Using POPAN-5 to Analyse Banding Data.

A. NEIL ARNASON, Department of Computer Science, University of Manitoba, Winnipeg, MB R3T 2N2 Canada.

CARL J. SCHWARZ, Department of Mathematics and Statistics, Simon Fraser University, Burnaby, BC V5A 1S6 Canada.

Short title: Using POPAN-5

Address of author for correspondence:

A. Neil Arnason
Department of Computer Science
University of Manitoba
Winnipeg, MB R3T 2N2
Canada
fax: (204) 269-9178 tel: (204) 474-6918 email: arnason@cs.umanitoba.ca

NOTE:

This is a semi-final text version; please see the published text cited below for the definitive version.

Summary:

We describe some recent developments in the POPAN system for the analysis of mark-recapture data from Jolly-Seber (JS) type experiments and how this system applies to the analysis of banding data. We discuss some of the extra data requirements of JS studies, which provide estimates of abundance and entry/birth rates, over survival (CJS) studies. We discuss how POPAN implements a unified likelihood approach using a constrained maximisation and show how this differs from a design-matrix approach used in CJS software. We illustrate the application of constraints and covariate models across groups with some examples drawn from the banding literature, including an example with age-class groups and we describe some of the resources in POPAN for carrying out standard goodness-of-fit testing.

Correct Citation:

Arnason, A. N. and Schwarz, C.J. (1999) Using POPAN-5 to analyse banding data. *Bird Study*, **46** (suppl.), S127-168.

USING POPAN-5

Introduction

The proceedings of the last three Euring technical meetings attest to the success of mark-recapture surveys in assessing population dynamics of bird populations. They also show the success of likelihood methods and the AIC criterion for fitting models accounting for survival and capture rates. These methods have been made accessible to biologists through very powerful software, such as programs SURGE[1], SURPH[2], and most recently, MARK[3]. These programs all allow analyses of single or multiple groups of capture histories over multiple sample times. Models of the Cormack-Jolly-Seber type (CJS, defined below) can then be fit with allowance for time or group effects on the survival or capture rates: i.e., rates can be constrained to be equal over some or all sample times, or among groups, or both. Models may also involve individual, group, or time covariate models for these rates. For example body weight at time of tagging is an individual covariate that might be used in a regression equation to explain survival, with or without additional group and time effects. Time-varying covariates, like weather variables, might affect all groups though possibly in different ways. Group covariates take different values in different groups; for example, effort expended in each of several nesting colonies might be used to explain capture rate.

A particular emphasis in bird studies is analyses involving age effects. Bird populations often exhibit age effects on survival and even on capture rates, and are studied using year-class banding methods. That is, birds are banded at known ages, usually in their first year, and sampling is annual. Thus annual age classes advance in lock-step with sample time so that models with age effects on survival and capture have a tractable structure.

The POPAN system brings the same advantages to mark-recapture studies of open populations where the biologist wants to estimate abundance and entry rates in addition to survival and capture rates. The models used in this case are called Jolly-Seber (JS) models. The distinction between the two is this: in CJS survival studies, the models condition on the number of marked animals released at each sample time and are only concerned with the fate of animals after they are first marked. Counts of unmarked captures are not used and estimates apply only to the marked subset. In JS studies, the proportion of marks in a sample must be an unbiased estimate of the proportion of marks in the population. It is this that permits estimation of abundance of the population as a whole and the number of new entries between sample times.

At the last EURING meeting, we described the history of the POPAN system and gave an overview[4] of the operation and capabilities of the then-current version, POPAN-4. This version only supports temporal constraints and time-varying covariate models. The syntax for extending this, in POPAN-5, to constraints across groups and to group covariate models was anticipated and described [4], but the actual capabilities had not been implemented. This has now been done, and in testing and developing the system, we have added a number of capabilities and have used the software to analyse a number of CJS and JS banding data sets.

We expand on the earlier description of the POPAN system, with emphasis on these new capabilities and on the extra data and assumption requirements of JS experiments over CJS experiments. We also describe a new set of test examples that lets users carry out important tests of assumptions and goodness of fit. The ability, in POPAN, to do general likelihood model fitting, including use of group and temporal constraints and covariates, is based on implementation of a general unified model [5]. The constrained likelihood approach of this implementation is quite different from the linear-model/design-matrix approach used in SURGE [1] and MARK [3]. We give further details on how this implementation was done and compare it to the design-matrix approach. We use POPAN on the classic, much-analysed [6,7] European Dipper dataset [8] to show the additional insights possible from a JS analysis. We also give an example of an age-class model.

Brief Overview of POPAN

POPAN is controlled by a command file made up of data manipulation and analysis tasks. Each task is specified by a paragraph, starting with the paragraph name (designated below by uppercase words) followed by a number of sentences of the form *keyword=keyword_values*; where the keywords are various reserved words designating paragraph options, and the values give choices or input data for those options. Reserved words can be shortened to as few as 1 or 2 letters, so we show the minimal part of a keyword in uppercase and may truncate (e.g. ATtribute, or ATtrib, or just AT). Tasks can be classified as performing data manipulation, data analysis, or simulation.

USING POPAN-5

POPAN has powerful data manipulation capabilities that provide for POPAN's a unique top-down approach to data organisation and analysis. Paragraph CREATE produces binary files combining attribute and capture histories by reading raw data (described more fully below) and metadata (descriptions of group attribute codes and sample occasions, supplied by keywords). Extensive checking of the raw data against the metadata helps ensures consistency and correctness for future analyses. The SELECT paragraph is used to select a binary file for subsequent analysis and, possibly, a subset of the histories, based on attribute and history conditions, and a subset of sample times, including abilities to pool samples together and treat as a single sample. Thus in POPAN, you keep all the data on a population together and split out subsets, as needed, for analyses. There is also a LIST paragraph to list histories from binary files in various ways, including as raw or grouped histories and sorted and blocked as required for the Leslie-Carothers test of equal catchability [9].

