

README

NCOMS-20-47536

April 6, 2021

Contents

1	Brief introduction	1
2	Configuration	2
3	File structure	3
4	Reading the code in <code>ReprodCodeNCOMS-20-47536.R</code>	3
4.1	General setup	3
4.2	Functions	4
4.3	Bayesian survival trajectory analysis (BaSTA)	4
4.4	Sensitivity analysis of life expectancy and lifespan equality to changes in Siler mortality parameters	6

1 Brief introduction

This README file provides a brief step-by-step tutorial to implement the demo code associated to the manuscript NCOMS-20-47536 "The evolutionary landscape of primate mortality".

The code provides an overview of the analyses used for the paper, including a survival analysis based on the general framework of Bayesian Survival Trajectory Analysis proposed by ColcheroClark2012 and later extended to open populations by Colchero:2016ch and Barthold:2016kvb. The package `BaSTA.ZIMS` is therefore an extension of the package `BaSTA` [1].

The survival analysis uses the parametric form for the mortality function or hazards rate proposed by Siler:1979p7140, given by

$$\mu(x) = \exp(a_0 - a_1x) + c + \exp(b_0 + b_1x), \quad (1)$$

where $a_0, b_0 \in \mathbb{R}$ and $a_1, c, b_1 > 0$ are mortality parameters to be estimated. From the function in Eq. (1), the cumulative hazards is calculated as

$$U(x) = \int_0^x \mu(t)dt, \quad (2)$$

which is then needed to calculate the survival function $S(x) = \exp(-U(x))$, the cumulative distribution function (CDF) for ages at death $F(x) = 1 - S(x)$, and the probability density function of ages at death, $f(x) = \mu(x)S(x)$.

The Bayesian inference on survival allows finding the vector of Siler parameters, $\theta = [a_0, a_1, c, b_0, b_1]$ that best fit the data. With these, the life expectancy is calculated as

$$e_0 = \int_0^\infty S(x)dx \quad (3)$$

while the lifespan inequality (often called Demetrius [2] or Keyfitz [3]) is

$$H = -\frac{\int_0^\infty S(x) \log[S(x)]dx}{e_0}. \quad (4)$$

Thus, the lifespan equality as proposed by Colchero:2016ch, is given by

$$\varepsilon_0 = -\log(H). \quad (5)$$

In the manuscript, we show that the sensitivity of life expectancy to changes in a given Siler parameter $\theta \in \theta$ is

$$e_\theta = \frac{\partial e}{\partial \theta} = \int_0^\infty S_\theta dx, \quad (6)$$

where $S_\theta = \partial S(x)/\partial \theta$, while the sensitivity of lifespan equality to θ is

$$\varepsilon_\theta = \frac{\partial \varepsilon}{\partial \theta} = \frac{e_\theta (1 + H^{-1}) - H^{-1} \int_0^\infty S_\theta U dx}{e}. \quad (7)$$

The code we provide here allows the user to run a survival analysis on a simulated dataset, extract the Siler mortality parameters, and calculate the sensitivities of life expectancy and lifespan equality to changes in mortality parameters, and the resulting gradient vectors.

2 Configuration

1. Configurations:

R version 4.0.3 (2020-10-10)
Platform: x86_64-apple-darwin15.6.0 (64-bit)
Running under: macOS Mojave Version 10.14.6
Running time: 1 minute.

2. Special hardware: None.

3 File structure

The .zip file includes the following files:

- README.pdf: This README file;
- BaSTA.ZIMS 1.0.2.tar.gz: the compressed R package BaSTA.ZIMS;
- ReprodCodeNCOMS-20-47536.R: the demo R code;
- testData.txt: the demo data to test the survival analysis.

4 Reading the code in ReprodCodeNCOMS-20-47536.R

4.1 General setup

You have to initialize the code by pointing it to your working directory, thus replace the following part “Directory where all files are saved” by the path to the directory, say, “ /TestData/”:

```
# Working directory:  
setwd("~/TestData/")
```

Then install the BaSTA.ZIMS and snowfall libraries, the first runs the Bayesian Survival Trajectory Analysis on census data, and the second allows these analyses to run in parallel:

```
# Install BaSTA.ZIMS package:  
install.packages("BaSTA.ZIMS_1.0.2.tar.gz",  
                 repos = NULL, type = "source")  
  
# Install package snowfall for parallel computing:  
install.packages("snowfall")
```

Next, load both libraries to the working environment:

```
# Load libraries:  
library(BaSTA.ZIMS)  
library(snowfall)
```

Next, load the test data for the survival analysis:

```
# Load survival test dataset:  
dat <- read.table("testData.txt", sep = "\\t", header = TRUE)
```

