An Introductory Tutorial to Cohort State-Transition Models in R

Fernando Alarid-Escudero, PhD* Eline Krijkamp, MSc† Eva A. Enns, PhD‡
Alan Yang, MSc§ Myriam G.M. Hunink, PhD†¶ Petros Pechlivanoglou, PhD‖
Hawre Jalal, MD, PhD**

2021-03-31

Abstract

Decision models can synthesize evidence from different sources to simulate the long-term consequences of different strategies in the presence of uncertainty. Cohort state-transition models (cSTM) are decision models commonly used in medical decision-making to simulate hypothetical cohorts' transitions across various health states over time. cSTMs can be time-independent, where transition probabilities are constant over time or depend on simulation time or time spent in a given state. This tutorial shows how to implement time-independent cSTMs in R, an open-source mathematical and statistical programming language, and is the first part of a two-part tutorial. We demonstrate how to calculate costs and effectiveness outcomes, conduct a cost-effectiveness analysis of multiple strategies and probabilistic sensitivity analyses using a previously published cSTM. We provide a link to a public repository with all the R code described in this tutorial that modelers can use to replicate the example or be adapted for various decision modeling applications.

*Division of Public Administration, Center for Research and Teaching in Economics (CIDE), Aguascalientes, AGS, Mexico
†Department of Epidemiology, Erasmus University Medical Center, Rotterdam, The Netherlands
‡Division of Health Policy and Management, University of Minnesota School of Public Health, Minneapolis, MN, USA
§The Hospital for Sick Children, Toronto
¶Center for Health Decision Sciences, Harvard T.H. Chan School of Public Health, Boston, USA
‖The Hospital for Sick Children, Toronto and University of Toronto, Toronto, Ontario, Canada
**University of Pittsburgh, Pittsburgh, PA, USA

1 Introduction

Healthcare policymakers are often tasked to allocate healthcare resources under constrained budgets and uncertain contexts. Their final choice can be informed by health economic evaluations, which often rely on decision models to synthesize evidence from different sources and project long-term outcomes of different alternatives. A commonly used decision model is the discrete-time cohort state-transition model (cSTM), often referred to as a Markov model.[1]

In a recent review, we illustrated the increased utilization of R's statistical programming framework in health decision sciences. We provided a collection of resources for its application in medical decision making.[2] Many packages have been developed specifically to estimate and construct cSTMs in R. However, these packages are necessarily inflexible and require the user to follow a specific cSTM structure. If the desired cSTM does not fit within this structure, using these packages can be challenging. For example, a realistic cSTM that accounts for simulation time-dependency (e.g., age dependency) and time dependency on state residence is challenging in the existing packages. Using an R package does not guide how to conceptualize and implement realistic cSTMs in a programming language, which is a current literature gap. This introductory tutorial is part of a two-part series that aim to (1) conceptualize time-independent cSTMs in a programming language and (2) guide how to implement these cSTMs in *base* R. We focus on using R *base* packages to avoid the limitation of constructing cSTMs in a pre-specified structure defined by a package and to ensure modelers understand the concept and structure of cSTMs rather than implementing them as a "black box".

A cSTM is a state transition model with a hypothetical cohort transitioning between different health states over time. In contrast, an individual-based state transition model (iSTM) is another variety of STM, where individuals are the modeling unit. Each of them can transition between health states over time.[3] We have previously published a tutorial on the implementation of iSTM in R.[4] This introductory tutorial covers time-independent cSTMs and how to use them to conduct a full cost-effectiveness analysis comparing multiple interventions and implementing probabilistic sensitivity analyses. The accompanying advanced tutorial covers cSTMs for which transition probabilities depend on simulation time and time spent in a state, calculation of various epidemiological outcomes, and incorporation of transition rewards.[5] We first describe each of the components of a time-independent cSTM. Then, we illustrate the implementation of these components with an example. Our general conceptualization should apply to other programming languages (e.g., MATLAB, Python, C++, Julia). The full R code used in this tutorial is provided in the supplementary materials so that readers can replicate and modify the example to fit their needs. The reader can find the most up-to-date model code and code to create the graphs of this tutorial in the accompanying GitHub repository (https://github.com/DARTH-git/Cohort-modeling-tutorial). We assume that the reader is familiar with the basics of decision modeling and coding in programming languages. Thus, a prior introduction to R and linear algebra for decision modelers is recommended.

2 Cohort state-transition models (cSTMs)

A cSTM is most appropriate when the decision problem can be described with a reasonable number of health states. cSTMs are often used because of their transparency, efficiency, ease of debugging, and ability to conduct specific value-of-information analyses.[3] cSTMs have been used to evaluate screening and surveillance programs,[6,7] diagnostic procedures,[8] disease management programs,[9], and interventions.[10,11]

A cSTM consists of a set of n_S mutually exclusive and collectively exhaustive health states. The cohort is assumed to be homogeneous within each health state. All persons residing in a particular health state are assumed to remain in the state for a cycle, have the same characteristics, and are indistinguishable from one another with respect to the cost and utility of being in the state. The cohort transitions between health states with defined probabilities, which are called "transition probabilities." A transition probability represents the chance that individuals in the cohort residing in a state in a given cycle transition to another state or remain in the same state for the next cycle. Transition probabilities only depend on the current health state in a given cycle. They do not depend on the history before that cycle, often referred to as the "Markovian assumption."[12–14] This means that in a Markovian cSTM, transition probabilities do not depend on the history of past transitions or time spent in a given state.

cSTMs are classified as either time-independent (time-homogeneous) or time-dependent (time-inhomogeneous). Time-independent cSTMs have constant transition probabilities (i.e., the probability of any state transition is independent of time). In contrast, time-dependent cSTMs have transition probabilities or rewards that vary over time. Time-independent models are simpler to implement than time-dependent ones, but most problems in healthcare are best modeled with time-dependent cSTMs. For example, time-dependent cSTMs can capture the increasing age-specific background mortality as the cohort ages (age dependency) and dependency on the amount of time spent in a given state (state residence). In this tutorial, we cover time-dependent cSTMs. In an advanced tutorial, we describe how to construct time-independent cSTM.[5]

2.1 Rates versus probabilities

Probabilities and rates are commonly used input parameters in decision-analytic models. While they are often numerically similar in practice, there is a subtle but important conceptual difference. A rate represents the *instantaneous* force of an event occurrence per unit time, while a probability represents the cumulative risk of an event over a defined time period.

To discuss these concepts further, let's assume that after 10,000 person-years of observation of healthy individuals (e.g., 10,000 observed for an average of 1 year, 5,000 individuals observed for an average of 2 years, etc.), we observe 500 events of getting sick. The annual event rate of becoming sick, μ_{yearly}, is then equal to $\mu_{yearly} = 500/10,000 = 0.05$.

If we then wanted to know what proportion of an initially healthy cohort becomes sick at the end of the year, then we can convert the annual rate of becoming sick into an annual probability of becoming sick using the following relationship:

$$p_{yearly} = 1 - \exp(-\mu_{yearly}). \tag{1}$$

This equation assumes that the rate of becoming sick is constant over the year, which implies that the time until a healthy person becomes sick is exponentially distributed. The parameter p_{yearly} is the transition probability from healthy to sick in a cSTM when using an annual cycle length.

If we were concerned that an annual cycle length was too long to capture disease dynamics accurately, we could instead convert to a monthly probability of becoming sick (a monthly transition probability). To do this, we first calculate the monthly rate of becoming sick. Because rates are instantaneous, the monthly rate is just the annual rate divided by 12:

$$\mu_{monthly} = \mu yearly/12. \tag{2}$$

The monthly probability can then be calculated from the monthly rate using equation @ref(eq: rate-to-prob-ann):

$$p_{monthly} = 1 - \exp\left(-\mu_{monthly}\right). \tag{3}$$

Sometimes, a transition probability might be available from published literature in one time frame (e.g., annual), whereas the model cycle length is different (e.g., monthly). In this case, one can first convert the transition probability to a rate using the following formula:

$$\mu_{yearly} = -\ln\left(1 - p_{yearly}\right). \tag{4}$$

The annual rate can then be divided by the appropriate number to convert to a new time unit (e.g., 12 for monthly, 52 for weekly, etc.) and then converted back to a probability. In cSTMs, a one-cycle transition probability reflects a conditional probability of transitioning during the cycle, given that the person is alive at the beginning of the cycle.[15]

3 Case study: Sick-Sicker model

We describe how to construct the previously published 4-state "Sick-Sicker" model for conducting a CEA of multiple strategies to illustrate the various aspects of cSTM implementation in R.[4,16] Figure 1 represents the state-transition diagram of the Sick-Sicker model.

