Software Documentation

GUTS: An $\mathcal{R}$ Package for the Calculation of the Likelihood Function of the GUTS Model

Carlo Albert* Sören Vogel**

6 May 2012

GUTS is a software for the fast calculation of the logarithm of the likelihood of an empirical survival model.

Contents

Preliminaries

This document was created using "LaTeX" and "Sweave" (package Sweave, Leisch, 2002) with $\mathcal{R}$, version 2.15 R Development Core Team (2012). A function is written `function()`, a package is written Package, $\mathcal{R}$ input is marked *`R: input…`*, and $\mathcal{R}$ output is marked `output`. All $\mathcal{R}$ code is set in a framed box.

```
R: version

               _
platform       x86_64-apple-darwin9.8.0
arch           x86_64
os             darwin9.8.0
system         x86_64, darwin9.8.0
status
major          2
minor          15.0
year           2012
month          03
day            30
svn rev        58871
language       R
version.string R version 2.15.0 (2012-03-30)
nickname
```

*EAWAG, 8600 Dübendorf, Switzerland, mailto:carlo.albert@eawag.ch
**University of Zurich, Switzerland, soeren.vogel@uzh.ch

1 Theoretical Background

GUTS Jager, Albert, Preuss and Ashauer (2011) is a model for survival of organisms, exposed to any kind of quantifiable stress. The time-dependent stressor, $C(t)$, is assumed to cause a time-dependent damage, $D(t)$, which is described by the linear differential equation

$$\dot{D}(t) = k_r(C(t) - D(t))\,, \tag{1}$$

where k_r is called *recovery rate*. The damage is the same for all individuals. However, the individuals are assumed to have different thresholds, beyond which the damage increases their probability to die. Thus, the model combines two sources of stochasticity: On the one hand, death is considered a stochastic event, whose probability increases linearly with the damage, once it exceeds a certain threshold. That is, there is stochasticity at individual level. On the other hand, this threshold is assumed to vary stochastically over the population. Thus, there is stochasticity at population level.

The hazard, $h_z(t)$, of an individual with threshold z is determined by the formula

$$h_z(t) = k_k \max(D(t) - z, 0) + h_b\,, \tag{2}$$

where k_k is called *killing rate* and h_b is the *background mortality*. The hazard, in turn, determines the individual's probability to survive until time t, $S_z(t)$, via the linear differential equation

$$\dot{S}_z(t) = -h_z(t)S_z(t)\,. \tag{3}$$

Finally, each individual is assumed to draw its thresholds z from a distribution, $f_{\boldsymbol{\theta}}(z)$, on the positive real axis. Hence, the parameter vector of the model reads as

$$\boldsymbol{\theta} = (h_b, k_r, k_k, \dots)\,, \tag{4}$$

where the additional arguments are supposed to determine the distribution $f_{\boldsymbol{\theta}}(z)$.

Combining equations (2) and (3), we find that the probability for an arbitrarily chosen member of the population to survive until time t is given by the formula

$$S_{\boldsymbol{\theta}}(t) = \int \exp\left(-k_k \int_0^t \max(D(\tau) - z, 0)d\tau - h_b t\right) f_{\boldsymbol{\theta}}(z)dz\,. \tag{5}$$

Let $\mathbf{y} = (y_0, y_1, \dots, y_n)$ denote a time series of survivors, counted at times $(t_0 = 0, t_1, \dots, t_n)$, and set $y_{n+1} = 0$. Then, the logarithm of the likelihood, $f(\mathbf{y}|\boldsymbol{\theta})$, of the model output $\mathbf{y}$ given the parameters $\boldsymbol{\theta}$ is, up to $\boldsymbol{\theta}$-independent terms, given by the formula

$$\ln f(\mathbf{y}|\boldsymbol{\theta}) = \sum_{i=1}^{n+1}(y_{i-1} - y_i)\ln(S_{\boldsymbol{\theta},i-1} - S_{\boldsymbol{\theta},i})\,, \tag{6}$$

where we have set

$$S_{\boldsymbol{\theta},i} = S_{\boldsymbol{\theta}}(t_i)\,, \quad S_{\boldsymbol{\theta},n+1} = 0\,. \tag{7}$$

