Appendix C: Individual heterogeneity in capture-recapture models - Frequentist approach using E-SURGE

## Introduction

In this appendix, we introduce three methods to cope with individual heterogeneity in capture-recapture models, which we implement in a frequentist framework using maximum likelihood methods. First, we present multistate models in which heterogeneity is measured on individuals using states. Then, we illustrate models with individual random effects and finite mixtures that can help in dealing with hidden heterogeneity. We refer to the paper for a formal presentation of these models and a list of references using them. Throughout this appendix, we use R to simulate data and program E-SURGE is used to fit models. We refer to the relevant literature for an introduction to the use of this program. Note that the frequentist approach can also be implemented with program Mark (appendix A). We refer to this section for comparison of the results we obtain below. Alternatively, the Bayesian approach can be employed (appendix B).

## Multistate models

In this section, we aim at illustrating how not accounting for individual heterogeneity may obscure the detection of life-history tradeoffs. In details, we consider two states for the individuals of our fake population, non-breeding (NB) and breeding (B). To mimic individual heterogeneity, we simulate a bunch of good individuals with survival and and a bunch of bad individuals with survival and . Overall, the cost of breeding on survival should be detected only in bad individuals after accounting for individual heterogeneity through quality. For each group of bad vs. good individuals, we consider the same detection probability , the same transition probabilities between breeding states and , and 100 newly marked individuals for each group in each year of the 6-year study.

### Data simulation

Using R code from [Kéry and Schaub (2012)](http://www.vogelwarte.ch/de/projekte/publikationen/bpa/) book (chapter 9), we first define a function to simulate multistate capture-recapture data:

# Define function to simulate multistate capture-recapture data  
simul.ms <- function(PSI.STATE, PSI.OBS, marked, unobservable = NA){  
 # Unobservable: number of state that is unobservable  
 n.occasions <- dim(PSI.STATE)[4] + 1  
 CH <- CH.TRUE <- matrix(NA, ncol = n.occasions, nrow = sum(marked))  
 # Define a vector with the occasion of marking  
 mark.occ <- matrix(0, ncol = dim(PSI.STATE)[1], nrow = sum(marked))  
 g <- colSums(marked)  
 for (s in 1:dim(PSI.STATE)[1]){  
 if (g[s]==0) next # To avoid error message if nothing to replace  
 mark.occ[(cumsum(g[1:s])-g[s]+1)[s]:cumsum(g[1:s])[s],s] <-  
 rep(1:n.occasions, marked[1:n.occasions,s])  
 } #s  
 for (i in 1:sum(marked)){  
 for (s in 1:dim(PSI.STATE)[1]){  
 if (mark.occ[i,s]==0) next  
 first <- mark.occ[i,s]  
 CH[i,first] <- s  
 CH.TRUE[i,first] <- s  
 } #s  
 for (t in (first+1):n.occasions){  
 # Multinomial trials for state transitions  
 if (first==n.occasions) next  
 state <- which(rmultinom(1, 1, PSI.STATE[CH.TRUE[i,t-1],,i,t-1])==1)  
 CH.TRUE[i,t] <- state  
 # Multinomial trials for observation process  
 event <- which(rmultinom(1, 1, PSI.OBS[CH.TRUE[i,t],,i,t-1])==1)  
 CH[i,t] <- event  
 } #t  
 } #i  
 # Replace the NA and the highest state number (dead) in the file by 0  
 CH[is.na(CH)] <- 0  
 CH[CH==dim(PSI.STATE)[1]] <- 0  
 CH[CH==unobservable] <- 0  
 id <- numeric(0)  
 for (i in 1:dim(CH)[1]){  
 z <- min(which(CH[i,]!=0))  
 ifelse(z==dim(CH)[2], id <- c(id,i), id <- c(id))  
 }  
 return(list(CH=CH[-id,], CH.TRUE=CH.TRUE[-id,]))  
# CH: capture histories to be used  
# CH.TRUE: capture histories with perfect observation  
}

Second, we use this function to simulate the two datasets of good and bad individuals:

