



Future directions

- what lies ahead?

Anne Viallefont - CEFE/CNRS

As you become more and more experienced using SURGE, you'll realize fairly quickly that (1) there are things you cannot do with the current version of the software (tabulated in Chapter 1), and (2) there are things that SURGE forces you to do that you'd rather not be bothered with (like manually counting parameters!). In this chapter, we discuss upcoming changes to SURGE which will, to varying degrees, address these two considerations, as well as several others.

Bill's Revenge (or...SURGE and Windows)

- In the first edition of this book, I suggested that a Windows version of SURGE is currently being developed, and would be available fairly soon. Well, it is here. It is not a full-blown Windows application - merely, it is SURGE recompiled as a Windows program (i.e., makes use of Windows memory management capabilities). However, as noted in several places in the second edition, it is "SURGE with a difference" - automatic parameter counting and calculation of AIC.
- Why not a full-blown true Windows application? As anyone who has written applications for Windows will tell you, it can be a rather daunting task. However, as noted in the first edition, there are several advantages in writing a native Windows application (as opposed to running SURGE in a DOS window). First, a native Windows application will be able to access the enhanced memory management capabilities of Windows ("enhanced" at least relative to DOS). In some senses, with the advent of robust DOS extenders (like Phar Lap, or Rational), this is less of an issue. It is possible to compile SURGE using these extenders so that it can access virtually unlimited memory models, so this alone is no longer much of a reason to write a Windows

version.

- However, a true-Windows version would offer 2 major advantages over the current DOS version: (1) a more intuitive user interface, and one which is increasingly familiar to most users, and (2) the ability to provide a more refined and logical separation of the "logical components" of SURGE. For example, the ability to make changes to the VAR and PAR, or MOD files, simply by access the appropriate window, without having to run through the entire program as the DOS version of SURGE now requires. Model specification, data input, external file manipulations, all can be treated separately. This will be a clear improvement over the current version. Unfortunately, WinSURGE has not fully implemented the Windows interface - for the moment, WinSURGE simply uses Windows to manage memory - no more, no less.
- Of course, to some degree these are only "structural", or even "aesthetic" changes. The more important changes will concern how SURGE works as an analysis tool, particularly in terms of model selection (Chapter 4, 5) and constraints (Chapter 6).

Model Selection - Counting Parameters

- As first discussed in Chapter 4, model selection tools such as Akaike's Information Criterion (AIC) or Likelihood Ratio Tests (LRT) require that you know exactly the number of estimable parameters in the models. Of course, it is also useful to know which parameters are not estimable (e.g., β terms), in order to draw meaningful conclusions from the estimates.
- To facilitate counting the number of estimable parameters, the new WinSURGE includes a numerical routine to compute the number of estimable parameters, and help to identify those which are non-estimable because of "structural problems" in the data (as discussed in Chapter 4). This is particularly important since, as discussed, there can be a difference between the number of parameters which could, in theory, be separately estimated for a given model, and the actual number which can be estimated for a given data set. So some current software (e.g., SURVIV) uses the former approach, which can lead to

overestimation of the number of parameters if there are, in fact, non-identifiable parameters because of problems with the data. The routine implemented in SURGE will attempt as much as possible to deal with this problem.

- Of course, if the number of parameters can be numerically determined, then clearly so too can the AIC be calculated automatically. WinSURGE calculates AIC for each model.
- One potential benefit of this “automatic calculation” of the number of parameters, and the AIC, is that it may be possible to create a “batch” version of SURGE such that all models with a particular structure (e.g., all models with or without group, time and interactions of group and time) can be fitted with one command, and the model with the lowest AIC automatically selected. Such a “batch mode” would significantly aid initial data exploration. Of course, we would not recommend relying exclusively on an “automated all possible models” approach, since this eliminates the thought and process of testing biological hypotheses. This has not yet been implemented, but should be relatively straightforward.

Constraints - adding flexibility

- At present, SURGE allows you to fix free parameters (i.e., parameters that are not included in a constraint) to a specific value. This can be quite useful for some purposes (e.g., if individuals aren’t in the sample area until a certain age after marking as young, then you would want to fix the recapture probability for some ages to be zero). However, you cannot fix biological parameters (survival, recapture) that are constrained (if you think about it for a moment, this should make sense), or the mathematical parameters (intercept, slopes) in the constraint itself.
- However, in the next version of SURGE, it will be possible to fix the mathematical parameters to a given value. As we will see shortly, such constraints, in combination with a different type of link function (the log link), are especially useful to estimate a constant survival rate when intervals between capture occasions are unequal (recall that the current version of SURGE assumes that the intervals between occasions are equal in length).

Choosing the link function in a constraint

- In the current version of SURGE, the only available link functions are LOGIT and IDENTITY. Unless you specify that you want to use the IDENTITY link (which is sometimes useful - for example, to get more precise confidence intervals when the parameter estimate is close to either 0 or 1), the default function used in any parameter estimation is LOGIT. Recall that this function has the advantage of leading to parameter estimates that fall between 0 and 1.
- In the future, SURGE will offer two additional link functions: the **log-log** link function, and the **log** link function.
- The **log-log** link function permits the user to use Cox’s proportional hazards model in capture-recapture analysis. This model is most widely used in the epidemiological control/treatment experiments (generally in “time to death” or “time to failure” analyses). It assumes that the “hazard rate” (i.e., the proportion of individuals dying at a particular instant), varies in time, but is proportional across groups. This framework is quite useful for expressing the differences in survival between groups in a fairly straightforward way, even in the presence of stochastic variation in survival over time.
- Unlike the logit or log-log link functions, the log link does not have the advantage of always yielding parameter estimates that fall on the 0-1 range. However, the log link function does have the great advantage of yielding parameter estimates that are related to each other by a simple proportionality relation. For example, you may find that the survival of females is x% smaller than that of males, or the survival of offspring is y% smaller than that of adults, even when there is a significant “time” effect (i.e., significant variation in survival among years). This is especially useful in estimating the underlying survival rate when time intervals between capture occasions are unequal.
- For example, suppose you capture animals on a regular basis, but that you cannot capture them when it is raining. Suppose you have a particularly wet period during your field season, such that often you want to capture on a particular day, but are unable to, because of the weather. So, perhaps you go out the day after you’d originally planned to, or the day after that. What results is a data set where the intervals

capturing individuals, recapture rate must equal 0!

