PSA: Markov Sick-Sicker model in R

with age-specific mortality

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.

```
rm(list = ls())      # clear memory (removes all the variables from the workspace)
```

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", "scales", "ellipse", "ggplot2", "lazyeval", "igraph", "truncnorm",
# load (install if required) packages from GitHub
# install_github("DARTH-git/dampack", force = TRUE) Uncomment if there is a newer version
# install_github("DARTH-git/dampack", force = TRUE) Uncomment if there is a newer version
# 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 functions needed
```

03 Input model parameters

```
# Strategy names
v_names_str <- c("No Treatment", "Treatment")

# Number of strategies
n_str <- length(v_names_str)

# Markov model parameters
age     <- 25                      # age at baseline
max_age <- 55                      # maximum age of follow up
n_t     <- max_age - age           # time horizon, number of cycles
v_n     <- c("H", "S1", "S2", "D")  # the 4 states of the model: Healthy (H), Sick (S1),
                                   # Sicker (S2), Dead (D)
n_s     <- length(v_n)             # number of health states

# Tunnels
tunnel_size <- n_t
# Sick state
Sick_tunnel <- paste("S1_", seq(1, tunnel_size), "Yr", sep = "")
### Create variables for time-dependent model
v_n_td      <- c("H", Sick_tunnel, "S2", "D")   # state names
n_s_td      <- length(v_n_td)                   # number of states

# Transition probabilities (per cycle) and hazard ratios
# Read age-specific mortality rates from csv file
```

```r
lt_usa_2005 <- read.csv("HMD_USA_Mx_2015.csv")
v_r_HD <- lt_usa_2005 %>%
  filter(Age >= age & Age <= (max_age-1)) %>%
  select(Total) %>%
  as.matrix()

p_HD    <- 1 - exp(- v_r_HD)          # probability to die when healthy
p_HS1   <- 0.15                         # probability to become sick when healthy
p_S1H   <- 0.5                          # probability to become healthy when sick

# Weibull parameters
l       <- 0.08 # scale
g       <- 1.1  # shape
# Weibull function
p_S1S2  <- l*g*(1:tunnel_size)^{g-1} # probability to become sicker when sick
                                     # (time-dependent)

hr_S1   <- 3                            # hazard ratio of death in sick vs healthy
hr_S2   <- 10                           # hazard ratio of death in sicker vs healthy
r_HD    <- - log(1 - p_HD)           # rate of death in healthy
r_S1D   <- hr_S1 * r_HD                 # rate of death in sick
r_S2D   <- hr_S2 * r_HD                 # rate of death in sicker
p_S1D   <- 1 - exp(-r_S1D)           # probability to die in sick
p_S2D   <- 1 - exp(-r_S2D)           # probability to die in sicker

# Cost and utility inputs
c_H     <- 2000                      # cost of remaining one cycle in the healthy state
c_S1    <- 4000                      # cost of remaining one cycle in the sick state
c_S2    <- 15000                     # cost of remaining one cycle in the sicker state
c_trt   <- 12000                     # cost of treatment(per cycle)
c_D     <- 0                         # cost of being in the death state
u_H     <- 1                         # utility when healthy
u_S1    <- 0.75                      # utility when sick
u_S2    <- 0.5                       # utility when sicker
u_D     <- 0                         # utility when dead
u_trt   <- 0.95                      # utility when being treated

# Discounting factor
d_r     <- 0.03                      # equal discount of costs and QALYs by 3%
# calculate discount weights for costs for each cycle based on discount rate d_c
v_dwc   <- 1 / (1 + d_r) ^ (0:n_t)
# calculate discount weights for effectiveness for each cycle based on discount rate d_e
v_dwe   <- 1 / (1 + d_r) ^ (0:n_t)
```

04 Define and initialize matrices and vectors

04.1 Cohort trace

```r
# create the markov trace matrix M capturing the proportion of the cohort in each state
# at each cycle
m_M_notrt <- m_M_trt <- matrix(NA,
```