set.seed(1) # for reproducibility  
p = 0.9  
R = 100  
#------------------------------  
#---- good quality individuals  
#------------------------------  
# Define mean survival, transitions, recapture, as well as number of occasions, states, observations and released individuals  
phiA <- 0.7  
phiB <- 0.8  
psiAB <- 0.8  
psiBA <- 0.3  
pA <- p  
pB <- p  
n.occasions <- 6  
n.states <- 3  
n.obs <- 3  
marked <- matrix(NA, ncol = n.states, nrow = n.occasions)  
marked[,1] <- rep(R, n.occasions)  
marked[,2] <- rep(R, n.occasions)  
marked[,3] <- rep(0, n.occasions)  
# Define matrices with survival, transition and recapture probabilities  
# 1. State process matrix  
totrel <- sum(marked)\*(n.occasions-1)  
PSI.STATE <- array(NA, dim=c(n.states, n.states, totrel, n.occasions-1))  
for (i in 1:totrel){  
 for (t in 1:(n.occasions-1)){  
 PSI.STATE[,,i,t] <- matrix(c(  
 phiA\*(1-psiAB), phiA\*psiAB, 1-phiA,  
 phiB\*psiBA, phiB\*(1-psiBA), 1-phiB,  
 0, 0, 1 ), nrow = n.states, byrow = TRUE)  
 } #t  
} #i  
# 2.Observation process matrix  
PSI.OBS <- array(NA, dim=c(n.states, n.obs, totrel, n.occasions-1))  
for (i in 1:totrel){  
 for (t in 1:(n.occasions-1)){  
 PSI.OBS[,,i,t] <- matrix(c(  
 pA, 0, 1-pA,  
 0, pB, 1-pB,  
 0, 0, 1 ), nrow = n.states, byrow = TRUE)  
 } #t  
} #i  
  
# Execute function  
sim <- simul.ms(PSI.STATE, PSI.OBS, marked)  
CH <- sim$CH  
his1 = CH[!apply(CH,1,sum)==0,] # remove lines of 0s  
  
#------------------------------  
#---- bad quality individuals  
#------------------------------  
# Define mean survival, transitions, recapture, as well as number of occasions, states, observations and released individuals  
phiA <- 0.7  
phiB <- 0.6  
psiAB <- 0.8  
psiBA <- 0.3  
pA <- p  
pB <- p  
n.occasions <- 6  
n.states <- 3  
n.obs <- 3  
marked <- matrix(NA, ncol = n.states, nrow = n.occasions)  
marked[,1] <- rep(R, n.occasions)  
marked[,2] <- rep(R, n.occasions)  
marked[,3] <- rep(0, n.occasions)  
# Define matrices with survival, transition and recapture probabilities  
# 1. State process matrix  
totrel <- sum(marked)\*(n.occasions-1)  
PSI.STATE <- array(NA, dim=c(n.states, n.states, totrel, n.occasions-1))  
for (i in 1:totrel){  
 for (t in 1:(n.occasions-1)){  
 PSI.STATE[,,i,t] <- matrix(c(  
 phiA\*(1-psiAB), phiA\*psiAB, 1-phiA,  
 phiB\*psiBA, phiB\*(1-psiBA), 1-phiB,  
 0, 0, 1 ), nrow = n.states, byrow = TRUE)  
 } #t  
} #i  
# 2.Observation process matrix  
PSI.OBS <- array(NA, dim=c(n.states, n.obs, totrel, n.occasions-1))  
for (i in 1:totrel){  
 for (t in 1:(n.occasions-1)){  
 PSI.OBS[,,i,t] <- matrix(c(  
 pA, 0, 1-pA,  
 0, pB, 1-pB,  
 0, 0, 1 ), nrow = n.states, byrow = TRUE)  
 } #t  
} #i  
  
# Execute function  
sim <- simul.ms(PSI.STATE, PSI.OBS, marked)  
CH <- sim$CH  
his2 = CH[!apply(CH,1,sum)==0,] # remove lines of 0s

Last, we pool these two datasets together:

his = rbind(his1,his2)

Format the data for analysis in E-SURGE. We use RMark to get a file in the inp format (check out [these notes](https://sites.google.com/site/workshoponcmr/) by Mike Conroy for more details):

k = ncol(his) # nb of capture occasions  
n = nrow(his) # nb of individuals  
out = array(dim=n)  
for (i in 1:n){  
 y = (his[i,] > 0) \* his[i,]  
 out[i] = paste(y,collapse="")  
}  
capt.hist = data.frame(ch = out)  
  
# export  
library(RMark)

## This is RMark 2.2.0

mstrata.processed=process.data(capt.hist,model="Multistrata")  
export.MARK(mstrata.processed, "multistate")

## NULL

### Model fitting

Now we fit a multistate model using the dataset *multistate.inp*: we assume that survival depends on the breeding states, transition probabilities are constant over time, as well as the detection probability. The states are alive in state A, alive in state B, and dead, and the events are non-detected, seen in state A, seen in state B. We use 1 age class. Transition between successive states can be viewed as involving two distinct processes: survival S followed by transition conditional on survival ψ.

We now present the model components in mathematical terms.