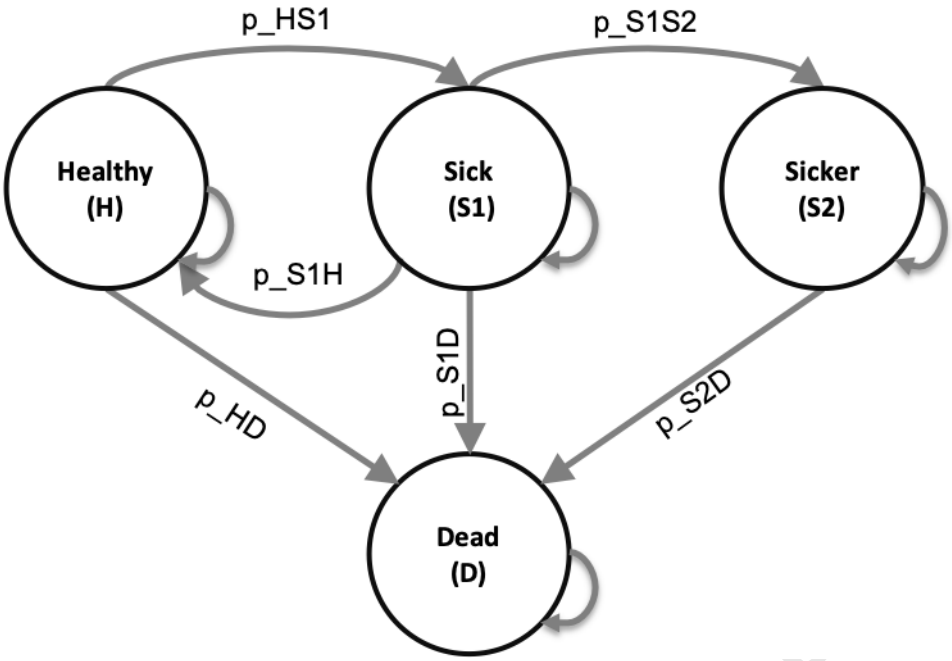

Figure 1: State-transition diagram of the time-independent Sick-Sicker cohort state-transition model with the name of the health states and possible transitions with their corresponding transition probabilities.

The model simulates a cohort to quantify the expected costs and quality-adjusted life years (QALYs) for individuals at risk of a hypothetical disease with two stages: "Sick" and "Sicker". All the parameters of the Sick-Sicker model and the corresponding R variable names are presented in Table 1 and follow the notation described in the DARTH coding framework.[17] Briefly, we define variables by `<x>_<y>_<var_name>`, where `x` is the prefix that indicates the data type (e.g., scalar (no prefix), `v` for vector, `m` for matrix, `a` for array, `df` for data frame, etc.), `y` is the prefix indicating variable type (e.g., `p` for probability, `r` for rate, `hr` for hazard ratio, `lor` for log-odds ratio, `c` for cost `c`, ufor utility, etc.), and `var_name` is some description of the variable presented separated by underscores. For example, `v_p_HD` denotes the vector of transition probabilities from health state "H" to health state "D". In later sections we will define and name all the other parameters.

In the Sick-Sicker model, we simulate a hypothetical cohort of 25-year-old individuals over their lifetime (until a maximum age of 100 years) who all start in the "Healthy" state (denoted "H"). This means that we will simulate the cohort for 75 cycles. The total number of cycles is denoted as n_T and defined in R as `n_cycles`. Healthy individuals are at risk of developing the disease when they transition to the "Sick" state (denoted by "S1"). Sick individuals are at risk of further progressing to a more severe disease stage, the "Sicker" health state (denoted by "S2"). Individuals in S1 can recover and return to H. However, once individuals reach S2, they cannot recover; that is, the probability of transitioning to S1 or H from S2 is zero. Individuals in H face constant background mortality. Individuals in S1 and S2 face an increased hazard of death, compared to healthy individuals, in the form of a hazard ratio (HR) of 3 and 10, respectively, relative to the background mortality hazard rate. Individuals in S1 and S2 also experience increased health care costs and reduced QoL compared to individuals in H. Once simulated individuals die, they transition to the absorbing "Dead" state (denoted by "D"), where they remain. All transitions between non-death states are assumed to be conditional on surviving each cycle. The evolution of the cohort is simulated in one-year discrete-time cycles. Both costs

and QALYs are discounted at an annual rate of 3%.

We are interested in evaluating the cost-effectiveness of four strategies: Strategy A, strategy B, a combination of A and B (Strategy AB), and the standard of care (strategy SoC). Strategy A involves administering treatment A that increases the QoL of individuals in S1 from 0.75 (utility without treatment, `u_S1`) to 0.95 (utility with treatment A, `u_trtA`) and costs $12,000 per year (`c_trtA`).[4] This strategy does not impact the QoL of individuals in S2, nor does it change the risk of becoming sick or progressing through the sick states. Strategy B uses treatment B to reduce only the rate of Sick individuals progressing to the Sicker state with a hazard ratio (HR) of 0.6 (`hr_S1S2_trtB`) and costs $13,000 per year (`c_trtB`), and does not affect QoL. Strategy AB involves administering both treatments A and B.

We assume that it is not possible to distinguish between Sick and Sicker patients; therefore, individuals in both disease states receive the treatments. Note that for strategy A, the model has the same structure and identical transition probabilities to SoC. The only difference is the added cost of the treatment for S1 or S2, and QoL increases for S1. After comparing the four strategies in terms of expected QALYs and costs, we calculate the incremental cost per QALY gained between non-dominated strategies.

Table 1: Description of parameters, their R variable name, base-case values and distribution.

Parameter	**R name**	**Base-case**	**Distribution**
Number of cycles (n_cycles)	`n_cycles`	75 years	-
Names of health states (n)	`v_names_states`	H, S1, S2, D	-
Annual discount rate for costs	`d_c`	3%	-
Annual discount rate for QALYs	`d_e`	3%	-
Number of PSA samples (K)	`n_sim`	1,000	-
Annual transition probabilities conditional on surviving			
- Disease onset (H to S1)	`p_HS1`	0.15	beta(30, 170)
- Recovery (S1 to H)	`p_S1H`	0.5	beta(60, 60)
- Disease progression (S1 to S2)	`p_S1S2`	0.105	beta(84, 716)
Annual mortality			
- Background mortality rate (H to D)	`r_HD`	0.002	-
- Hazard ratio of death in S1 vs H	`hr_S1`	3.0	lognormal(log(3.0), 0.01)
- Hazard ratio of death in S2 vs H	`hr_S2`	10.0	lognormal(log(10.0), 0.02)
Annual costs			
- Healthy individuals	`c_H`	$2,000	gamma(100.0, 20.0)
- Sick individuals in S1	`c_S1`	$4,000	gamma(177.8, 22.5)
- Sick individuals in S2	`c_S2`	$15,000	gamma(225.0, 66.7)
- Dead individuals	`c_D`	$0	-
Utility weights			
- Healthy individuals	`u_H`	1.00	beta(200, 3)
- Sick individuals in S1	`u_S1`	0.75	beta(130, 45)

Parameter	R name	Base-case	Distribution
- Sick individuals in S2	`u_S2`	0.50	beta(230, 230)
- Dead individuals	`u_D`	0.00	-
Treatment A effectiveness			
- Utility for treated individuals in S1	`u_trtA`	0.95	beta(300, 15)
Treatment B effectiveness			
- Reduction in rate of disease progression (S1 to S2) as hazard ratio (HR)	`hr_S1S2_trtB`	log(0.6)	lognormal(log(0.6), 0.1)

The following sections include R code snippets. All the code is stored as a GitHub repository and can be accessed from https://github.com/DARTH-git/Cohort-modeling-tutorial. The R code below describes the initialization of the input parameters.

```r
## General setup
cycle_length <- 1 # cycle length equal one year
n_age_init <- 25  # age at baseline
n_age_max  <- 100 # maximum age of follow up
n_cycles <- n_age_max - n_age_init # number of cycles
v_names_states <- c("H", "S1", "S2", "D") # the 4 health states of the model:
                                # Healthy (H), Sick (S1), Sicker (S2), Dead (D)
n_states <- length(v_names_states) # number of health states
d_e <- 0.03 # discount rate for QALYs of 3% per cycle
d_c <- 0.03 # discount rate for costs of 3% per cycle
v_names_str <- c("Standard of care", # store the strategy names
                 "Strategy A",
                 "Strategy B",
                 "Strategy AB")

## Transition probabilities (per cycle), hazard ratios and odds ratio (OR)
r_HD    <- 0.002 # constant rate of dying when Healthy (all-cause mortality rate)
p_HS1   <- 0.15  # probability of becoming Sick when Healthy
p_S1H   <- 0.5   # probability of becoming Healthy when Sick
p_S1S2  <- 0.105 # probability of becoming 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

# Effectiveness of treatment B
hr_S1S2_trtB <- 0.6 # hazard ratio of becoming Sicker when Sick under treatment B

## State rewards
## Costs
c_H    <- 2000  # cost of being Healthy for one cycle
c_S1   <- 4000  # cost of being Sick for one cycle
```

```
c_S2   <- 15000 # cost of being Sicker for one cycle
c_D    <- 0     # cost of being dead for one cycle
c_trtA <- 12000 # cost of receiving treatment A for one cycle
c_trtB <- 13000 # cost of receiving treatment B for one cycle
# Utilities
u_H    <- 1     # utility of being Healthy for one cycle
u_S1   <- 0.75  # utility of being Sick for one cycle
u_S2   <- 0.5   # utility of being Sicker for one cycle
u_D    <- 0     # utility of being dead for one cycle
u_trtA <- 0.95  # utility when receiving treatment A for one cycle
```