POPAN's analysis capabilities fit various JS models to the SELECTed data. The ANALYSIS paragraph can carry out any of 32 different "black box" analyses. These include the standard Jolly-Seber (open) model allowing entries ("births") and losses ("death"); various closure models ("birth-only", "death-only"); various time-constant parameter models of Jolly and Dickson [10]; and a non-parametric smoothing method [11] that is particularly useful for long-term monitoring experiments. These models cannot be customised: closure or constancy of rates applies to all sample times or to none. ANALYSIS also includes a general Chi-square test analysis that allows it to be used with a preceding STATISTICS paragraph to do general tests based on a 2-by-2 table of counts. The STATISTICS paragraph provides a very general, but clear and comprehensible syntax for accumulating the counts needed to form estimates and tests. POPAN generates the sufficient statistics for most ANALYSIS tasks automatically, but the usefulness of having a general statistics gathering capability is more evident with testing, as described below.

POPAN provides two further analysis paragraphs for carrying out customised model fits. TEST fits the log-linear models of Cormack [12] allowing some customising by dropping or adding terms to the model. The method only works with limited numbers of samples (k<10) and produces estimates and goodness-of-fit diagnostics, including residuals, but does not provide the se of the estimates. Much more powerful and complete is the UFIT paragraph. UFIT uses the unified model [5] with user-specified constraints, covariates, and covariate models that may be within (temporal) or among groups. Groups are defined, as in SELECT, by logical conditions. UFIT reports the maximised log-likelihood value, *mll*, and the number of restrictions imposed, *r*. This permits computation of likelihood ratio tests and the change in AIC for assessing which of two fitted models better describes the data. The full set of parameter estimates (capture, survival and entry rates) is reported for each sample time, *i*, along with the estimated se and a number of derived parameter estimates (abundance at time *i*, gross entries in *i*, *i*+1 and total net and gross entries).

The SIMULATE paragraph provides a general means of generating replicated, stochastic sampling experiments applied to a population with user-specified demographic rates. It is fully integrated with all the analysis paragraphs, and reports the means and sd over replicates of all statistics and all estimates and their se. Mechanisms can be specified that satisfy, or that violate, assumptions such as homogeneity of rates over individuals. Thus SIMULATE is a powerful tool for investigating precision of sampling plans, robustness of models to assumption failure, and (because testing is carried out by an ANALYSIS paragraph) the power of tests to detect failures.

There is a Windows (3.1 or later) interface, called RUNPOPAN, that makes it easy to construct command and data files. It provides paragraph templates and on-line help files and lets you create and edit paragraphs, or copy and paste from a growing library of examples. An advantage of the command file approach is that these libraries can be developed to perform a specific, re-usable sequence of tasks (paragraphs) more-or-less independently of the data; the user then just switches in another SELECT paragraph to point to the desired data set and the task sequence is re-applied to the new data. Once a command file is composed, RUNPOPAN displays menus that allow the user to Run the code, to browse the resulting Log file where commands are reflected back and errors reported, and to browse the Results file where statistics and estimates are reported. This has the advantage, over purely point-and-click program control, of creating a written, re-usable record of how results were obtained.

USING POPAN-5

Data requirements

In this section we show how POPAN handles the raw data for JS experiments. Their requirements are somewhat more stringent than for CJS experiments. We also use this section to illustrate the advantages of the top-down approach to group definition.

Most data formats for CJS programs are similar to that of program RELEASE[13]. Data from a *k*-sample experiment is provided as a vector of length *k* of 0's (not seen) and 1's (seen). This is followed by a group count for the number of animals in each of the *g* groups that share this history. Symbolically, this can be designated as: $D_1\ D_2 \ldots D_k\ \ c_1\ c_2 \ldots c_g$

For example, if 3 females and 5 males were seen at times 3, 4, and 6, the history would be designated as: 00110100 3 5. Because histories don't need to be unique, individual histories can be denoted using counts, c_i, that are always 0 or 1.

If the animals are removed from the population (lost on capture) at the last capture time (here, at time 6), the count is negated (-3 or -5). This changes the meaning of the trailing D values after the last capture (which must all be 0). In our example, $D_5 = 0$ indicates the animals were alive but not captured, whereas with a loss on capture at t = 6, $D_7 = D_8 = 0$ indicate that the animals were not available for capture. Clearly this distinction is important for estimating capture and recovery rates. In JS experiments, a similar distinction may have to be made for the leading D values before the first capture time. For example, if the data come from a year-class banding experiment, it is known that the animals are not present in the population prior to first capture (here, at time 3). In other situations, the birds may or may not have been present but didn't happen to be captured. The two situations cannot be distinguished from the history but this information is not needed for CJS models because they describe the survival and capture rates of the marked sub-population only. However, for the JS models, it is necessary to estimate the capture rate of the entire population, so these two situations must be distinguished. POPAN does this by accounting for "injections", the opposite of a "loss on capture". Table 1 shows how the POPAN data formats support this, both in fixed-length data format similar to that above, and in a variable length format specifying the list of capture times (useful in experiments with large numbers of samples but low capture rates). Note that an animal could be injected and lost at the same sample time…such animals contribute nothing to the analysis but POPAN must allow for it because this situation can arise when sample times are pooled. All POPAN analyses, except for the Jolly-Dickson models[10], allow for injections and all allow for losses on capture.