First survival, with in rows the states at time t, in columns the states at time t+1:



Now the transitions, with among survivors, transitions between states:



The observation process is driven by:



where the states are in rows, the observations in columns. At initial capture:



Initial states are:



In GEPAT, we have:

Transitions:

Step1 (survival)



Step2 (transition)



Event: 

Initial state: 

In GEMACO, the syntax is:

For Initial State: IS = i

For Transition: S (i.e. step 1) = f T (i.e. step 2) = f.to

For Event: E = firste+nexte

The results are:

piA

Par# 1# IS( 1, 1)( 1, 1)( 1 1) | 0.500000000 0.478100968 0.521899032 0.011180128

SA

Par# 13# S( 1, 1)( 1, 1)( 1 1) | 0.692093902 0.666444442 0.716609698 0.012805506

SB

Par# 14# S( 2, 2)( 1, 1)( 1 1) | 0.699862036 0.679730248 0.719251037 0.010085710

Psi(B->A)

Par# 39# T( 2, 1)( 1, 1)( 1 2) | 0.308345294 0.285689496 0.331962784 0.011810894

Psi(A->B)

Par# 40# T( 1, 2)( 1, 1)( 1 2) | 0.777655122 0.750027275 0.803031347 0.013523467

p

Par# 96# E( 1, 2)( 2, 2)( 1 1) | 0.900413342 0.883776245 0.914898215 0.007921038

AIC = 11615.1

Now we run the same model without a state effect on survival. We just need to amend the GEMACO syntax as follows:

For Initial State: IS = i

For Transition: S (i.e. step 1) = i T (i.e. step 2) = f.to

For Event: E = firste+nexte

The results are:

piA

Par# 1# IS( 1, 1)( 1, 1)( 1 1) | 0.500000000 0.478100968 0.521899032 0.011180128

S

Par# 13# S( 1, 1)( 1, 1)( 1 1) | 0.696844560 0.681337562 0.711916010 0.007802448

Psi(B->A)

Par# 39# T( 2, 1)( 1, 1)( 1 2) | 0.308357742 0.285694711 0.331983005 0.011814729

Psi(A->B)

Par# 40# T( 1, 2)( 1, 1)( 1 2) | 0.777651731 0.750026841 0.803025509 0.013522089

p

Par# 96# E( 1, 2)( 2, 2)( 1 1) | 0.900374454 0.883729698 0.914865964 0.007924680

AIC = 11613.3

Let's add individual heterogeneity through an individual covariate for bad vs. good individuals:

quality=c(rep(0,nrow(his1)),rep(1,nrow(his2))) # 0 for good, 1 for bad

capt.hist$quality=c(rep('good',nrow(his1)),rep('bad',nrow(his2)))  
mstrata.processed=process.data(capt.hist,model="Multistrata",groups = 'quality')

## Warning in process.data(capt.hist, model = "Multistrata", groups = "quality"):   
## quality is not a factor variable. Coercing to factor.

export.MARK(mstrata.processed, "multistate\_cov")

## NULL

Now we fit again the two models from above, including the effect of individual heterogeneity. You first need to read the file *multistate\_cov.inp* with an individual covariate for quality (group here).

In GEMACO:

For Initial State: IS = i

For Transition: S (i.e. step 1) = f.g T (i.e. step 2) = f.to

For Event: E = firste+nexte

The results are:

piA

Par# 1# IS( 1, 1)( 1, 1)( 1 1) | 0.500000000 0.478104493 0.521895507 0.011178326

SA bad

Par# 25# S( 1, 1)( 1, 1)( 1 1) | 0.698644472 0.660832631 0.733938084 0.018674111

SB bad

Par# 26# S( 2, 2)( 1, 1)( 1 1) | 0.578269957 0.547097745 0.608830677 0.015767719

SA good

Par# 50# S( 1, 1)( 1, 1)( 2 1) | 0.687400734 0.652286741 0.720488223 0.017419531

SB good

Par# 51# S( 2, 2)( 1, 1)( 2 1) | 0.800245625 0.775338044 0.823022099 0.012162493

Psi(B->A)

Par# 76# T( 2, 1)( 1, 1)( 1 2) | 0.308126327 0.285494396 0.331719420 0.011798555

Psi(A->B)

Par# 77# T( 1, 2)( 1, 1)( 1 2) | 0.777683687 0.750048031 0.803066097 0.013527034

p

Par# 158# E( 1, 2)( 2, 2)( 1 1) | 0.900017665 0.883411800 0.914487274 0.007909458

AIC = 11494.7

Let us fit the same model without a state effect on survival.

In GEMACO:

For Initial State: IS = i

For Transition: S (i.e. step 1) = g T (i.e. step 2) = f.to

For Event: E = firste+nexte

The results are:

piA

Par# 1# IS( 1, 1)( 1, 1)( 1 1) | 0.500000000 0.478104493 0.521895507 0.011178326

Sbad

Par# 25# S( 1, 1)( 1, 1)( 1 1) | 0.626213649 0.602717237 0.649129396 0.011847691

Sgood

Par# 50# S( 1, 1)( 1, 1)( 2 1) | 0.757126606 0.737215797 0.775987212 0.009892566

Psi(B->A)

Par# 76# T( 2, 1)( 1, 1)( 1 2) | 0.308357754 0.285699775 0.331977526 0.011812035

Psi(A->B)

Par# 77# T( 1, 2)( 1, 1)( 1 2) | 0.777651734 0.750035176 0.803018483 0.013518169

p

Par# 158# E( 1, 2)( 2, 2)( 1 1) | 0.899885314 0.883231523 0.914394116 0.007931599

AIC = 11541.1

Clearly, the inclusion of quality improves the AIC. Also, the model with a difference in survival between breeders and non-breeders is better supported by the data when individual heterogeneity is accounted for.