```r
                                nrow = n_t + 1, ncol = n_s_td,
                                dimnames = list(paste("cycle", 0:n_t, sep = " "), v_n_td))

head(m_M_notrt) # show first 6 rows of the matrix

# The cohort starts as healthy
# initialize first cycle of Markov trace accounting for the tunnels
m_M_notrt[1, ] <- m_M_trt[1, ] <- c(1, rep(0, tunnel_size), 0, 0)
```

04.2 Transition probability array

```r
# create the transition probability array for NO treatment
a_P_notrt <- array(0,                                         # Create 3-D array
                   dim = c(n_s_td, n_s_td, n_t),
                   dimnames = list(v_n_td, v_n_td, 0:(n_t-1))) # name dimensions
```

Fill in the transition probability array:

```r
# from Healthy

# from Sick

# from Sicker

# from Dead

# create transition probability matrix for treatment same as NO treatment
a_P_trt <- a_P_notrt
```

05 Run Markov model

```r
# your turn
```

06 Compute and Plot Epidemiological Outcomes

06.1 Cohort trace

```r
# your turn
```

06.2 Overall Survival (OS)

```r
# your turn
```

06.2.1 Life Expectancy (LE)

```r
# your turn
```

06.3 Disease prevalence

```r
# your turn
```

06.4 ratio of sick(S1) vs sicker(S2)

```r
# your turn
```

07 Compute Cost-Effectiveness Outcomes

07.1 Mean Costs and QALYs for Treatment and NO Treatment

```r
# your turn
```

07.2 Discounted Mean Costs and QALYs

```r
# your turn
```

07.3 Compute ICERs of the Markov model

```r
# your turn
```

07.4 Plot frontier of the Markov model

```r
# your turn
```

08 Deterministic Sensitivity Analysis

08.1 List of input parameters

Create list "l_params_all" with all input probabilities, cost and utilities.

```r
l_params_all <- as.list(data.frame(
  p_HS1   = 0.15,                # probability to become sick when healthy
  p_S1H   = 0.5,               # probability to become healthy when sick
  hr_S1   = 3,                 # hazard ratio of death in sick vs healthy
  hr_S2   = 10,                # hazard ratio of death in sicker vs healthy
  c_H     = 2000,              # cost of remaining one cycle in the healthy state
  c_S1    = 4000,              # cost of remaining one cycle in the sick state
  c_S2    = 15000,             # cost of remaining one cycle in the sicker state
  c_trt   = 12000,             # cost of treatment(per cycle)
  c_D     = 0,                 # cost of being in the death state
  u_H     = 1,                 # utility when healthy
  u_S1    = 0.75,              # utility when sick
  u_S2    = 0.5,               # utility when sicker
  u_D     = 0,                 # utility when dead
  u_trt   = 0.95,              # utility when treated
  d_e    = 0.03,               # discount factor for effectiveness
  d_c    = 0.03                # discount factor for costs
))
```

08.2 Load Sick-Sicker Markov model function

```r
source("Functions_markov_sick-sicker_tunnels.R")
```

08.3 One-way sensitivity analysis (OWSA)

```r
# your turn
```

08.3.1 Plot OWSA

```r
# your turn
```

08.3.2 Optimal strategy with OWSA

```r
# your turn
```

08.3.3 Tornado plot

```r
# your turn
```

08.4 Two-way sensitivity analysis (TWSA)

```
# your turn
```

08.4.1 Plot TWSA

```
# your turn
```

09 Probabilistic Sensitivity Analysis (PSA)

```
# Function to generate PSA input dataset
```

09.1 Conduct probabilistic sensitivity analysis

```
# your turn
```

09.2 Create PSA object for dampack

```
# your turn
```

09.2.1 Save PSA objects

```
# your turn
```

09.3 Create probabilistic analysis graphs

```
# your turn
```

Vector with willingness-to-pay (WTP) thresholds.

```
# your turn
```

09.3.1 Cost-Effectiveness Scatter plot

```
# your turn
```

09.4 Conduct CEA with probabilistic output

```
# your turn
```

09.4.1 Plot cost-effectiveness frontier

```
# your turn
```

09.4.2 Cost-effectiveness acceptability curves (CEACs) and frontier (CEAF)

```
# your turn
```

09.4.3 Expected Loss Curves (ELCs)

```
# your turn
```

09.4.4 Expected value of perfect information (EVPI)

```
# your turn
```