Table 1. POPAN-5 data formats and symbolic form for data histories. Symbols (A1, T, C, X1, etc.) can be used in selection conditions; X1, XT are the absolute values of Z1 and ZT. Example formats are for a dataset with 2 attributes (age, coded J, Y or A; nesting sites coded 1, 2, 3) and 7 sample times. The examples are for 37 Young from site 1 that were captured at times 2, 5, and 6, showing how injections and/or losses on capture are encoded. Free format is similar, except attributes must be enclosed in quotes (e.g. 'Y') and FORMAT=FREE is specified in CREATE instead of a FORTRAN format string. See Table 2 for a CREATE paragraph example.

		POPAN format (variable length)	CMR format (fixed length)
		FORMAT= '(F3.0,2A2,I3,1X,A1,7I3)';	FORMAT= '(F3.0,2A2,7I3)'
Injected	Lost on Capture	Symbolic form:	Symbolic form:
		`N A1 A2 T C Z1... ZT`	`N A1 A2 D1 D2 ...       D7`
NO	NO	`37 Y 1  3    2  5  6`	`37 Y 1  0  1  0  0  1  1  0`
NO	YES	`37 Y 1  3    2  5 -6`	`37 Y 1  0  1  0  0  1  2  0`
YES	NO	`37 Y 1  3 *  2  5  6`	`37 Y 1  0 -1  0  0  1  1  0`
YES	YES	`37 Y 1  3 *  2  5 -6`	`37 Y 1  0 -1  0  0  1  2  0`

POPAN designates attributes explicitly using codes that can be used later to split out groups. The example history above could be represented using a single attribute (for Sex, coded, as 'M' and 'F', say) as two history records:

```
3  F   00110100    and    5  M   00110100
```

and Table 1 gives another example involving multiple attributes. The advantage of POPAN's attribute method over the RELEASE format is that it is easy to cross-classify animals by multiple attributes; for example, if birds can be classified by sex (2 values, say M and F), banding site (4 values, say, 1,2,3, and 4), tag type (2 values, say, N and S) and age cohort at first banding (3 values, say, J, Y, A), then there is a large number of ways to define the groups of interest, and the group counts need to be re-assembled for each definition of the groups. POPAN makes it easy to do this, on the fly, using general logical conditions. There are 2 paragraphs where this can be done: in SELECT, the ATtribute keyword is used to select out a subset for subsequent analyses; in UFIT keywords G1, G2, etc. are used to define each group . For example, using the attribute codes from Table 1, we could define a group as the juveniles in nesting area 1 using:

```
G1  = (A1 .EQ. ‘J’ .AND. A2 .EQ. ‘1’) ;
```

or form a group from juveniles and young together and also pool areas 1 and 2 using:

```
G1 = ((A1 .EQ. ‘J’ .OR. A1 .EQ. ‘Y’) .AND. A2 .LE. ‘2’);
```

The ATtribute and Group keyword_values follow the FORTRAN syntax for logical statements (e.g. .LE. is the relational operator ≤) and can involve the symbolic variables in the capture history (X1, XT, T, N in Table 1) as well as the attribute symbols (A1, A2). This means that in year-class banding experiments you can select out the individual year classes using the time of first capture (X1). For example, ATtribute = (X1 .EQ. 3); selects out the 1993 year class in an experiment that began in 1991. Of course, all the histories selected will have D1 = D2 = 0 and so the sample size at these first two times will be zero. When a single year class is SELECTed, ANALYSIS will eliminate the null sample times automatically. As we shall see, when several year classes are analysed together in UFIT, special steps must be taken to deal with the null samples.

Constraint Implementation

UFIT implements a very general constrained likelihood model[5]. The innovation of this model is its use of a super-population model of N animals whose entry is distributed over sample times proportional to the entry rate parameters b_i, where the b_i sum to 1 over $i = 0\ldots k\text{-}1$. The usual birth counts of the Jolly-Seber[14,15] parameterisation are derived as $B_i = N\, b_i$ for $i = 1\ldots k\text{-}1$. The remaining parameters of the model are, as in CJS models, the survival rates: ϕ_i $i = 1\ldots k\text{-}1$ and the capture rates: p_i $i = 1\ldots k$. There are 3k-1 parameters but some constraints must be imposed to resolve identifiability. What is identifiable depends on what further constraints are imposed (see Cooch *et al.*[7] for a thorough discussion of methods) but for the full time-dependent model (p_t, ϕ_t, b_t), b_0 and p_1 are not separately estimable, nor are $\phi_{k\text{-}1}$ and p_k. In UFIT, the user should resolve this by constraining p_1 and p_k to be 1. POPAN automatically constrains the entry rates, b_i, called Birth Proportions in POPAN, to sum to 1.

The model is formulated in terms of the logits of the rate parameters, whose range is from $-\infty$ to $+\infty$ rather than [0, 1] so that parameter estimates can never be inadmissible. This includes the derived parameters such as the net births ($B_i \geq 0$) and population size ($N_i \geq n_i + z_i$ = minimum number alive, where z_i is the number seen before and after time i but not in i). It also makes it easy to translate constraints on the biological parameters into constraints on their logits because the transformation is unique and invertable.

The maximized log-likelihood is obtained by an iterative scoring method with constraints imposed using the Lagrange multiplier method. This means that any constraint, linear or non-linear, can be imposed on any single model parameter or any combination of parameters. The iteration algorithm only needs to be able to evaluate the constraint as a function of the parameter vector θ, and to evaluate the partial derivatives of the constraint $G(\theta)$ with respect to each model parameter. A particularly useful non-linear constraint is forcing survival rates per unit time to be equal when sample intervals δ_i are unequal, for example: $G(\theta) = \phi_1^{1/\delta 1} - \phi_2^{1/\delta 2}$. In practice, POPAN provides a syntax for specifying constraints in terms of the biological parameters, and some non-linear transformations of the parameters, which limits the generality somewhat, but saves the user the trouble of constructing $G(\theta)$ and its derivatives.