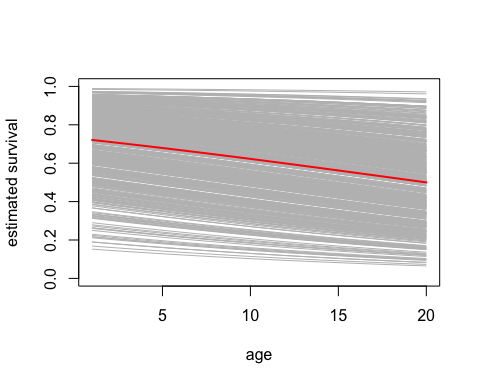
# Models with individual random effects

Here, we aim at illustrating how not accounting for individual heterogeneity may obscure the detection of senescence in survival. More specifically, we consider a single cohort of 500 individuals with survival decreasing as they age over a 20-year study. We also add a frailty for each individual under the form of a normal distribution. Specifically, we specify where . We use , and . If we condition upon the random effect, survival is decreasing as age increases. Note that we consider the same detection probability for all individuals.

### Data simulation

First, we simulate survival for each individual then plot the individual trajectories (in grey) as well as survival conditional on the random effect (in red):

rm(list=ls())  
r = set.seed(3) # for reproducibility  
p = 0.5 # detection  
intercept\_phi = 1   
slope\_phi = -0.05  
sigmaphi = 1  
nind = 500 # nb of individuals  
nyear = 20 # duration of the study  
expit<-function(x){exp(x)/(1+exp(x))} # reciprocal logit function  
z<-data<-x<-matrix(NA,nrow=nind,ncol=nyear)  
first<-rep(1,nind)  
age = matrix(NA,nind,nyear)  
phi = matrix(NA,nind,nyear)  
# simulate age-varying survival for each individual  
for (i in 1:nind){  
 mask <- first[i]:nyear  
 age[i,mask] <- mask - first[i] + 1  
 phi[i,mask] <- expit(intercept\_phi + slope\_phi \* age[i,mask] + rnorm(1,0,sigmaphi))  
}  
plot(age[1,],phi[1,],type='l',col='grey',ylim=c(0,1),xlab='age',ylab='estimated survival')  
for (i in 2:nind){  
 lines(age[i,],phi[i,],type='l',col='grey')  
}  
lines(1:nyear,expit(intercept\_phi + slope\_phi \* 1:nyear),col='red',lwd=2)



Now simulate the encounter histories:

for(i in 1:nind){  
 z[i,first[i]] <- x[i,first[i]] <- 1  
 for(j in (first[i]+1):nyear){  
 z[i,j]<-rbinom(1,1,phi[i,j-1]\*z[i,j-1])  
 x[i,j]<-rbinom(1,1,z[i,j]\*p)  
 }  
}  
his = x  
his[is.na(his)]=0 # remove lines with 0's

Format data

k = ncol(his) # nb of capture occasions  
n = nrow(his) # nb of individuals  
out = array(dim=n)  
for (i in 1:n){  
 y = (his[i,] > 0) \* his[i,]  
 out[i] = paste(y,collapse="")  
}  
capt.hist = data.frame(ch = out)  
  
# export  
library(RMark)  
cjs.processed=process.data(capt.hist,model="CJS")  
export.MARK(cjs.processed, "random")

## NULL

### Model fitting

We fit the model with an age effect but no individual heterogeneity to the simulated dataset.

The states are alive and dead. The events are non-detected (0) or detected (1). Therefore 2 states and 2 events.

In mathematical notation:









* In GEPAT:

Transitions: 

Event: 

Initial state: 

* In GEMACO:

For Initial State: i

For Transition: i+a\*x(1)

For Event: firste+nexte

Age-varying survival

Par# 21# T( 1, 1)( 1, 1)( 1 1) | 0.723806618 0.692784709 0.752813691 0.015325031

Par# 78# T( 1, 1)( 2, 2)( 1 1) | 0.740054200 0.714396361 0.764167931 0.012702308

Par# 132# T( 1, 1)( 3, 3)( 1 1) | 0.755668630 0.733968241 0.776138790 0.010760322

Par# 183# T( 1, 1)( 4, 4)( 1 1) | 0.770635823 0.751157005 0.789018153 0.009659580