The first lines of the data should look as:

```

> ID Birth.Date Min.Birth.Date Max.Birth.Date
> 1 1 1962-08-08 1962-08-08 1962-08-08
> 2 2 1955-08-23 1955-08-23 1955-08-23
> 3 3 1958-12-31 1958-12-31 1958-12-31
> 4 5 1961-09-23 1961-09-23 1961-09-23
> 5 6 1963-03-23 1963-03-23 1963-03-23
> 6 7 1955-12-10 1955-12-10 1955-12-10
> Entry.Date Depart.Date Depart.Type
> 1 1962-08-08 1963-07-18 D
> 2 1955-08-23 1958-02-19 D
> 3 1958-12-31 1988-10-17 D
> 4 1961-09-23 1995-10-30 D
> 5 1963-03-23 1963-03-26 D
> 6 1955-12-10 1956-09-03 D

```

4.2 Functions

The next code section includes general demographic functions described in Eqs.(1) and (2) and the remaining functions (survival, CDF and PDF), the life expectancy and lifespan equality functions in Eqs. (3)- (5), as well as the functions used to calculate the sensitivities described in the manuscript in Eqs. (6) and (7). We recommend to run this part of the code without changing it. This section starts as:

```

# ===== #
# ===== FUNCTIONS: =====
# ===== #

```

4.3 Bayesian survival trajectory analysis (BaSTA)

The following code runs the Bayesian inference model for survival based on XX

```

# Run BaSTA on test data for Siler model:
out <- bastaZIMS(dat, model = "GO", shape = "bathtub", ncpus = 4, nsim = 4,
                 niter = 21001, burnin = 1000, parallel = TRUE)

```

You can visually inspect the convergence of the MCMC chains as

```

# Plot traces:
plot(out)

```

To visualize the general information of the model outputs you can use function `summary()` as

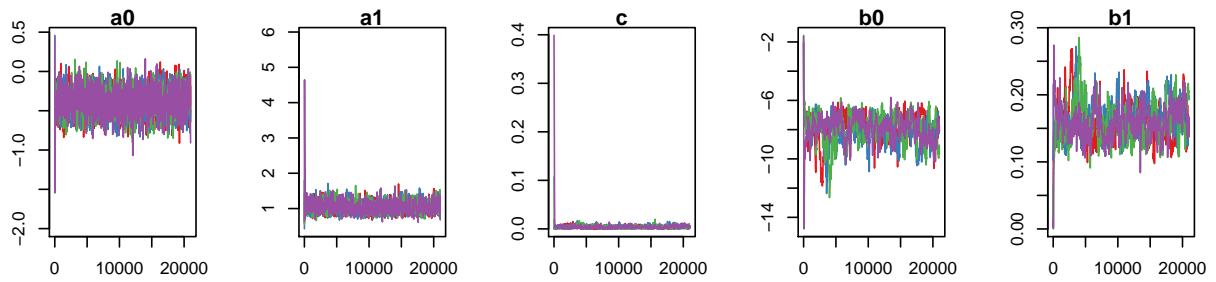


Figure 1: Plot of parameter traces.

```
summary(out)
```

```
>
> Call:
> Model                : G0
> Shape                : bathtub
> Covars. structure    : 0
> Minimum age         : fused
> Cat. covars.         :
> Cont. covars.        :
>
> Model settings:
>   niter  burnin thinning  nsim
>   21001   1000     20     4
>
> Mean Kullback-Leibler
> discrepancy calibration (KLDC):
> KLDC was not calculated due to lack of convergence,
> or because covariates were not included in the model.
>
> Coefficients:
>      Mean      SE      2.5%     97.5%
> a0 -0.389337 0.16296 -0.718337 -0.07578
> a1  1.037312 0.15323  0.770993  1.37296
> c   0.003862 0.00280  0.000179  0.01028
> b0 -7.989301 0.93906 -10.093545 -6.52231
> b1  0.160866 0.02696  0.116838  0.22067
> PotScaleReduc
> a0      0.9997
> a1      1.0006
> c      1.0075
```

```

> b0      1.0146
> b1      1.0149
>
> Convergence:
> Appropriate convergence reached for all parameters.
>
> DIC:
> 907.4019

```

To extract the resulting Siler mortality parameters, just run the following snippet of code:

```

# Extract Siler mortality parameters:
theta <- out$coefficients[, 1]

```

which consist of the vector θ of Siler parameters (rounded to the third digit):

```

>      a0      a1      c      b0      b1
> -0.389  1.037  0.004 -7.989  0.161

```

4.4 Sensitivity analysis of life expectancy and lifespan equality to changes in Siler mortality parameters

Given the mortality parameters, the code uses the functions to calculate life expectancy and lifespan equality as:

```

# Vector of age intervals:
dx <- 0.001
xv <- seq(0, 200, dx)

# Calculate life expectancy:
e0 <- ex(x = xv, dx = dx, th = theta)

# Calculate lifespan equality:
eps0 <- epx(x = xv, dx = dx, th = theta)

```

where xv is a vector of partitions of ages between 0 and 200 years, while dx is the width of each interval, necessary for the approximations of the integrals in Eqs. (3) and (5).