Table 2. Example of binary file CREATE and analysis for the two-group (male and female) European Dipper data from Cooch *et al.*[7] The FORTRAN FORMAT makes it possible to read fields in any order. No SELECT is needed after CREATE if data subsetting is not required. The first UFIT paragraph fits the final model (p, ϕ_{fn}, b_{g*t}) of Lebreton *et al.*[6] and the comments (lines beginning with C) show how to modify constraints to fit the other 2 nested birth models: b_t and b.

```
  CREATE:
     NAME = 'European Dipper data from Cooch et al., Males and Females';
C  Seven equally spaced (annual) sample times using grouped history counts (all are 1)
     BEGIN = 1;   END= 7;   SVALUE = (1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0);
     IDENT = GROUPED ;    HIST = FULL ;
C Card layout is history (D) vector, then history count, then sex attribute code
     INPUT = CARDS;   FORMAT = '(T9, F2.0,1X,A1,T1, 7I1)' ;
C Read data with 1 attribute (sex) and give  coding for the (2) values
     ANUM = 1(2);    ALIST = SEX ;    AVALUES = SEX (M) 'MALE'  (F) 'FEMALE' ;
     SAVE = ASIS;   DATASET = 'eurtest\ed.bin';/
1111110  1 M
1111100  1 F
1111000  1 M
1111000  1 F
   :
0000001  1 F
$ENDDATA

  UFIT:
     TITLE='First model is the final model of Lebreton et al' ;
     NGROUPS = 2 ;  LSEL = 7;  ANALYSIS = 4 ;
C Group 1 is males,  Group 2 is everything else (females)
     G1 = (A1 .EQ. 'M') ;   G2 = OTHER ;
C Flood covariate is same for both groups, so give as a vector of length k=LSEL
     C1 = (0,1,1,0,0,0,0) ;
C Capture Probability CONSTrained to be the same over all groups and times
     CPCONST = CONSTANT ;
C  Survival Probability CONSTrained to follow a linear model in C1 with
C      same intercept and slope in all groups
     SPCONST = LOGITP - (G1:G2C0, G1:G2C1) ;
C  Birth Proportion CONST   no constraint gives model g * t  (group and time effects)
C  then try model t  (time effect) by uncommenting next line
C     BPCONST = TEffect ;
C  then no time or group effects by uncommenting this next line
C     BPCONST = CONSTANT ;
;/
```

Table 3. Syntax for global and detailed constraints in UFIT. xPCON means the syntax applies to constraints on the p_i (CPconstraint), the ϕ_i (SPconstraint) or the b_i (BPconstraint). yPCON applies to SPCON and BPCON only. The detailed constraints are given for a situation with g=2 groups and k=5 sample times. Detailed additive models are created using dummy covariates (see Chapter 6 of Cooch *et al.*[7]).

Model	**Global Constraint**	**Detailed Constraint**
t * g	xPCON = NOne	(default)
g	xPCON = GEffect	CPCON = (G1P1:P4-G1P5)(G2P1:P4-G2P5)
		yPCON = (G1P1:P3-G1P4)(G2P1:P3-G2P4)
t	xPCON = TEffect	CPCON = (G1P1:P5 - G2P1:P5)
		yPCON = (G1P1:P4 - G2P1:P4)
	xPCON = COnstant	CPCON = (G1P1:P4-G1P5)(G2P1:P5-G1P5)
		yPCON = (G1P1:P3-G1P4)(G2P1:P4-G1P4)
•	yPCON = CLosed	SPCON = (G1P1:P4 - 1)(G2P1:P4 - 1)
		BPCON = (G1P1:P4 - 0)(G2P1:P4 - 0)
g + t	xPCON = LNpArallel	additive model on log scale
	xPCON = LParallel	additive model on logit scale
	xPCON = PRoportional	additive model on natural scale

The iterative technique does not use individual histories (and hence cannot use individual covariates), but constructs the likelihood using a set of sufficient statistics which, in POPAN, are generated along with initial parameters by an internal call to an ANALYSIS routine. This gives the user multiple starting points for testing the stability of the iterative solution. This extends readily to multiple groups: the statistics arrays are generated within each group (defined divisively within UFIT by keywords G1=, G2=, ...etc. whose syntax is the same as the ATTRIBUTE keyword in SELECT...see Table 2 for an example) and the parameter vector θ is expanded to $g\,(3k - 1)$ elements. The log-likelihood is the sum of the log likelihoods from each group, and Lagrange constraints can be added to reflect constraints within or among groups.

The syntax for imposing constraints is summarised in Table 3. All constraints upon the parameters are specified using the same syntax whether applied to Capture Probabilities (CPconstraint=) or Survival Probabilities (SPconstraint=) or Birth Proportions (BPconstraint=). The keyword_value syntax consists of a set of contrasts involving a Group index (if there is more than 1 group) and a Parameter index for the sample time. For example, SPcon = (G1P1-G2P5) constrains the ϕ_1 in Group1 to be equal to ϕ_5 in group 2. Specifying the keyword ADjust = YEs allows for constraints on a per-unit time basis by applying the 1/ δ_i transformation using sample time spacings stored in the binary file (keyword SValue in Table 2). You can also override these values by specifying a covariate (e.g., ADjust = C1) . The covariate can be a time covariate (length k) or a group covariate (length gk) ; this latter case allows for physically separated groups, such as nesting sites, that were not sampled on the same days.