Par# 231# T( 1, 1)( 5, 5)( 1 1) | 0.784947084 0.765859458 0.802879155 0.009444050

Par# 276# T( 1, 1)( 6, 6)( 1 1) | 0.798598758 0.778390232 0.817396781 0.009949811

Par# 318# T( 1, 1)( 7, 7)( 1 1) | 0.811591827 0.789302718 0.832024827 0.010895601

Par# 357# T( 1, 1)( 8, 8)( 1 1) | 0.823931475 0.799091944 0.846290673 0.012033680

Par# 393# T( 1, 1)( 9, 9)( 1 1) | 0.835626629 0.808091272 0.859896148 0.013201915

Par# 426# T( 1, 1)( 10, 10)( 1 1) | 0.846689493 0.816503330 0.872684179 0.014307462

Par# 456# T( 1, 1)( 11, 11)( 1 1) | 0.857135087 0.824449835 0.884586663 0.015300853

Par# 483# T( 1, 1)( 12, 12)( 1 1) | 0.866980796 0.832005993 0.895587945 0.016157986

Par# 507# T( 1, 1)( 13, 13)( 1 1) | 0.876245942 0.839220375 0.905703041 0.016869562

Par# 528# T( 1, 1)( 14, 14)( 1 1) | 0.884951384 0.846125970 0.914964922 0.017435065

Par# 546# T( 1, 1)( 15, 15)( 1 1) | 0.893119139 0.852746379 0.923416921 0.017859263

Par# 561# T( 1, 1)( 16, 16)( 1 1) | 0.900772053 0.859099381 0.931108003 0.018150095

Par# 573# T( 1, 1)( 17, 17)( 1 1) | 0.907933486 0.865199038 0.938089724 0.018317340

Par# 582# T( 1, 1)( 18, 18)( 1 1) | 0.914627054 0.871056977 0.944414223 0.018371762

Par# 588# T( 1, 1)( 19, 19)( 1 1) | 0.920876386 0.876683196 0.950132875 0.018324520

p

Par# 653# E( 1, 2)( 2, 2)( 1 1) | 0.496648851 0.468198435 0.525120985 0.014536772

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Index Beta | Lower & Upper 95 percent CI | S.E.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Intercept of the age effect

Beta# 1# | +0.880595669 +0.705870714 +1.055320625 +0.089145385

Slope of the age effect

Beta# 2# | +0.082827304 +0.050223273 +0.115431335 +0.016634710

##############################################################

p on logit scale

Beta# 3# | -0.013404796 -0.127378210 +0.100568618 +0.058149701

##############################################################

Having a look to the parameter estimates, it sounds like the slope of the age effect on survival is estimated positive... Which means that at the population level, whenever individual heterogeneity is ignored, then senescence is completely masked. Even worse, survival is increasing with increasing age.

Now we fit the model with a random effect in the survival process.

* In GEMACO:

For Initial State: i

For Transition: i+a\*x(1) + ind

For Event: firste+nexte

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Index Beta | Lower & Upper 95 percent CI | S.E.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Intercept of the age effect

Beta# 1# | +1.054558008 +0.801511207 +1.307604809 +0.129105511

Slope of the age effect

Beta# 2# | -0.068181499 -0.182575340 +0.046212342 +0.058364205

Beta# 3# | +1.102869039 +0.801536407 +1.404201672 +0.153741139

################### IND.R.E. ESTIMATES UNDER NORM. ASSUMPTIONS

Standard deviation of the random effect

Beta al. ind. # 1# SE | +1.216320118 +0.551659257 +1.880980979 +0.339112684

###############################################################

p on the logit scale

Beta# 4# | +0.016201370 -0.100161997 +0.132564737 +0.059369065

###############################################################

The intercept and slope of the age-survival relationship are quite close to the values we used to simulate the data.

## Models with finite mixtures

Here, we again aim at illustrating how not accounting for individual heterogeneity may obscure the detection of senescence in survival. In contrast with the previous section, we now use finite mixtures to deal with heterogeneity. More specifically, we consider a cohort of 1000 individuals that are split into a group of robust individuals in proportion with constant high survival and a group of frail individuals with survival that senesce over the 20 years of the study according to the relationship . We use , , and . Note that we consider the same detection probability for all individuals.