- While this method works, it is impractical if the number of “missing occasions”, or the degree of “inequality” in intervals, is large. Introducing one or two columns of fictitious dummy variables into the capture histories is one thing - introducing 10-20 is another, and this is not at all an unusual situation in long-term studies with a number of missing occasions.
- The addition of the new log link function in the next version of SURGE will allow users to use a different approach to dealing with unequal intervals between occasions.
- How does this work? The aim is to estimate survival across unequal time intervals, such that survival in interval (i), ϕ_i , is equal to $\phi_0^{l_i}$, where ϕ_0 is the survival per unit time we want to estimate, and l_i is the length of interval (i) (expressed in the same time units). Since $\phi_i = \phi_0^{l_i}$, we can write $\log(\phi_i) = l_i * \log(\phi_0)$.
- This equality can be included in a model by defining a constraint on the survival parameters, with a log link function, and including the l_i values in the VAR file. For example, suppose you had a study with 6 occasions, with the intervals between occasions being 5, 1, 4, 2, and 5 respectively. The VAR file is simply

5	1
\$	
\$	
5	
1	
4	
2	
5	

- With the intercept (α) fixed to zero (which will be possible in the next release of SURGE), the estimated parameter SLOPE n^o1 gives an

estimate of $\beta = \log(\phi_0)m$ from which ϕ_0 , the survival rate per time unit, is easily reconstituted.

What road to take?

- Back in Chapter 1, we briefly noted that while SURGE is an excellent program, it does have some limits, many of which will be “fixed” in the next version (as we’ve noted in this Chapter). One key limitation to earlier versions of SURGE has been addressed in WinSURGE - the automatic counting of parameters.
- We also noted that SURGE is not the only program “on the market”. Many of the first generation of applications, particularly JOLLY and JOLLYAGE, are still fairly widely used. However, both these applications have very real limitations, particularly in terms of model testing. For most purposes, SURGE (or one of the other newer applications - see below) is to be preferred. Program SURVIV (created by Gary White), is a notable exception to the other “early programs”. SURVIV is, for all intents and purposes, a “general model fitting” language, which allows you to build and test models of virtually any design. It is because of this generality that SURVIV is still widely used, especially among biometricians working in the field. SURVIV has been described as “user-hostile”, but if you’re motivated, and especially if you have some background in programming, learning to use SURVIV can pay great dividends, especially if you want to start moving away from the “standard” CJS-type models. Many of the “new ideas” in mark-recapture analysis are being explored with SURVIV. For example, suppose marked individuals in your population temporarily leave your sample area with probability γ_i . If a marked individual is out of the sampler area, then $p=0$. Thus, if $\gamma_i > 0$, then the recapture rate is in fact a function of the true recapture rate (the probability of seeing the marked individual conditional on it being alive *and* in the sample region) and the probability of temporarily emigrating the sample. SURGE (and most other applications) only deal with 2 primary parameters: ϕ and p . You cannot use SURGE (or any other application) to model the temporary emigration problem (with parameters ϕ , p and γ).

However, you can with SURVIV (although it is often NOT a trivial exercise).

- While SURVIV can be used for virtually any kind of modeling, it can be cumbersome to use, and is often extremely slow (the speed with which solutions are reached is generally inversely related to the generality of the optimization algorithm). For all of the models discussed in this book, SURGE is much faster than SURVIV, and is far easier to use.
- To some degree, SURGE represented the end of the first generation, and was the impetus for the next generation of software. SURGE was the first program to fully incorporate a simple and user-friendly means of applying linear models to mark-recapture analysis, perhaps because of the hypothesis-testing orientation of its developers. Much of the “newer” second generation software for mark-recapture analysis, especially programs POPAN, SURPH, and most recently MARK, have taken much of the philosophy of SURGE (and much of the basic structure) as the starting point, and in many cases have provided logical enhancements, in terms of both utility and function, to the current version of SURGE. Each of these new programs differ in some fundamental ways, but by and large there is increasing convergence in the software available for mark-recapture analysis. Each has notable strengths (e.g., MARK can handle both recapture and recovery data, SURGE is very fast and flexible for CJS-type models, POPAN has superior data handling, and can estimate abundance, both MARK and SURPH can handle individual covariates) and weaknesses, many of which will in all likelihood disappear within the next year or so. In future, there is likely to be little obvious reason to choose one application over another, although this may change.
- However, regardless of whether you use SURGE or some other program for your analyses, the underlying logic of the material covered in this guide should be generally useful in your work.



Where can I get these other programs? The easiest way to find these programs is on World Wide Web pages maintained by the author(s). In the following table, the current URLs for some of these programs are listed. Alternatively, you can

access

<http://www.biol.sfu.ca/cmr>

for a generic listing of pointers to these and many other programs.

<i>Program</i>	<i>WWW site</i>
POPAN	http://www.cs.umanitoba.ca/~popan/popan_home.html
MARK	http://www.cnr.colostate.edu/~gwhite/mark/mark.htm
SURPH	http://www.cqs.washington.edu/~surph/surph.shtml
RELEASE	http://www.cnr.colostate.edu/~gwhite/software.html