

**Provisional title:** A new method for model-based health economic evaluation utilising and extending moment-generating functions

**Provisional running head:** Moment-generating functions for decision models

**Author and corresponding author:** Snowsill, Tristan PhD [[orcid.org/0000-0001-7406-2819](https://orcid.org/0000-0001-7406-2819)]

**Affiliation:** Health Economics Group, University of Exeter, Exeter UK

**Address:** Room 1.11, South Cloisters, St Luke's Campus, Heavitree Road, Exeter EX1 2LU, United Kingdom

This method was presented at the Health Economists' Study Group meeting in Bristol, June 20-22, 2018. The current manuscript is substantially redrafted from the previously presented manuscript following comments from colleagues and peer review.

Neither the author nor the department/institution received funding for this work. There are no potential conflicts of interest.

## **Abstract**

### **Background**

Health economic evaluations frequently include projections for lifetime costs and health effects using modelling frameworks such as Markov modelling or discrete event simulation (DES). Markov models typically cannot represent events whose risk is determined by the length of time spent in state (sojourn time) without the use of tunnel states. DES is very flexible, but introduces Monte Carlo variation which can significantly limit the complexity of model analyses.

### **Methods**

We present a new methodological framework for health economic modelling which is based on, and extends, the concept of moment-generating functions (MGFs) for time-to-event random variables. When future costs and health effects are discounted, MGFs can be used to very efficiently calculate the total discounted life years spent in a series of health states. Competing risks are incorporated into the method. This method can also be used to calculate discounted costs and health effects when these payoffs are constant per unit time, one-off or exponential with regard to time. MGFs are extended to additionally support costs and health effects which are polynomial with regard to time (as in a commonly used model of population norms for EQ-5D utility).

### **Worked example**

A worked example is used to demonstrate application of the new method in practice, and to compare it to Markov modelling and DES. Results are compared in terms of convergence and accuracy, and computation times are compared. R code and an Excel workbook are provided.

### **Conclusions**

The MGF method can be applied to health economic evaluations in the place of Markov modelling or DES and has certain advantages over both.

## Introduction

Mathematical models are frequently used in health economic evaluations to extrapolate beyond observed data for estimates of lifetime costs and effects.<sup>1</sup> The observed data may come from experimental trials, as well as from observational sources, such as registries.<sup>2</sup> Often the effects of interventions are measured in life years or quality-adjusted life years (QALYs), as particularly the latter allows for comparisons of cost-effectiveness across different interventions, populations and diseases.<sup>3</sup>

As spending and health effects usually do not happen simultaneously, there is a need to consider the time-preferences for costs and effects. It is near universal practice to use a constant rate of discounting<sup>4</sup> (e.g., 3.5% for evaluations following the National Institute for Health and Care Excellence [NICE] reference case<sup>5</sup>).

A very common approach in modelling is to represent the health status of any given individual using one of a finite set of mutually exclusive health states.<sup>6</sup> The individual may transition between these states as aspects of their health status or care provision change. Such models are termed *state transition models*. For example, in a model of treatments for renal failure, there may be separate health states for patients with a kidney transplant, patients receiving haemodialysis, patients receiving peritoneal dialysis, and patients who have died. Health states are associated with probabilities/rates/time-to-event distributions of transitions to other health states, as well as payoffs (typically costs and health state utility values). The most common implementations of state transition models are Markov models (cohort simulation and microsimulation) and Discrete Event Simulation (DES). Another modelling method, partitioned survival analysis, resembles Markov cohort simulation but transitions are not modelled, rather the state membership over time is explicitly modelled according to a set of survival curves.<sup>7</sup>

The cohort-based methods are generally very computationally tractable but lack flexibility, while the simulation-based methods are highly flexible but introduce Monte Carlo variation. Monte Carlo variation can be minimised by producing large numbers of simulations, but this can be computationally costly, and even when minimised Monte Carlo variation can represent a challenge for model verification.

Probabilistic sensitivity analysis (PSA) is frequently used to estimate the joint impact of uncertainty across all parameters, and requires an extra “loop” of calculations to explore the joint parameter space.<sup>8</sup> Traditional partial expected value of perfect information (pEVPI) analyses require an additional loop, which means a three-level loop in the case of a patient-level simulation model, and model calibration exercises require repeated evaluation of expected outputs from a model, creating a similar computational challenge (many numerical methods for optimisation require estimates of the partial derivative of the function with respect to the parameter space, which cannot be accurately estimated in the presence of random noise). While some advances have been made to avoid the additional loop for pEVPI,<sup>9</sup> it remains true that in general an analyst would prefer to obtain model outputs for a given set of input parameters which is unaffected by Monte Carlo variation.

This paper sets out a method which can be used to compute expected life years, lifetime costs and QALYs when there is discounting. The method uses and extends MGFs. By virtue of considering state transition models where the times to transition are modelled (as opposed to probabilities or rates), and utilising MGFs, the method shares many principles with statistical flowgraph modelling,<sup>10</sup> which explicitly describes processes in terms of branch probabilities (like a decision tree or Markov model) and MGFs for waiting time distributions. The method in this paper does not rely on statistical flowgraph modelling techniques, but instead is optimised to consider cumulative discounted costs and outcomes, as these are of interest in health economic applications.

The substantial advantage of the method described in this paper over Markov cohort simulation is the ability to model transitions, costs and QALY weights which are dependent on the length of time spent in a particular state (i.e., it overcomes the Markov memoryless property). Individual patient simulation methods (e.g., Markov microsimulation and DES) are not restricted by the Markov memoryless property but they introduce Monte Carlo variation, with the challenges outlined above. The method set out in this paper achieves greater flexibility over Markov cohort simulation while still producing deterministic results.

## Methods

This section begins with a description of MGFs and a simple motivating example for why MGFs arise naturally when considering discounted outcomes in models. An introduction to statistical flowgraph models is provided next, followed by a description of how the methods applied in the motivating example can be extended to consider not just discounted life years, but discounted costs and QALYs when the rates at which these are accrued vary according to state, time within state and time since the model origin. We next consider alternative competing risk formulations besides the one assumed in statistical flowgraph modelling. The section concludes with notes on calculating MGFs in practice and an overall summary of the method.

### MGFs

The MGF,  $M_X(t)$ , of a random variable,  $X$ , is defined as:

$$M_X(t) \stackrel{\text{def}}{=} \mathbb{E}_X[e^{tX}] \quad (1)$$

For example, the MGF of an exponentially distributed random variable,  $X \sim \text{Exp}(\lambda)$  is  $\lambda/(\lambda - t)$  as shown below:

$$\begin{aligned} M_X(t) &= \mathbb{E}_X[e^{tX}] \\ &= \int_0^{\infty} f_X(x)e^{tx} dx \\ &= \int_0^{\infty} \lambda e^{-\lambda x} e^{tx} dx \\ &= \frac{\lambda}{\lambda - t} \int_0^{\infty} (\lambda - t)e^{-(\lambda-t)x} dx \\ &= \frac{\lambda}{\lambda - t} \end{aligned} \quad (2)$$

The final step of this derivation can be derived through usual means or by recognising that the integrand is the probability density function for an exponential random variable with rate  $\lambda - t$ , and so the integral across all its domain must equal 1.

### Discounting

As noted in the introduction it is near universal practice to use a constant rate of discounting. This is often presented as an annual discount rate, such that the discounted value of a cost  $C$  incurred  $X$  years in the future is:

$$C_d = \frac{C}{(1 + r_a)^X} \quad (3)$$

Where  $r_a$  is the annual discount rate, e.g., 0.035 for a discount rate of 3.5% per year.

It is more mathematically convenient to use continuous discounting formula which is equivalent:

$$C_d = C e^{-rX} \quad (4)$$

Where  $r$  is the continuous discount rate. These discount rates are readily calculated from each other:

$$\begin{aligned} C_d &= \frac{C}{(1 + r_a)^X} = C e^{-rX} \\ (1 + r_a)^{-X} &= e^{-rX} \\ (1 + r_a)^{-X} &= (e^r)^{-X} \\ 1 + r_a &= e^r \\ \ln(1 + r_a) &= r \end{aligned} \quad (5)$$

Throughout this paper we use the continuous discounting formula.

Note that if  $X$ , the time at which the cost is incurred, is a random variable, we can take the expectation of the discounted cost and then express it in terms of the MGF of  $X$ :

$$\mathbb{E}_X[C_d] = C \mathbb{E}_X[e^{-rX}] = C M_X(-r) \quad (6)$$

This is indeed the fundamental observation underpinning the method outlined in this paper: that quantities in a health economic evaluation with discounting can be expressed in terms of MGFs.

### **Motivating example**

#### **Part 1**

Consider a very simple two-state model in which patients are either alive or dead, and the hazard rate of death is a constant ( $\lambda$ ), i.e., the time-to-event (TTE) distribution for death is an exponential distribution. If we let  $X$  denote the time to death then  $f_X(x)$  and  $F_X(x)$  are the probability density and cumulative distribution functions respectively:

$$f_X(x) = \lambda e^{-\lambda x} \quad (7)$$

$$F_X(x) = 1 - e^{-\lambda x} \quad (8)$$

The life years lived (for a particular value of  $X$ ) is simply:

$$LY = \int_0^X dx = X \quad (9)$$

I.e., life years lived is a random variable (or a function of a random variable). Given this, life expectancy is:

$$\begin{aligned} \mathbb{E}_X[LY] &= \mathbb{E}_X[X] \\ &= \int_0^{\infty} x f_X(x) dx \\ &= \int_0^{\infty} \lambda x e^{-\lambda x} dx \\ &= \underbrace{[-x e^{-\lambda x}]_0^{\infty}}_{=0} + \lambda^{-1} \underbrace{\int_0^{\infty} \lambda e^{-\lambda x} dx}_{=1} \\ &= \lambda^{-1} \end{aligned} \quad (10)$$

Now consider discounted life years lived using the continuous discounting function given in Equation (4):

$$LY_d = \int_0^X e^{-rx} dx = \frac{1}{r} (1 - e^{-rX}) \quad (11)$$

This is also a (function of a) random variable, and its expectation can be obtained:

$$\mathbb{E}_X[LY_d] = \mathbb{E}_X \left[ \frac{1}{r} (1 - e^{-rX}) \right] = \frac{1}{r} (1 - \mathbb{E}_X[e^{-rX}]) \quad (12)$$

The quantity  $\mathbb{E}_X[e^{-rX}]$  is equal to the MGF of  $X$ ,  $M_X(t) \stackrel{\text{def}}{=} \mathbb{E}_X[e^{tX}]$ , evaluated at  $t = -r$ . For the exponential distribution the MGF was shown above to be

$$M_X(t) = \frac{\lambda}{\lambda - t} \quad (13)$$

So we can substitute this into Equation (12):

$$\begin{aligned} \mathbb{E}_X[LY_d] &= \frac{1}{r} (1 - M_X(-r)) \\ &= \frac{1}{r} \left( 1 - \frac{\lambda}{\lambda + r} \right) \\ &= \frac{1}{r} \left( \frac{\lambda + r - \lambda}{\lambda + r} \right) \\ &= \frac{1}{\lambda + r}. \end{aligned} \quad (14)$$

Note that we have not considered any transition probabilities (as in a discrete-time Markov model) or rates (as in a continuous-time Markov model), but only time-to-event variables (as in a discrete event simulation).