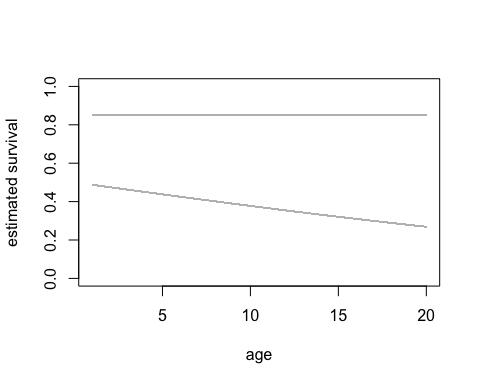
### Data simulation

First simulate data

rm(list=ls())  
r = set.seed(3) # for reproducibility  
p = 0.5 # detection  
prop\_class1 = 0.3 # pi  
phi\_class1 = 0.85 # survival or robust ind  
intercept\_phi\_class2 = 0 #beta\_0  
slope\_phi\_class2 = -0.05 # beta\_1  
nind = 1000 # nb of ind  
nyear = 20 # duration of the study  
expit<-function(x){exp(x)/(1+exp(x))} # reciprocal of the logit function  
z<-data<-x<-matrix(NA,nrow=nind,ncol=nyear)  
first<-rep(1,nind)  
age = matrix(NA,nind,nyear)  
phi = matrix(NA,nind,nyear)  
which\_mixture = rep(NA,nind)  
# simulate age-varying survival for each individual,   
# by first assigning them to the robust or frail class, then using the corresponding   
# survival   
for (i in 1:nind){  
 mask <- first[i]:nyear  
 age[i,mask] <- mask - first[i] + 1  
 which\_mixture[i] <- rbinom(1,1,prop\_class1) # assign ind i to a class with prob pi  
 if (which\_mixture[i] == 1){  
 phi[i,mask] <- phi\_class1 # robust  
 } else {   
 phi[i,mask] <- expit(intercept\_phi\_class2 + slope\_phi\_class2 \* age[i,mask])} # frail  
}

Represent graphically survival over time in the two classes:

plot(age[1,],phi[1,],type='l',col='grey',ylim=c(0,1),xlab='age',ylab='estimated survival')  
for (i in 2:nind){  
 lines(age[i,],phi[i,],type='l',col='grey')  
}



Now simulate the encounter histories:

for(i in 1:nind){  
 z[i,first[i]] <- x[i,first[i]] <- 1  
 for(j in (first[i]+1):nyear){  
 z[i,j]<-rbinom(1,1,phi[i,j-1]\*z[i,j-1])  
 x[i,j]<-rbinom(1,1,z[i,j]\*p)  
 }  
}  
his = x  
his[is.na(his)]=0

Format data

k = ncol(his) # nb of capture occasions  
n = nrow(his) # nb of individuals  
out = array(dim=n)  
for (i in 1:n){  
 y = (his[i,] > 0) \* his[i,]  
 out[i] = paste(y,collapse="")  
}  
capt.hist = data.frame(ch = out)  
  
# export  
library(RMark)  
cjs.processed=process.data(capt.hist,model="CJS")  
export.MARK(cjs.processed, "mixture")

## NULL

### Model fitting

Let's fit two models assuming homogeneity: first one with an age effect, second one with constant survival. The mathematical structure of the model is given in the previous section.

In GEMACO:

For Initial State: to

For Transition: i+a\*x(1)

For Event: firste+nexte

Age-varying survival estimates

Par# 21# T( 1, 1)( 1, 1)( 1 1) | 0.634502002 0.609101468 0.659175569 0.012783734

Par# 78# T( 1, 1)( 2, 2)( 1 1) | 0.664560709 0.643729318 0.684770533 0.010474694

Par# 132# T( 1, 1)( 3, 3)( 1 1) | 0.693342124 0.675555482 0.710571432 0.008935426

Par# 183# T( 1, 1)( 4, 4)( 1 1) | 0.720691941 0.704072792 0.736726761 0.008331994

Par# 231# T( 1, 1)( 5, 5)( 1 1) | 0.746494372 0.729339294 0.762915578 0.008566903

Par# 276# T( 1, 1)( 6, 6)( 1 1) | 0.770671662 0.751918395 0.788406236 0.009309127

Par# 318# T( 1, 1)( 7, 7)( 1 1) | 0.793181957 0.772414467 0.812514367 0.010229179

Par# 357# T( 1, 1)( 8, 8)( 1 1) | 0.814015923 0.791238494 0.834826954 0.011116020

Par# 393# T( 1, 1)( 9, 9)( 1 1) | 0.833192529 0.808633891 0.855163983 0.011860800

Par# 426# T( 1, 1)( 10, 10)( 1 1) | 0.850754381 0.824748161 0.873493398 0.012417022

Par# 456# T( 1, 1)( 11, 11)( 1 1) | 0.866762952 0.839680331 0.889870582 0.012773115

Par# 483# T( 1, 1)( 12, 12)( 1 1) | 0.881293983 0.853505895 0.904401415 0.012936700

Par# 507# T( 1, 1)( 13, 13)( 1 1) | 0.894433255 0.866289150 0.917219527 0.012925645

Par# 528# T( 1, 1)( 14, 14)( 1 1) | 0.906272847 0.878089131 0.928471722 0.012762790

