heemod TB population Markov model

N Green

17/01/2020

```r
# see: https://cran.r-project.org/web/packages/heemod/vignettes/d_non_homogeneous.html

# NOTE:
# transitions happen at the beginning of each year (equivalent to transition happening at
# the end + ignoring the first year) with method = "beginning".
# Since with this method the first year is actually the second,
# costs should be discounted from the start with the argument first = TRUE in discount().

library(heemod)
library(purrr)
library(dplyr)
```

```r
# age-dependent probability of death, TB and QoL weighting
pdeath_QoL <-
  read.csv(here::here("raw data", "pdeath_QoL.csv"))

# probabilistic realisations of starting state probabilities
# generated from decision tree
load(file = here::here("data", "init_states.RData"))
head(init_states)
```

```
##   noLTBI completeTx incompleteTx      noTx activeTB dead
## 1  0.643 0.04429927  0.068075534 0.2446252        0    0
## 2  0.683 0.02576789  0.019090692 0.2721414        0    0
## 3  0.672 0.04943117  0.030844729 0.2477241        0    0
## 4  0.742 0.05223975  0.001835084 0.2039252        0    0
## 5  0.726 0.02743526  0.010678055 0.2358867        0    0
## 6  0.650 0.05570397  0.024181029 0.2701150        0    0
```

```r
# define the model heemod parameters
param <- define_parameters(
  age_init = 34,                   # starting age
  age = age_init + markov_cycle,   # increment age annually

  # transition probabilities
  pReact_comp = 0.0006779,         # TB after completed LTBI treatment
  pReact_incomp = 0.0015301,       # TB after LTBI treatment dropout
  pReact = 0.0019369,              # TB after no treatment

  TB_cost = 4925.76,               # cost of TB treatment (£)
  d = 0.035,                       # annual discount factor

  # match prob death to age
  pdeath = look_up(data = pdeath_QoL,
                   value = "pDeath",
                   age = age),
  pdeathTB = look_up(data = pdeath_QoL,
                     value = "pDeath_TB",
                     age = age),

  # match QoL weight to age
  QoL = look_up(data = pdeath_QoL,
                value = "QOL_weight",
                age = age)
)
```

```
## Warning: markov_cycle was deprecated in heemod 0.16.0.
## i Please use model_time instead.
## i The deprecated feature was likely used in the base package.
##   Please report the issue to the authors.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
# create transition matrix
mat_trans <- define_transition(
  state_names = c(
    "noLTBI",
    "completeTx",
    "incompleteTx",
    "noTx",
    "activeTB",
    "dead"
  ),

  # from-to probability matrix
  # C represent complements
  C, 0, 0, 0, 0,             pdeath,
  0, C, 0, 0, pReact_comp,   pdeath,
  0, 0, C, 0, pReact_incomp, pdeath,
  0, 0, 0, C, pReact,        pdeath,
  C, 0, 0, 0, 0,             pdeathTB,
  0, 0, 0, 0, 0,             1
)

# define starting state populations
init_states <- select(.data = init_states,
                      noLTBI,
                      completeTx,
                      incompleteTx,
                      noTx)

init_states <- data.frame(init_states,
                          activeTB = 0,
                          dead = 0)

# define cost and utility values associated with each state
```

```
noLTBI <- define_state(
  cost = 0,
  utility = discount(QoL, d, first = TRUE)
)

completeTx <- define_state(
  cost = 0,
  utility = discount(QoL, d, first = TRUE)
)

incompleteTx <- define_state(
  cost = 0,
  utility = discount(QoL, d, first = TRUE)
)

noTx <- define_state(
  cost = 0,
  utility = discount(QoL, d, first = TRUE)
)

activeTB <- define_state(
  cost = discount(TB_cost, d, first = TRUE),
  utility = discount(QoL - 0.15, d, first = TRUE)
)

dead <- define_state(
  cost = 0,
  utility = 0
)

# combine all of the model elements to form
# a 'stratgey' consisting of a transition
# matrix and states states with properties attached
strat <- define_strategy(
  transition = mat_trans,
  noLTBI = noLTBI,
  completeTx = completeTx,
  incompleteTx = incompleteTx,
  noTx = noTx,
  activeTB = activeTB,
  dead = dead
)
```

```
save(strat, params, cost, utility,
     file = "data/ltbi_heemod.RData")
```

```
# run a single simulation
res_mod <-
  run_model(
    init = 1000 * init_states[1, ], # initial population sizes
    method = "end",
    strat,
    parameters = param,
    cycles = 66,                     # number of time steps
    cost = cost,
    effect = utility
  )
```