To compute the background mortality risk, `p_HD`, from the background mortality rate for the same cycle length (i.e.,`cycle_length=1`), we apply Eq.(1) to `r_HD`. To compute the mortality risks from S1 and S2, we multiply the background mortality rate `r_HD` by the hazard ratios `hr_S1` and `hr_S2`, respectively, and then convert back to probabilities using Eq.(1). These calculations are required because hazard ratios only apply to rates and not to probabilities. The code below performs the computation in R. In the supplementary material, we provide R functions that compute transformations between rates and probabilities since these transformations are frequently used.

```
## Mortality rates
r_S1D <- r_HD * hr_S1 # rate of dying when Sick
r_S2D <- r_HD * hr_S2 # rate of dying when Sicker
## Probabilities of dying
cycle_length <- 1
p_HD  <- 1 - exp(-r_HD*cycle_length)  # background mortality risk (i.e., probability)
p_S1D <- 1 - exp(-r_S1D*cycle_length) # probability of dying when Sick
p_S2D <- 1 - exp(-r_S2D*cycle_length) # probability of dying when Sicker
```

To compute the risk of progression from S1 to S2 under treatment B, we first transform `p_S1S2` to a rate, `r_S1S2`, using Eq.(4). Then, we multiply the hazard ratio of treatment B to the rate of progressing from S1 to S2 and transform it back to probabilities by applying Eq.(1).

```
## Transition probability of becoming Sicker when Sick for treatment B
# transform probability to rate
r_S1S2 <- -log(1-p_S1S2)/cycle_length
# apply hazard ratio to rate to obtain transition rate of becoming Sicker when Sick
# for treatment B
r_S1S2_trtB <- r_S1S2 * hr_S1S2_trtB
# transform rate to probability
p_S1S2_trtB <- 1-exp(-r_S1S2_trtB*cycle_length) # probability to become Sicker when Sick
                                                # under treatment B conditional on surviving
```

4 Conceptualizing and implementing time-independent cSTM dynamics

A cSTM consists of three core components: (1) a state vector, $\mathbf{m}_t$, that stores the distribution of the cohort across all health states in cycle t where $t = 0, \ldots, n_T$; (2) the cohort trace matrix, M, that stacks $\mathbf{m}_t$ for all t and represents the distribution of the cohort in the various states over time; and (3) a transition probability matrix, P.[18] If the cSTM is comprised of n_S discrete health states, $\mathbf{m}_t$ is a $1 \times n_S$ vector and P is a $n_S \times n_S$ matrix. The i-th element of $\mathbf{m}_t$, where $i = 1, \ldots, n_S$, represents the proportion of the cohort in the i-th health state in cycle t, referred to as $m_{[t,i]}$. Thus, $\mathbf{m}_t$ is written as:

$$\mathbf{m}_t = \begin{bmatrix} m_{[t,1]} & m_{[t,2]} & \cdots & m_{[t,n_S]} \end{bmatrix}.$$

The elements of P are the transition probabilities of moving from state i to state j, $p_{[i,j]}$, where $\{i, j\} = 1, \ldots, n_S$ and $t = 0, \ldots, n_T$

$$P = \begin{bmatrix} p_{[1,1]} & p_{[1,2]} & \cdots & p_{[1,n_S]} \\ p_{[2,1]} & p_{[2,2]} & \cdots & p_{[2,n_S]} \\ \vdots & \vdots & \ddots & \vdots \\ p_{[n_S,1]} & p_{[n_S,2]} & \cdots & p_{[n_S,n_S]} \end{bmatrix}.$$

Note that all rows of the transition probability matrix must sum to one, $\sum_{j=1}^{n_S} p_{[i,j]} = 1$ for all $i = 1, \ldots, n_S$.

The state vector at cycle $t+1$ ($\mathbf{m}_{t+1}$) is then calculated as the matrix product of the state vector at cycle t, $\mathbf{m}_t$, and the transition probability matrix, P, such that

$$\mathbf{m}_{t+1} = \mathbf{m}_t P \text{ for } t = 0, \ldots, (n_T - 1),$$

where $\mathbf{m}_1$ is computed from $\mathbf{m}_0$ and represents the initial state vector with the distribution of the cohort across all health states at the start of the simulation (cycle 0). Then, this equation is iteratively evaluated until $t = n_T$.

The cohort trace matrix, M, is a matrix of dimensions $(n_T + 1) \times n_S$ where each row is a state vector ($-\mathbf{m}_t-$), such that

$$M = \begin{bmatrix} -\mathbf{m}_0- \\ -\mathbf{m}_1- \\ \vdots \\ -\mathbf{m}_{n_T}- \end{bmatrix}.$$

Note that the initial cycle (i.e., cycle 0) corresponds to $t = 0$, which is on the first row of M. Thus, M stores the output of the cSTM, which could be used to compute various epidemiological outcomes, such as prevalence and survival probability over time, and economic outcomes, such as cumulative resource use and costs. Table 2 describes the elements related to the core components of cSTM and their suggested R code names. For a more detailed description of the variable types, data structure, R name for all cSTM elements, please see the Supplementary Material.

Table 2: Components of a cSTM with their R name.

Element	Description	R name
n_S	Number of states	`n_states`
$\mathbf{m}_0$	Initial state vector	`v_s_init`
$\mathbf{m}_t$	State vector in cycle t	`v_mt`
M	Cohort trace matrix	`m_M`
P	Time-independent transition probability matrix	`m_P`
$\mathbf{A}$	Transition-dynamics array	`a_A`

For the Sick-Sicker model, the entire cohort starts in the H state. Therefore, we create the $n_S \times 1$ initial state vector `v_s_init` with all of the cohort assigned to the H state:

```
v_s_init <- c(H = 1, S1 = 0, S2 = 0, D = 0) # initial state vector
v_s_init
```

```
##  H S1 S2  D
##  1  0  0  0
```

The variable `v_s_init` is used to initialize M represented by `m_M` for the cohorts under SoC strategy and strategy A because the transition probabilities are the same under both strategies, and by `m_M_trtB` for the cohorts under strategies B and AB because both transition probabilities under both strategies are affected by treatment B.

```
## Initialize cohort trace for SoC
m_M <- matrix(NA,
              nrow = (n_cycles + 1), ncol = n_states,
              dimnames = list(0:n_cycles, v_names_states))
# Store the initial state vector in the first row of the cohort trace
m_M[1, ] <- v_s_init
## Initialize cohort trace under treatment B
m_M_trtB <- m_M # structure and initial states remain the same.
```

Note that the initial state vector, `v_s_init`, can be modified to account for the cohort's distribution across the states at the start of the simulation and might vary by strategy.

Since the Sick-Sicker model consists of 4 states, we create a 4×4 transition probability matrix, `m_P`. We initialize the matrix with default values of zero for all transition probabilities and then populate it with the corresponding transition probabilities. To access an element of `m_P`, we specify first the row number (or name) and then the column number (or name) separated by a comma. For example, the transition probability of going from state Healthy (H) to state Sick (S1) could be accessed by `m_P[1, 2]` or by using the corresponding row or column state-names as characters `m_P["H", "S1"]`. We assume that all transitions to non-death states are conditional on not dying in a cycle. Thus, we first condition on surviving by multiplying the transition probabilities times `1-p_HD`, the probability of not dying in a cycle. For example, to obtain the probability of transitioning from H to S1, we multiply the transition probability from H to S1 conditional of being alive, `p_HS1` by `1-p_HD`.

```r
## Initialize transition probability matrix
m_P <- matrix(0,
              nrow = n_states, ncol = n_states,
              dimnames = list(v_names_states, v_names_states)) # row and column names
## Fill in matrix
# From H
m_P["H", "H"]   <- (1 - p_HD) * (1 - p_HS1)
m_P["H", "S1"]  <- (1 - p_HD) * p_HS1
m_P["H", "D"]   <- p_HD
# From S1
m_P["S1", "H"]  <- (1 - p_S1D) * p_S1H
m_P["S1", "S1"] <- (1 - p_S1D) * (1 - (p_S1H + p_S1S2))
m_P["S1", "S2"] <- (1 - p_S1D) * p_S1S2
m_P["S1", "D"]  <- p_S1D
# From S2
m_P["S2", "S2"] <- 1 - p_S2D
m_P["S2", "D"]  <- p_S2D
# From D
m_P["D", "D"]   <- 1
```

Because treatment B alters progression from S1 to S2, we created a different transition probability matrix to model this treatment, `m_P_trtB`. We initialize `m_P_trtB` as a copy of `m_P` and update only the transition probabilities from S1 to S2 (i.e., `p_S1S2` is replaced with `p_S1S2_trtB`).

```r
## Initialize transition probability matrix for treatment B
m_P_trtB <- m_P
## Update only transition probabilities from S1 involving p_S1S2
m_P_trtB["S1", "S1"] <- (1 - p_S1D) * (1 - (p_S1H + p_S1S2_trtB))
m_P_trtB["S1", "S2"] <- (1 - p_S1D) * p_S1S2_trtB
```

Once both transition matrices are created, we verify they are valid by checking that all of their rows sum to one and that each of the transition probabilities of both matrices is between 0 and 1 using the functions `check_sum_of_transition_array` and `check_transition_probability`, respectively, which have been used previously elsewhere[17] and are provided in the `darthtools` package (https://github.com/DARTH-git/darthtools). If a transition matrix is not valid, they will produce an error message.

```r
### Check if transition probability matrices are valid
## Check that transition probabilities are [0, 1]
check_transition_probability(m_P)
check_transition_probability(m_P_trtB)
## Check that all rows sum to 1
check_sum_of_transition_array(m_P,      n_states = n_states, n_cycles = n_cycles)
check_sum_of_transition_array(m_P_trtB, n_states = n_states, n_cycles = n_cycles)
```

Next, we obtain the cohort distribution across the 4 states over 75 cycles using a time-independent cSTM

under SoC and treatment B. To achieve this, we iteratively compute the matrix product between each of the rows of `m_M` and `m_P`, and between `m_M_trtB` and `m_P_trtB`, respectively, using the `%*%` symbol in R at each cycle using a `for` loop

```
# Iterative solution of time-independent cSTM
for(t in 1:n_cycles){
  # For SoC
  m_M[t + 1, ] <- m_M[t, ] %*% m_P
  # For treatment B
  m_M_trtB[t + 1, ] <- m_M_trtB[t, ] %*% m_P_trtB
}
```

Table 3 shows the cohort trace matrix M of the Sick-Sicker model under the SoC strategy for the first six cycles. The whole cohort starts in the H state and transitions to the rest of the states over time. Given that the D state is absorbing, the proportion in this state increases over time. A graphical representation of the cohort trace for all the cycles is shown in Figure 2.