## Part 2

Now we consider a slightly more complex model in which there are three states: healthy, diseased and dead. We assume that transitions are possible between healthy and diseased, and between diseased and dead (i.e., we do not include recovery from disease or death from other causes). Let  $X_1$  denote the TTE variable for transitioning from healthy to diseased, and let  $X_2$  denote the TTE variable for transitioning from diseased to dead. Assume  $X_1$  and  $X_2$  are independent. Life years lived is then  $X_1 + X_2$ .

Life expectancy is calculated simply as:

$$\begin{aligned}\mathbb{E}[LY] &= \mathbb{E}_{X_1, X_2} \left[ \int_0^{X_1+X_2} dx \right] \\ &= \mathbb{E}_{X_1, X_2} [X_1 + X_2] \\ &= \mathbb{E}_{X_1} [X_1] + \mathbb{E}_{X_2} [X_2].\end{aligned}\tag{15}$$

Discounted life years are calculated as:

$$LY_d = \int_0^{X_1+X_2} e^{-rx} dx = \frac{1}{r} (1 - e^{-r(X_1+X_2)})\tag{16}$$

And discounted life expectancy is:

$$\begin{aligned}\mathbb{E}[LY_d] &= \mathbb{E}_{X_1, X_2} \left[ \frac{1}{r} (1 - e^{-r(X_1+X_2)}) \right] \\ &= \mathbb{E}_{X_1, X_2} \left[ \frac{1}{r} (1 - e^{-rX_1} e^{-rX_2}) \right] \\ &= \frac{1}{r} (1 - \mathbb{E}_{X_1, X_2} [e^{-rX_1} e^{-rX_2}]) \\ &= \frac{1}{r} (1 - \mathbb{E}_{X_1} [e^{-rX_1}] \mathbb{E}_{X_2} [e^{-rX_2}]) \\ &= \frac{1}{r} (1 - M_{X_1}(-r) M_{X_2}(-r)).\end{aligned}\tag{17}$$

This illustrates a key property of MGFs, that the MGF of the sum of two independent random variables is the product of their MGFs. It also illustrates that it is only necessary to evaluate the MGFs at single points.

In *Appendix 1* we demonstrate that the method applied above, with exponential TTE distributions, gives identical results to an equivalent Markov cohort simulation (in the



limit as the cycle length tends to zero) and to direct calculation of the state occupancy equations.

Exponential TTE distributions are easy to work with in most contexts, but the MGF method described in this manuscript can be applied with any distribution with a finite MGF at the necessary points.

### **Statistical flowgraph models**

Statistical flowgraph models<sup>10</sup> are a type of state transition model which can be understood in a similar manner to discrete-time Markov models, in that from any state there is a probability of transitioning to another state, or remaining in the current state. The key point of divergence from Markov models is that the time steps are not of fixed length, but are random time-to-event variables.

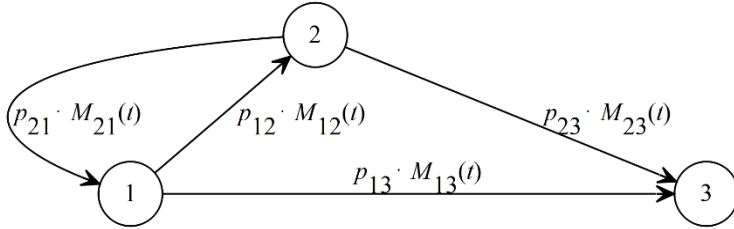
Although developed independently of statistical flowgraph models, there are similarities between the method proposed in this paper and statistical flowgraph models, so it is instructive to introduce them as an area for potential cross-fertilisation and to highlight differences for readers who may already be familiar with them.

Statistical flowgraph models have seen use in engineering applications, and have also been applied to health, but have not seen notable use in health economic modelling. Their focus is typically on deriving an overall MGF for a waiting time distribution of interest (e.g., survival distribution) and then “inverting” this into a probability density function.<sup>10</sup> Note that this is different from the approach described in this paper in which the MGF itself is evaluated and used within expressions to derive discounted outcomes.

Statistical flowgraph models are stochastic multistate models represented by directed graphs in which nodes represent states and edges represent transitions between those states (i.e., events). Each edge has a *transmittance*, which is the product of a transition probability and an MGF. The transition probability represents the probability that a modelled item (e.g., an individual) will transition along that edge (i.e., experience the given event) as opposed to any alternative edge from the current node. The MGF describes the time-to-event variable for how long the item waits in the state before transitioning along the edge, conditional on the item transitioning along that edge.

**Fig. 1** presents an example statistical flowgraph model with three states labelled 1, 2 and 3. These could respectively represent healthy, diseased and dead in a health

economic model where it is possible to recover from disease and to die while healthy or diseased. The transmittance labels are such that  $p_{ij}$  is the transition probability of transitioning from state  $i$  to state  $j$  and  $M_{ij}(t)$  is the MGF for the waiting time in state  $i$  prior to transition to state  $j$ .



**Figure 1: Example statistical flowgraph model**

In such an example we are likely interested to know the distribution of time spent prior to reaching state 3, given an item starts in state 1 (or 2). There are manual reduction procedures for solving a statistical flowgraph model (determining the MGF for a waiting time of interest), as well as a procedure based on linear algebra, Mason's rule.<sup>11</sup>

If we are interested in finding the MGF for the waiting time distribution from state 1 to state 3, we will eventually identify that

$$\mathcal{M}(t) = \frac{p_{13}M_{13}(t) + p_{12}p_{23}M_{12}(t)M_{23}(t)}{1 - p_{12}p_{21}M_{12}(t)M_{21}(t)}. \quad (18)$$

At this stage, an analyst would typically use numerical methods to invert the MGF into a PDF for the waiting time distribution, however, if we are interested in the expected discounted life years from state 1 to state 3, we can simply employ the approach we have used before:

$$\begin{aligned} \mathbb{E}[LY_d] &= \frac{1}{r}(1 - \mathcal{M}(-r)) \\ &= \frac{1}{r} \left( 1 - \frac{p_{13}M_{13}(-r) + p_{12}p_{23}M_{12}(-r)M_{23}(-r)}{1 - p_{12}p_{21}M_{12}(-r)M_{21}(-r)} \right). \end{aligned} \quad (19)$$

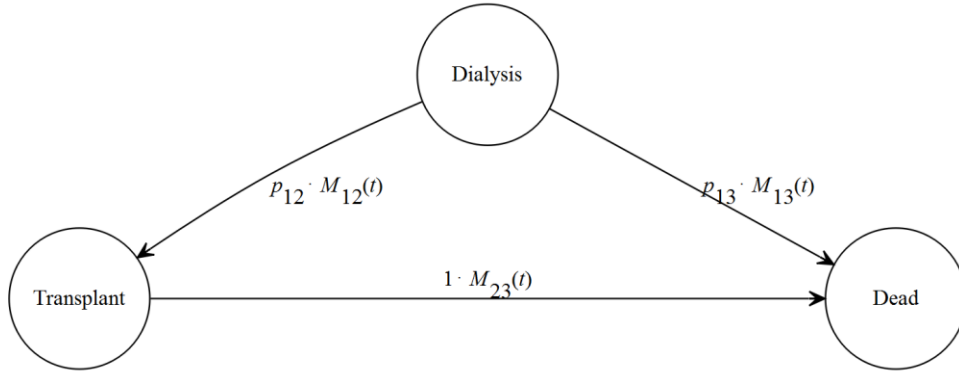
Where  $\mathcal{M}(\cdot)$  is the MGF for the solved flowgraph model.

### **Beyond discounted life years**

In many health economic evaluations we are interested to know the (incremental) discounted costs and quality-adjusted life years (QALYs) associated with an intervention. Costs and QALYs (or other "payoffs" of interest) are typically not accrued

at a constant rate but vary according to the health state of an individual, how long the individual has been in a health state and how old the individual is.

Consider a simple model of renal failure as shown in **Fig. 2**. Individuals start the model with end-stage renal failure and in receipt of dialysis, which has significant ongoing costs and poor quality of life. Some individuals receive a transplant, which has a significant upfront cost and some ongoing costs, a better quality of life than dialysis and better life expectancy. Let  $c_1$  and  $u_1$  denote the cost rate and QALY weight in dialysis, let  $c_{12}$  denote the cost of transplantation,  $c_2$  and  $u_2$  the cost rate and QALY weight for transplanted patients. For now assume no cost of death and that cost rates and QALY weights are constant within each state.



**Figure 2: Simple renal failure model**

Since there are no cycles it is easy to deduce that there are two paths to death: Dialysis  $\rightarrow$  Transplant  $\rightarrow$  Death (with probability  $p_{12}$ ) and Dialysis  $\rightarrow$  Death (with probability  $p_{13} = 1 - p_{12}$ ).

If the patient dies without transplantation, they live for  $X_{13}$  years accruing costs at a rate of  $c_1$  and QALYs at a rate of  $u_1$ . Their discounted costs and QALYs can be written as a function of  $X_{13}$ :

$$\text{Costs} = \int_0^{X_{13}} c_1 e^{-rx} dx = \frac{c_1}{r} (1 - e^{-rX_{13}}) \quad (20)$$

$$\text{QALYs} = \int_0^{X_{13}} u_1 e^{-rx} dx = \frac{u_1}{r} (1 - e^{-rX_{13}}) \quad (21)$$

The expectation of these is taken by replacing  $e^{-rX_{13}}$  with  $\mathbb{E}_{X_{13}}[e^{-rX_{13}}] = M_{X_{13}}(-r)$ .