```
## No named model -> generating names.
```

Run multiple simulations

Using the sample of starting state probabilities

```
res_mod <- list()

for (i in 1:nrow(init_states)) {

  res_mod[[i]] <-
    suppressMessages(
      run_model(
        # init = c(674.0588764, # hard-code values
        #          168.0253748,
        #          42.42724895,
        #          115.4884998,
        #          0,
        #          0),
        init = 1000 * init_states[i, ],  # population sizes
        method = "end",
        strat,
        parameters = param,
        cycles = 66,
        cost = cost,
        effect = utility
      ))
}
```

Results

```
res_mod[[1]]
```

```
## 1 strategy run for 66 cycles.
##
## Initial state counts:
##
## noLTBI = 643
## completeTx = 44.2992721299233
## incompleteTx = 68.0755335064717
## noTx = 244.625194363605
## activeTB = 0
## dead = 0
##
## Counting method: 'end'.
##
## Values:
##
##       cost  utility
## I 64939.64 18735.25
```

```r
# extract the cost and utility values
c1 <- map_df(res_mod, "run_model")$cost
h1 <- map_df(res_mod, "run_model")$utility

get_counts(res_mod[[1]])
```

```
## # A tibble: 396 x 4
##    .strategy_names model_time state_names count
##    <chr>                <int> <chr>       <dbl>
##  1 I                        1 noLTBI       642.
##  2 I                        2 noLTBI       643.
##  3 I                        3 noLTBI       643.
##  4 I                        4 noLTBI       642.
##  5 I                        5 noLTBI       642.
##  6 I                        6 noLTBI       642.
##  7 I                        7 noLTBI       642.
##  8 I                        8 noLTBI       642.
##  9 I                        9 noLTBI       641.
## 10 I                       10 noLTBI       641.
## # i 386 more rows
```

```r
get_values(res_mod[[1]])
```

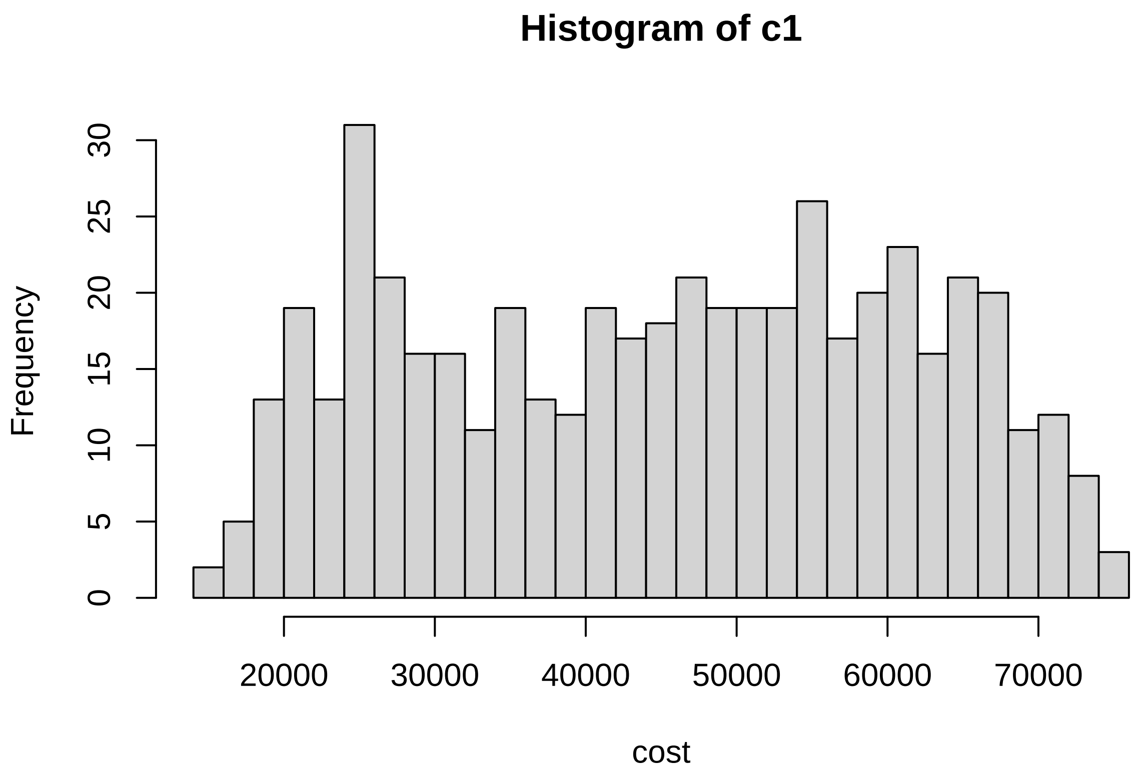
```