After calculating e_0 and ϵ_0 , the following code provides examples of calculating the sensitivities of e_0 and ϵ_0 to changes in a_1 and b_1 , based on the functions in Eqs. (6) and (7):

```

# Sensitivity of life expectancy to a1:
sensE0a1 <- CalcSensElastPaceShape(x = xv, dx = dx, th = theta, idth = "a1")

# Sensitivity of lifespan equality to a1:
sensEps0a1 <- CalcSensElastPaceShape(x = xv, dx = dx, th = theta, idth = "a1",
                                     lifeEqual = TRUE, sensLifeExp = sensE0a1)

# Sensitivity of life expectancy to b1:
sensE0b1 <- CalcSensElastPaceShape(x = xv, dx = dx, th = theta, idth = "b1")

# Sensitivity of lifespan equality to b1:
sensEps0b1 <- CalcSensElastPaceShape(x = xv, dx = dx, th = theta, idth = "b1",
                                     lifeEqual = TRUE, sensLifeExp = sensE0b1)

```

The code that follows only prints to the console the results:

```

# Print to the console the results:
cat(sprintf("Sensitivity of life expectancy to:\na1 = %s\nb1 = %s\n",
            round(abs(sensE0a1), 2), round(abs(sensE0b1), 2)))

> Sensitivity of life expectancy to:
> a1 = 10.06
> b1 = 500.06

cat(sprintf("Sensitivity of lifespan equality to:\na1 = %s\nb1 = %s\n",
            round(abs(sensEps0a1), 2), round(abs(sensEps0b1), 2)))

> Sensitivity of lifespan equality to:
> a1 = 0.65
> b1 = 4.37

```

The gradient vectors, this is the vectors of change in e_0 and ε_0 as a function of each parameter in θ can be obtained from:

```

# Gradient vector (columns for life expectancy and lifespan equality,
#                  rows for each Siler parameter):
gradVec01 <- cbind(e0 = CalcGradientPS(x = xv, dx = dx, th = theta),
                  eps0 = CalcGradientPS(x = xv, dx = dx, th = theta,
                                         lifeEqual = TRUE))

```

The following code plots the vectors on the e_0 ε_0 space:

Finally, the code provides estimates of the sensitivities of each mortality parameter in θ to changes in life expectancy and lifespan equality, as shown in the manuscript. The resulting vectors of change per parameter are plotted as

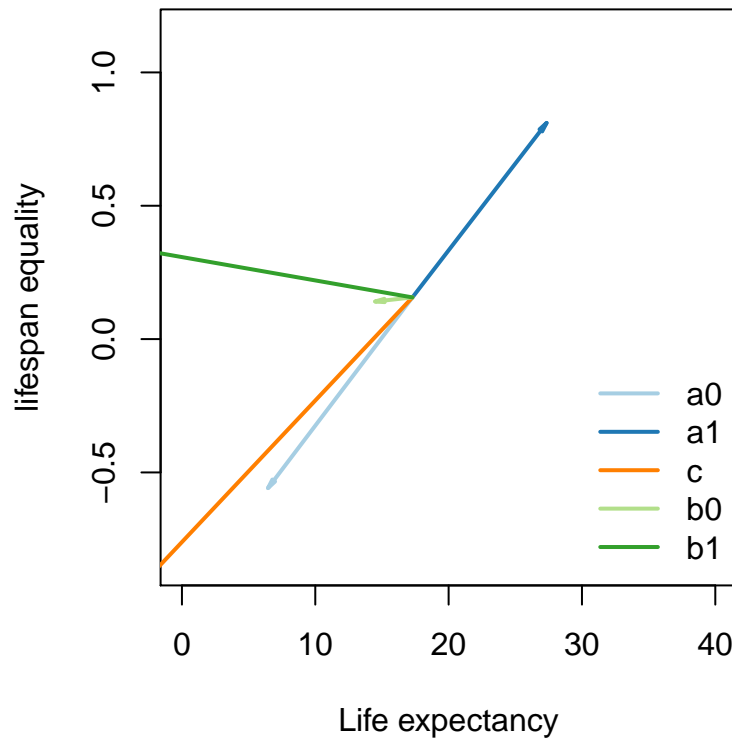


Figure 2: Vectors of change in life expectancy and lifespan equality as a function of each mortality parameter

References

- [1] Colchero, F., Jones, O. R. & Rebke, M. BaSTA: an R package for Bayesian estimation of age-specific survival from incomplete mark-recapture/recovery data with covariates. *Methods in Ecology and Evolution* **3**, 466–470 (2012).
- [2] Demetrius, L. Adaptive value, entropy and survivorship curves. *Nature* **275**, 213–214 (1978).
- [3] Keyfitz, N. & Caswell, H. Applied Mathematical Demography. Springer-Verlag, New York (2005).