If instead the patient is transplanted, they live on dialysis for  $X_{12}$  years and then with a transplant for  $X_{23}$  years. The cost of their transplantation is incurred at time  $X_{12}$  and is

therefore expected to be discounted by  $\mathbb{E}_{X_{12}}[e^{-rX_{12}}] = M_{X_{12}}(-r)$ . The discounted costs and QALYs accrued post-transplantation are:

$$\begin{aligned} \text{Costs} &= \int_{X_{12}}^{X_{12}+X_{23}} c_2 e^{-rx} dx \\ &= \frac{c_2}{r} (e^{-rX_{12}} - e^{-r(X_{12}+X_{23})}) \end{aligned} \quad (22)$$

$$\begin{aligned} \text{QALYs} &= \int_{X_{12}}^{X_{12}+X_{23}} u_2 e^{-rx} dx \\ &= \frac{u_2}{r} (e^{-rX_{12}} - e^{-r(X_{12}+X_{23})}) \end{aligned} \quad (23)$$

Once again, the expectation is taken by replacing  $e^{-rX_{12}}$  with  $\mathbb{E}_{X_{12}}[e^{-rX_{12}}] = M_{X_{12}}(-r)$  and also replacing  $e^{-r(X_{12}+X_{23})}$  with  $\mathbb{E}_{X_{12}, X_{23}}[e^{-r(X_{12}+X_{23})}] = M_{X_{12}}(-r)M_{X_{23}}(-r)$ .

In total therefore, we have the following lifetime expected discounted costs and QALYs:

$$\begin{aligned} \mathbb{E}[\text{Costs}] &= p_{12} \left[ \frac{c_1}{r} (1 - M_{X_{12}}(-r)) + c_{12} M_{X_{12}}(-r) \right. \\ &\quad \left. + \frac{c_2}{r} (M_{X_{12}}(-r) (1 - M_{X_{23}}(-r))) \right] \\ &\quad + (1 - p_{12}) \left[ \frac{c_1}{r} (1 - M_{X_{13}}(-r)) \right] \end{aligned} \quad (24)$$

$$\begin{aligned} \mathbb{E}[\text{QALYs}] &= p_{12} \left[ \frac{u_1}{r} (1 - M_{X_{12}}(-r)) \right. \\ &\quad \left. + \frac{u_2}{r} (M_{X_{12}}(-r) (1 - M_{X_{23}}(-r))) \right] \\ &\quad + (1 - p_{12}) \left[ \frac{u_1}{r} (1 - M_{X_{13}}(-r)) \right] \end{aligned} \quad (25)$$

In addition to one-off payoffs (e.g., the cost of transplantation above) and constant per-state payoffs (e.g., the QALY weight in dialysis above), we can quite easily incorporate payoff functions which include exponential functions of time.

For example, if the QALY weight following surgery is initially low but soon improves to a higher baseline, we may use a utility function  $u(x) = u_h - (u_h - u_l)e^{-ax}$ , where  $u_l$  and  $u_h$  are the low and high QALY weights and  $a$  is a parameter that determines how quickly QALY weights recover. In this example  $x$  would represent time since surgery (i.e., time in a post-surgery state), so expected discounted QALYs could be (assuming we are considering QALYs accrued in a state entered at time  $X_{12}$  and exited at time  $X_{12} + X_{23}$ ):

$$\begin{aligned}
\mathbb{E}[\text{QALYs}] &= \mathbb{E} \left[ \int_{X_{12}}^{X_{12}+X_{23}} u(x - X_{12}) e^{-rx} dx \right] \\
&= \mathbb{E} \left[ \int_{X_{12}}^{X_{12}+X_{23}} (u_h - (u_h - u_l) e^{-a(x-X_{12})}) e^{-rx} dx \right] \\
&= \mathbb{E} \left[ \int_{X_{12}}^{X_{12}+X_{23}} u_h e^{-rx} - (u_h - u_l) e^{-a(x-X_{12})-rx} dx \right] \\
&= \mathbb{E} \left[ \int_{X_{12}}^{X_{12}+X_{23}} u_h e^{-rx} - (u_h - u_l) e^{aX_{12}} e^{-(a+r)x} dx \right] \\
&= \mathbb{E} \left[ u_h \int_{X_{12}}^{X_{12}+X_{23}} e^{-rx} dx - (u_h - u_l) e^{aX_{12}} \int_{X_{12}}^{X_{12}+X_{23}} e^{-(a+r)x} dx \right] \\
&= \mathbb{E} \left[ u_h \int_{X_{12}}^{X_{12}+X_{23}} e^{-rx} dx \right] \\
&\quad - \mathbb{E} \left[ (u_h - u_l) e^{aX_{12}} \int_{X_{12}}^{X_{12}+X_{23}} e^{-(a+r)x} dx \right] \\
&= \mathbb{E} \left[ \frac{u_h (e^{-rX_{12}} - e^{-r(X_{12}+X_{23})})}{r} \right] \\
&\quad - \mathbb{E} \left[ \frac{(u_h - u_l) e^{aX_{12}} (e^{-(a+r)X_{12}} - e^{-(a+r)(X_{12}+X_{23})})}{a+r} \right] \\
&= \mathbb{E} \left[ \frac{u_h (e^{-rX_{12}} - e^{-r(X_{12}+X_{23})})}{r} \right] \\
&\quad - \mathbb{E} \left[ \frac{(u_h - u_l) (e^{-rX_{12}} - e^{-r(X_{12}+X_{23})-aX_{23}})}{a+r} \right] \\
&= \frac{u_h (M_{X_{12}}(-r) (1 - M_{X_{23}}(-r)))}{r} \\
&\quad - \frac{(u_h - u_l) (M_{X_{12}}(-r) (1 - M_{X_{23}}(-r - a)))}{a+r}
\end{aligned} \tag{26}$$

### Extended moment-generating functions and polynomial payoffs

We may also want to incorporate polynomial functions for payoffs. A common example would be to have baseline QALY weights be a polynomial function of age, e.g., the quadratic model for population norms of EQ-5D utility described by Ara and Brazier.<sup>12</sup> Attempting to employ the method described so far in the case of polynomial payoffs, after repeated application of integration by parts, results in terms which are the product of a power and exponential function of a random variable, i.e.,  $X^j e^{-rX}$ . The expectation of this is *not*  $\mathbb{E}[X^j] \mathbb{E}[e^{-rX}]$ , since these two components are not independent. Instead, we newly define the extended moment-generating function (EMGF) as:

$$M_X^n(t) := \mathbb{E}[X^n e^{tX}] \quad (27)$$

And derive the following property of the EMGF:

If  $S = \sum_i a_i X_i$  is a linear combination of independent random variables, and  $n$  is a non-negative integer, then:

$$\begin{aligned} M_S^n(t) &= \mathbb{E} \left[ \left( \sum_i a_i X_i \right)^n e^{t \sum_i a_i X_i} \right] \\ &= \sum_{k_1+k_2+\dots+k_m=n} \frac{n!}{k_1! k_2! \dots k_m!} a_i^{k_i} M_{X_i}^{k_i}(a_i t) \end{aligned} \quad (28)$$

This summation is conducted over all possible combinations of non-negative integer  $k_i$  where they sum to  $n$ .

This means the EMGF for a linear combination of independent random variables can be written as a linear combination of their EMGFs.

Consider an example where health state utility value declines linearly once in a state:

$$\begin{aligned} Q_d &= \int_{X_1}^{X_1+X_2} (a - b(x - X_1)) e^{-rx} dx \\ &= \frac{e^{-rX_1}}{r^2} (brX_2 e^{-rX_2} + (ar - b)(1 - e^{-rX_2})) \end{aligned} \quad (29)$$

$$\mathbb{E}[Q_d] = \frac{M_{X_1}(-r)}{r^2} \left( brM_{X_2}^1(-r) + (ar - b)(1 - M_{X_2}(-r)) \right) \quad (30)$$

By combining constant, one-off, exponential and polynomial payoffs we can define models with significant flexibility that would not be possible in traditional Markov cohort simulations.

### Flowgraph cycles

In the previous example (**Fig. 2**) there were no cycles ('cycles' is used here in the graph theoretical sense, i.e., paths in the flowgraph which return to a previously visited state, in contrast to time cycles as used in discrete Markov models). We might consider this unrealistic and want to include failure of a kidney graft, after which patients return to dialysis. To incorporate this we could add another edge to our model with a transmittance of  $p_{21} \cdot M_{21}(t)$ . The transmittance for the edge denoting death with a kidney transplant would change to  $p_{23} \cdot M'_{23}(t)$  (we use  $M'_{23}(t)$  since the conditional time-to-event distribution may change after introducing a new competing risk). Patients

can now in theory cycle infinitely many times between dialysis and a kidney transplant before dying, but in practice a geometric limit applies. In *Appendix 2* we go through the necessary steps to produce a formula for the discounted life years lived in dialysis. There is no reason why similar steps cannot be taken to calculate other discounted payoffs.

We note, however, that the mathematics are quite involved, and that it is likely that little is gained from having a model with a cycle versus a model where there are a finite number of retransplantations allowed (indeed this could be made more realistic than the model with cycles since the probability of obtaining a second or third transplant is likely different to the probability of obtaining an initial transplant).

Cycles should only be incorporated into a model where it is realistic to believe patients could cycle many times and that the parameters governing the transitions would not change with the number of cycles completed.

### **Alternative competing risk formulations**

So far we have considered the competing risk formulation used in statistical flowgraph modelling, which has also been described as the pattern mixture approach to competing risks.<sup>13</sup> In this formulation we use transition probabilities to determine which event takes place (i.e., a categorical distribution) and then a TTE distribution is defined (by its MGF) for the waiting time prior to that event. Alternative competing risk formulations have been used in data modelling and in simulations.

A simple (though not as flexible) alternative is to model event times as latent independent event times, and the earliest of these is the event which takes place (with the corresponding event time). This is a very typical approach in discrete event simulations.<sup>14</sup> These event times can be sampled according to shared covariates to account for population heterogeneity. For example, in a renal failure model we could model time to death on dialysis and time to transplantation as dependent on age (younger patients are sometimes prioritised for transplants but also have a higher life expectancy on dialysis). This approach is particularly desirable when producing a health economic model from aggregate data from multiple data sources. The approach is readily incorporated into the framework described in this paper since it is not computationally difficult to convert this competing risks formulation into the pattern mixture formulation. Further details are provided in *Appendix 3*.

Other notable frameworks are the Fine and Gray method of competing risk subhazard distributions<sup>15</sup> and the vertical modelling approach.<sup>16</sup> These have not yet been incorporated into the framework described in this paper.

### **Methods for evaluating MGFs**

Some probability distributions used in economic modelling have MGFs with closed forms. We have already seen the MGF of the exponential distribution and in *Appendix 4* we derive the MGFs and extended MGFs of the exponential, gamma, degenerate and continuous uniform distributions. Other parametric survival distributions frequently used in health economic evaluations (e.g., Weibull, log-normal, generalised gamma) do not have finite closed form MGFs.