##     model_time .strategy_names value_names        value
## 1            1               I        cost 2893.6217181
## 2            2               I        cost 2788.4685562
## 3            3               I        cost 2686.9630890
## 4            4               I        cost 2589.0048654
## 5            5               I        cost 2494.3029316
## 6            6               I        cost 2402.9158522
## 7            7               I        cost 2314.6079384
## 8            8               I        cost 2229.3845530
## 9            9               I        cost 2147.0354045
## 10          10               I        cost 2067.5602321
## 11          11               I        cost 1990.6994981
## 12          12               I        cost 1916.4105829
## 13          13               I        cost 1844.6932493
## 14          14               I        cost 1775.3801128
## 15          15               I        cost 1708.4451018
## 16          16               I        cost 1643.6450770
## 17          17               I        cost 1580.9234648
## 18          18               I        cost 1520.2998884
## 19          19               I        cost 1461.6662675
## 20          20               I        cost 1404.8697286
## 21          21               I        cost 1349.8350639
## 22          22               I        cost 1296.4699955
## 23          23               I        cost 1244.7507045
## 24          24               I        cost 1194.4873875
## 25          25               I        cost 1145.7691008
## 26          26               I        cost 1098.3958630
## 27          27               I        cost 1052.2756949
## 28          28               I        cost 1007.5182217
## 29          29               I        cost  964.0011471
## 30          30               I        cost  921.6721968
## 31          31               I        cost  880.5389973
## 32          32               I        cost  840.5856673
## 33          33               I        cost  801.9526687
## 34          34               I        cost  764.1848705
## 35          35               I        cost  727.2389153
## 36          36               I        cost  691.0301963
## 37          37               I        cost  655.6061620
## 38          38               I        cost  620.9247593
## 39          39               I        cost  586.5875311
## 40          40               I        cost  552.9029239
## 41          