Table 3: The distribution of the cohort under SoC for the first six cycles of the time-independent Sick-Sicker model. The first row, labeled with cycle 0, contains the distribution of the cohort at time zero.

Cycle	H	S1	S2	D
0	1.000	0.000	0.000	0.000
1	0.848	0.150	0.000	0.002
2	0.794	0.186	0.016	0.005
3	0.766	0.192	0.035	0.008
4	0.745	0.190	0.054	0.011
5	0.726	0.186	0.073	0.015

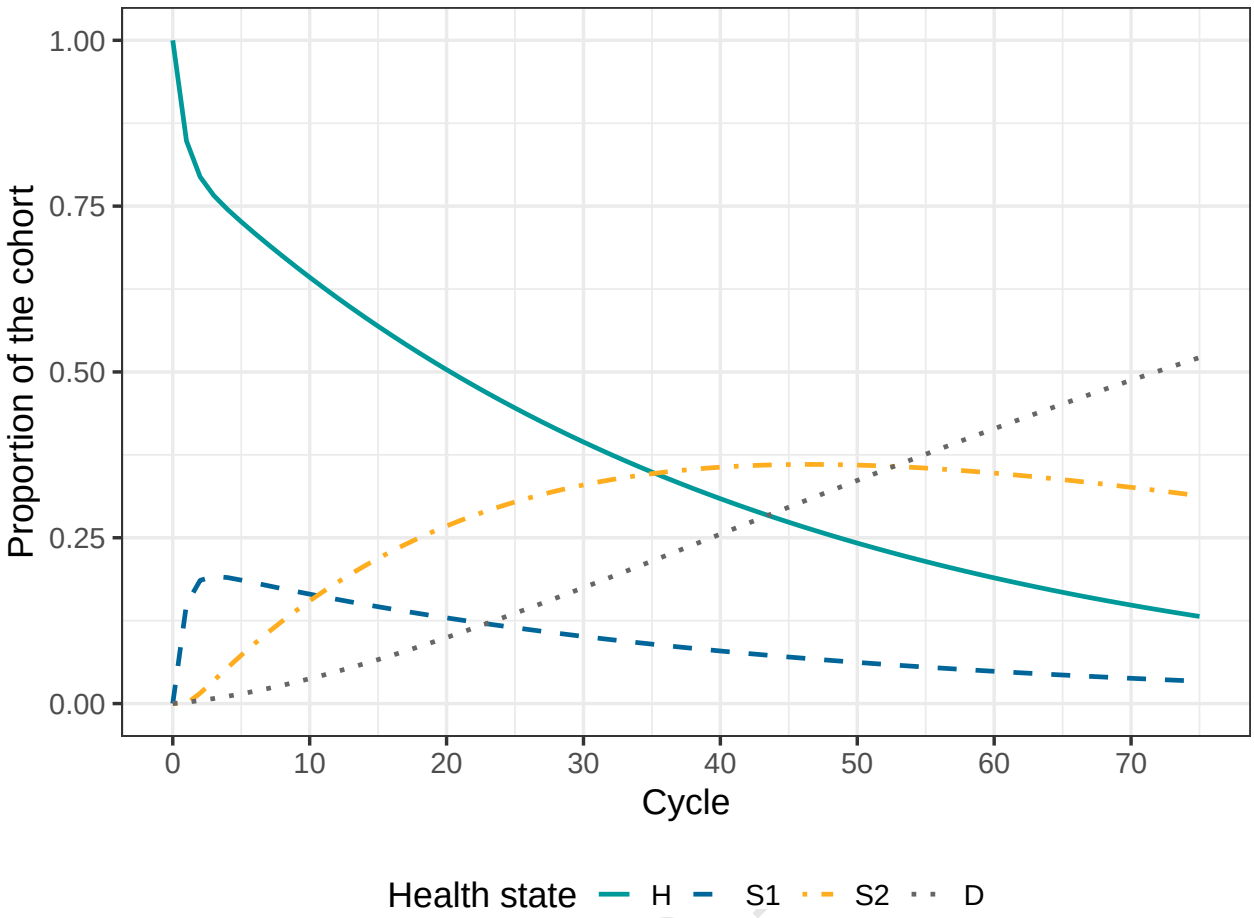

Figure 2: Cohort trace of the time-independent cSTM

5 Economic and Epidemiological outcomes

cSTMs can be used to generate different effectiveness and economic outputs. In a CEA, the outcomes are typically the total expected QALYs and total costs accrued by the cohort over a predefined time horizon. However, epidemiological outcomes are often used to produce other measures of interest or model calibration and validation. Some common epidemiological outcomes include survival, prevalence, incidence, the average number of events, and lifetime risk of events.[3] In this tutorial, we describe how to generate the effectiveness and economic outcomes. We describe how to compute epidemiological outcomes in an accompanying tutorial.[5]

5.1 Effectiveness and economic outcomes

5.1.1 State rewards

A state reward refers to a value assigned to individuals for being in a given state. In a cost-utility context, these could be either utilities or costs associated with remaining in a certain health state for one cycle. The total expected reward of an outcome of interest for the entire cohort at each cycle can be represented by a column vector $\mathbf{y}$ of size $n_T + 1$. To calculate $\mathbf{y}$, we compute the matrix product of the cohort trace matrix

times a *vector* of state rewards $\mathbf{r}$ of the same dimension as the number of states (n_S), such that

$$\mathbf{y} = M\mathbf{r}. \tag{5}$$

Note that rewards can also be time- or age-dependent; in such cases, the vector of state rewards will be time-dependent, $\mathbf{r}_t$. For the Sick-Sicker model, we create a vector of utilities and costs for each of the four strategies considered. The vectors of utilities and costs in R, `v_u_SoC` and `v_c_SoC`, respectively, contain the utilities and costs corresponding with being in each of the four health states under SoC, which are shown in Table 1.

```
# Vector of state utilities under SOC
v_u_SoC <- c(H = u_H, S1 = u_S1, S2 = u_S2, D = u_D)
# Vector of state costs under SoC
v_c_SoC <- c(H = c_H, S1 = c_S1, S2 = c_S2, D = c_D)
```

We account for the benefits and costs of both treatments individually and their combination to create the state-reward vectors under treatments A and B (strategies A and B, respectively) and when applied jointly (strategy AB). Only treatment A affects QoL, so we create a vector of utilities for strategy A, `v_u_strA`, where we substitute the utility of being in S1 under SOC, `u_S1`, with the utility associated with the benefit of treatment A in being in that state, `u_trtA`. Treatment B does not affect QoL, so the vector of utilities for strategy B, `v_u_strB`, is the same as for SoC However, when both treatments A and B are applied jointly (strategy AB), the resulting vector of utilities `v_u_strAB` equals that of strategy A.

```
# Vector of state utilities for strategy A
v_u_strA <- c(H = u_H, S1 = u_trtA, S2 = u_S2, D = u_D)
# Vector of state utilities for strategy B
v_u_strB <- v_u_SoC
# Vector of state utilities for strategy AB
v_u_strAB <- v_u_strA
```

Both treatments A and B incur a cost. To create the vector of state costs for strategy A, `v_c_strA`, we add the cost of treatment A, `c_trtA`, to the state costs of S1 and S2. Similarly, when constructing the vector of state costs for strategy B, `v_c_strB`, we add the cost of treatment B, `c_trtB`, to the state costs of S1 and S2. Finally, for the vector of state costs for strategy AB, `v_c_strAB`, we add both treatment costs to the state costs of S1 and S2.

```
# Vector of state costs for strategy A
v_c_strA <- c(H  = c_H,
              S1 = c_S1 + c_trtA,
              S2 = c_S2 + c_trtA,
              D  = c_D)
# Vector of state costs for strategy B
v_c_strB <- c(H  = c_H,
              S1 = c_S1 + c_trtB,
              S2 = c_S2 + c_trtB,
              D  = c_D)
```

```r
# Vector of state costs for strategy AB
v_c_strAB <- c(H  = c_H,
               S1 = c_S1 + (c_trtA + c_trtB),
               S2 = c_S2 + (c_trtA + c_trtB),
               D  = c_D)
```