Constraining parameters to numeric values is done through a similar syntax; e.g. CPcon = (G1P1-1); fixes p_1 in Group 1 to the value 1. Ranges can also be used to fix several values at once: e.g. BPCON=(G1P1:P6 - 0) constrains the birth parameters in group1 at sample times 1 through 6 to the value 0. These constraints are particularly important in JS models for imposing selective closure involving no births (b_i = 0) or no deaths (ϕ_i = 1). These constraints are also important for resolving non-identifiability. POPAN-5 adds the ability to handle non-identifiability resulting from null sample size, $n_i = 0$. POPAN-4 fails if this condition occurs but, because POPAN-4 doesn't allow groups, one can simply change the SELECT to OMit the null sample time. However, with multiple groups in POPAN-5, eliminating the sample time may discard a non-zero sample in another group, and, as we have seen, age-class models necessarily involve null samples for all age classes after the first. For open (birth and death) models, with $n_i = 0$, you can only estimate the survival product $\phi_{i-1}\ \phi_i$ and the birth sum $b_{i-1} + b_i$. You must constrain ϕ_i to 1 and b_i to 0. In addition, for numerical reasons, you must constrain p_i to 0.

Covariate models are also easy to specify as constraints. The user lists the temporal (length k) or group (length gk) covariates involved, and then specifies that the parameter (or its logit or its $1/\delta_i$ transform) be expressed as a linear combination of these covariates. POPAN allows up to 9 covariates specified by the keywords C1=, C2=, etc. Covariate regression models for a parameter or its logit are then specified as being composed of terms that may include an intercept (C0), a particular covariate (e.g. C2) or the product of two covariates (e.g. C12 or C11). Coefficients for each term can be constrained equal across groups be preceding the term with a group range (Table 2 gives an example). For the simple linear model in 1 covariate and 2 groups we can list all 4 cases, using the survival parameter as an example:

1. SPcon = Logitp - (C0, C1); fits the model: logit $(\phi_i) = \beta_0 + \beta_1\ \mathrm{C1}_i$ with different β coefficients in each group; You can specify ADjust=YEs to apply the $1/\delta_i$ transform: i.e. to fit the model logit$(\phi_i^{1/\delta_i}) = \beta_0 + \beta_1\ \mathrm{C1}_i$.
2. SPcon = Logitp - (G1:G2C0, C1); constrains the intercepts, β_0 , to be equal in the 2 groups;
3. SPcon = Logitp - (C0, G1:G2C1) constrains the slope, β_1, to be equal; and
4. SPcon = Logitp - (G1:G2C0, G1:G2C1), as used in Table 2, constrains both, giving a common covariate response model across groups.

We described elsewhere[5] how the covariate models are transformed into constraints on the parameters to permit estimation of the coefficients and their se within the framework of the constrained likelihood used by the unified model.

The design-matrix approach taken by most other CJS programs gives a less direct specification of covariate models. For example, the design matrix for the European Dipper with Flood covariate and unequal slope and intercept for males and females (Cooch *et al.*[7]) requires a design matrix with 3 columns, the first for the SEX group (1 1 1 1 1 1 0 0 0 0 0), the second for the FLOOD dummy variable (0 1 1 0 0 0 0 1 1 0 0) to mark years 2 and 3 as flood years (within each sex group), and the interaction term (their product). Thus males (first 6 rows) have the covariate model:

$\phi_i = \beta_0 + \beta_2$ FLOOD$_i$ whereas females have the model:

$\phi_i = (\beta_0 + \beta_1) + (\beta_2 + \beta_3)$ FLOOD$_i$. This example shows that it is not directly obvious from the design matrix which coefficients are equal, and that the group-specific β coefficients and their se are not always directly obtained. Moreover, constructing the design matrix can get very complex with more than 2 groups and more than 1 covariate and it may have to be re-constructed when new restrictions are placed on the coefficients. The POPAN approach (Table 2) is simpler and more direct and extends easily to higher numbers of groups and covariate terms.

Users need a quick way to specify constraints globally to all times and/or groups. These are standard models that are useful as first-run screenings of a dataset. POPAN-5 now provides keyword-values in UFIT to do this. In Table 3 we show these and the equivalent detailed syntax that the user could modify to apply the constraints selectively. We also added the keyword_values PArallel and PRoportional for the additive models *g*+*t* on the logit- and log-transformed scale, respectively: the constraint is imposed as $T(G_jP_i) - T(G_{j+1}P_i) - \alpha_{j+1}$ where $j = 1...g-1$ and T is the transform and α is the additive constant parameter, giving the offset between the parameters in one group over the first group.

Other recent changes to POPAN-5

Model selection tools like AIC and LRT need reliable counts of the number of identifiable parameters. POPAN-5 determines the number of identifiable parameters by using the Singular Value Decomposition method to invert the Hessian matrix in the iterative maximisation procedure. This makes the iteration robust to redundant constraints and helps to identify which parameters are involved. POPAN-4 simply failed with a ‘singular matrix’ error in the presence of redundancies but POPAN-5 will continue iterating and prints out the number of singularities. To compare 2 models, say model A and model B, the 2 models would be fit using separate UFIT tasks. Each will report the maximized log likelihood (mll$_i$), number of restrictions (rest$_i$), and number of singularities (sing$_i$), for i = A,B. If B is a submodel of A, you can test if B is a significantly worse fit than A by computing

$$\mathrm{LRT} = 2(\mathrm{mll}_A - \mathrm{mll}_B) \text{ and } df = (\mathrm{rest}_A - \mathrm{sing}_A) - (\mathrm{rest}_B - \mathrm{sing}_B)$$

and assessing the significance of LRT as a χ^2 variate with df degrees of freedom. Similarly the change in AIC, which does not require that B be a submodel of A, is computed as

$$\Delta\mathrm{AIC} = -\mathrm{LRT} + 2\, df.$$

It is especially important to test for assumption failures and identify the likely biases they produce because JS experiments require stronger assumptions then CJS experiments and can be more sensitive to failures. The 2 standard tests used in both the JS context[16] and the CJS context[6,7] are based on tests developed for the RELEASE monograph[13] and subsequently extended by Pradel[17]. One test has two components called, 2.Ct and 2.Cm, that are geared toward detecting capture heterogeneity; the other has two components 3.Sr and 3.Sm related to survival heterogeneity. POPAN-5 is distributed with a test suite of example tasks that will produce all 4 of these tests. Arnason & Schwarz[4] give an example of running one of these tests. All 4 tests are based on animals known to be alive in the population so if the user knows there is closure (e.g., no deaths), the tests should be modified to reflect this. This can’t be done with “black box” procedures, but POPAN allows it by giving the user complete control over the definition of cell statistics. The examples in the test suite are commented to show how to make these changes using the STATISTICS paragraph.
test

Experiences in applying POPAN-5 to banding data.