41               I        cost  519.7202225
## 42          42               I        cost  487.3073234
## 43          43               I        cost  455.5072152
## 44          44               I        cost  424.3748705
## 45          45               I        cost  393.7646992
## 46          46               I        cost  363.6181185
## 47          47               I        cost  333.6484722
## 48          48               I        cost  304.3522703
## 49          49               I        cost  275.4194877
## 50          50               I        cost  247.2334057
## 51          51               I        cost  219.8060264
## 52          52               I        cost  193.2614457
## 53          53               I        cost  168.0387193
## 54          54               I        cost  144.1360339
## 55          55               I        cost  121.8242521
## 56          56               I        cost  101.2858046
## 57          57               I        cost   82.6780498
## 58          58               I        cost   66.3118896
## 59          59               I        cost   51.9160178
## 60          60               I        cost   39.5652209
## 61          61               I        cost   29.5376199
## 62          62               I        cost   21.4867125
## 63          63               I        cost   15.2944047
## 64          64               I        cost   10.5076250
## 65          65               I        cost    6.9552518
## 66          66               I        cost    4.4602567
## 67           1               I     utility  878.4298395
## 68           2               I     utility  847.9796437
## 69           3               I     utility  818.5383220
## 70           4               I     utility  790.0196004
## 71           5               I     utility  762.4472174
## 72           6               I     utility  735.7517380
## 73           7               I     utility  709.9398032
## 74           8               I     utility  684.9494409
## 75           9               I     utility  660.7851933
## 76          10               I     utility  637.3687936
## 77          11               I     utility  