2 The Algorithm

The calculation of the log-likelihood requires two numerical integrations (see eq. (5)), and has, therefore, two large numbers, N and M. The following algorithm is of the order $\mathcal{O}(N) + \mathcal{O}(M)$. It is based on

the approximation

$$S_i = \int \exp\left[-k_k \int_0^{t_i} \max(0, D(\tau) - z) d\tau - h_b t_i\right] f_{\boldsymbol{\theta}}(z) dz$$
$$\approx \frac{1}{N} \sum_{j=1}^{N} \exp\left[-k_k \Delta\tau \sum_{D_l > z_j} (D_l - z_j) - h_b t_i\right]$$
$$= \frac{1}{N} e^{-h_b t_i} \Big(e^{-k_k \Delta\tau (e_N - z_N f_N)} + e^{-k_k \Delta\tau (e_N + e_{N-1} - z_{N-1}(f_N + f_{N-1}))} + \dots$$
$$+ e^{-k_l \Delta\tau (e_N + \dots + e_1 - z_1 (f_N + \dots + f_1))} \Big), \quad (8)$$

for an ordered sample $z_1 < \dots < z_N$ from $f_{\boldsymbol{\theta}}(z)$, and with $D_l = D(\tau_l)$ on a grid $\tau_0 < \dots < \tau_{M-1}$. The inner sum in the second line extends over all D_l, for which $\tau_l < t_i$, and we have set $\Delta\tau = t_n/M$. Furthermore,

$$e_j = \sum_{z_j < D_l < z_{j+1}} D_l \,, \quad (9)$$

and

$$f_j = \sharp\{D_l | z_j < D_l < z_{j+1}\} \,, \quad (10)$$

for $1 \le j \le N$ (Set $z_{N+1} = \infty$).

The corresponding algorithm for the calculation of (6) reads as follows:

1. Draw N thresholds from $f_{\boldsymbol{\theta}}(z)$ and order them $z_1 < \dots < z_N$.
2. Refine the grid $t_0 < \dots < t_n$ to a fine grid $\tau_0 < \dots < \tau_{M-1}$.
3. Set $i = 0$.
4. Solve eq. (1), for $t_i \le \tau_l \le t_{i+1}$, using equation

$$D_l = D(\tau_l) = D(s_k) e^{-k_r(\tau_l - s_k)} + C_k \left(1 - e^{-k_r(\tau_l - s_k)}\right)$$
$$+ \frac{C_{k+1} - C_k}{s_{k+1} - s_k} \left(\tau_l - s_k - k_r^{-1} + k_r^{-1} e^{-k_r(\tau_l - s_k)}\right), \quad (11)$$

   for $s_k \le \tau_l \le s_{k+1}$.
5. Update (9) and (10), for $1 \le j \le N$. (This can be done in time $\mathcal{O}(1)$, for each D_l.)
6. Calculate S_i using the recursion:

$$F_j = F_{j+1} + f_j \,, \quad (12)$$
$$E_j = E_{j+1} + e_j \,, \quad (13)$$
$$S_{i,j} = S_{i,j+1} + \exp\left(-k_k \Delta\tau (E_j - F_j z_j)\right) \,, \quad (14)$$

   for $j = N-1, \dots, 1$ and with $S_{i,N} = \exp(-k_k \Delta\tau (E_N - F_N z_N))$ and $F_N = f_N$, $E_N = e_N$. Then,

$$S_i = \frac{1}{N} e^{-h_b t_i} S_{i,1} \,. \quad (15)$$

7. Increment i and go to 4.
8. Calculate the log-likelihood function according to equation (6).

3 The C++ Class and Its Methods

The **GUTS** class allows to store the time series of exterior concentrations, $\mathbf{C} = (C(s_0), \ldots, C(s_m))$, the data, i.e., the time series of survivors, $\mathbf{y} = (y(t_0), \ldots, y(t_n))$, parameter values, $\boldsymbol{\theta} = (h_b, k_r, k_k, \ldots)$, of the model and the distribution, $f_{\boldsymbol{\theta}}(z)$, from which the thresholds of the model are sampled. Furthermore, it provides a method to calculate the logarithm of the likelihood (see section 2).

