PSA: Three-strategy decision tree in R - HVE

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. 2020 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/dectree", force = TRUE) # Uncomment if there is a newer version
p_load_gh("DARTH-git/dampack", "DARTH-git/dectree") # load one or more GitHub packages
```

02 Load functions

```
source(here('functions','Functions.R'))
```

03 Define parameter input values

```
v_names_str   <- c("No Tx", "Tx All", "Biopsy")    # names of strategies
n_str         <- length(v_names_str)               # number of strategies
wtp           <- 100000                            # willingness to pay threshold

# Probabilities
p_HVE         <- 0.52   # prevalence of HVE
p_HVE_comp    <- 0.71   # complications with untreated HVE
p_OVE_comp    <- 0.01   # complications with untreated OVE
p_HVE_comp_tx <- 0.36   # complications with treated HVE
p_OVE_comp_tx <- 0.20   # complications with treated OVE
p_biopsy_comp <- 0.05   # probability of complications due to biopsy

# Costs
c_VE          <- 1200   # cost of viral encephalitis care without complications
c_VE_comp     <- 9000   # cost of viral encephalitis care with complications
c_tx          <- 9500   # cost of treatment
c_biopsy      <- 25000  # cost of brain biopsy

# QALYs
q_VE          <- 20     # remaining QALYs for those without VE-related complications
q_VE_comp     <- 19     # remaining QALYs for those with VE-related complications
q_loss_biopsy <- -0.01  # one-time QALY loss due to brain biopsy

# store the parameters into a list
l_params_all <- as.list(data.frame(p_HVE, p_HVE_comp, p_OVE_comp, p_HVE_comp_tx, p_OVE_comp_tx, p_biops
```

```
                                   c_VE, c_VE_comp, c_tx, c_biopsy,
                                   q_VE, q_VE_comp, q_loss_biopsy))
# store the names of the parameters into a vector
v_names_params <- c('p_HVE', 'p_HVE_comp', 'p_OVE_comp', 'p_HVE_comp_tx', 'p_OVE_comp_tx', 'p_biopsy_co
                    'c_VE', 'c_VE_comp',  'c_tx', 'c_biopsy', 'q_VE', 'q_VE_comp', 'q_loss_biopsy')
```

04 Create and run decision tree model

```
decision_tree_HVE_output <- with(as.list(l_params_all), {

  # Create vector of weights for each strategy

  v_w_no_tx  <- c(  p_HVE  *    p_HVE_comp  ,  # HVE, complications
                    p_HVE  * (1-p_HVE_comp) ,  # HVE, no complications
                 (1-p_HVE) *    p_OVE_comp  ,  # OVE, complications
                 (1-p_HVE) * (1-p_OVE_comp))   # OVE, no complications

  v_w_tx     <- # your turn

  v_w_biopsy <- # your turn

  # Create vector of outcomes (QALYs) for each strategy

  v_qaly_no_tx  <- c(q_VE_comp ,  # HVE, complications
                     q_VE      ,  # HVE, no complications
                     q_VE_comp ,  # OVE, complications
                     q_VE)        # OVE, no complications

  v_qaly_tx     <- # your turn

  v_qaly_biopsy <- # your turn

  # Create vector of costs for each strategy

  v_cost_no_tx  <- c(c_VE_comp ,  # HVE, complications
                     c_VE      ,  # HVE, no complications
                     c_VE_comp ,  # OVE, complications
                     c_VE)        # OVE, no complications

  v_costs_tx     <- # your turn

  v_costs_biopsy <- # your turn

  # Calculate total utilities for each strategy ####
  total_qaly_no_tx  <- v_w_no_tx  %*%  v_qaly_no_tx
  total_qaly_tx     <- v_w_tx     %*%  v_qaly_tx
  total_qaly_biopsy <- v_w_biopsy %*%  v_qaly_biopsy

  # Calculate total costs for each strategy ####
  total_cost_no_tx  <- v_w_no_tx  %*%  v_cost_no_tx
  total_cost_tx     <- v_w_tx     %*%  v_cost_tx
```

```r
  total_cost_biopsy <- v_w_biopsy %*%  v_cost_biopsy

  v_total_qaly <- c(total_qaly_no_tx, total_qaly_tx, total_qaly_biopsy)  # vector of total QALYs
  v_total_cost <- c(total_cost_no_tx, total_cost_tx, total_cost_biopsy)  # vector of total costs
  v_nmb        <- v_total_qaly * wtp - v_total_cost                      # calculate vector of nmb

  # Name outcomes
  names(v_total_qaly) <- v_names_str  # names for the elements of the total QALYs vector
  names(v_total_cost) <- v_names_str  # names for the elements of the total cost vector
  names(v_nmb)        <- v_names_str  # names for the elements of the nmb vector

  df_output <- data.frame(Strategy =  v_names_str,
                          Cost     =  v_total_cost,
                          Effect   =  v_total_qaly,
                          NMB      =  v_nmb)
  return(df_output)
})

# model output
decision_tree_HVE_output
```

04.1 Plot the decision tree

```r
# your turn
```

05 Cost-Effectiveness Analysis

```r
# your turn
```

05.1 Plot frontier of Decision Tree

```r
# your turn
```

06 Deterministic Sensitivity Analysis

06.1 List of input parameters

```r
# your turn
```

06.2 Load decision tree model function

```r
# your turn
```

06.3 One-way sensitivity analysis (OWSA)

```r
# your turn
```

06.3.1 Plot OWSA

```r
# your turn
```

06.3.2 Optimal strategy with OWSA

```r
# your turn
```

06.3.3 Tornado plot

```r
# your turn
```

06.4 Two-way sensitivity analysis (TWSA)

```r
# your turn
```

06.4.1 Plot TWSA

```r
# your turn
```

07 Probabilistic Sensitivity Analysis (PSA)

```r
# your turn
```

07.2 Create PSA object for dampack

```r
# your turn
```

07.2.1 Save PSA objects

```r
# your turn
```

07.3 Create probabilistic analysis graphs

```r
# your turn
```

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

```r
# your turn
```

07.3.1 Cost-Effectiveness Scatter plot

```r
# your turn
```

07.4 Conduct CEA with probabilistic output

```r
# your turn
```

07.4.1 Plot cost-effectiveness frontier

```r
# your turn
```

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

```r
# your turn
```

07.4.3 Expected Loss Curves (ELCs)

```r
# your turn
```

07.4.4 Expected value of perfect information (EVPI)

```r
# your turn
```