Simple 3-state Digitized Partitioned Survival model in R

The DARTH workgroup

Developed by the Decision Analysis in R for Technologies in Health (DARTH) workgroup:

Fernando Alarid-Escudero, PhD (1)

Eva A. Enns, MS, PhD (2)

M.G. Myriam Hunink, MD, PhD (3,4)

Hawre J. Jalal, MD, PhD (5)

Eline M. Krijkamp, MSc (3)

Petros Pechlivanoglou, PhD (6)

Alan Yang, MSc (7)

In collaboration of:

1. Drug Policy Program, Center for Research and Teaching in Economics (CIDE) - CONACyT, Aguascalientes, Mexico
2. University of Minnesota School of Public Health, Minneapolis, MN, USA
3. Erasmus MC, Rotterdam, The Netherlands
4. Harvard T.H. Chan School of Public Health, Boston, USA
5. University of Pittsburgh Graduate School of Public Health, Pittsburgh, PA, USA
6. The Hospital for Sick Children, Toronto and University of Toronto, Toronto ON, Canada
7. The Hospital for Sick Children, Toronto ON, Canada

Please cite our publications when using this code:

- Jalal H, Pechlivanoglou P, Krijkamp E, Alarid-Escudero F, Enns E, Hunink MG. An Overview of R in Health Decision Sciences. Med Decis Making. 2017; 37(3): 735-746. https://journals.sagepub.com/doi/abs/10.1177/0272989X16686559

- Krijkamp EM, Alarid-Escudero F, Enns EA, Jalal HJ, Hunink MGM, Pechlivanoglou P. Microsimulation modeling for health decision sciences using R: A tutorial. Med Decis Making. 2018;38(3):400–22. https://journals.sagepub.com/doi/abs/10.1177/0272989X18754513

- Krijkamp EM, Alarid-Escudero F, Enns E, Pechlivanoglou P, Hunink MM, Jalal H. A Multidimensional Array Representation of State-Transition Model Dynamics. BioRxiv 670612 2019.https://www.biorxiv.org/content/10.1101/670612v1

Change `eval` to `TRUE` if you want to knit this document.

01 Load packages

```
if (!require('pacman')) install.packages('pacman'); library(pacman) # use this package to conveniently
# load (install if required) packages from CRAN
p_load("here", "dplyr", "devtools", "gems", "flexsurv", "survHE", "ggplot2", "msm", "igraph", "mstate",
# install_github("DARTH-git/darthtools", force = TRUE) Uncomment if there is a newer version
p_load_gh("DARTH-git/dampack", "DARTH-git/darthtools")
```

02 Load functions

```
# No function needed
```

03 Input model parameters

```
v_names_states <- c("Healthy", "Sick", "Dead")  # state names
n_states  <- length(v_names_states)   # No of states
c_l       <- 1 / 12                   # cycle length (a month)
n_t       <- 20                       # number of years (20 years)
times     <- seq(0, n_t, c_l)         # the cycles in years
d_r       <- 0.03                     # discount rate
set.seed(2009)                        # set the seed

c_H       <- 200                      # cost of remaining one cycle healthy
c_S       <- 500                      # cost of remaining one cycle sick
c_D       <- 0                        # cost of remaining one cycle dead
v_c       <- c(c_H, c_S, c_D)         # store in a vector

u_H       <- 0.75                     # utility when healthy
u_S       <- 0.30                     # utility when sick
u_D       <- 0                        # utility when dead
v_u       <- c(u_H, u_S, u_D)         # store in a vector

v_dw      <- 1 / (1 + d_r) ^ (times)  # discount weight
```

04 Digitized Data

Use the function `digitise()` to translate the digitised OS and PFS data into patient level information.

```
# Create IPD and KM data for the OS curves
digitise(surv_inp   = "OS_Examp.txt",
         nrisk_inp  = "OS_Examp_AtRisk.txt",
         km_output  = "KMdata_OS.txt",
         ipd_output = "IPDdata_OS.txt")

# Create IPD and KM data for the PFS curves (there have been 99 events in the cohort)
digitise(surv_inp   = "PFS_Examp.txt",
         nrisk_inp  = "PFS_Examp_AtRisk.txt",
         km_output  = "KMdata_PFS.txt",
         ipd_output = "IPDdata_PFS.txt")

# Link the IPD files across the two arms of the trial for OS and PFS
IPD_OS  <- make.ipd(ipd_files = c("IPDdata_OS.txt"), ctr = 1,
                    var.labs  = c("time","event","arm"))
IPD_PFS <- make.ipd(ipd_files = c("IPDdata_PFS.txt"), ctr = 1,
                    var.labs  = c("time","event","arm"))
```

05 Analysis

05.1 Partitioned Survival model

```
# fit all parametric models to the data and extract the AIC/BIC
# Select the one with the most appropriate fit
# Repeat for PFS and OS
fit_PFS  <- fit.fun(time = "time", status = "event", data = IPD_PFS, times = times,
                    extrapolate = T)
fit_OS   <- fit.fun(time = "time", status = "event", data = IPD_OS, times = times,
                    extrapolate = T)

# Check AIC of each model to assess goodness-of-fit
GoF_PFS <- data.frame(AIC = fit_PFS$AIC, BIC = fit_PFS$BIC)
GoF_OS  <- data.frame(AIC = fit_OS$AIC,  BIC = fit_OS$BIC)

# "Exponential", "Weibull (AFT)", "Gamma", "log-Normal", "log-Logistic", "Gen. Gamma"
choose_PFS <- rownames(GoF_PFS)[which.min(GoF_PFS$AIC)]
choose_OS  <- rownames(GoF_OS)[which.min(GoF_OS$AIC)]

# construct a partitioned survival model out of the chosen models
m_M_PSM <- partsurv(pfs_survHE = fit_PFS, os_survHE  = fit_OS,
                    choose_PFS = choose_PFS,  choose_OS  = choose_OS,
                    time = times, v_names_states = v_names_states)

# plot the results of PSM and the true data
plot_trace_PSM(time = times, partsurv.model = m_M_PSM, v_names_states = v_names_states)
```

Calculate total cost and QALYs per cycle.

```
v_c_t <- m_M_PSM$trace %*% v_c
v_u_t <- m_M_PSM$trace %*% v_u

tot_c <-  t(v_c_t) %*% v_dw
tot_u <-  t(v_u_t) %*% v_dw
```

```r
# display results
results <- data.frame(Total_Cost = tot_c, Total_QALYs = tot_u)
results
```