3.1 Fields

The **GUTS** class has no public fields. Modifications of an existing **GUTS** object must therefore be done using setter methods. However, private fields represent the attributes and the state of an object, where the attributes identify one specific experiment and the state holds current values. Of particular interest are the following attributes (due to programming conventions C++ field names may differ from the mathematical notations above):

`C`: vector of (exterior) concentrations $(C_0, C_1, \ldots, C_m)$.

`Ct`: vector of time points of concentrations $(0 = s_0 < s_1 < \cdots < s_m)$.

`y`: vector of survivors $(y_0, y_1, \ldots, y_n)$.

`yt`: vector of time points of survivors $(0 = t_0 < t_1 < \cdots < t_n \leq s_m)$.

`par`: parameter vector ($\boldsymbol{\theta} = (h_b, k_r, k_k, \ldots)$) with the following parameters:

1. background mortality rate (h_b)
2. recovery rate (k_r)
3. killing rate (k_k)

The additional arguments ($par_4 \ldots$) determine parameters of the distribution from which thresholds are sampled. Currently, only the *lognormal* distribution is implemented, and the additional parameters are its *mean* and *standard deviation*. (Note that this differs from the implementation in $\mathcal{R}$ where the parameters denote mean and standard deviation of the *corresponding normal* distribution.) If attribute `dist` (see below) is "empirical", parameters for the distribution are ignored.

`M`: number of grid points on the time axis for the numerical integration (numerical exactness)

`dist`: name of the distribution to sample from (currently implemented "normal", "lognormal", or "empirical")

`N`: number of threshold samples (numerical exactness)

`z`: the actual sample of size `N`, either generated from `dist` with parameters from `par`, or provided as ascendingly ordered positive numeric vector

Note that in the source code each attribute is prefixed with an `m` indicating that this is a member variable set by the corresponding method.

3.2 Methods

4 Implementation in $\mathcal{R}$

5 Usage of the $\mathcal{R}$ Package

(Example: MCMC with GUTS)

As a real-world example we perform a Bayesian parameter inference (with uniform priors) using the survival data of *Gammarus pulex* exposed to *Diazinon* Ashauer, Hintermeister, Caravatti, Kretschmann and Escher (2010).

In these experiments three different exposure patterns (treatments) have been applied. Since we want to use all the data for the parameter inference, we represent the three exposure patterns by three instances of the GUTS class and use the sum of the three respective loglikelihood functions for the Metropolis algorithm.

(Here comes the R code and the plots)

6 Command Line Version

6.1 Options

`--C!` : a vector of doubles holding the concentrations.

`--Ct!` : a vector of doubles holding the concentration time points. The units provided here must also be used with survivor time points (see below).

`--y!` : a vector of integers holding the number of survivors.

`--yt!` : a vector of doubles of survivor time points. The units provided here must also be used with concentration time points.

`--par!` : a vector of doubles holding the parameters.

`--M!` : an integer holding the number of time grid points on the time axis.

`--dist!` : a string holding the name of the distribution to sample from. Can be either "lognormal" or "empirical". Ignored if the z-option is used.

`--z!` : a vector of doubles holding the actual sample.

`--z-file!` : a character string. The string is the name of the file with doubles of values of the sample.

`--title!` : a single line character string holding the title of the experiment.

`-v!` : verbosity. Set to 0 = quiet or 1 = verbose, where 1 means to display intermediate information while running.

`--help, -h!` : display basic usage of the command.

7 Notes on Current Version and Further Development

References

Ashauer, R., Hintermeister, A., Caravatti, I., Kretschmann, A. & Escher, B. I. (2010). Toxicokinetic and toxicodynamic modeling explains carry-over toxicity from exposure to diazinon by slow organism recovery. *Environmental Science & Technology*, *44*(10), 3963–3971.

Jager, T., Albert, C., Preuss, T. G. & Ashauer, R. (2011). General unified theory of survival – a toxicokinetic toxicodynamic framework for ecotoxicology. *Environmental Science & Technology*, *45*(7), 2529–2540. doi:10.1021/es103092a

Leisch, F. (2002). Sweave: dynamic generation of statistical reports using literate data analysis. In W. Härdle & B. Rönz (Eds.), *Compstat 2002—proceedings in computational statistics* (575–580). Heidelberg: Physica Verlag.

R Development Core Team. (2012, April). *R: a language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. Retrieved April 19, 2013, from R Foundation for Statistical Computing: http://www.R-project.org