Par# 546# T( 1, 1)( 15, 15)( 1 1) | 0.916907952 0.888962354 0.938308381 0.012472733

Par# 561# T( 1, 1)( 16, 16)( 1 1) | 0.926434262 0.898963963 0.946877123 0.012079847

Par# 573# T( 1, 1)( 17, 17)( 1 1) | 0.934945891 0.908148071 0.954318713 0.011607075

Par# 582# T( 1, 1)( 18, 18)( 1 1) | 0.942533803 0.916567709 0.960764571 0.011075243

Par# 588# T( 1, 1)( 19, 19)( 1 1) | 0.949284670 0.924274621 0.966335432 0.010502719

p

Par# 653# E( 1, 2)( 2, 2)( 1 1) | 0.498743562 0.474462571 0.523030481 0.012399528

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Index Beta | Lower & Upper 95 percent CI | S.E.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Intercept of the age effect

Beta# 1# | +0.419474040 +0.291573267 +0.547374812 +0.065255496

Slope of the age effect

Beta# 2# | +0.132105603 +0.104773345 +0.159437860 +0.013945029

###############################################################

p on the logit scale

Beta# 3# | -0.005025762 -0.102238679 +0.092187155 +0.049598427

###############################################################

AIC = 5602.4

The parameter estimates of the model with constant survival are:

S

Par# 21# T( 1, 1)( 1, 1)( 1 1) | 0.725899443 0.709680514 0.741542274 0.008129685

P

Par# 653# E( 1, 2)( 2, 2)( 1 1) | 0.475812335 0.451522130 0.500217519 0.012432103

AIC = 5715.0

Again, as in the previous section, it's striking to see that survival is increasing when age increases if individual heterogeneity is ignored. In other words, senescence is masked.

Now let's fit a model with heterogeneity - two-finite mixture - in the survival probability, with constant parameters over time. The states are alive class 1, alive class 2 and dead. The events are non-detected (0) or detected (1). Therefore 3 states and 2 events.

In mathematical notation,



Individuals can be allowed to move from a heterogeneity class to the other through the addition of a transition matrix:



The observation process:





The initial states:

;

* In GEPAT:

Transitions:  if no transitions among classes,

and  (step 1: survival)  (step 2: transition) otherwise.

Event: 

Initial state: 

* In GEMACO:

For Initial State: i

For Transition: f if no transitions, f (step 1) and from.to (step 2) otherwise.

For Event: firste + nexte

The results are (model without transition):

Pi

Par# 1# IS( 1, 1)( 1, 1)( 1 1) | 0.748543574 0.653145204 0.824744130 0.043980224

S in class 1

Par# 41# T( 1, 1)( 1, 1)( 1 1) | 0.519903016 0.464158678 0.575156158 0.028432652

S in class 2

Par# 42# T( 2, 2)( 1, 1)( 1 1) | 0.875371408 0.843116523 0.901767410 0.014901018

p

Par# 1094# E( 1, 2)( 2, 2)( 1 1) | 0.507100506 0.482518751 0.531647978 0.012543062

AIC = 5586.7

Obviously, for frail individuals, we miss the age effect to be able to detect senescence. Now let's add age to this model. The syntax in GEMAC needs to be amended as follows:

For Initial State: i

For Transition: from.[i+a\*x(1)]

For Event: firste+nexte

Pi

Par# 1# IS( 1, 1)( 1, 1)( 1 1) | 0.294256551 0.165112317 0.467812500 0.079027988

S in class 1 age 1

Par# 41# T( 1, 1)( 1, 1)( 1 1) | 0.875054721 0.760952848 0.939055475 0.043984603

S in class 2 age 1

Par# 42# T( 2, 2)( 1, 1)( 1 1) | 0.549419208 0.486878288 0.610436543 0.031680222

S in class 1 age 2

Par# 136# T( 1, 1)( 2, 2)( 1 1) | 0.873680874 0.771445476 0.934092071 0.040395671

S in class 2 age 2… and so on

Par# 137# T( 2, 2)( 2, 2)( 1 1) | 0.457972782 0.359820192 0.559499665 0.051627316

Par# 226# T( 1, 1)( 3, 3)( 1 1) | 0.872294124 0.781485444 0.928803259 0.036775417

Par# 227# T( 2, 2)( 3, 3)( 1 1) | 0.369272919 0.214092346 0.557187982 0.090917607

Par# 311# T( 1, 1)( 4, 4)( 1 1) | 0.870894401 0.791039496 0.923195478 0.033139612

Par# 312# T( 2, 2)( 4, 4)( 1 1) | 0.288605164 0.113920623 0.561430205 0.120373358

Par# 391# T( 1, 1)( 5, 5)( 1 1) | 0.869481631 0.800058733 0.917291656 0.029512714

Par# 392# T( 2, 2)( 5, 5)( 1 1) | 0.219428679 0.056839754 0.567339001 0.134577053