Evaluating the MGF of a distribution at a particular value involves performing a single 1-dimensional integration with one improper limit (the lower limit is 0 because time-to-event distributions are non-negative, the upper limit is infinite because a lifetime horizon is assumed), which is not challenging for modern statistical programs which can automatically perform appropriate transformations and apply Gaussian quadrature techniques (e.g., the `integrate` function in R). Spreadsheet software typically does not include such functionality, but the transformations and Gaussian quadrature can nevertheless be readily implemented provided the nodes and weights for Gaussian quadrature (constants) are calculated in a suitable package. For simplicity, the transformation  $x = (1 + u)/(1 - u)$  maps the interval  $[0, \infty)$  onto  $(-1, 1)$ , which is suitable for Gauss–Legendre quadrature (see *Appendix 5*).

We can also calculate the MGF for non-parametric TTE distributions, specifically Kaplan–Meier curves (see *Appendix 4*). This can be very useful when observed survival is not well fitted by a parametric survival function.

### **Summary of the MGF method**

We now briefly summarise the procedure for conducting a model-based health economic evaluation using MGFs, the MGF method.

**Step 1: Conceptualise model in terms of health states and events**  
Following suitable methodology (e.g., Roberts et al.<sup>17</sup>), identify health states and any events which need to be represented. Events typically lead to transitions between health states.



### Step 2: Identify suitable waiting time distributions

Depending on data availability, waiting time distributions for events should preferably be identified using the pattern-mixture formulation as used in statistical flowgraph models. Failing this, independent TTE distributions should be identified for the events in the model.

### Step 3: Identify suitable payoff functions

Costs and QALYs are accumulated in the model either at a constant rate (for each state), as one-off (e.g., surgery cost), or with rates which are expressed in polynomial or exponential terms with regards to time (waiting time in the state or time since the start of the model). Linear combinations are also allowable. Suitable payoff functions should be identified, bearing in mind that polynomial functions will require calculations of EMGFs.

### Step 4: Develop expressions for total discounted payoffs

Expressions for total discounted payoffs are developed in terms of (extended) MGFs.

Take each payoff in each state one at a time. Let  $S$  be the sum of the time-to-event variables for the events which resulted in reaching the current state and let  $X^*$  be the time-to-event variable for the event which results in exiting the current state. Let  $x$  denote the time since the start of the model. The (undiscounted) payoff function identified in Step 3 is now written as  $f(x, S)$ , noting that time in the current state is  $x - S$ . To discount it, multiply by  $e^{-rx}$ . The discounted payoff accrued in the state is then found by algebraically integrating this discounted payoff function from  $S$  to  $S + X^*$ . This produces the accrued payoff as a function of the random variables  $S$  and  $X^*$ . The expectation is then taken, relying on  $\mathbb{E}[e^{-rS}] = M_S(-r)$ , which will be the products of the MGFs for the random TTE variables which sum to  $S$  (assuming these are independent) and similar expressions based on the extended MGF.

Total discounted payoffs are obtained by combining all such expressions, weighted according to the probabilities of those payoffs being accrued.

### Step 5: Evaluate numerically

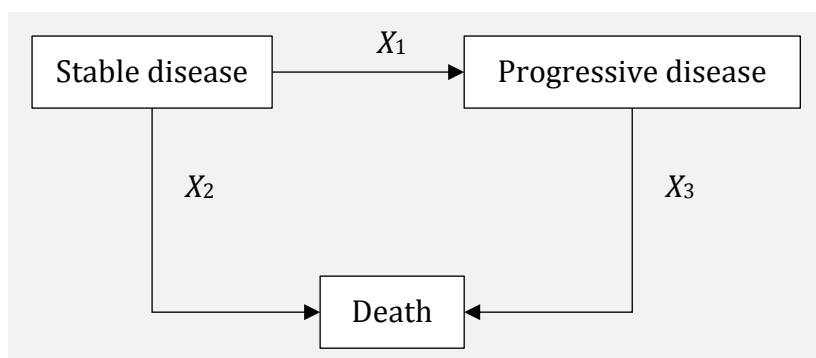
(Extended) MGFs with closed forms should be evaluated algebraically and numerical integration techniques should be used for MGFs without closed forms to produce numerical estimates of total discounted payoffs.



## Example application

We present a worked example of the application of the MGF method (in particular Steps 4 and 5 described above), with comparison to evaluation using a Markov cohort simulation (with tunnel states), Markov microsimulation and DES. We include code listings in *Appendix 6* for all four methods in R version 3.5,<sup>18</sup> making use of the *heemod* package for the Markov modelling.<sup>19</sup> Furthermore fully working code has been uploaded to GitHub for the MGF and DES methods (<https://www.github.com/tristansnowsill/mgf-example/>). We additionally provide an Excel 2013 (Microsoft Corporation; Redmond, WA) implementation of the MGF method utilising Gauss-Legendre quadrature throughout.<sup>20</sup>

The example model includes two health states (stable disease and progressive disease) and the death state (**Fig. 3**). The sojourn-dependent time-to-event distributions are  $X_1$  (representing disease progression),  $X_2$  (death from stable disease) and  $X_3$  (post-progression survival). These are modelled by Weibull, Gompertz and log-normal distributions respectively. Note that in this model these TTE distributions are assumed to be independent of each other and we do not specify branch probabilities as would be typical in a statistical flowgraph model.



**Figure 3: Model diagram**

The model includes one-off costs for progression and for death, and constant cost rates in the stable disease and progressive disease states. It also includes age-dependent baseline utility using a quadratic formula and constant utility multipliers for stable and progressive disease.

It should be noted that there is only one aspect of this model which cannot be represented faithfully in a Markov cohort simulation without tunnel states, which is the

log-normal distribution for survival in the progressive disease state, so this is in no way a pathological example.

## Methods

### MGF method

In the model there are two different paths which individuals can take. They either suffer from disease progression and then death, or they die without disease progression. We let  $p$  denote the probability of the first path being taken.

With probability  $p$  the following costs are incurred:

- $C_I$ : A steady cost accrued prior to progression (between  $x = 0$  and  $x = X_1$ )
- $C_{II}$ : A one-off cost at time of progression (at  $x = X_1$ )
- $C_{III}$ : A steady cost accrued following progression (between  $x = X_1$  and  $x = X_1 + X_3$ )
- $C_{IV}$ : A one-off cost at time of death (at  $x = X_1 + X_3$ )

Also with probability  $p$  the following QALYs are accrued:

- $B_I$ : QALYs accrued prior to progression (between  $x = 0$  and  $x = X_1$ )
- $B_{II}$ : QALYs accrued following progression (between  $x = X_1$  and  $x = X_1 + X_3$ )

With probability  $(1 - p)$  the following costs are incurred:

- $C_V$ : A steady cost accrued prior to death (between  $x = 0$  and  $x = X_2$ )
- $C_{VI}$ : A one-off cost at time of death (at  $x = X_2$ )

Also with probability  $(1 - p)$  the following QALYs are accrued:

- $B_{III}$ : QALYs accrued prior to death (between  $x = 0$  and  $x = X_2$ )

We consider the costs first:

$$C_I = \int_0^{X_1} c_{stable} e^{-rx} dx = \frac{c_{stable}}{r} (1 - e^{-rX_1}) \quad (31)$$

$$C_{II} = c_{progression} e^{-rX_1} \quad (32)$$

$$\begin{aligned}
C_{III} &= \int_{X_1}^{X_1+X_3} c_{progressive} e^{-rx} dx \\
&= \frac{c_{progressive}}{r} (e^{-rX_1} - e^{-r(X_1+X_3)})
\end{aligned} \tag{33}$$

$$C_{IV} = c_{death} e^{-r(X_1+X_3)} \tag{34}$$

$$C_V = \int_0^{X_2} c_{stable} e^{-rx} dx = \frac{c_{stable}}{r} (1 - e^{-rX_2}) \tag{35}$$

$$C_{VI} = c_{death} e^{-rX_2} \tag{36}$$

Where  $c_{stable}$  and  $c_{progressive}$  are the cost rates in the stable and progressive disease states,  $c_{progression}$  is the one-off cost associated with disease progression and  $c_{death}$  is the one-off cost associated with death.

Next we consider the QALYs, where baseline age-dependent utility is given by  $u(x) = u_0 + u_1x + u_2x^2$  and utility is scaled by  $v_{stable}$  in the stable disease state and  $v_{progressive}$  in the progressive disease state.

First we solve a ‘‘helper’’ integral:

$$\begin{aligned}
U(A, B) &= \int_A^B u(x) e^{-rx} dx \\
&= \int_A^B (u_0 + u_1x + u_2x^2) e^{-rx} dx \\
&= \frac{e^{-rA}}{r^3} ((u_0 + u_1A + u_2A^2)r^2 + (u_1 + 2u_2A)r + (2u_2)) \\
&\quad - \frac{e^{-rB}}{r^3} ((u_0 + u_1B + u_2B^2)r^2 + (u_1 + 2u_2B)r + (2u_2))
\end{aligned} \tag{37}$$

Now:

$$B_I = \int_0^{X_1} v_{stable} u(x) e^{-rx} dx = v_{stable} U(0, X_1) \tag{38}$$

$$\begin{aligned}
B_{II} &= \int_{X_1}^{X_1+X_3} v_{progressive} u(x) e^{-rx} dx \\
&= v_{progressive} U(X_1, X_1 + X_3)
\end{aligned} \tag{39}$$

$$B_{III} = \int_0^{X_2} v_{stable} u(x) e^{-rx} dx = v_{stable} U(0, X_2) \tag{40}$$

Before we take the expected values of these quantities using MGFs we need to realise that there has been some abuse of notation: we have used  $X_1$  and  $X_2$  in our equations above where in fact these should be  $(X_1 | X_1 < X_2)$  and  $(X_2 | X_2 < X_1)$  since progression and death without progression are competing events. Following the methods in *Appendix 4* we find:

$$\begin{aligned}
p &= \mathbb{E}[\mathbf{1}_{X_1 < X_2}] \\
&= \iint_{X_1, X_2} \mathbf{1}_{x_1 < x_2} f_{X_1}(x_1) f_{X_2}(x_2) dx_2 dx_1 \\
&= \int_0^{\infty} f_{X_1}(x_1) (1 - F_{X_2}(x_1)) dx_1
\end{aligned} \tag{41}$$

$$\begin{aligned}
f_{(X_1 | X_1 < X_2)}(x) &\propto f_{X_1}(x) (1 - F_{X_2}(x)) \\
&= \frac{1}{p} f_{X_1}(x) (1 - F_{X_2}(x))
\end{aligned} \tag{42}$$

$$\begin{aligned}
f_{(X_2 | X_2 < X_1)}(x) &\propto f_{X_2}(x) (1 - F_{X_1}(x)) \\
&= \frac{1}{1-p} f_{X_2}(x) (1 - F_{X_1}(x))
\end{aligned} \tag{43}$$

We produce a table of extended MGFs evaluated at  $t = -r$  for  $n = 0, 1, 2$ , as shown in **Table 1**. The bottom row is formed from the rows above following Equation (28).