574.1453476
## 78          12               I     utility  553.6448161
## 79          13               I     utility  533.7923691
## 80          14               I     utility  514.5837727
## 81          15               I     utility  495.9494006
## 82          16               I     utility  477.8753901
## 83          17               I     utility  460.3707879
## 84          18               I     utility  443.4060424
## 85          19               I     utility  426.9378036
## 86          20               I     utility  410.9459951
## 87          21               I     utility  372.1436459
## 88          22               I     utility  357.9371158
## 89          23               I     utility  344.0983823
## 90          24               I     utility  330.6551502
## 91          25               I     utility  317.5519374
## 92          26               I     utility  304.7640740
## 93          27               I     utility  292.3250666
## 94          28               I     utility  280.2013944
## 95          29               I     utility  268.3796387
## 96          30               I     utility  256.8637124
## 97          31               I     utility  239.4907084
## 98          32               I     utility  228.8785935
## 99          33               I     utility  218.4770364
## 100         34               I     utility  208.2747816
## 101         35               I     utility  198.2486822
## 102         36               I     utility  188.4132946
## 103         37               I     utility  178.7575629
## 104         38               I     utility  169.1677942
## 105         39               I     utility  159.7332680
## 106         40               I     utility  150.4112925
## 107         41               I     utility  132.2225432
## 108         42               I     utility  123.8134117
## 109         43               I     utility  115.5567282
## 110         44               I     utility  107.4136681
## 111         45               I     utility   99.3688475
## 112         46               I     utility   91.3444637
## 113         47               I     utility   83.4764501
## 114         