through individuals potentially available for 2 or more primary periods
 for(i in 1:length(N.ix2)){
   #state process for latent state after their first primary period
   #draw z conditional on being seen during previous primary period
   z[N.ix2[i],first[N.ix2[i]]+1]~ dcat(trmat[1:3,3,first[N.ix2[i]]])
   #loop through secondary periods
   for(tt_ in 1:T2[first[N.ix2[i]]+1]){
       #likelihood of secondary periods observation, conditional on z=3
       y[N.ix2[i],tt_,first[N.ix2[i]]+1] ~ dbern(pd[first[N.ix2[i]]+1,tt_] *
                                              equals(z[N.ix2[i],first[N.ix2[i]]+1],3))
   } #t_
} #N.ix2
#PART3: loop through individuals potentially available for 3 or more primary periods
for(i in 1:length(N.ix3)){
  #loop through remain primary periods after first and second primary periods
  for(t_ in (first[N.ix3[i]]+2):T){
      #state process: draw z(t_) conditional on  z(t_-1)
      z[N.ix3[i],t_]~ dcat(trmat[1:3, z[N.ix3[i],t_-1], t_-1]) #
      #Observation error: Bernoulli
      for(tt_ in 1:T2[t_]){
          #likelihood of secondary periods observation, conditional on z=3
          y[N.ix3[i],tt_,t_] ~ dbern(pd[t_,tt_]*equals(z[N.ix3[i],t_],3))
      } #tt_
   } #t_
 } #N.ix3
# estimate number of individuals available for capture (inside) per primary period
for(t_ in 1:T){
  Nin[t_] <- n.vector[t_]/p.eff[t_]
} #t_
}
```

5.6. JAGS code: PCRD Hierarchical Bayes

The following shows JAGS code for the Hierarchical Bayesian version of the PCRD. For the "fixed-effects" version that conditions on the first-capture, see Appendix section 5.5; for the full-capture "fixed-effects" version, see Appendix section 5.4. For an online demonstration with real data, see the R code at: `https://github.com/faraway1nspace/PCRD_JAGS_demo`.

The necessary input data (in R or Matlab) are: `T` is the number of primary periods; `T2` is a vector of the number of secondary periods per primary period, of length `T`; `M` is the total number of individuals, including both observed individuals and pseudo-individuals; `Y` is a 3D array where rows are `M` individuals,

columns are the secondary periods, and the 3rd dimension represents `T` primary periods, where every cell is either 1 for observed, 0 for unobserved, or `NA` for missing. Note, if there are a different number of secondary periods per primary period, then the number of columns of `Y` will be the maximum number of secondary periods, and the extra cell entries for primary periods with fewer secondary periods should be padded with NA (recall that the vector `T2` instructs JAGS about the number of secondary periods per primary period). Users must augment their observed capture histories with "all-zero" capture histories, for a total of `M` observed individuals and pseudo-individuals.

In addition, users must input the hyperparameters governing the time-varying state parameters; see the variables with the `pr.` suffix. `pr.phiunif` is the min and max for the hyperprior controlling the mean survival probability. `pr.g1mu`, `pr.g2mu`, `pr.pdmu` are the precision parameters for the logit-Normal hyperprior governing the values of $\mu_{\gamma'}$, $\mu_{\gamma''}$, and $\mu_{p(t)}$, respectively, and should be approximately 0.4 to ensure a uniform and uninformative distribution on the probability scale. `pr.tauphi`, `pr.taug1`, `pr.taug2`, `pr.taupdmu`, `pr.taupd2nd` and `pr.taueps` are the hyperparameters controlling the dispersion of time-varying parameters, respectively, σ_{ϕ}, $\sigma_{\gamma'}$, $\sigma_{\gamma''}$, $\sigma_{\gamma'}$, $\sigma_{p(t)}$, $\sigma_{p(t,s)}$, $\sigma_{p(i)}$. Each is a vector of two elements, the first being the inverse-scale parameter s and the second being the degrees-of-freedom ν (or shape parameter) of the scaled half Student-t distribution.

A special challenge for the JAGS version of the PCRD is initializing the latent states' matrix `z` in a manner consistent with the observed data. See Appendix section 5.7 for example R code.