We now have all the pieces we need to derive total expected discounted costs and QALYs:

$$\begin{aligned}
\mathbb{E}[\text{Discounted costs}] &= p\mathbb{E}[C_I + C_{II} + C_{III} + C_{IV}] \\
&\quad + (1-p)\mathbb{E}[C_V + C_{VI}]
\end{aligned} \tag{44}$$

$$\mathbb{E}[\text{Discounted QALYs}] = p\mathbb{E}[B_I + B_{II}] + (1-p)\mathbb{E}[B_{III}] \tag{45}$$

By linearity of expectation we can consider  $\mathbb{E}[C_I]$ ,  $\mathbb{E}[C_{II}]$ , ... and  $\mathbb{E}[B_I]$ , ... separately. For example:

$$\begin{aligned}
\mathbb{E}[C_{III}] &= \mathbb{E}\left[\frac{C_{progressive}}{r} (e^{-rX_1} - e^{-r(X_1+X_3)})\right] \\
&= \frac{C_{progressive}}{r} (\mathbb{E}[e^{-rX_1}] - \mathbb{E}[e^{-r(X_1+X_3)}]) \\
&= \frac{C_{progressive}}{r} (m_1^{(0)} - m_1^{(0)} m_3^{(0)})
\end{aligned} \tag{46}$$

$$\mathbb{E}[B_{II}] = \mathbb{E}[v_{progressive} U(X_1, X_1 + X_3)] \tag{47}$$

$$\begin{aligned}
&= v_{progressive} \mathbb{E} \left[ \frac{e^{-rX_1}}{r^3} \left( (u_0 + u_1X_1 + u_2X_1^2)r^2 + (u_1 + 2u_2X_1)r + (2u_2) \right) \right. \\
&\quad \left. - \frac{e^{-r(X_1+X_3)}}{r^3} \left( (u_0 + u_1(X_1 + X_3) + u_2(X_1 + X_3)^2)r^2 \right. \right. \\
&\quad \left. \left. + (u_1 + 2u_2(X_1 + X_3))r + (2u_2) \right) \right] \\
&= \frac{v_{progressive}}{r^3} \left[ \left( (u_0m_1^{(0)} + u_1m_1^{(1)} + u_2m_1^{(2)})r^2 + (u_1m_1^{(0)} + 2u_2m_1^{(1)})r \right. \right. \\
&\quad \left. \left. + (2u_2m_1^{(0)}) \right) \right. \\
&\quad \left. - \left( (u_0m_1^{(0)}m_3^{(0)} + u_1(m_1^{(0)}m_3^{(1)} + m_1^{(1)}m_3^{(0)}) \right. \right. \\
&\quad \left. \left. + u_2(m_1^{(0)}m_3^{(2)} + 2m_1^{(1)}m_3^{(1)} + m_1^{(2)}m_3^{(0)}) \right) r^2 \right. \\
&\quad \left. + (u_1m_1^{(0)}m_3^{(0)} + 2u_2(m_1^{(0)}m_3^{(1)} + m_1^{(1)}m_3^{(0)}))r \right. \\
&\quad \left. + (2u_2m_1^{(0)}m_3^{(0)}) \right) \left. \right]
\end{aligned}$$

### Markov cohort simulation

For comparison, a discrete time Markov model was constructed using the `heemod` package in R with the use of tunnel states.

Sojourn-dependent transition probabilities were estimated by calculating the cumulative hazards for competing risks at the start and end of the cycle, and converting these into transition probabilities assuming constant competing hazard rates within each cycle. This is more accurate than, e.g., applying transition probabilities estimated from hazard rates at the start, midpoint or end of each cycle.

The implementation of tunnel states in `heemod` means that the transition probabilities in the end state of the tunnel are based on the maximum sojourn time covered by the tunnel states, i.e., there is no attempt to fit the final transition probability to the behaviour of the survival curve beyond the tunnel duration.

Cycle lengths of 1 to 12 months were explored as well as maximum sojourn cycle memory of 1 (no tunnel states), 2 (one tunnel state), 4 (three tunnel states, etc.), 8, 16

and 32. The life table method of estimating state membership between cycles was used.<sup>21</sup>

### Markov microsimulation

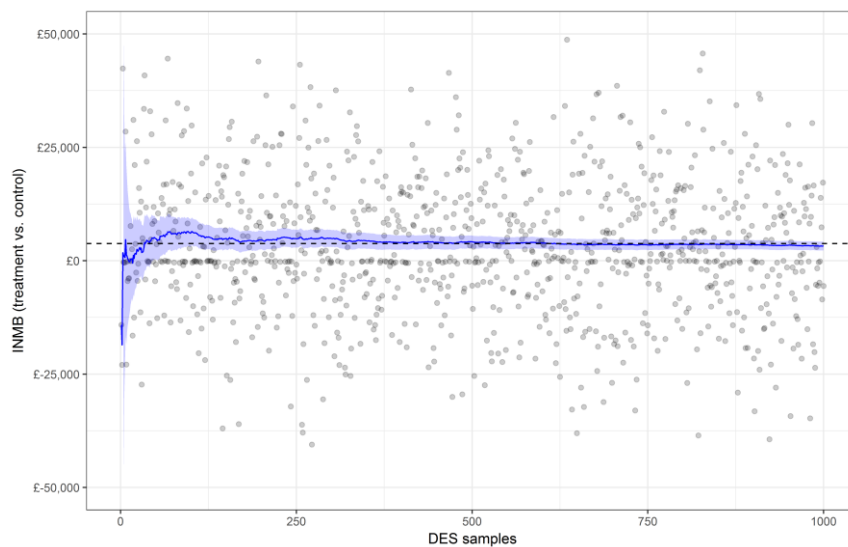
Also for comparison, a discrete time Markov microsimulation was built in R using the vectorisation approach described by Krijkamp et al.<sup>22</sup> Sojourn-dependent transition probabilities were calculated as for the Markov cohort simulation. A cycle length of three months was used. Costs and QALYs were calculated assuming that transitions occur at the end of each cycle. No half cycle correction was applied.

### DES

As a final comparator, a DES model was constructed using efficient vectorised operations. Variance reduction was included by using common samples of  $X_2$  and  $X_3$  across the control and treatment arms.

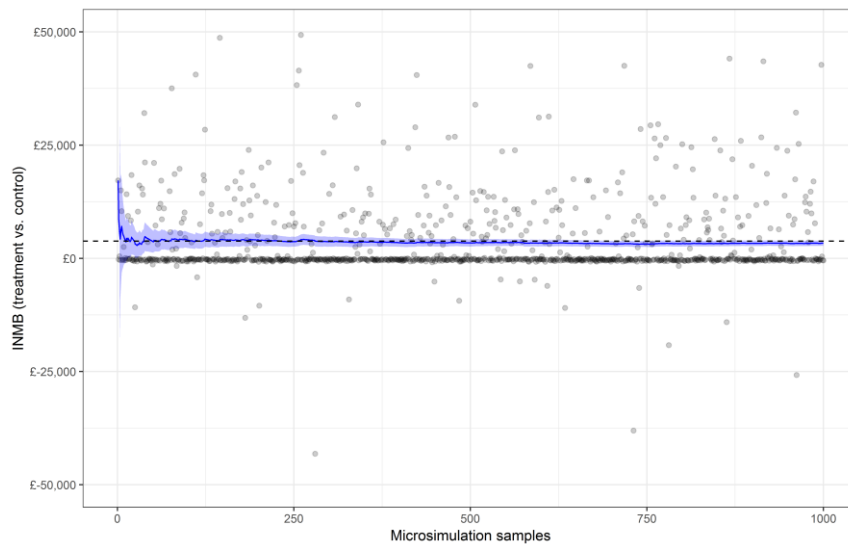
## Results

The MGF method gave consistent results with the discrete event simulation (see **Table 2** and **Fig. 4**). It was very efficient, requiring only 17 1-dimensional numerical integrals to be evaluated through Gaussian quadrature.



(a)





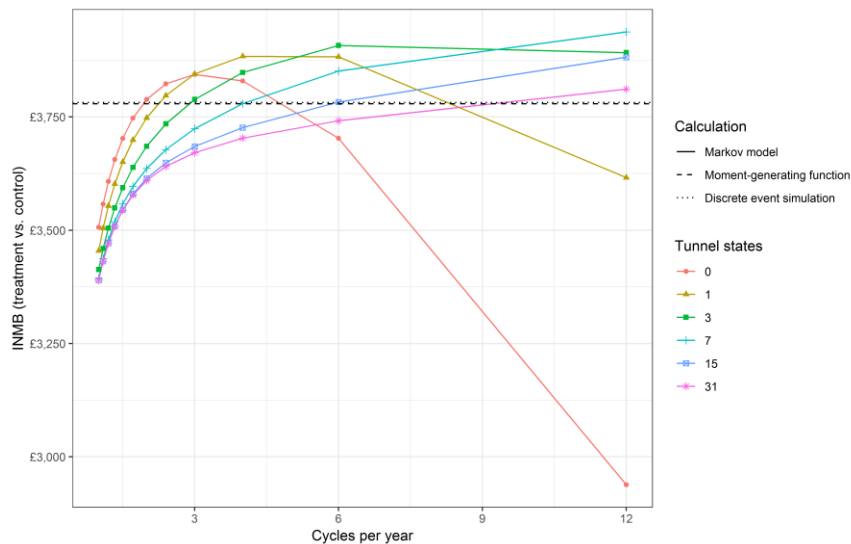
(b)

**Figure 4: Simulation ((a) DES and (b) Markov microsimulation) results**

Key: Points are individual simulation samples, the solid line is the cumulative mean incremental net monetary benefit, the shaded ribbon is the cumulative 95% confidence interval based on 1-sample t statistic, the dashed line is the mean incremental net monetary benefit as calculated by the MGF method

Even with 100,000 simulations and reasonable variance reduction measures, the discrete event simulation still produced a somewhat imprecise estimate of economic value.

The Markov cohort simulation approach was able to reach a fair approximation of the true results when a large number of tunnel states were used and a short cycle length was employed (see **Table 2** and **Fig. 5**), but the approximation was poor when a longer cycle length was used and/or no/few tunnel states were employed. There is a trade-off that as the cycle length is reduced, more tunnel states are required to represent the same portion of the survival curve.