To compute the expected QALYs and costs for the Sick-Sicker model under SoC and strategy A, we apply Eq. (5) by multiplying the cohort trace matrix, `m_M`, times the corresponding strategy-specific state vectors of rewards. Similarly, To compute the expected rewards for strategies B and AB, we multiply the cohort trace matrix accounting for the effectiveness of treatment B, `m_M_trtB`, times their corresponding state vectors of rewards.

```r
# Vector of QALYs under SoC
v_qaly_SoC <- m_M %*% v_u_SoC
# Vector of costs under SoC
v_cost_SoC <- m_M %*% v_c_SoC
# Vector of QALYs for strategy A
v_qaly_strA <- m_M %*% v_u_strA
# Vector of costs for strategy A
v_cost_strA <- m_M %*% v_c_strA
# Vector of QALYs for strategy B
v_qaly_strB <- m_M_trtB %*% v_u_strB
# Vector of costs for strategy B
v_cost_strB <- m_M_trtB %*% v_c_strB
# Vector of QALYs for strategy AB
v_qaly_strAB <- m_M_trtB %*% v_u_strAB
# Vector of costs for strategy AB
v_cost_strAB <- m_M_trtB %*% v_c_strAB
```

5.1.2 Within-cycle correction

Discretizing a continuous-time cSTM with a discrete-time cSTM might introduce biases in the time spent on each state.[19] One approach to reducing these biases is to reduce cycle length, which will require simulating the model for a larger number of cycles, which sometimes is not computationally efficient. Another approach to better approximate the expected rewards of cumulative health and cost outcomes from a continuous-time process using a discrete-time model is to use within-cycle corrections (WCC).[20] In this tutorial, we use Simpson's 1/3rd rule as a WCC by multiplying the rewards (e.g., costs and effectiveness) by 1/3 in the first and last cycles, by 4/3 if cycle number is odd, and by 2/3 if the cycle number is even.[22] We implement the WCC by generating a column vector **wcc** of size $n_T + 1$ with values corresponding to the first, $t = 0$, and last cycle, $t = n_T + 1$, equal to 1/3, and the entries corresponding to the odd and even cycles with 4/3 and 2/3, respectively.

$$\mathbf{wcc} = \left[\frac{1}{3}, \frac{2}{3}, \frac{4}{3}, \cdots, \frac{1}{3}\right]$$

The within-cycle correction vector is the same for both costs and QALYs; thus, only one vector, `v_wcc`, is required.

```
## Vector with cycles
v_cycles <- seq(1, n_cycles+1)
## Generate 2/3 and 4/3 multipliers for even and odd entries, respectively
v_wcc <- ((v_cycles %% 2)==0)*(2/3) + ((v_cycles %% 2)!=0)*(4/3)
## Substitute 1/3 in first and last entries
v_wcc[1] <- v_wcc[n_cycles + 1] <- 1/3
```

5.1.3 Discounting future rewards

Future costs and benefits are often discounted by a specific rate to calculate the net present value of these rewards. To account for discounting, the generated rewards per cycle are multiplied by the cycle-specific discount weight. The vector of expected rewards, $\mathbf{y}$, is multiplied by a discounting column vector $\mathbf{d}$ of size $n_T + 1$ where each of its t-th entry represents the discounting for cycle t

$$\mathbf{d} = \left[1, \frac{1}{(1+d)^1}, \frac{1}{(1+d)^2}, \cdots, \frac{1}{(1+d)^{n_T}}\right],$$

where d is the cycle-length discount rate. Therefore, the total expected discounted outcome over the n_T cycles, y, is obtained by the inner product between $\mathbf{y}$ transposed, $\mathbf{y}'$, and $\mathbf{d}$,

$$y = \mathbf{y}'\mathbf{d}. \tag{6}$$

The discount vectors for costs and QALYs for the Sick-Sicker model, `v_dwc` and `v_dwe`, respectively, are

```
# Discount weight for effects
v_dwe <- 1 / ((1 + d_e) ^ (0:(n_cycles)))
# Discount weight for costs
v_dwc <- 1 / ((1 + d_c) ^ (0:(n_cycles)))
```

To account for both discounting and within-cycle correction, we incorporate **wcc** in equation (6) using an element-wise multiplication with $\mathbf{d}$, indicated by the $\odot$ sign.

$$y = \mathbf{y}^{'} \left(\mathbf{d} \odot \mathbf{wcc}\right). \tag{7}$$

To compute the total expected discounted QALYs and costs under all four strategies accounting for within-cycle correction, we apply equation (7) to the vectors with the rewards, discounts, and within-cycle correction.

```
## Expected discounted QALYs under SoC
n_tot_qaly_SoC <- t(v_qaly_SoC) %*% (v_dwe * v_wcc)
## Expected discounted costs under SoC
n_tot_cost_SoC <- t(v_cost_SoC) %*% (v_dwc * v_wcc)
## Expected discounted QALYs for strategy A
n_tot_qaly_strA <- t(v_qaly_strA) %*% (v_dwe * v_wcc)
## Expected discounted costs for strategy A
```

```r
n_tot_cost_strA <- t(v_cost_strA) %*% (v_dwc * v_wcc)
## Expected discounted QALYs for strategy B
n_tot_qaly_strB <- t(v_qaly_strB) %*% (v_dwe * v_wcc)
## Expected discounted costs for strategy B
n_tot_cost_strB <- t(v_cost_strB) %*% (v_dwc * v_wcc)
## Expected discounted QALYs for strategy AB
n_tot_qaly_strAB <- t(v_qaly_strAB) %*% (v_dwe * v_wcc)
## Expected discounted costs for strategy AB
n_tot_cost_strAB <- t(v_cost_strAB) %*% (v_dwc * v_wcc)
```

Table 4: Total expected discounted QALYs and costs per average individual in the cohort of the Sick-Sicker model by strategy accounting for within-cycle correction.

	Costs	QALYs
Standard of care	$148,657	20.990
Strategy A	$275,937	21.717
Strategy B	$248,571	22.482
Strategy AB	$361,341	23.354

The total expected QALYs and costs for the Sick-Sicker model under the four strategies accounting for within-cycle correction are shown in Table 4.

6 Cost-effectiveness analysis and incremental cost-effectiveness ratios (ICERs)

We combine the total expected discounted costs and QALYs for all four strategies into outcome-specific vectors, `v_cost_str` for costs and `v_qaly_str` for QALYs. We use the R package `dampack` (https://cran.r-project.org/web/packages/dampack/)[23] to calculate the incremental costs and effectiveness and the incremental cost-effectiveness ratio (ICER) the non-dominated strategies and create the data frame `df_cea` with this information.

```r
### Vector of costs
v_cost_str <- c(n_tot_cost_SoC, n_tot_cost_strA, n_tot_cost_strB, n_tot_cost_strAB)
### Vector of effectiveness
v_qaly_str <- c(n_tot_qaly_SoC, n_tot_qaly_strA, n_tot_qaly_strB, n_tot_qaly_strAB)

### Calculate incremental cost-effectiveness ratios (ICERs)
df_cea <- dampack::calculate_icers(cost = v_cost_str,
                                   effect = v_qaly_str,
                                   strategies = v_names_str)
```

The CEA results of the Sick-Sicker model are presented in Table 5. SoC is the least costly and effective strategy, followed by Strategy B producing an expected benefit of 1.492 QALYs per individual for an additional expected cost of $99,913 with an ICER of $66,960/QALY followed by Strategy AB with an ICER $129,354/QALY.

Strategy A is a dominated strategy.

Table 5: Cost-effectiveness analysis results for the Sick-Sicker model. ND: Non-dominated strategy; D: Dominated strategy.

Strategy	Costs ($)	QALYs	Incremental Costs ($)	Incremental QALYs	ICER ($/QALY)	Status
Standard of care	148,657	20.990	NA	NA	NA	ND
Strategy B	248,571	22.482	99,913	1.492	66,960	ND
Strategy AB	361,341	23.354	112,771	0.872	129,354	ND
Strategy A	275,937	21.717	NA	NA	NA	D

Figure 3 shows the cost-effectiveness efficient frontier of all four strategies for the Sick-Sicker model.