Modeling the European Dipper birth component : The final model (p , ϕ_{fn}) adopted by Lebreton *et al.* [6] for the European Dippers has the same capture rate and response of survival to the flood/no flood covariate for males and females. As a JS model, this is (p , ϕ_{fn} , b_{g*t}), and fitting this model (Table 2) gives identical estimates to those reported in Lebreton *et al.*[6] but POPAN also reports population sizes and annual numbers of new recruits to the breeding population.

Table 4. Results of fitting restricted recruitment models to the European Dipper data. Model Selection Criteria statistics, including the number of identifiable parameters (np), the maximum log likelihood (mll) and change in AIC and estimates for time-constant parameters are given in (a) with se in parentheses below. Estimates of new recruits (B) and abundance (N) are given in (b) for each sex for the 2 time-varying models (g*t and t). The se for N and B were all between 3.0 and 6.0 and so are not reported here.

(a) Model Selection Criteria statistics and time-constant parameters

Model	np	-mll	ΔAIC	ϕ_f	ϕ_n	p	B(m)	B(f)
(p, ϕ_{fn}, b_{g*t})	15	598.6	16.6	0.496	0.607	0.900	---	---
				(0.043)	(0.031)	(0.024)		
(p, ϕ_{fn}, b_{t})	11	599.2	9.8	0.469	0.607	0.900	---	---
				(0.043)	(0.031)	(0.029)		
(p, ϕ_{fn}, b)	4	601.3	0	0.470	0.605	0.902	22.8	24.8
				(0.045)	(0.031)	(0.029)	(0.71)	(0.76)

(b) Time varying parameter estimates

	B(m)		B(f)		N(m)		N(f)	
year	g * t	t	g * t	t	g * t	t	g * t	t
1	21.4	23.1	31.6	29.5	13.3	19.3	11.1	10.4
2	26.7	26.5	28.5	28.7	29.5	31.8	38.3	35.9
3	23.1	22.6	24.2	24.6	40.6	41.4	46.4	45.6
4	22.9	20.4	19.6	22.1	42.2	42.1	45.9	45.9
5	24.1	23.2	24.3	25.2	48.6	46.0	47.4	50.0
6	17.3	18.9	22.9	20.9	53.5	51.1	53.1	55.5

Further restricting the birth parameter, model b_t permits testing if the relative recruitment pattern over time is the same for both sexes, or equivalently, that the sex ratio of new recruits is constant over time. Model *b* permits testing if recruits per year is constant for each sex. Because the total number of new animals in each group (N) is unconstrained, both restricted models allow an unequal sex ratio (females appear to be favoured slightly, at 52.0% of the new recruits). Results and parameters are reported in Table 4: clearly none of the LRT between any of the models is significant, and the last model has the lowest AIC. The data seem to support a fixed number of recruits per year, even in flood years (i = 2, 3), at a rate that has caused steady growth in the size of the breeding population.

Analysing age cohort classes: We use a simulated set of CJS data, file F_AGE.REL from Chapter 7 of Cooch *et al.*[7] This represents 7 years of releases of birds banded as juveniles. Over the same 7 years the recaptures of previously banded birds are recorded. Table 5 shows how to CREATE the data from the RELEASE data file: no changes are needed to the raw data because the data form a single group. To illustrate the ease of subsetting, we select out the first 3 cohorts over the first 6 sample times for analysis (SELECT task in Table 5). Within each of the 3 cohorts, birds move from juvenile to adult status in one year. We will fit a model allowing for a difference in survival between the two age classes (juveniles and adults), with possible differences among cohorts and with capture rate time-dependent but common to all cohorts (regardless of age class). In the usual CJS notation, we are fitting model $(\phi_{a(2)}, p_t)$.

The 3 cohort groups are defined in UFIT at the line starting with NGROUPS in Table 5. The chief problem is that, in POPAN, all groups must have the same number of sample times, so the null samples at time 1 in Cohort 2 and at times 1 and 2 in Cohort 3 have to be dealt with using constraints. Table 6 shows the equivalence of the age- and time-specific parameters in the usual CJS array with the POPAN parameters.