**Figure 5: Comparison of Markov models with different cycle lengths and numbers of tunnel states**

#### Execution time

Execution time was measured using the `microbenchmark` package, with one hundred replications on a laptop running R v3.5.1 (R Foundation for Statistical Computing) using RStudio v1.2 (RStudio, Inc.). The laptop was running Windows 7 (Microsoft Corporation) with an Intel Core i7 processor running at 2.6 GHz and with 16 GB RAM.

As shown in **Table 3**, the MGF method is faster than the other methods, even being comparable to DES with a low number of simulations (1,000). With 1,000 simulations, DES retained a Monte Carlo standard error of 12.7% of incremental net monetary benefit.

## Discussion

We have presented the moment generating function (MGF) method, a new method for calculating lifetime discounted costs and outcomes for health economic models, which is distinct from Markov modelling and discrete event simulation and which has advantages over both paradigms (see **Table 4** and *Appendix 7* for a detailed comparison). It can represent sojourn-dependent transition times and payoffs in a very computationally tractable manner, in contrast to Markov modelling. It provides precision with fast convergence and no Monte Carlo variation, in contrast to discrete event simulation. The ability to provide precise answers quickly also makes the MGF method well suited to analyses which are challenging when using discrete event simulation, such as value of information analyses and model calibration. In terms of how it fits in with an analyst's "workflow" the closest analogy is a Markov cohort simulation – there is no need to produce multiple iterations to obtain an answer for a single set of parameters, but when conducting sensitivity analyses (including probabilistic sensitivity analysis) it is necessary to re-run the model using the different sets of parameters.

The MGF method can be readily implemented in specialist statistical packages such as R and Stata, as well as in widely available spreadsheet software such as Excel (without the use of VBA code). There is no reason to think it could not also be incorporated into specialist modelling software such as TreeAge. Future work may include procedural generation of code for the MGF framework based on a description of the underlying model.

The MGF method has a noteworthy limitation, which it is hoped will be addressed in the future. It is not currently possible to include transitions based on wall time (except in the trivial case where wall time is equal to sojourn time, i.e., for an initial state with no possibility of return), or for risks to compete from different starting times. These limitations make it challenging to include, for example, general mortality within a model. In certain situations it is, however, more important to capture sojourn time-dependent transitions, such as in advanced cancers.

The method also assumes independence of the time-to-event random variables in the specification. This does not mean, though, that the risks of different events must be treated as independent, as explained in *Appendix 8*, because conditional independence

can be leveraged in this regard. For example, the risk of a patient with a haematological cancer receiving a stem cell transplant and the risk of the same patient dying from other causes are related according to the age and frailty of the patient, but may be independent across patients conditional on their age and frailty.

In addition to future work to address the issues described above, we will also attempt to develop methods for incorporating flexible spline TTE distributions<sup>23</sup> and alternative competing risk specifications.<sup>15, 16</sup>

## **Acknowledgements**

I wish to thank Antonieta Medina-Lara and Anne Spencer for their comments on this manuscript and an earlier manuscript. I thank Mark W Pennington for his comments on this manuscript and for discussing the earlier manuscript at the Health Economists' Study Group in Bristol, June 2018, and Nicky Welton for chairing that session and providing helpful comments, and all the attendees who contributed to the discussion. I also thank the anonymous peer reviewers for their incredibly helpful suggestions for redrafting the manuscript and pointing me towards statistical flowgraph model research.

### **Declaration of conflicting interests**

The author declares that there is no conflict of interest.

## References

1. Caro JJ, Briggs AH, Siebert U, Kuntz KM, ISPOR-SMDM Modeling Good Research Practices Task Force. Modeling good research practices--overview: a report of the ISPOR-SMDM Modeling Good Research Practices Task Force-1. *Med Decis Making*. 2012;32(5):667-77. doi: 10.1177/0272989X12454577
2. Cooper N, Sutton A, Ades A, Paisley S, Jones D. Use of evidence in economic decision models: Practical issues and methodological challenges. *Health Econ*. 2007;16(12):1277-86. doi: 10.1002/hec.1297
3. Drummond MF, Sculpher MJ, Torrance GW, O'Brien B, Stoddart GL. *Methods for the economic evaluation of health care programmes*. Oxford: Oxford University Press; 2005.
4. Weinstein MC, Siegel JE, Gold MR, Kamlet MS, Russell LB. Recommendations of the panel on cost-effectiveness in health and medicine. *JAMA*. 1996;276(15):1253-8.
5. National Institute for Health and Care Excellence. The reference case. 2013. In: *Guide to the methods of technology appraisal 2013* [Internet]. London: NICE. Available from: <https://www.nice.org.uk/process/pmg9/chapter/the-reference-case>.
6. Siebert U, Alagoz O, Bayoumi AM, Jahn B, Owens DK, Cohen DJ, et al. State-transition modeling: a report of the ISPOR-SMDM Modeling Good Research Practices Task Force-3. *Med Decis Making*. 2012;32(5):690-700. doi: 10.1177/0272989X12455463
7. Woods B, Sideris E, Palmer S, Latimer N, Soares M. NICE DSU Technical Support Document 19. Partitioned survival analysis for decision modelling in health care: a critical review 2017. Available from: <http://www.nicedsu.org.uk/>.
8. Briggs AH. Handling uncertainty in cost-effectiveness models. *Pharmacoeconomics*. 2000;17(5):479-500.
9. Strong M, Oakley JE, Brennan A. Estimating multiparameter partial expected value of perfect information from a probabilistic sensitivity analysis sample: a nonparametric regression approach. *Med Decis Making*. 2014;34(3):311-26. doi: 10.1177/0272989X13505910
10. Huzurbazar AV. *Flowgraph models for multistate time-to-event data*. New Jersey: John Wiley & Sons; 2005.

11. Ren Y. The methodology of flowgraph models: a thesis submitted for the degree of Doctor of Philosophy. London: London School of Economics and Political Science; 2011.
12. Ara R, Brazier JE. Populating an economic model with health state utility values: moving toward better practice. *Value Health*. 2010;13(5):509-18. doi: 10.1111/j.1524-4733.2010.00700.x
13. Larson MG, Dinse GE. A Mixture Model for the Regression Analysis of Competing Risks Data. *Journal of the Royal Statistical Society Series C (Applied Statistics)*. 1985;34(3):201-11. doi: 10.2307/2347464
14. Karnon J, Stahl J, Brennan A, Caro JJ, Mar J, Moller J. Modeling using discrete event simulation: a report of the ISPOR-SMDM Modeling Good Research Practices Task Force-4. *Med Decis Making*. 2012;32(5):701-11. doi: 10.1177/0272989X12455462
15. Fine JP, Gray RJ. A Proportional Hazards Model for the Subdistribution of a Competing Risk. *Journal of the American Statistical Association*. 1999;94(446):496-509. doi: 10.1080/01621459.1999.10474144
16. Nicolaie MA, van Houwelingen HC, Putter H. Vertical modeling: A pattern mixture approach for competing risks modeling. *Statistics in Medicine*. 2010;29(11):1190-205. doi: 10.1002/sim.3844
17. Roberts M, Russell LB, Paltiel AD, Chambers M, McEwan P, Krahn M, et al. Conceptualizing a model: a report of the ISPOR-SMDM Modeling Good Research Practices Task Force-2. *Med Decis Making*. 2012;32(5):678-89. doi: 10.1177/0272989X12454941
18. R Core Team. R: A language and environment for statistical computing Vienna, Austria: R Foundation for Statistical Computing; 2017. Available from: <https://www.R-project.org/>.
19. Filipović-Pierucci A, Zarca K, Durand-Zaleski I. Markov Models for Health Economic Evaluation: The R Package heemod. *ArXiv e-prints*. 2017;1702.03252.
20. Snowsill T. A new method for model-based health economic evaluation utilising and extending moment-generating functions: Example model. *Open Research Exeter*. 2019. doi: 10.24378/exe.643
21. Barendregt JJ. The half-cycle correction: banish rather than explain it. *Med Decis Making*. 2009;29(4):500-2. doi: 10.1177/0272989X09340585

22. 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. doi: 10.1177/0272989X18754513
23. Royston P, Parmar MK. Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. *Stat Med*. 2002;21(15):2175-97. doi: 10.1002/sim.1203
24. van Rosmalen J, Toy M, O'Mahony JF. A Mathematical Approach for Evaluating Markov Models in Continuous Time without Discrete-Event Simulation. *Medical Decision Making*. 2013;33(6):767-79. doi: 10.1177/0272989x13487947
25. Davis P, Rabinowitz P. *Methods of numerical integration*. New York: Academic Press; 1975.
26. DeVore R, Scott L. Error-bounds for Gaussian quadrature and weighted-L1 polynomial approximation. *SIAM J Numer Anal*. 1984;21(2):400-12.
27. Stoer J, Bulirsch R. *Introduction to numerical analysis*. 3rd ed. New York: Springer-Verlag; 2002.



## Tables

**Table 1: Table of extended MGFs in the example model**

$S$	$M_S^n(-r)$		
	$n = 0$	$n = 1$	$n = 2$
$(X_1   X_1 < X_2)$	$m_1^{(0)}$	$m_1^{(1)}$	$m_1^{(2)}$
$(X_2   X_2 < X_1)$	$m_2^{(0)}$	$m_2^{(1)}$	$m_2^{(2)}$
$X_3$	$m_3^{(0)}$	$m_3^{(1)}$	$m_3^{(2)}$
$(X_1 + X_3   X_1 < X_2)$	$m_1^{(0)} m_3^{(0)}$	$(m_1^{(0)} m_3^{(1)} + m_1^{(1)} m_3^{(0)})$	$(m_1^{(0)} m_3^{(2)} + 2m_1^{(1)} m_3^{(1)} + m_1^{(2)} m_3^{(0)})$

**Table 2. Worked example results**

Arm	QALYs		Costs		Net monetary benefit	
	Absolute	Incremental	Absolute	Incremental	Absolute	Incremental
<b>MGF method</b>						
Control	1.678		£8437		£25,119	
Treatment	1.871	0.194	£8529	£92	£28,898	£3779
<b>Discrete event simulation (100,000 simulations) [Mean (95% CI)]</b>						
Control	1.676 (1.670, 1.683)		£8432 (£8419, £8444)		£25,090 (£24,969, £25,211)	
Treatment	1.870 (1.863, 1.877)	0.194 (0.189, 0.199)	£8531 (£8518, £8544)	£99 (£89, £109)	£28,872 (£28,743, £29,000)	£3782 (£3684, £3880)
<b>Markov cohort simulation (1-month cycle length, up to 31 tunnel states)</b>						
Control	1.667		£8414		£24,921	
Treatment	1.862	0.195	£8509	£95	£28,733	£3811
<b>Markov cohort simulation (1-year cycle length, no tunnel states)</b>						
Control	1.883		£8729		£28,927	
Treatment	2.061	0.179	£8794	£65	£32,434	£3506
<b>Markov microsimulation (1-month cycle length, 100,000 simulations) [Mean (95% CI)]</b>						
Control	1.732 (1.726, 1.739)		£8458 (£8446, £8470)		£26,184 (£26,064, £26,304)	
Treatment	1.921 (1.914, 1.928)	0.189 (0.185, 0.192)	£8562 (£8549, £8575)	£103 (£96, £110)	£29,857 (£29,729, £29,985)	£3673 (£3608, £3738)