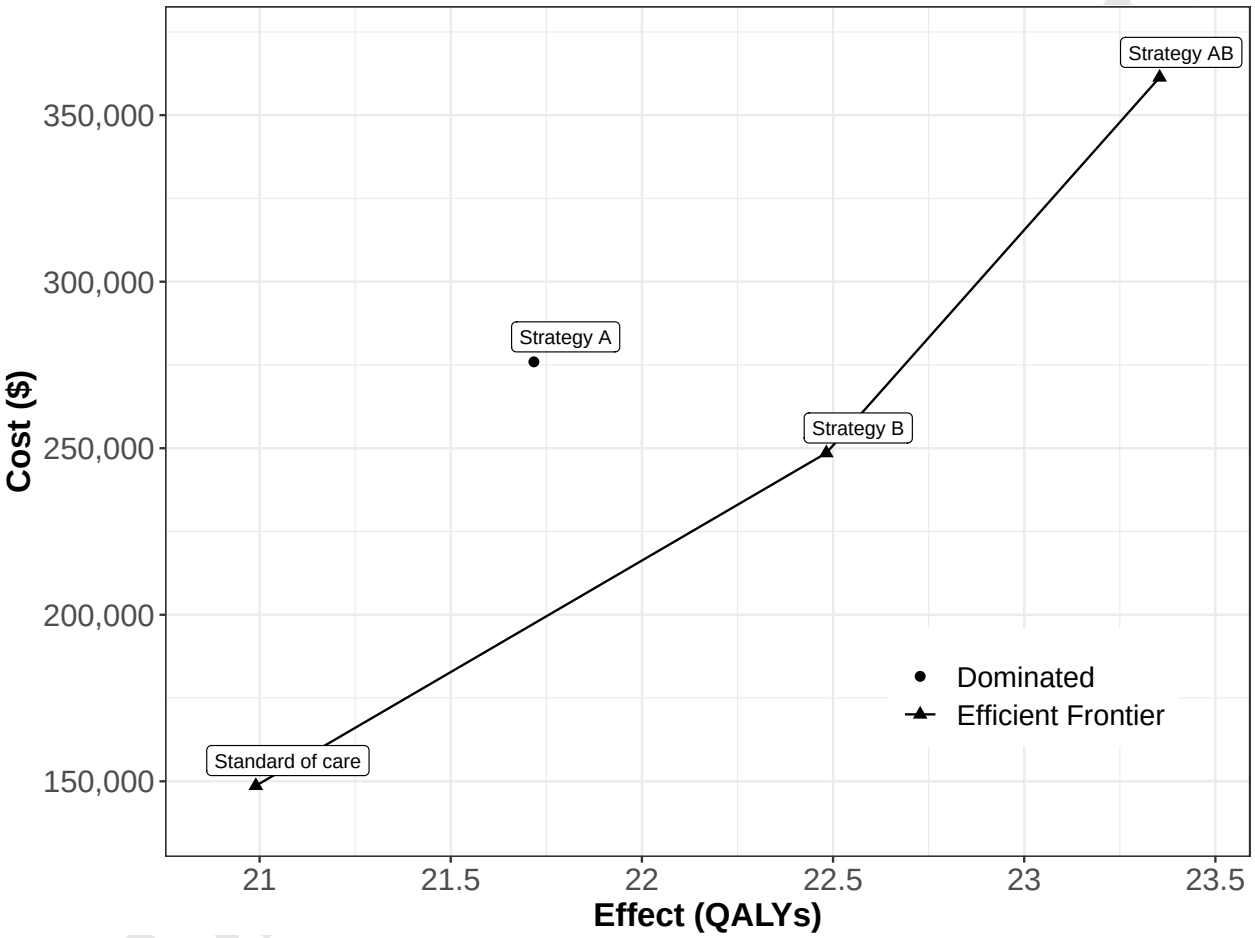

Figure 3: Cost-effectiveness efficient frontier of all four strategies for the Sick-Sicker model.

7 Probabilistic sensitivity analysis

To quantify the effect of model parameter uncertainty on cost-effectiveness outcomes, we conducted a probabilistic sensitivity analysis (PSA).[24] In a PSA, we randomly draw parameter sets from distributions that reflect the current uncertainty in model parameter estimates. The distribution for all the parameters and their

values are described in Table 1 and more detailed in the Supplementary Material. For each sampled set of parameter values, we compute model outcomes (e.g., total discounted cost and QALYs) for each strategy. In a previously published manuscript, we describe the implementation of these steps in R.[17] Briefly, to conduct the PSA, we create three R functions:

1. A function called `generate_psa_params(n_sim, seed)` that generates a sample of size `n_sim` for the model parameters from their distributions defined in Table 1. The function also takes a seed number as input, which ensures reproducibility of the PSA results. By calling this function, we generate the sample of parameter sets for the PSA: `df_psa_input <- generate_psa_params(n_sim = n_sim)`.
2. A function called `decision_model` that wraps the R code of the cSTM described in section Conceptualizing and implementing time-independent cSTM dynamics and requires as input a list of model parameter values.
3. A function called `calculate_ce_out` that calculates total discounted costs and QALYs based on the `decision_model` function output.

These functions are provided in the Supplementary Material and the accompanying GitHub repository of this manuscript.

In Figure 4, we present the cost-effectiveness acceptability curves (CEACs), which shows the probability that each strategy is cost-effective, and the cost-effectiveness frontier (CEAF), which shows the strategy with the highest expected monetary benefit, over a range of willingness-to-pay (WTP) thresholds. Each strategy's net monetary benefit (NMB) is defined as the product of total discounted QALYs and the WTP threshold minus the total discounted costs, calculated for each PSA parameter set sample. At WTP thresholds less than \$75,000 per QALY, SoC is the strategy with the highest probability of being cost-effective and the highest expected NMB. Strategy B has the highest probability of being cost-effective and the highest expected NMB for WTP thresholds between \$75,000 and \$125,000 per QALY. Strategy AB has the highest expected NMB for WTP thresholds greater than or equal to \$125,000 per QALY and is the strategy with the highest probability of being cost-effective.

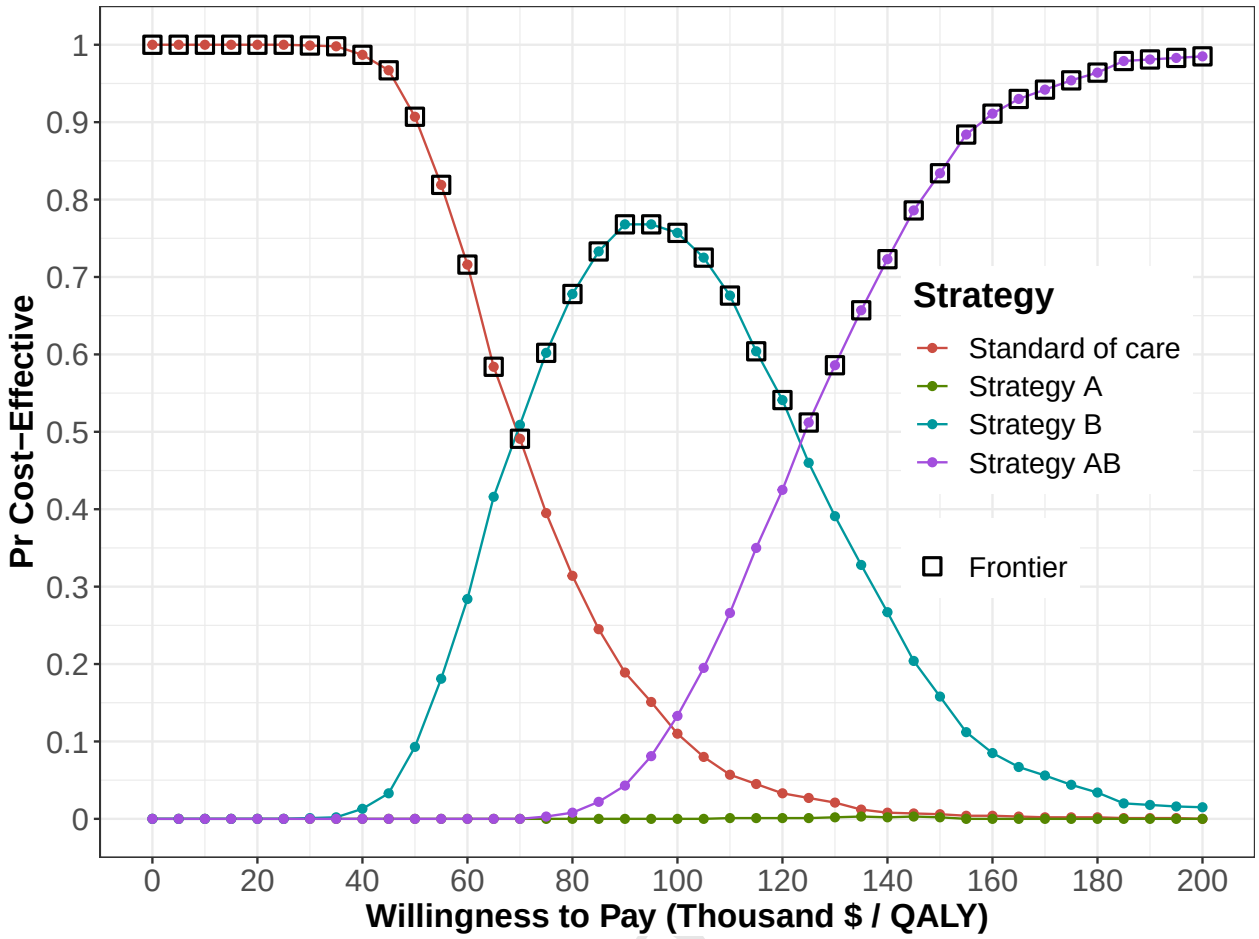

Figure 4: Cost-effectiveness acceptability curves (CEACs) and frontier (CEAF).