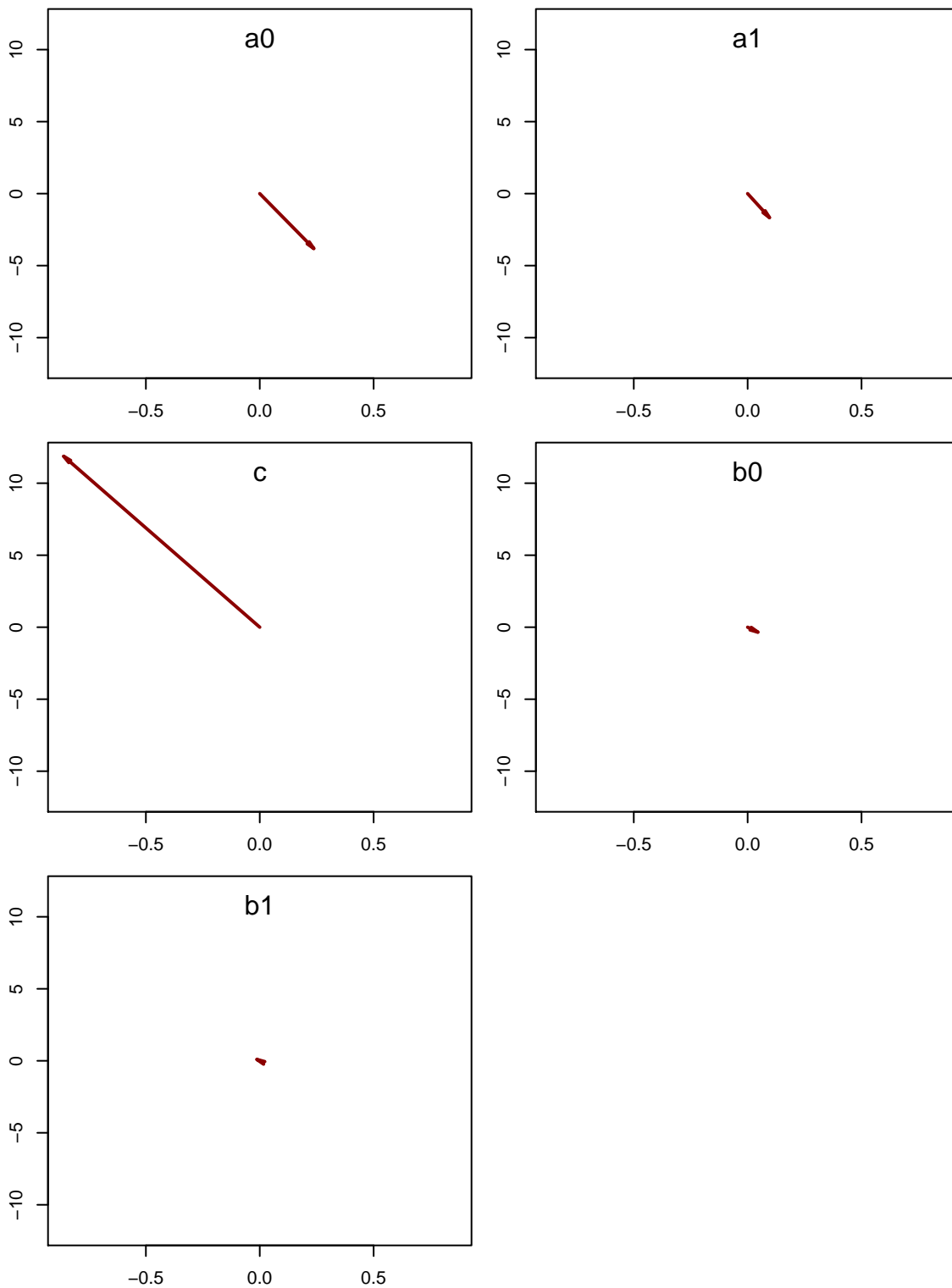


Figure 3: Vectors of change in mortality parameters as a function of changes in life expectancy and lifespan equality