**Table 3. Execution times for different methods**

<b>Method</b>	<b>Method parameters</b>		<b>Time per iteration (ms) [Mean (SD), Median (IQR)]</b>
MGF			11.0 (29.2), 6.59 (6.25–7.75)
Markov cohort <sup>[a]</sup>	Cycles	800	166 (39.5), 152 (147–171)
	Tunnel states	0	
Markov cohort <sup>[a]</sup>	Cycles	2,400	775 (119), 731 (692–832)
	Tunnel states	23	
Markov microsimulation <sup>[a]</sup>	Simulations	1,000	326 (70.6), 304 (290–335)
Markov microsimulation <sup>[a]</sup>	Simulations	100,000	26,124 (2513), 25,829 (24,361–27,373)
DES	Simulations	1,000	5.86 (12.3), 4.13 (3.96–4.69)
DES	Simulations	100,000	234 (34.1), 223 (215–238)

Notes: [a] Excludes all preparation prior to the `run_model` command (Markov cohort model) and the function factory step (Markov microsimulation)

**Table 4. Capabilities of different methods**

<b>Method</b>	<b>MGF</b>	<b>Markov cohort without tunnel states</b>	<b>Markov cohort with tunnel states</b>	<b>Markov micro- simulation</b>	<b>DES</b>
Sojourn time-dependent transitions  Transitions between states can depend on the length of time in the current state. Can use non-exponential time-to-event distributions.	Yes	No	Partial	Yes	Yes
Wall time-dependent transitions  Transitions between states can depend on the length of time since the model start. Typical example is age-related other cause mortality.	No	Yes	Yes	Yes	Yes
Kaplan–Meier survival  Directly use Kaplan–Meier survival curves, without fitting a parametric model or piecewise exponential model. Particularly valuable when parametric models give a poor fit.	Yes	Partial	Partial	Partial	Yes
Sojourn time-dependent payoffs  Payoffs depend on the time in the current state, e.g., costs are initially high within a state but then diminish.	Partial	No	Partial	Yes	Yes
Wall time-dependent payoffs  Payoffs depend on the time since the model start, e.g., health state utility values decline with age.	Partial	Yes	Yes	Yes	Yes
Monte Carlo variation	No	No	No	Yes	Yes

---

There is random error introduced into results, which must either be minimised with a large number of simulations, or accounted for within statistical analyses.

Convergence behaviour	See	$\mathcal{O}(n^{-1})$	$\mathcal{O}(n^{-1/2})$	$\mathcal{O}(n^{-1/2})$	$\mathcal{O}(n^{-1/2})$
The global truncation error as a function of $n$ , which is the number of quadrature nodes for the MGF method, the cycle length for the Markov cohort simulations, and the number of simulations for the microsimulation methods.					
appendix					

Notes: See *Appendix 6* for further details.

## Appendix 1: Demonstration of equivalence of MGF method, Markov cohort simulation and state occupancy equations in a simple example

### Background

This appendix gives further details supporting the demonstration of the equivalence of different methods in a simple model.

We have a conceptual state transition model with three states, with transitions only from the first to the second state (at rate  $\lambda_1$ ), and the second state to the third state (at rate  $\lambda_2$ ).

### Markov cohort simulation

An equivalent Markov model, where the cycle length is  $\delta x$ , is as follows:

$$\mathbf{y}_{n+1} = \begin{pmatrix} e^{-\lambda_1 \delta x} & 1 - e^{-\lambda_1 \delta x} & 0 \\ 0 & e^{-\lambda_2 \delta x} & 1 - e^{-\lambda_2 \delta x} \\ 0 & 0 & 1 \end{pmatrix}^T \mathbf{y}_n$$

$$\mathbf{y}_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Based on the recurrence relation we then find:

$$\mathbf{y}_n = \left( \begin{pmatrix} e^{-\lambda_1 \delta x} & 1 - e^{-\lambda_1 \delta x} & 0 \\ 0 & e^{-\lambda_2 \delta x} & 1 - e^{-\lambda_2 \delta x} \\ 0 & 0 & 1 \end{pmatrix}^T \right)^n \mathbf{y}_0$$

Using eigendecomposition we find that

$$\begin{pmatrix} e^{-\lambda_1 \delta x} & 1 - e^{-\lambda_1 \delta x} & 0 \\ 0 & e^{-\lambda_2 \delta x} & 1 - e^{-\lambda_2 \delta x} \\ 0 & 0 & 1 \end{pmatrix}^T = V \Lambda V^{-1}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ \frac{1 - e^{-\lambda_1 \delta x}}{e^{-\lambda_1 \delta x} - e^{-\lambda_2 \delta x}} & 1 & 0 \\ \frac{e^{-\lambda_2 \delta x} - 1}{e^{-\lambda_1 \delta x} - e^{-\lambda_2 \delta x}} & -1 & 1 \end{pmatrix} \begin{pmatrix} e^{-\lambda_1 \delta x} & 0 & 0 \\ 0 & e^{-\lambda_2 \delta x} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ \frac{e^{-\lambda_1 \delta x} - 1}{e^{-\lambda_1 \delta x} - e^{-\lambda_2 \delta x}} & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Meaning that

$$\begin{aligned}
\mathbf{y}_n &= \begin{pmatrix} 1 & 0 & 0 \\ \frac{1 - e^{-\lambda_1 \delta x}}{e^{-\lambda_1 \delta x} - e^{-\lambda_2 \delta x}} & 1 & 0 \\ \frac{e^{-\lambda_2 \delta x} - 1}{e^{-\lambda_1 \delta x} - e^{-\lambda_2 \delta x}} & -1 & 1 \end{pmatrix} \begin{pmatrix} e^{-n\lambda_1 \delta x} & 0 & 0 \\ 0 & e^{-n\lambda_2 \delta x} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ \frac{e^{-\lambda_1 \delta x} - 1}{e^{-\lambda_1 \delta x} - e^{-\lambda_2 \delta x}} & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \mathbf{y}_0 \\
&= \begin{pmatrix} e^{-n\lambda_1 \delta x} \\ \frac{(1 - e^{-\lambda_1 \delta x})(e^{-n\lambda_1 \delta x} - e^{-n\lambda_2 \delta x})}{e^{-\lambda_1 \delta x} - e^{-\lambda_2 \delta x}} \\ \frac{e^{-n\lambda_2 \delta x}(1 - e^{-\lambda_1 \delta x}) - e^{-n\lambda_1 \delta x}(1 - e^{-\lambda_2 \delta x})}{e^{-\lambda_1 \delta x} - e^{-\lambda_2 \delta x}} + 1 \end{pmatrix}
\end{aligned}$$

Discounted QALYs are calculated as a sum of the Markov occupancies across all cycles weighted by the health state utility value and the discounting function. Let  $u_{\text{Unaffected}}$  be the health state utility value for the healthy state and  $u_{\text{Affected}}$  be the health state utility value in the diseased state. As we are working analytically we assume an infinite number of cycles:

$$QALY_d = u_{\text{Unaffected}} \sum_{n=0}^{\infty} y_{n1} e^{-rn\delta x} \delta x + u_{\text{Affected}} \sum_{n=0}^{\infty} y_{n2} e^{-rn\delta x} \delta x$$

Where  $y_{n1}$  and  $y_{n2}$  are the first and second components of  $\mathbf{y}_n$  (i.e., the state membership for the healthy and diseased states in cycle  $n$ ).

Let us consider the first component of this:

$$\begin{aligned}
\sum_{n=0}^{\infty} e^{-rn\delta x} u_{\text{Unaffected}} e^{-n\lambda_1 \delta x} \delta x &= \sum_{n=0}^{\infty} e^{-n\delta x(r+\lambda_1)} u_{\text{Unaffected}} \delta x \\
&= u_{\text{Unaffected}} \delta x (1 + e^{-\delta x(r+\lambda_1)} + e^{-2\delta x(r+\lambda_1)} + e^{-3\delta x(r+\lambda_1)} \\
&\quad + \dots) \\
&= u_{\text{Unaffected}} \delta x \sum_{n=0}^{\infty} (e^{-\delta x(r+\lambda_1)})^n
\end{aligned}$$

This is a geometric series, and since  $\sum_{n=0}^{\infty} x^n = (1 - x)^{-1}$ , we obtain

$$\frac{u_{\text{Unaffected}} \delta x}{1 - e^{-\delta x(r+\lambda_1)}}$$

We now take the limit as  $\delta x \rightarrow 0$  using L'Hôpital's rule

$$\begin{aligned}
\lim_{\delta x \rightarrow 0} \frac{u_{\text{Unaffected}} \delta x}{1 - e^{-\delta x(r+\lambda_1)}} &= \lim_{\delta x \rightarrow 0} \frac{\frac{d}{d\delta x} (u_{\text{Unaffected}} \delta x)}{\frac{d}{d\delta x} (1 - e^{-\delta x(r+\lambda_1)})} \\
&= \lim_{\delta x \rightarrow 0} \frac{u_{\text{Unaffected}}}{(r + \lambda_1) e^{-\delta x(r+\lambda_1)}} \\
&= \frac{u_{\text{Unaffected}}}{r + \lambda_1}
\end{aligned}$$

This is exactly the component of discounted QALYs in the unaffected state as produced using the MGF method.