The code provided in the GitHub repository also produces expected loss curves (ELCs). These curves quantify the expected loss from each strategy over a range of WTP thresholds (Figure 5). The expected loss considers both the probability of making the wrong decision and the magnitude of the loss due to this decision, representing the foregone benefits of choosing a suboptimal strategy. The expected loss of the optimal strategy represents the lowest envelope of the ELCs because, given current information, the loss cannot be minimized further. The lower envelope also represents the expected value of perfect information (EVPI), which quantifies the value of eliminating parameter uncertainty. The strategy SoC has the lowest expected loss for WTP thresholds less than $70,000 per QALY, strategy B has the lowest expected loss for WTP threshold greater than or equal to $75,000 and less than $125,000. Strategy AB has the lowest expected loss for WTP threshold greater than or equal to $125,000 per QALY. At a WTP threshold of $70,000 per QALY, the EVPI is highest at $10,847. For a more detailed description of these outputs and the R code to generate them, we refer the reader to a previous publication by our group.[25]

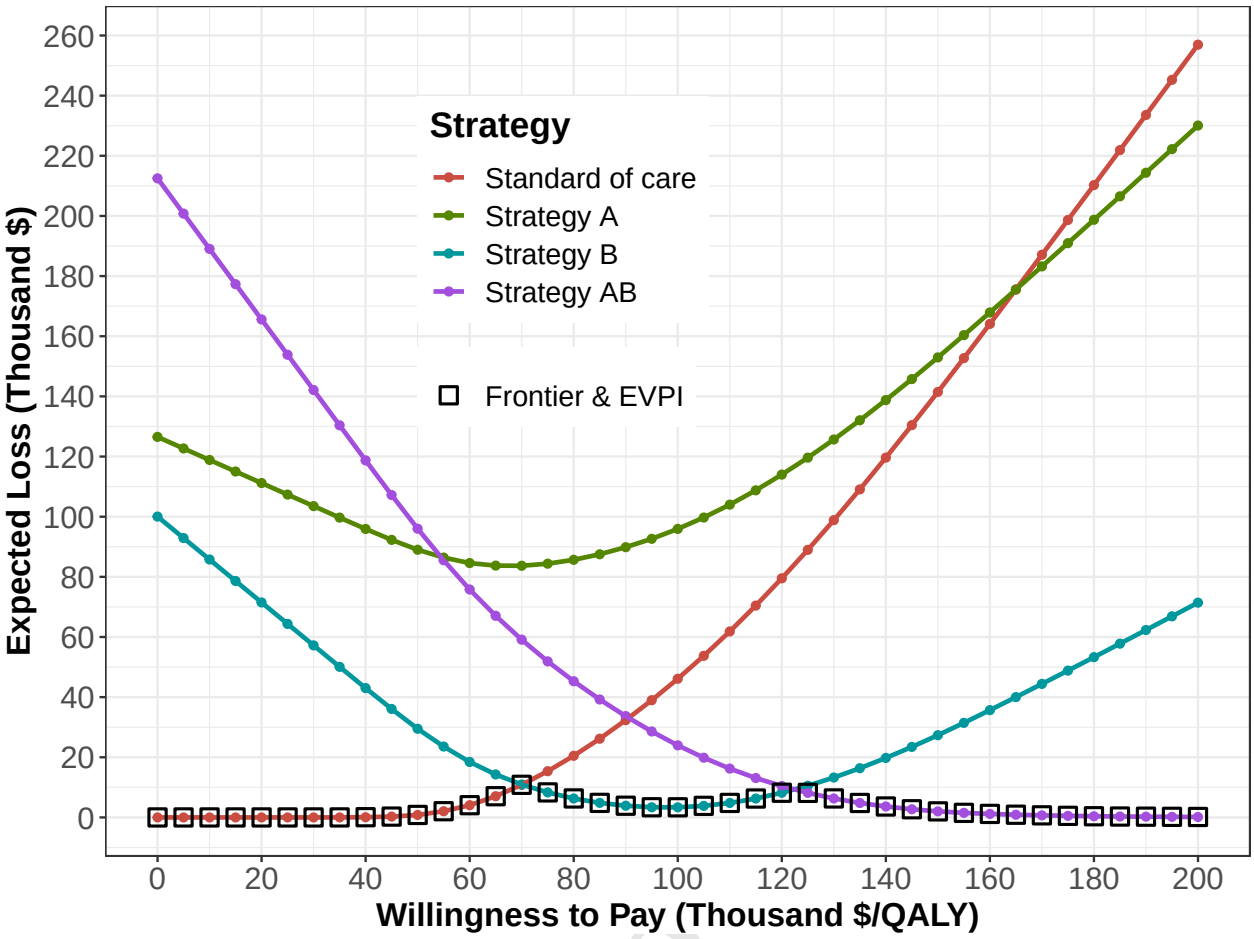

Figure 5: Expected loss curves (ELCs) and expected value of perfect information (EVPI).

8 Discussion

In this tutorial, we provided a conceptualization of time-independent cSTMs with their mathematical description and a walk-through of their implementation for CEA in R using a previously published example. We used R as the programming language of choice to show these models' implementation with accompanying code throughout the tutorial.

The parameterization of our example model assumes all parameters are known, or at least, the characterization of their uncertainty is known (i.e., we know their distributions). However, to construct a real-world cSTM, modelers must conduct a thorough synthesis of current evidence to determine these models' appropriate structure and inform all parameters. For example, determining whether transitions between non-death health states are estimated conditional on being alive or mortality risks are also considered competing risks.[24] Similarly, our PSA analysis simplifies reality where all model parameters are assumed to be independent of each other. However, parameters could be correlated with each other or have a rank order, and appropriate statistical methods that simulate these correlations or rank order might be needed.[26] We encourage modelers to use appropriate statistical methods to synthesize and quantify model parameters uncertainty accurately. Besides, modelers should correctly specify all model parameters for the cycle length of the model. For example,

some probability revision is required to adjust an annual mortality rate to a weekly probability correctly.[20]

In general, cSTMs are recommended when the number of states is "not too large".[3] This recommendation arises because as the number of states increases, it becomes more difficult to keep track of their construction but not because of the added computational expense. It is possible to build fairly complex cSTMs in R as long as the size of the transition probability matrix and outputs of interest can be stored in the RAM memory of the computer running the analysis. For example, a typical PC with 8GB of RAM can handle a transition probability array of about 1000 states and 600 slices. However, these matrices can grow quickly, and if the required number of state descriptions gets too large and difficult to manage its coding, it becomes preferable to use a stochastic (Monte Carlo) version of the state-transition model –often called individual-based state-transition models (iSTM) or microsimulation models– rather than a cohort simulation model.[3] In iSTM, the risks and rewards of simulated individuals do not need to depend only on a specific health state and can depend on their individual characteristics and attributes. Besides, modelers can store health state history and other events over time for each individual to determine the risk of new events and corresponding costs and effects. It is recommended to think about the required model structure before implementing the model in R or any other tool because turning a cSTM into an iSTM requires a different code structure. However, most input parameters and some model structures might remain the same. Still, an iSTM will also require additional functions to describe the dependency of transition probabilities and rewards on individuals' history. In a previous tutorial, we showed how to write these additional functions for the Sick-Sicker example model.[4]

With increasing model complexity and interdependency of functions to conduct various analyses like PSA, it is important to ensure all code and functions work as expected and all elements of cSTM are valid. We can achieve this by creating functions that help with model debugging and validation and thorough unit testing. In the accompanying GitHub repository, we provided functions to check that transition probability matrices and their elements are valid. However, unit testing is beyond the scope of this tutorial. Still, we refer the reader to a previously published manuscript where we describe unit testing in more detail and provide accompanying code.[17]

We focused on discrete-time matrix-form cSTMs, but these can also be implemented via a set of difference equations and continuous-time using differential equations in R.[27,28] We refer readers interested in learning more on continuous-time cSTMs to previously published manuscripts[19,29–31] and a tutorial using R.[32] There are other approaches to construct Markov models in R, most of them through purpose-specific R packages. For example, the `heemod`[33] package is designed to build Markov models that account for dependency using a pre-defined structure. Although `heemod` is a well-structured package, it requires users to set up the Markov model structure, specify the parameters, and run analyses in a pre-specified approach. This, however, limits the understanding of how cSTMs and Markov models work and are constructed. This tutorial shows how cSTMs are constructed, parameterized, and run by using only base R, so readers get a deep understanding of this type of decision models. Finally, the variable names used in this paper reflect our own style. While we provide standardized variable names, adopting these conventions is ultimately a personal preference.

In summary, this tutorial provides a conceptualization of cSTMs and a step-by-step guide to implement them in R. We aim to add to the current body of literature and material on building this type of decision models, so health decision scientists and health economists can develop cSTMs in a more flexible, efficient, open-source manner, and encouraging increased transparency and reproducibility.