```
model{
  #hyperpriors: logit(theta_mu)~Normal and theta_sd ~ half-t
  phi.mu ~ dunif(pr.phiunif[1],pr.phiunif[2]) #mean survival with a Uniform prior
  sigma.phi ~ dt(0,pr.tauphi[1],pr.tauphi[2]) T(0,) #mean survival dispersion, half-t hyperprior
  g1.mu ~ dnorm(0,pr.g1mu) #mean gamma1, temp migration out | out
  sigma.g1~dt(0,pr.taug1[1],pr.taug1[2]) T(0,) #mean gamma1 dispersion, half-t hyperprior
  g2.mu ~ dnorm(0,pr.g2mu) #mean gamma2,  temp migration out | in
  sigma.g2~dt(0,pr.taug2[1],pr.taug2[2]) T(0,) #mean gamma2 dispersion, half-t hyperprior
  pd.mu ~ dnorm(0,pr.pdmu) #mean detection prob, overall
  sigma.pdmu~dt(0,pr.taupdmu[1],pr.taupdmu[2]) T(0,) #primary period detection prob dispersion
```

```
sigma.pd2nd~dt(0,pr.taupd2nd[1],pr.taupd2nd[2]) T(0,) #secondary periods detection prob dispersion
sigma.eps ~ dt(0,pr.taueps[1],pr.taueps[2]) T(0,) #individual detection prob dispersion
#time-variant parameters
for(t_ in 1:T){ #loop through primary periods
  pd_mu[t_]~dnorm(pd.mu,pow(sigma.pdmu,-2)) #primary period mean detaction prob (logit)
  lgamma1[t_]~dnorm(g1.mu,pow(sigma.g1,-2)) #prob migrate out|out (logit)
  gamma1[t_] <- ilogit(lgamma1[t_]) #prob migrate out|out (probability)
  lgamma2[t_]~dnorm(g2.mu,pow(sigma.g2,-2)) #prob migrate out|in (logit)
  gamma2[t_] <- ilogit(lgamma2[t_]) #prob migrate out|in (probability)
  #RECRUITMENT: psi is the 'inclusion probability' and lambda is the 'recruitment ratio'
  psi[t_]~dunif(0,1) #inclusion probability
  lambda[t_] <- (1-gamma1[t_])/(gamma2[t_]-gamma1[t_]+1) #long-term probability inside study area
  #NOTE, lambda could also be a random variable with a beta prior
  #secondary-period detection probabilities
  for(tt_ in 1:T2[t_]){ #loop through secondary periods
    pd[t_,tt_] ~ dnorm(pd_mu[t_],pow(sigma.pd2nd,-2))
  } #tt_
} #tt
#first state transition (pure nusance; strictly from outside-pop to part of marked-pop)
trmat0[1] <- (1-psi[1]) #remains not-yet-in-pop
trmat0[2] <- 0
trmat0[3] <- psi[1]*(1-lambda[1]) #inclusion into pop, goes outside study are
trmat0[4] <- psi[1]*lambda[1] #inclusion into pop, goes inside study
#state transitions (2:T)
for(t_ in 1:(T-1)){
  lphi[t_]~dnorm(log(phi.mu/(1-phi.mu)), pow(sigma.phi,-2)) #survival prob (logit)
  phi[t_]<-ilogit(lphi[t_])
  #state transitions
   #trmat: transition matrix for Markovian latent-states
   #1 =not yet in population; 2=dead;3=offsite;4=onsite (only observable state)
   #transition are from the column --> rows
   #trmat[row,column,time] = [state at time=t_; state at time t_-1; time=t_]
   #notice that the primary periods are offset by 1 (because we already dealt with T=1)
   trmat[1,1,t_]<- 1-psi[t_+1] #excluded from pop
   trmat[2,1,t_] <- 0 #dead
   trmat[3,1,t_] <- psi[t_+1]*(1-lambda[t_+1]) #inclusion into pop,outside study
   trmat[4,1,t_] <- psi[t_+1]*lambda[t_+1] #inclusion into pop,inside study
   trmat[1,2,t_]<- 0
   trmat[2,2,t_]<- 1 #stay dead
   trmat[3,2,t_]<- 0
   trmat[4,2,t_]<- 0
   trmat[1,3,t_]<- 0
   trmat[2,3,t_]<- 1-phi[t_] #dies outside
   trmat[3,3,t_]<- gamma1[t_+1]*phi[t_] #stays outside | outside
   trmat[4,3,t_]<- (1-gamma1[t_+1])*phi[t_] #reenters study area | outside
   trmat[1,4,t_]<- 0 #
   trmat[2,4,t_]<- 1-phi[t_] #dies inside