Par# 466# T( 1, 1)( 6, 6)( 1 1) | 0.868055744 0.808466542 0.911143364 0.025933967

Par# 467# T( 2, 2)( 6, 6)( 1 1) | 0.163033321 0.027401186 0.573886545 0.134612431

Par# 536# T( 1, 1)( 7, 7)( 1 1) | 0.866616669 0.816136245 0.904853295 0.022468726

Par# 537# T( 2, 2)( 7, 7)( 1 1) | 0.118923874 0.012982355 0.580731224 0.124477476

Par# 601# T( 1, 1)( 8, 8)( 1 1) | 0.865164336 0.822849453 0.898617060 0.019229590

Par# 602# T( 2, 2)( 8, 8)( 1 1) | 0.085528922 0.006098728 0.587726599 0.108703766

Par# 661# T( 1, 1)( 9, 9)( 1 1) | 0.863698678 0.828223841 0.892795622 0.016413068

Par# 662# T( 2, 2)( 9, 9)( 1 1) | 0.060863793 0.002853126 0.594798655 0.090993113

Par# 716# T( 1, 1)( 10, 10)( 1 1) | 0.862219624 0.831624921 0.888003767 0.014344226

Par# 717# T( 2, 2)( 10, 10)( 1 1) | 0.042977390 0.001332033 0.601905266 0.073794596

Par# 766# T( 1, 1)( 11, 11)( 1 1) | 0.860727109 0.832208366 0.885068078 0.013454384

Par# 767# T( 2, 2)( 11, 11)( 1 1) | 0.030178460 0.000621246 0.609020147 0.058433586

Par# 811# T( 1, 1)( 12, 12)( 1 1) | 0.859221066 0.829372485 0.884574801 0.014049219

Par# 812# T( 2, 2)( 12, 12)( 1 1) | 0.021107065 0.000289587 0.616125624 0.045433815

Par# 851# T( 1, 1)( 13, 13)( 1 1) | 0.857701429 0.823294953 0.886333504 0.016034341

Par# 852# T( 2, 2)( 13, 13)( 1 1) | 0.014721066 0.000134949 0.623209051 0.034831448

Par# 886# T( 1, 1)( 14, 14)( 1 1) | 0.856168135 0.814690775 0.889619546 0.019039551

Par# 887# T( 2, 2)( 14, 14)( 1 1) | 0.010246943 0.000062876 0.630260898 0.026409477

Par# 916# T( 1, 1)( 15, 15)( 1 1) | 0.854621120 0.804234323 0.893752500 0.022715244

Par# 917# T( 2, 2)( 15, 15)( 1 1) | 0.007122795 0.000029292 0.637273662 0.019848161

Par# 941# T( 1, 1)( 16, 16)( 1 1) | 0.853060321 0.792365477 0.898290509 0.026832541

Par# 942# T( 2, 2)( 16, 16)( 1 1) | 0.004946395 0.000013645 0.644241217 0.014810879

Par# 961# T( 1, 1)( 17, 17)( 1 1) | 0.851485677 0.779341741 0.902978670 0.031257457

Par# 962# T( 2, 2)( 17, 17)( 1 1) | 0.003432704 0.000006356 0.651158420 0.010987269

Par# 976# T( 1, 1)( 18, 18)( 1 1) | 0.849897128 0.765314906 0.907672877 0.035912276

Par# 977# T( 2, 2)( 18, 18)( 1 1) | 0.002381123 0.000002961 0.658020846 0.008110822

Par# 986# T( 1, 1)( 19, 19)( 1 1) | 0.848294615 0.750380525 0.912290836 0.040750672

Par# 987# T( 2, 2)( 19, 19)( 1 1) | 0.001651151 0.000001379 0.664824620 0.005962504

p

Par# 1094# E( 1, 2)( 2, 2)( 1 1) | 0.502494541 0.477418750 0.527557789 0.012801305

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Index Beta | Lower & Upper 95 percent CI | S.E.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

pi on the logit scale

Beta# 1# | -0.874799776 -1.620671255 -0.128928297 +0.380546673

###############################################################

Intercept of the age effect for class 1

Beta# 2# | +1.958917382 +1.098782490 +2.819052274 +0.438844333

Intercept of the age effect for class 2

Beta# 3# | +0.565155134 +0.040178984 +1.090131285 +0.267844975

Slope of the age effect for class 1

Beta# 4# | -0.012506830 -0.086831557 +0.061817897 +0.037920779

Slope of the age effect for class 2

Beta# 5# | -0.366830798 -0.764287874 +0.030626277 +0.202784222

###############################################################

p on the logit scale

Beta# 6# | +0.009978247 -0.090386484 +0.110342979 +0.051206496

###############################################################

AIC = 5584.4

Seems like we've managed to capture the main patterns in the simulated data.