9 Acknowledgements

Dr. Alarid-Escudero was supported by a grant from the National Cancer Institute (U01-CA-199335) as part of the Cancer Intervention and Surveillance Modeling Network (CISNET) and the Gordon and Betty Moore Foundation. Miss Krijkamp was supported by a fellowship from a grant by the Gordon and Betty Moore Foundation (GBMF7853) through the Society for Medical Decision Making (SMDM). Dr. Enns was supported by a grant from the National Institute of Allergy and Infectious Diseases of the National Institutes of Health under award no. K25AI118476. Dr. Hunink receives Royalties from Cambridge University Press for a textbook on Medical Decision Making, reimbursement of expenses from the European Society of Radiology (ESR) for work on the ESR guidelines for imaging referrals, reimbursement of expenses from the European Institute for Biomedical Imaging Research (EIBIR) for membership of the Scientific Advisory Board, and research funding from the American Diabetes Association, the Netherlands Organization for Health Research and Development, the German Innovation Fund, Netherlands Educational Grant ("Studie Voorschot Middelen"), and the Gordon and Betty Moore Foundation. Dr. Jalal was supported by a grant from the National Institute on Drug Abuse of the National Institute of Health under award no. K01DA048985. The content is solely the authors' responsibility and does not necessarily represent the official views of the National Institutes of Health. The funding agencies had no role in the study's design, interpretation of results, or writing of the manuscript. The funding agreement ensured the authors' independence in designing the study, interpreting the data, writing, and publishing the report. We also want to thank the anonymous reviewers of *Medical Decision Making* for their valuable suggestions and the students that took our classes to try our materials.

References

1. Kuntz KM, Russell LB, Owens DK, et al. Decision Models in Cost-Effectiveness Analysis. In: Neumann PJ, Sanders GD, Russell LB, et al. (eds) *Cost-effectiveness in health and medicine*. New York, NY: Oxford University Press, 2017, pp. 105–136.

2. Jalal H, Pechlivanoglou P, Krijkamp E, et al. An Overview of R in Health Decision Sciences. *Medical Decision Making*; 37: 735–746, http://journals.sagepub.com/doi/10.1177/0272989X16686559 (2017).

3. Siebert U, Alagoz O, Bayoumi AM, et al. State-Transition Modeling: A Report of the ISPOR-SMDM Modeling Good Research Practices Task Force-3. *Medical Decision Making*; 32: 690–700, http://mdm.sagepub.com/cgi/doi/10.1177/0272989X12455463 (2012).

4. Krijkamp EM, Alarid-Escudero F, Enns EA, et al. Microsimulation Modeling for Health Decision Sciences Using R: A Tutorial. *Medical Decision Making*; 38: 400–422, http://journals.sagepub.com/doi/10.1177/0272989X18754513 (2018).

5. Alarid-Escudero F, Krijkamp E, Enns EA, et al. A Tutorial on Time-Dependent Cohort State-Transition Models in R. 2021.

6. Suijkerbuijk AWM, Van Hoek AJ, Koopsen J, et al. Cost-effectiveness of screening for chronic hepatitis B and C among migrant populations in a low endemic country. *PLoS ONE* 2018; 13: 1–16.

7. Sathianathen NJ, Konety BR, Alarid-Escudero F, et al. Cost-effectiveness Analysis of Active Surveillance Strategies for Men with Low-risk Prostate Cancer. *European Urology*; 75: 910–917, https://linkinghub.elsevier.com/retrieve/pii/S0302283818308534 (2019).

8. Lu S, Yu Y, Fu S, et al. Cost-effectiveness of ALK testing and first-line crizotinib therapy for non-small-cell lung cancer in China. *PLoS ONE* 2018; 13: 1–12.

9. Djatche LM, Varga S, Lieberthal RD. Cost-Effectiveness of Aspirin Adherence for Secondary Prevention of Cardiovascular Events. *PharmacoEconomics - Open*; 2: 371–380, https://doi.org/10.1007/s41669-018-0075-2 (2018).

10. Pershing S, Enns EA, Matesic B, et al. Cost-Effectiveness of Treatment of Diabetic Macular Edema. *Annals of Internal Medicine* 2014; 160: 18–29.

11. Smith-Spangler CM, Juusola JL, Enns EA, et al. Population Strategies to Decrease Sodium Intake and the Burden of Cardiovascular Disease: A Cost-Effectiveness Analysis. *Annals of Internal Medicine*; 152: 481–487, http://annals.org/article.aspx?articleid=745729 (2010).

12. Kuntz KM, Weinstein MC. Modelling in economic evaluation. In: Drummond MF, McGuire A (eds) *Economic evaluation in health care: Merging theory with practice*. New York, NY: Oxford University Press, 2001, pp. 141–171.

13. Sonnenberg FA, Beck JR. Markov models in medical decision making: A practical guide. *Medical Decision Making*; 13: 322–338, http://mdm.sagepub.com/cgi/doi/10.1177/0272989X9301300409 (1993).

14. Beck JR, Pauker SG. The Markov process in medical prognosis. *Medical Decision Making*; 3: 419–458, http://mdm.sagepub.com/cgi/doi/10.1177/0272989X8300300403 (1983).

15. Miller DK, Homan SM. Determining Transition Probabilities: Confusion and Suggestions. *Medical Decision Making*; 14: 52–58, http://mdm.sagepub.com/cgi/doi/10.1177/0272989X9401400107 (1994).

16. Enns EA, Cipriano LE, Simons CT, et al. Identifying Best-Fitting Inputs in Health-Economic Model Calibration: A Pareto Frontier Approach. *Medical Decision Making*; 35: 170–182, http://www.ncbi.nlm.nih.gov/pubmed/24799456 (2015).

17. Alarid-Escudero F, Krijkamp E, Pechlivanoglou P, et al. A Need for Change! A Coding Framework for Improving Transparency in Decision Modeling. *PharmacoEconomics*; 37: 1329–1339, https://doi.org/10.1007/s40273-019-00837-x (2019).

18. Iskandar R. A theoretical foundation of state-transition cohort models in health decision analysis. *PLOS ONE*; 13: e0205543, https://www.biorxiv.org/content/early/2018/09/28/430173 (2018).

19. Rosmalen J van, Toy M, O'Mahony JF. A Mathematical Approach for Evaluating Markov Models in Continuous Time without Discrete-Event Simulation. *Medical Decision Making*; 33: 767–779, http://mdm.sagepub.com/cgi/doi/10.1177/0272989X13487947 (2013).

20. Hunink MGGM, Weinstein MC, Wittenberg E, et al. *Decision Making in Health and Medicine*. 2nd ed. Cambridge: Cambridge University Press, http://ebooks.cambridge.org/ref/id/CBO9781139506779 (2014).

21. Elbasha EH, Chhatwal J. Theoretical foundations and practical applications of within-cycle correction methods. *Medical Decision Making* 2016; 36: 115–131.

22. Elbasha EH, Chhatwal J. Myths and misconceptions of within-cycle correction: a guide for modelers and decision makers. *PharmacoEconomics* 2016; 34: 13–22.

23. Alarid-Escudero F, Easterly CA, Knowlton G, et al. dampack: Decision-Analytic Modeling Package, https://cran.r-project.org/web/packages/dampack/%20https://github.com/DARTH-git/dampack (2021).

24. Briggs AH, Weinstein MC, Fenwick EAL, et al. Model Parameter Estimation and Uncertainty Analysis: A Report of the ISPOR-SMDM Modeling Good Research Practices Task Force Working Group-6. *Medical Decision Making* 2012; 32: 722–732.

25. Alarid-Escudero F, Enns EA, Kuntz KM, et al. "Time Traveling Is Just Too Dangerous" But Some Methods Are Worth Revisiting: The Advantages of Expected Loss Curves Over Cost-Effectiveness Acceptability Curves and Frontier. *Value in Health* 2019; 22: 611–618.

26. Goldhaber-Fiebert JD, Jalal HJ. Some Health States Are Better Than Others: Using Health State Rank Order to Improve Probabilistic Analyses. *Medical Decision Making*; 36: 927–940, http://mdm.sagepub.com/cgi/doi/10.1177/0272989X15605091 (2015).

27. Grimmett G, Welsh D. Markov Chains. In: *Probability: An introduction*. Oxford University Press, pp. 203–, www.statslab.cam.ac.uk/ grg/teaching/chapter12.pdf (2014).

28. Axler S, Gehring FW, Ribet KA. Difference Equations. New York, NY: Springer, http://link.springer.com/10.1007/0-387-27645-9 (2005).

29. Cao Q, Buskens E, Feenstra T, et al. Continuous-Time Semi-Markov Models in Health Economic Decision Making: An Illustrative Example in Heart Failure Disease Management. *Medical Decision Making*; 36: 59–71, http://mdm.sagepub.com/cgi/doi/10.1177/0272989X15593080 (2016).

30. Begun A, Icks A, Waldeyer R, et al. Identification of a multistate continuous-time nonhomogeneous Markov chain model for patients with decreased renal function. *Medical Decision Making*; 33: 298–306, http://www.ncbi.nlm.nih.gov/pubmed/23275452 (2013).

31. Soares MO, Canto E Castro L. Continuous time simulation and discretized models for cost-effectiveness analysis. *PharmacoEconomics*; 30: 1101–1117, http://www.ncbi.nlm.nih.gov/pubmed/23116289 (2012).

32. Frederix GWJ, Hasselt JGC van, Severens JL, et al. Development of a framework for cohort simulation in cost-effectiveness analyses using a multistep ordinary differential equation solver algorithm in R. *Medical Decision Making*; 33: 780–92, http://www.ncbi.nlm.nih.gov/pubmed/23515213 (2013).

33. Filipović-Pierucci A, Zarca K, Durand-Zaleski I. Markov Models for Health Economic Evaluation: The R Package heemod. *arXiv:170203252v1*; April: 30, http://arxiv.org/abs/1702.03252 (2017).