Table 5. Example of creating a file from RELEASE format data and carrying out an age-class constrained model fit on a sub-set of the age cohorts and sample times.

```
CREATE:
    NAME = 'Females..F_AGE.REL data from Cooch et al.';
    INPUT = CARDS;  FORMAT = '(T15, F3.0, T1, 7I1)' ;
    BEGIN = 1;   END= 7;   IDENT = GROUPED ;  HIST = FULL ;
    ANUM = 0;  SVALUE = (1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0);
    SAVE = ASIS;  DATASET = 'eurtest\fage.bin';/
1111111       2;
1111110       4;
1111010       1;
  :
0000100     135;
0000011      30;
0000010     155;
$ENDDATA

  SELECT:
    TITLE = 'Selecting first three year classes and first 6 times...' ;
    INPUT = 'eurtest\fage.bin';
    ATTRIB = (X1 .LE. 3) ;     END = 6 ;/

  UFIT:
    TITLE = 'Fit model PHI(Age(2)), P(t) ' ;    LSEL = 6;
C Start with a Jolly-Dickson constant p and phi model
    ANALYSIS = 12 ;
C 3 year-class cohorts
    NGROUPS = 3 ;   G1 = (X1 .EQ. 1); G2 = (X1 .EQ. 2); G3 = (X1 .EQ. 3);
C Trace iteration cycles, otherwise minimal output
    TRACE = 100000000000000000; OUT = TABLE;  SUPPRESS = BOTH ;
C Constraints for 0 sample times, then non-identifiable first P,
C   then constant P across cohorts at each sample time (last 4 rows)
    CPCONST  =    (G2P1-0)(G3P1-0)(G3P2-0)
                  (G1P1-1)(G2P2-1)(G3P3-1)
                  (G1P3-G2P3)
                  (G1P4-G2P4)(G2P4-G3P4)
                  (G1P5-G2P5)(G2P5-G3P5)
                  (G1P6-G2P6)(G2P6-G3P6) ;
C Constraints for survival...first for 0 sample times,
C   then constrain over times within cohorts (using ranges)
    SPCONST  =    (G2P1-1)(G3P1-1)(G3P2-1)
                  (G1P12:P4-G1P5)(G2P3:P4-G2P5)(G3P4-G3P5);
C Constrain for no births in all groups
    BPCONST  =    CLOSED ;/
```

Table 6. Equivalence of the triangular array of age- and time-specific rates in a CJS experiment with the rectangular array of group- and time-specific rates required by POPAN. Survival rates (a) for the former case are designated by $\phi_J^{(i)}, \phi_A^{(i)}$ for the Juvenile and Adult rates in cohort i, respectively. Below them are the equvalent POPAN parameter ϕ_{ij} for the survival from year j to year j+1 in (cohort) group i. Capture rate equivalences (b) are shown similarly, but do not have cohort superscripts because they are assumed time dependent only. The POPAN parameter p_{ij} gives the capture rate in (cohort) group i in year j.

(a) Survival rate

	Sample times				
	1	2	3	4	5
Cohort					
1	$\phi_J^{(1)}$	$\phi_A^{(1)}$	$\phi_A^{(1)}$	$\phi_A^{(1)}$	$\phi_A^{(1)}$
	ϕ_{11}	ϕ_{12}	ϕ_{13}	ϕ_{14}	ϕ_{15}
2		$\phi_J^{(2)}$	$\phi_A^{(2)}$	$\phi_A^{(2)}$	$\phi_A^{(2)}$
	ϕ_{21}	ϕ_{22}	ϕ_{23}	ϕ_{24}	ϕ_{25}
3			$\phi_J^{(3)}$	$\phi_A^{(3)}$	$\phi_A^{(3)}$
	ϕ_{31}	ϕ_{32}	ϕ_{33}	ϕ_{34}	ϕ_{35}

(b) Capture rates

	Sample times					
	1	2	3	4	5	6
Cohort						
1		p_2	p_3	p_4	p_5	p_6
	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}
2			p_3	p_4	p_5	p_6
	p_{21}	p_{22}	p_{23}	p_{24}	p_{25}	p_{26}
3				p_4	p_5	p_6
	p_{31}	p_{32}	p_{33}	p_{34}	p_{35}	p_{36}

To fit this model, the following constraints must be imposed on the popan parameters:

- $\phi_{21} = 1$, $\phi_{31} = 1$, $\phi_{32} = 1$. These constraints are "artifacts" of the experimental design imposed because cohort i cannot be seen before time i.
- $\phi_{12} = \phi_{13} = \phi_{14} = \phi_{15}$, $\phi_{23} = \phi_{24} = \phi_{25}$, $\phi_{34} = \phi_{35}$ imposed to constrain adult survival to be equal over time within cohorts but different among cohorts.

In a similar fashion, the parameter structure for the capture rates in Table 6b indicates that the following constraints should be imposed:

- $p_{21} = 0$, $p_{31} = 0$, $p_{32} = 0$ because no birds are observed in cohort *i* before time *i*;
- $p_{11} = 1$, $p_{22} = 1$, $p_{33} = 1$ because these are deliberate releases and are not captures sampled from the larger population;
- $p_{13} = p_{23}$, $p_{14} = p_{24} = p_{34}$, $p_{15} = p_{25} = p_{35}$, $p_{16} = p_{26} = p_{36}$ which constrains the capture rates to be equal among cohorts but allows them to vary over time.

The only other consideration is that additional constraints may have to be imposed because of identifiability problems as outlined in Cooch *et al.*[7] These are typically of the form $p_{ik} = 1$ where k is the final capture time. These are not needed in this particular model because the last capture rates are identifiable.

The corresponding UFIT constraint equations are shown in Table 5. The estimates produced by this task are the same as those produced by SURGE (after re-editing all the data to do the equivalent subsetting). The estimated juvenile survival rate for cohort 1 is 0.339 (se = 0.037) while its adult survival rate is 0.846 (0.035). These differ from the corresponding values for cohort 2 of 0.243 (0.035) and 0.775 (0.057) respectively, and from those of cohort 3 of 0.190 (0.031) and 0.991 (0.076) respectively. The capture rates are the same across groups at the same times, as required by model p_t except when constrained to 0 or 1 as noted above. The estimates are, for time 2: 0.733 (0.063); time 3: 0.702 (0.053); time 4: 0.454 (0.049); time 5: 0.678 (0.058); and time 6: 0.692 (0.082). POPAN also returns estimates of the number of each cohort alive at each sample time, but these are of minor interest.

Further simplifications in the model structure can be translated into POPAN constraints by writing out the index matrix and then associating the POPAN parameters with each index following the example above.