48               I     utility   75.6805993
## 115         49               I     utility   68.0624950
## 116         50               I     utility   60.6263134
## 117         51               I     utility   53.4071482
## 118         52               I     utility   46.5273300
## 119         53               I     utility   39.9880969
## 120         54               I     utility   33.8662986
## 121         55               I     utility   28.2148059
## 122         56               I     utility   23.0799343
## 123         57               I     utility   18.5512301
## 124         58               I     utility   14.5562346
## 125         59               I     utility   11.1189004
## 126         60               I     utility    8.3205496
## 127         61               I     utility    6.0674994
## 128         62               I     utility    4.3297883
## 129         63               I     utility    2.9825219
## 130         64               I     utility    1.9796768
## 131         65               I     utility    1.2731980
## 132         66               I     utility    0.7895506
```

```r
summary(res_mod[[4]])
```

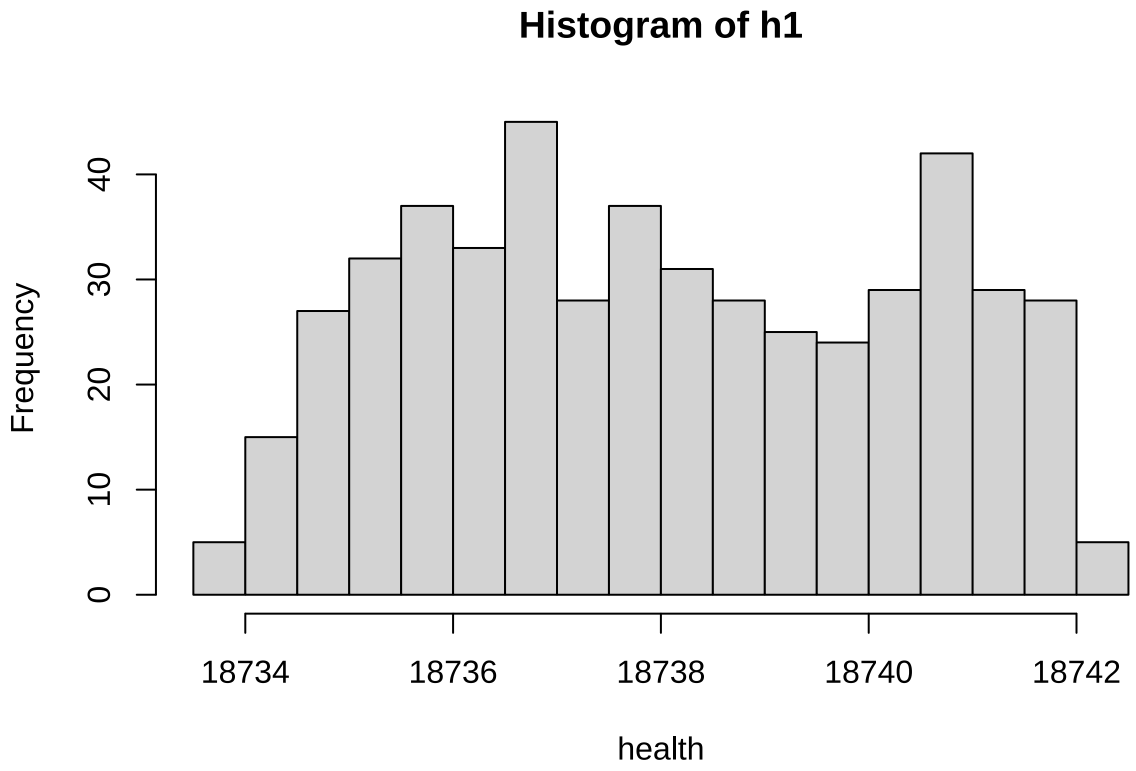
```
## 1 strategy run for 66 cycles.
##
## Initial state counts:
##
```

```
## noLTBI = 742
## completeTx = 52.2397525531171
## incompleteTx = 1.83508364108649
## noTx = 203.925163805796
## activeTB = 0
## dead = 0
##
## Counting method: 'end'.
##
## Values:
##
##        cost  utility
## I 46249.04 18737.91
```

```
# plots
hist(c1, breaks = 30, xlab = "cost")
```



```
hist(h1, breaks = 30, xlab = "health")
```

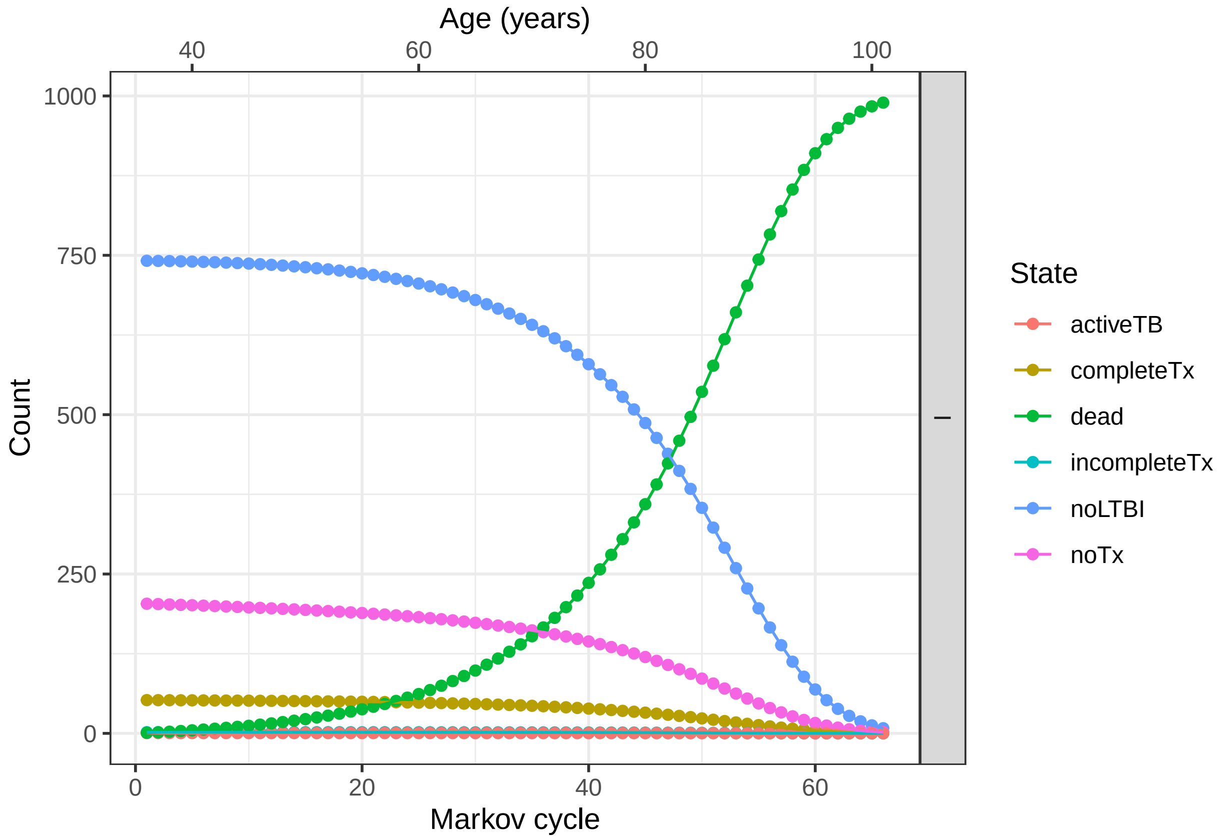


```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

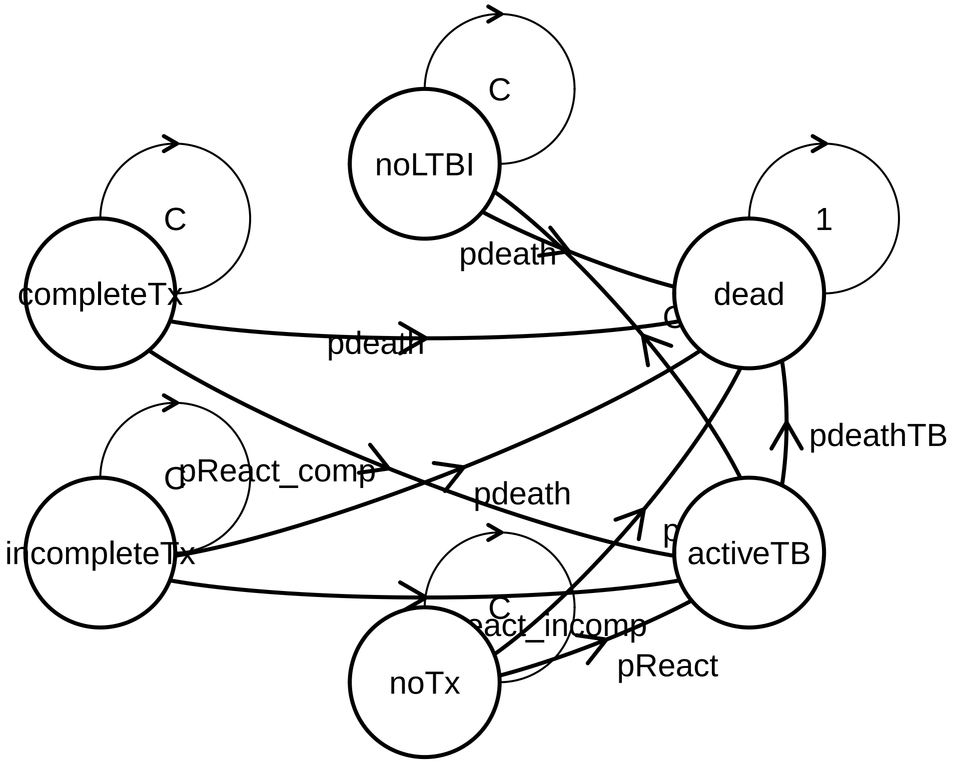
```
plot(res_mod[[4]]) +
   scale_x_continuous(sec.axis = sec_axis(~ . + 35, name = "Age (years)")) +
  theme_bw()
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```



```
# state-edge graph
plot(mat_trans, arr.type = "simple")
```

```
## Loading required namespace: diagram
```


C
noLTBI
C
1
completeTx
pdeath
dead
C
pdeath
pdeathTB
C
pReact_comp
pdeath
incompleteTx
p
activeTB
C
eact_incomp
noTx
pReact