README Version 2 for return-time capture-recapture models

Fitting the models in "Breeding return times and abundance in capture-recapture models"

Shirley Pledger, Edward Baker and Kim Scribner (2013)
Biometrics **69**, 991–1001 DOI: 10.1111/biom12094

This is Version 2 of the readme pdf for fitting return-time capture-recapture models.

The early part of this pdf is essentially the same as the earlier readme file in Version 1. Section 4 gives information about using this code if a year of sampling was missed, e.g. from Covid-19 lock-downs.

The new **R**code is in the files `JSreturnFun2.R` and `JSreturnRun2.R`. They include the modifications needed to allow for a possible missed year of data.

1 The Files

Updated **R** code is in the files `JSreturnFun2.R` and `JSreturnRun2.R`. The Fun2 file contains definitions of the functions needed, and the Run2 file is used for the actual model fitting. Two test data sets, with $K = 10$ and $K = 6$ sampling occasions are in `testdataK10.csv` and `testdataK6.csv` respectively. The smaller data set is more quickly analysed, but for some models the results are unsatisfactory because of too few samples. The larger data set gives more satisfactory analyses, but is slower to run.

2 Details of the Models

There are 24 models fitted in the run file `JSreturnRun2.R`. This run file is designed for copying and pasting commands into R. It stores the results, as well as giving printouts of model fitting results and times taken. To run this file automatically, just modify the line which names the .csv data file to read your own data file, set the constant M, then in **R** type `source("JSreturnRun2.R")`

2.1 The 24 models

- There are eight basic return-time (RET) models, which add return-time parameters τ_r to the Jolly-Seber model. We use the Schwarz-Arnason parameterisation of the Jolly-Seber model (JSSA) which has parameters N (superpopulation), β_j (entry proportion), ϕ_j (survival probability) and p_j (capture probability). The run file includes fitting eight versions of this model, with $\boldsymbol{\beta}$, $\boldsymbol{\phi}$ and $\boldsymbol{p}$ either constant (c) or varying over time (t). Note that "constant beta", written $\beta(c)$, means one value for β_0, to allow for an accumulation of animals already present at the start of the study, with another constant for $\beta_1, \ldots, \beta_{K-1}$ to allow for a steady-state population in which there is a constant number of recruits each year. The recruitment (birth) parameters satisfy the constraint $\sum_{j=0}^{K-1} \beta_j = 1$, as they are the proportions of the superpopulation N first entering the current population at samples $1, 2, \ldots K$ (and the proportions must add to one).

- The eight JSSA models corresponding to the RET models are fitted later in the file. Their likelihoods have been constructed to make them comparable with the RET models, i.e. basically a RET model but with the τ vector simply (1), of length 1 (always return each year). These JSSA models may be compared with the RET models using either an information criterion or a likelihood ratio test, to establish if the data are signalling a delayed return time. A comparison of population parameters will show the amount of positive bias if data sets with delayed returns are wrongly modelled with a JS-type model.
- The are also eight reparameterisations of the models with time-varying entry ($\beta(t)$), where we replace the parameters $\{\beta_j\}$ with the estimated current population sizes $N_1, N_2, \ldots, N_K$. These models yield standard errors for N_j directly from the Hessian matrix.

2.2 Model labels in the R code

The first three letters specify constant (c) or time-varying (t) assumptions for the entry parameters $\boldsymbol{\beta}$, the survival probabilities $\boldsymbol{\phi}$ and the capture probabilities $\boldsymbol{p}$ respectively. After that, an "r" signals a RET model, and an "Nj" signals the alternative parameterisation using the current population.

For example, `tccr` indicates $\{\beta(t), \phi(c), p(c), \tau(r)\}$ while `tccrNj` indicates the N_j version of the $\{\beta(t), \phi(c), p(c), \tau(r)\}$ model.

2.3 Choosing M

The maximum number of non-breeding states, M, must be specified near the start of the run file. Models may be re-fitted with different M values and compared using AIC or BIC, to decide the appropriate maximum number of non-breeding states, as indicated by the data.

3 Fitting the Models

To fit the models, ensure the data files and the fun and run files are in the same directory. Open R, and cut and paste groups of commands from `JSreturnRun2.R` into R. The run file has been annotated to provide information about the model fitting process. Note that early in the file the data set and M must be defined.

For a large data set and slow model fitting, the run file may be modified to run automatically. In the run file enter the data set as a .csv file, set the chosen value of M, then source the file in **R** and leave it to run automatically.

3.1 Interpreting Results

After model fitting, the AIC and BIC table may be constructed, as at the end of the run file. These give model selection (lowest AIC or BIC) provided the same data set is used for all the models.

Alternatively, likelihood ratio tests may be constructed to compare any model with a submodel. Use the change in residual deviance divided by the change in number of parameters as a χ^2 test statistic, with degrees of freedom equal to the change in number of parameters.

Reruns may be done with different M values and AIC used to decide which value of M is chosen by the data to give the best fit.

Also in the run file are the Jolly-Seber models, assuming the animals return every year ($M = 0$). These may be compared with the return-time models using either AIC or BIC, or a likelihood ratio test. The AIC table at the end of JSreturnRun2.R shows that including the return-time parameters for the example data set `testdataK10` strongly supports the inclusion of the return time modification to the Jolly-Seber model.

3.2 Troubleshooting

R will give warnings or error messages about failures to run.

Note: Some models, especially those with time-dependent survival probabilities, phi(t), may give a warning that not all standard errors are estimable. The warning is:

```
"In sqrt(diag(vcovmat)) :  NaNs produced"
```

All the point estimates are valid, as are the confidence intervals which are given in the output. Unavailable standard errors can occur if a parameter is really close to a boundary. With the sturgeon data, some survival probabilities are so close to one that the upper limit of the confidence interval tries to go above one, which is not allowed. It also forces the lower confidence limit to zero, so a usable lower confidence limit is also unavailable. Only the point estimate is valid, but there is no information about the confidence limits.

4 Updates March 2021

In 2020, Covid 19 lock-downs prevented sampling of freshwater sturgeon at spawning time, leaving a gap in sampling years. Fortunately a small modification of the code has provided usable results. It gives very low (essentially zero) estimates of capture probability for the missing year. Be warned that any missing year may give rise to some inestimable parameters, especially with the more complicated models and multiple missing years.

Using this updated **R** code, simply run the programs with zeros for the missing year. Do include the column of zeros in the data matrix, so the survival estimates will correctly count the years elapsed between captures. For any model with p(t), check that the capture probability estimates look right - very low in the missing year.

Background statistical information: The **R** code is treating missing data as **sampling zeros** with such low probability of capture that none were seen in that year because of low search levels. The alternative is to specify **structural zeros**, when the design of the experiment means that there must be zeros in the capture histories for that year. If structural zeros are assumed, modifications to degrees of freedom should be done. This has not been done here, as a complete rewrite of the input-output code would be needed. With sampling zeros, AIC comparisons between competing models are still valid, as they deal with differences of degrees of freedom, not their absolute values. However, p values would need some modification - too many missing years could compromise the results.