   trmat[3,4,t_]<- gamma2[t_+1]*phi[t_] #leaves study area | inside
   trmat[4,4,t_]<- (1-gamma2[t_+1])*phi[t_] #stays inside | inside
} #t_
```

```
  #loop through M individuals
  for (i in 1:M){
    #state transitions and likelihood for the first primary period
    z[i,1]~ dcat(trmat0) #z at time 0 is strictly 'not-yet-in-pop'
    alive_i[i,1] <- step(z[i,1]-3) #count if i is alive or not
    Nin_i[i,1] <- equals(z[i,1],4) #count if i is within study area
    eps_i[i] ~ dnorm(0,pow(sigma.eps,-2)) #random effects at individual levels
    #Observation error y[i,tt_,t_] ~ Bernoulli conditional on being inside z=4
    for(tt_ in 1:T2[1]){  #loop through secondary periods
       y[i,tt_,1] ~ dbern(equals(z[i,1],4)/(1+exp(-pd[1,tt_]-eps_i[i]))) #inverse-logit transform
    }
    #state transition and likelihood for primary periods 2:T
    for(t_ in 2:T){
      #State process: draw z(t_) conditional on  z(t_-1)
      z[i,t_] ~ dcat(trmat[1:4, z[i,t_-1] , t_-1])
      #Observation error y[i,tt_,t_] ~ Bernoulli condition on being inside z=4
      for(tt_ in 1:T2[t_]){ #loop through secondary periods
        y[i,tt_,t_] ~ dbern(equals(z[i,t_],4)/(1+exp(-pd[t_,tt_]-eps_[i]))) #inverse-logit transform
      }
      #check whether i individual is alive and inside
      alive_i[i,t_] <- step(z[i,t_]-3) #check alive
      Nin_i[i,t_] <- equals(z[i,t_],4) #count if i is within study area
    } #t_
  } #i
  #tally population size
  for(t_ in 1:T){
     alive[t_] <- sum(alive_i[,t_]) #number alive
     Nin[t_] <- sum(Nin_i[,t_]) #number in study area
  } #t_
} #end model
```

5.7. R code: initializing latent states for JAGS

JAGS requires initial guesses of all random variables to start each MCMC chain, and to do so in a way that is consistent with the observed data and model assumptions. The latent state matrix `z` is perhaps the most difficult variable to initialize, so we have provided an example R function `generate.z.psi` to facilitate this initialization. The function uses a classic Hidden Markov Model forward-messaging/backwards-sampling algorithm to sample from `z`, conditional on PCRD parameters and the observed capture histories. The user must re-run the function for each MCMC chain, and append the resulting z matrix to a named list of other initialized parameters (which are trivial to generate). See the `rjags` manual for how to pass initial values to JAGS. Here, the user must input arguments: `y`, a 3D array of the observed capture histories (including all-zero

pseudo-individuals), identical to that used in the JAGS model; `T2` a vector of the number of secondary periods per primary period; `first.capture` a boolean integer whether or not the user intends to condition on first-capture (`TRUE`) or intends to model the full-capture histories (`FALSE`), in which case the function also returns initial guesses for the recruitment parameter `psi`. `z.priors` are a named list of Beta parameters used to generate random values of the PCRD parameters (`phi`, `g1`, `g2`, `pd`) which are then used to generate the latent states `z`. The default values should work well and the JAGS model should quickly leave the state initializations and converge on the posterior expectations.

