SA: Markov Sick-Sicker 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 (6,7)

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. 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. Med Decis Making. Online First https://doi.org/10.1177/0272989X19893973

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/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

```
# 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

# Transition probabilities (per cycle)
p_HD    <- 0.005                   # probability to die when healthy
p_HS1   <- 0.15                      # probability to become sick when healthy, conditional on survivi
p_S1H   <- 0.5                       # probability to become healthy when sick, conditional on survivi
p_S1S2  <- 0.105                     # probability to become sicker when sick, conditional on survivin
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
```

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

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

# The cohort starts as healthy
m_M_notrt[1, ] <- m_M_trt[1, ] <- c(1, 0, 0, 0) # initiate first cycle of cohort trace
```

04.2 Transition probability matrix

```r
# create the transition probability matrix for NO treatment (notrt)
m_P_notrt  <- matrix(0,
                     nrow = n_s,
                     ncol = n_s,
                     dimnames = list(v_n, v_n)) # name the columns and rows of the matrix
m_P_notrt
```

Fill in the transition probability matrix:

```r
# from Healthy
m_P_notrt["H", "H"  ] <- (1 - p_HD) * (1 - p_HS1)
m_P_notrt["H", "S1" ] <- (1 - p_HD) * p_HS1
m_P_notrt["H", "D"  ] <- p_HD
# from Sick
m_P_notrt["S1", "H" ] <- (1 - p_S1D) * p_S1H
m_P_notrt["S1", "S1"] <- (1 - p_S1D) * (1 - (p_S1H + p_S1S2))
m_P_notrt["S1", "S2"] <- (1 - p_S1D) * p_S1S2
m_P_notrt["S1", "D" ] <- p_S1D
# from Sicker
m_P_notrt["S2", "S2"] <- 1 - p_S2D
m_P_notrt["S2", "D" ] <- p_S2D
# from Dead
m_P_notrt["D", "D"  ] <- 1

# create transition probability matrix for treatment same as no treatment
m_P_trt <- m_P_notrt
```

05 Run Markov model

```r
for (t in 1:n_t){     # loop through the number of cycles
  m_M_notrt[t + 1, ] <- t(m_M_notrt[t, ]) %*% m_P_notrt   # estimate the Markov trace
                                                          # for the next cycle (t + 1)
  m_M_trt[t + 1, ]   <- t(m_M_trt[t, ])    %*% m_P_trt     # estimate the Markov trace
                                                          # for the next cycle (t + 1)
} # close the loop

head(m_M_notrt)  # show the first 6 lines of the matrix
```

06 Compute and Plot Epidemiological Outcomes

06.1 Cohort trace

```r
# create a plot of the data
matplot(m_M_notrt, type = 'l',
        ylab = "Probability of state occupancy",
        xlab = "Cycle",
        main = "Cohort Trace")
# add a legend to the graph
legend("topright", v_n, col = 1:n_s, lty = 1:n_s, bty = "n")
```

06.2 Overall Survival (OS)

```r
# calculate the overall survival (OS) probability for no treatment
v_os_notrt <- 1 - m_M_notrt[, "D"]
```

```r
# alternative way of calculating the OS probability
v_os_notrt <- rowSums(m_M_notrt[, 1:3])
# create a simple plot showing the OS
plot(0:n_t, v_os_notrt, type = 'l',
     ylim = c(0, 1),
     ylab = "Survival probability",
     xlab = "Cycle",
     main = "Overall Survival")
# add grid
grid(nx = n_t, ny = 10, col = "lightgray", lty = "dotted", lwd = par("lwd"),
     equilogs = TRUE)
```

06.2.1 Life Expectancy (LE)

```r
v_le <- sum(v_os_notrt)     # summing probablity of OS over time  (i.e. life expectancy)
```

06.3 Disease prevalence

```r
v_prev <- rowSums(m_M_notrt[, c("S1", "S2")]) / v_os_notrt
plot(v_prev,
     ylim = c(0, 1),
     ylab = "Prevalence",
     xlab = "Cycle",
     main = "Disease prevalence")
```

06.4 Proportion of sick in S1 state

```r
v_prop_S1 <- m_M_notrt[, "S1"] / v_prev
plot(0:n_t, v_prop_S1,
     xlab = "Cycle",
     ylab = "Proportion",
     main = "Proportion of sick in S1 state",
     col  = "black", type = "l")
```

07 Compute Cost-Effectiveness Outcomes

```r
# Vectors with costs and utilities by treatment
v_u_notrt   <- c(u_H, u_S1, u_S2, u_D)
v_u_trt     <- c(u_H, u_trt, u_S2, u_D)

v_c_notrt   <- c(c_H, c_S1, c_S2, c_D)
v_c_trt     <- c(c_H, c_S1 + c_trt, c_S2 + c_trt, c_D)
```

07.1 Mean Costs and QALYs for Treatment and NO Treatment

```
v_tu_notrt  <- m_M_notrt   %*%  v_u_notrt
v_tu_trt    <- m_M_trt     %*%  v_u_trt

v_tc_notrt  <- m_M_notrt   %*%  v_c_notrt
v_tc_trt    <- m_M_trt     %*%  v_c_trt
```

07.2 Discounted Mean Costs and QALYs

```
tu_d_notrt  <- t(v_tu_notrt)   %*%  v_dwe
tu_d_trt    <- t(v_tu_trt)     %*%  v_dwe

tc_d_notrt  <- t(v_tc_notrt)   %*%  v_dwc
tc_d_trt    <- t(v_tc_trt)     %*%  v_dwc

# store them into a vector
v_tc_d      <- c(tc_d_notrt, tc_d_trt)
v_tu_d      <- c(tu_d_notrt, tu_d_trt)

# Dataframe with discounted costs and effectiveness
df_ce       <- data.frame(Strategy = v_names_str,
                          Cost     = v_tc_d,
                          Effect   = v_tu_d)
df_ce
```

07.3 Compute ICERs of the Markov model

```
df_cea <- calculate_icers(cost       = df_ce$Cost,
                          effect     = df_ce$Effect,
                          strategies = df_ce$Strategy)
df_cea
```

07.4 Plot frontier of the Markov model

```
plot(df_cea, effect_units = "Quality of Life", xlim = c(15.6, 16.6))
```

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_HD     = 0.005,  # 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
  p_S1S2   = 0.105,  # probability to become sicker 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
))

# store the parameter names into a vector
v_names_params <- names(l_params_all)
```

08.2 Load Sick-Sicker Markov model function

```r
source("Functions_markov_sick-sicker.R")
# Test function
calculate_ce_out(l_params_all)
```

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

```
# 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
gen_psa <-
```

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.

```r
# your turn
```

09.3.1 Cost-Effectiveness Scatter plot

```r
# your turn
```

09.4 Conduct CEA with probabilistic output

```r
# your turn
```

09.4.1 Plot cost-effectiveness frontier

```r
# your turn
```

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

```r
# your turn
```

09.4.3 Expected Loss Curves (ELCs)

```r
# your turn
```

09.4.4 Expected value of perfect information (EVPI)

```r
# your turn
```