Conclusions

The JS model is a more general model than the CJS models typically used to analyze bird population data as it allows the experimenter to estimate abundance as well as survival and capture rates. Few studies have used JS models to analyse bird population data, probably for two reasons. First, until now, CJS model software allowed a wider class of models to be fit to experiments. However with the current release of POPAN, most of the models of interest in CJS experiments can be fit with analogous models in a JS context. As well, additional models that investigate patterns of abundance and recruitment can be fit that cannot be fit with CJS software. Second, problems of inference and effects of assumption violations are less troublesome in survival studies. Definition of the target population estimated is clear: for both survival and capture rates it is the marked subclass. The biologist can then work to ensure that this sub-class is sufficiently representative of the population of interest. With JS models, there must be a well-defined population of roughly equally catchable marked and unmarked animals; differences in capture rate, if any, should largely be accounted for by the cohort and covariate effects, and given equal attributes, marked and unmarked must be equally catchable. This is perhaps why JS models have been particularly successful with fish populations which are confined in a body of water. In many bird population studies, the experiment is conducted on a sub-area of a larger population, where edge effects, transients, and temporary emigration make definition of the target population meaningless. Nesting sites or colonies that are reasonably well confined in space at the sampling times and that are small enough to be sampled (fairly) randomly are better candidates for JS model analysis.

Over the past decade, the sophistication in design and analysis of CJS experiments has increased dramatically - partially as an effect of powerful analysis tools becoming available. With this new release of POPAN, we look forward to a similar increase in the number and sophistication of JS experiments designed to investigate changes in abundance as well as survival.

Software availability

POPAN-5 software is available from the POPAN web site (http://www.cs.umanitoba.ca/~popan) as of late 1997. This includes on-line help that describes the revised syntax for POPAN-5. Revisions of the User's manual[18] and full POPAN manual[19] have been available on the web site since spring of 1998

Acknowledgments

We acknowledge the work of Gord Boyer in developing POPAN-4 and POPAN-5 and of Lai Shar in developing RUNPOPAN. This work was supported by grants from the Natural Sciences and Engineering Research Council of Canada.

References

1. Pradel, R. & Lebreton, J.-D. (1993) *User's manual for program SURGE Version 4.2*. Centre d'Ecologie Fonctionelle et Evolutive-CNRS, Montpellier, France.
2. Smith, S.G., Skalski, J.R., Schlechte, J. W., Hoffman, A., & Cassen,V. (1994) *SURPH.1 Statistical survival analysis of fish and wildlife tagging studies.* Centre for Quantitative Sciences, University of Washington, Seattle.
3. White, G.C. & Burnham, K. P. (1997) Program MARK - survival estimation from populations of marked animals. (*To appear*: Proceedings of this conference)
4. Arnason, A. N. & Schwarz, C. J. (1995) POPAN-4: enhancements to a system for the analysis of mark-recapture data from open populations. *Journal of Applied Statistics*, **22**, 785-800.
5. Schwarz, C. J. & Arnason, A. N. (1996) A general methodology for the analysis of capture-recapture experiments in open populations. *Biometrics*, **52**, 860-873.
6. Lebreton, J.-D., Burnham, K. P., Clobert, J. & Anderson, D. R. (1992) Modeling survival and testing biological hypotheses using marked animals: a unified approach with case studies, *Ecological Monographs*, **62**, 67-118.
7. Cooch, E. G., Pradel, R. & Nur, N. (1996) *A practical guide to mark-recapture analysis*. Centre d'Ecologie Fonctionelle et Evolutive-CNRS, Montpellier, France.

8. Marzolin, G. (1988) Polygynie du Cincle plongeur (*Cinclus cinclus*) dans les côtes de Lorraine. *L'oiseau et la revue Française d'ornithologie,* **58**, 277-286.

9. Carothers, A. D. (1971) An examination and extension of Leslie's test of equal catchability. *Biometrics*, **27**, 615-630.

10. Jolly, G. M. (1982) Mark-recapture models with parameters constant in time. *Biometrics*, **38**, 301-321.

11. Hargrove, J. W. & Borland, C. W. (1994) Pooled population parameter estimates from mark-recapture data. *Biometrics,* **50,** 1129-1141.

12. Cormack, R. M. (1989) Loglinear models for capture-recapture. *Biometrics*, **41**, 385-413.

13. Burnham, K. P., Anderson, D.R., White, G.C., Brownie, C., & Pollock, K. H. (1987) *Design and analysis methods for fish survival experiments based on release-recapture.* Monograph **5**, American Fisheries Society, Bethesda MD.

14. Jolly, G. M. (1965) Explicit estimates from capture-recapture data with both death and immigration - stochastic model. *Biometrika*, **52**, 225-247.

15. Seber, G. A. F. (1965) A note on the multiple-recapture census. *Biometrika*, **52**, 249-259.

16. Pollock, K. H., Nichols, J. D., Brownie, C., & Hines, J. E. (1990) Statistical inference for capture-recapture experiments. *Wildlife Monograph* No. 107, 1-97.

17. Pradel, R. (1993) Flexibility in survival analysis from recapture data: handling trap-dependence. In *Marked individuals in the study of bird population* (ed. J-D. Lebreton & P.M. North), pp 29-37. Birkhäuser Verlag, Basel.

18. Arnason, A. N., Shar, L., & Boyer, G. (1995) *RUNPOPAN: Installation and user's manual for running POPAN-4 on IBM PC microcomputers under Windows 3.1/32S or Windows 95*. Scientific report, Department of Computer Science, University of Manitoba, Winnipeg, ii+33p.

19. Arnason, A. N., Schwarz, C. J., & Boyer, G. (1995) *POPAN-4: A data maintenance and analysis system for mark-recapture data.* Scientific report, Department of Computer Science, University of Manitoba, Winnipeg, viii+267p.