```
generate.z.psi <- function(y,T2,first.capture=FALSE,
  z.priors = list(phi.beta=c(shape1=30,shape2=5),
       g1.beta=c(shape1=20,shape2=20),
       g2.beta=c(shape1=20,shape2=20),
       pd.beta=c(shape1=12,shape2=65))){
    exclude_=1; dead_=2;out_=3;in_=4; nstates = 4; T=length(T2)
    Y.t <-apply(y,c(1,3),function(x) sum(x,na.rm=TRUE)) # captures per primary period
    mm<-nrow(Y.t)
    first.capt <- apply(Y.t,1,function(x){ if(any(x>0)){ min(which(x>0))} else {T+1}})
    priors.str = strsplit(names(z.priors),split=".",fixed=TRUE) # get name of parameter and its densi
    random.var = lapply(priors.str, function(x) x[1]) # get names of random variables
    for(par_ in 1:length(z.priors)){
        rdraw=do.call(get(paste0("r",priors.str[[par_]][2])),
                      args=c(list(n=T),as.list(z.priors[[par_]])))
        assign(priors.str[[par_]][1],value=rdraw)
    }
    # estimate psi values
    psi.counts = tabulate(first.capt,(T+1))[1:T] # number of entries per primary periods
    psi= rbeta(T,10+psi.counts,5+(mm + mm*first.capture)-psi.counts) # random draws from beta
    t.mat = array(0,c(4,4,T)) # transition matrix
    if(!any(random.var=="lambda")){ # esimate lambda, only if its not a random variable
        lambda = (1-g1)/(g2-g1+1) # probability of transitioning to 'onsite' state
    }
    for(t_ in 1:T){
        t.mat[,1,t_]<-c(1-psi[t_],0,psi[t_]*(1-lambda[t_]),psi[t_]*lambda[t_])
        t.mat[,2,t_]<-c(0,1,0,0)
        t.mat[,3,t_]<-c(0,1-phi[t_],g1[t_]*phi[t_],(1-g1[t_])*phi[t_])
        t.mat[,4,t_]<-c(0,1-phi[t_],g2[t_]*phi[t_],(1-g2[t_])*phi[t_])
    }
    z = matrix(0,mm,T) # latent states matrix
    a = matrix(0,nstates,T+1);a[,1]=1*((1:4)==exclude_) # forward messages
    z.vec = numeric(T) # latent states
    for(i in 1:mm){ # loop through individuals
        for(t_ in 1:T){ # loop through primary periods
            alph = dbinom(x = Y.t[i,t_],size=T2[t_],prob = pd[t_]*((1:nstates) == in_)) *
```

```
                (t.mat[,,t_])%*%a[,t_]
            a[,t_+1] = alph/sum(alph) # forward messages
        } # t_
        # backwards sampling z_{t} ~ z_{t+1},Y
        z.vec[T] <- sample(1:nstates,1,replace=FALSE,prob=a[,T+1])
        for(t_ in (T-1):1){
            p_z = dbinom(x = Y.t[i,t_+1],size=T2[t_+1],prob = pd[t_+1]*(z.vec[t_+1] == in_)) *
                  t.mat[z.vec[t_+1],,t_+1]*a[,t_+1]/a[z.vec[t_+1],t_+2]
            z.vec[t_]<-sample(1:nstates,1,replace=FALSE,prob=p_z/sum(p_z))
        }
        z[i,] <- z.vec
    } # i
    # if conditioning on first capture
    if(first.capture){ # replace z values as NA, if only modelling fullcapture
        z <- z-1 # need to remove unseen 'not-yet-entered' state
        for(i in 1:mm){
            z[i,1:first.capt[i]]<-rep(NA,first.capt[i])
        }
        RET = list(z=z)
    } else { #
    # if modelling full capture histories, need psi estimates too
        RET = list(z = z, psi = psi) # return latents states and psi
    }
    return(RET) # inner function
}
```

6. Works Cited

Darroch, J. N. (1958). The multiple-recapture census: I. Estimation of a closed population. *Biometrika*, *45*, 343–359. doi:10.2307/2333183.

Fujiwara, M., & Caswell, H. (2002). Estimating population projection matrices from multi-stage mark–recapture data. *Ecology*, *83*, 3257–3265. doi:10.1890/0012-9658(2002)083[3257:EPPMFM]2.0.CO;2.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian analysis*, *1*, 515–533. doi:10.1214/06-BA117A.

Gelman, A. (2013). Two simple examples for understanding posterior p-values whose distributions are far from unform. *Electronic Journal of Statistics*, *7*, 2595–2602. doi:10.1214/13-EJS854.

Gelman, A., Meng, X., & Stern, H. (1996). Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica*, *6*, 733–759.

Nicholson, K., Bejder, L., Allen, S. J., Krützen, M., & Pollock, K. H. (2012). Abundance, survival and temporary emigration of bottlenose dolphins (*Tursiops sp.*) off Useless Loop in the western gulf of Shark Bay, Western Australia. *Marine and Freshwater Research*, *63*, 1059–1068. URL: http://dx.doi.org/10.1071/MF12210.

Silva, M., Magalhães, S., Prieto, R., Santos, R., & Hammond, P. (2009). Estimating survival and abundance in a bottlenose dolphin population taking into account transience and temporary emigration. *Marine Ecology Progress Series*, *392*, 263–276. doi:10.3354/meps08233.

Wen, Z., Pollock, K., Nichols, J., & Waser, P. (2011). Augmenting superpopulation capture-recapture models with population assignment data. *Biometrics*, *67*, 691–700. doi:10.1111/j.1541-0420.2010.01522.x.