For the second component, we proceed exactly as for the first component:

$$\begin{aligned}
u_{\text{Affected}} \sum_{n=0}^{\infty} y_{n2} e^{-rn\delta x} \delta x &= \frac{u_{\text{Affected}} (1 - e^{-\lambda_1 \delta x}) \delta x}{e^{-\lambda_1 \delta x} - e^{-\lambda_2 \delta x}} \sum_{n=0}^{\infty} (e^{-n\lambda_1 \delta x} - e^{-n\lambda_2 \delta x}) e^{-rn\delta x} \\
&= \frac{u_{\text{Affected}} (1 - e^{-\lambda_1 \delta x}) \delta x}{e^{-\lambda_1 \delta x} - e^{-\lambda_2 \delta x}} \sum_{n=0}^{\infty} (e^{-n\delta x(\lambda_1+r)} - e^{-n\delta x(\lambda_2+r)}) \\
&= \frac{u_{\text{Affected}} (1 - e^{-\lambda_1 \delta x}) \delta x}{e^{-\lambda_1 \delta x} - e^{-\lambda_2 \delta x}} \left( \sum_{n=0}^{\infty} (e^{-\delta x(\lambda_1+r)})^n \right. \\
&\quad \left. - \sum_{n=0}^{\infty} (e^{-\delta x(\lambda_2+r)})^n \right) \\
&= \frac{u_{\text{Affected}} (1 - e^{-\lambda_1 \delta x}) \delta x}{e^{-\lambda_1 \delta x} - e^{-\lambda_2 \delta x}} \left( \frac{1}{1 - e^{-\delta x(\lambda_1+r)}} - \frac{1}{1 - e^{-\delta x(\lambda_2+r)}} \right) \\
&= \frac{u_{\text{Affected}} (1 - e^{-\lambda_1 \delta x}) \delta x (e^{-\delta x(\lambda_1+r)} - e^{-\delta x(\lambda_2+r)})}{(e^{-\lambda_1 \delta x} - e^{-\lambda_2 \delta x}) (1 - e^{-\delta x(\lambda_1+r)}) (1 - e^{-\delta x(\lambda_2+r)})} \\
&= \frac{u_{\text{Affected}} (1 - e^{-\lambda_1 \delta x}) \delta x e^{-\delta x r}}{(1 - e^{-\delta x(\lambda_1+r)}) (1 - e^{-\delta x(\lambda_2+r)})} \\
&= \frac{u_{\text{Affected}} (1 - e^{-\lambda_1 \delta x}) \delta x}{(e^{\delta x r} - e^{-\delta x \lambda_1}) (e^{\delta x r} - e^{-\delta x \lambda_2})}
\end{aligned}$$

Taking the limit as  $\delta x \rightarrow 0$ :

$$u_{\text{Affected}} \lim_{\delta x \rightarrow 0} \frac{(1 - e^{-\lambda_1 \delta x}) \delta x}{(e^{\delta x r} - e^{-\delta x \lambda_1}) (e^{\delta x r} - e^{-\delta x \lambda_2})} = u_{\text{Affected}} \frac{\lambda_1}{(r + \lambda_1)(r + \lambda_2)}$$

Which again exactly matches the results from the MGF method.



### State occupancy equations

It can also be demonstrated that the same result is obtained if equations for the state occupancy over time are identified analytically. Let  $\mathbf{y}(x) = [y_1(x) \quad y_2(x) \quad y_3(x)]^T$  denote the probability that an individual is in each of the three states at a given time  $x$ .

$$\begin{aligned}
 y_1(x) &= \Pr(X_1 > x) \\
 &= 1 - F_{X_1}(x) \\
 &= e^{-\lambda_1 x} \\
 y_2(x) &= \Pr(X_1 < x < X_1 + X_2) \\
 &= \int_0^x f_{X_1}(x_1) (1 - F_{X_2}(x - x_1)) dx_1 \\
 &= \int_0^x \lambda_1 e^{-\lambda_1 x_1} e^{-\lambda_2(x-x_1)} dx_1 \\
 &= \frac{\lambda_1}{\lambda_1 - \lambda_2} e^{-\lambda_2 x} (1 - e^{-x(\lambda_1 - \lambda_2)}) \\
 y_3(x) &= \Pr(x > X_1 + X_2) \\
 &= \int_0^x f_{X_1}(x_1) F_{X_2}(x - x_1) dx_1 \\
 &= \int_0^x \lambda_1 e^{-\lambda_1 x_1} (1 - e^{-\lambda_2(x-x_1)}) dx_1 \\
 &= 1 - e^{-\lambda_1 x} + \frac{\lambda_1}{\lambda_1 - \lambda_2} (e^{-\lambda_1 x} - e^{-\lambda_2 x})
 \end{aligned}$$

The lifetime discounted QALYs in the unaffected state are therefore:

$$\int_0^{\infty} y_1(x) u_{\text{Unaffected}} e^{-rx} dx = u_{\text{Unaffected}} \int_0^{\infty} e^{-(\lambda_1 + r)x} dx = \frac{u_{\text{Unaffected}}}{\lambda_1 + r}$$

And in the affected state:

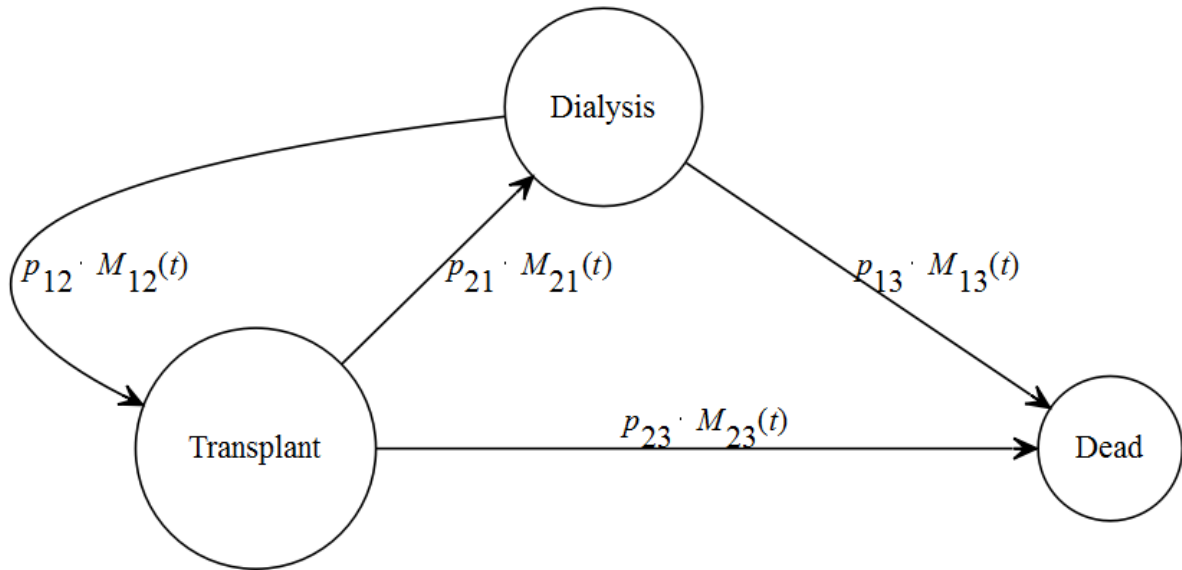
$$\begin{aligned}
 \int_0^{\infty} y_2(x) u_{\text{Affected}} e^{-rx} dx &= u_{\text{Affected}} \int_0^{\infty} \frac{\lambda_1}{\lambda_1 - \lambda_2} e^{-\lambda_2 x} (1 - e^{-x(\lambda_1 - \lambda_2)}) e^{-rx} dx \\
 &= u_{\text{Affected}} \frac{\lambda_1}{\lambda_1 - \lambda_2} \left( \int_0^{\infty} e^{-\lambda_2 x} e^{-rx} dx - \int_0^{\infty} e^{-\lambda_2 x} (e^{-x(\lambda_1 - \lambda_2)}) e^{-rx} dx \right) \\
 &= u_{\text{Affected}} \frac{\lambda_1}{\lambda_1 - \lambda_2} \left( \frac{1}{r + \lambda_2} - \frac{1}{r + \lambda_1} \right) = u_{\text{Affected}} \frac{\lambda_1}{\lambda_1 - \lambda_2} \left( \frac{r + \lambda_1 - (r + \lambda_2)}{(r + \lambda_2)(r + \lambda_1)} \right) \\
 &= u_{\text{Affected}} \frac{\lambda_1}{(r + \lambda_2)(r + \lambda_1)}
 \end{aligned}$$

These both agree exactly with the results of the MGF method.

## Appendix 2: Example of including cycles in the MGF method

In this appendix we expand on an example given within the main paper to incorporate a cycle.

The figure for the model is given below (contrast with **Fig. 2** which does not include the possibility of returning from the Transplant state to the Dialysis state). It uses the statistical flowgraph model / pattern mixture approach to define competing risks.



To account for the fact that there could be any number of transplantations, we introduce an additional subscript for the TTE variables corresponding to transplantation and graft failure.  $X_{12k}$  is the waiting time in dialysis before undergoing the  $k^{\text{th}}$  transplantation and  $X_{21k}$  is the time with a transplant prior to graft failure and returning to dialysis. There is no need to introduce additional subscripts for  $X_{13}$  and  $X_{23}$  since Dead is an absorbing state.

We introduce two new random variables for convenience:  $K$  denotes the number of transplants a patient receives in their lifetime, and  $D$  is 1 if they die with a transplant or 0 if they die while on dialysis (note that if  $D = 1$  then  $K \geq 1$ ).

In this example we focus on calculating the discounted life years lived on dialysis. The same methodology can be employed to calculate other discounted payoffs.

The expected discounted life years lived on dialysis is factorised across the different combinations of  $K$  and  $D$ :

$$\mathbb{E} \left[ LY_d^{(Dialysis)} \right] = \sum_{k=0}^{\infty} \sum_{d=0}^1 \Pr(K = k, D = d) \mathbb{E} \left[ LY_d^{(Dialysis)} \mid K = k, D = d \right]$$

By simple consideration of the branch probabilities, we find:

$$\begin{aligned} \Pr(K = k, D = 0) &= p_{12}^k p_{21}^k p_{13} \\ \Pr(K = k, D = 1) &= p_{12}^k p_{21}^{k-1} p_{23}, \quad k \geq 1 \end{aligned}$$

We then develop formulae for the discounted life years lived in dialysis according to the values of  $K$  and  $D$ :

$$\begin{aligned} LY_d^{(Dialysis)} \mid (K = k, D = 0) &= \left( \sum_{m=0}^{k-1} \int_{\sum_{n=1}^m X_{12n} + X_{21n}}^{[\sum_{n=1}^m X_{12n} + X_{21n}] + X_{12(m+1)}} e^{-rx} dx \right) + \int_{\sum_{n=1}^k X_{12n} + X_{21n}}^{[\sum_{n=1}^k X_{12n} + X_{21n}] + X_{13}} e^{-rx} dx \\ LY_d^{(Dialysis)} \mid (K = k, D = 1) &= \sum_{m=0}^{k-1} \int_{\sum_{n=1}^m X_{12n} + X_{21n}}^{[\sum_{n=1}^m X_{12n} + X_{21n}] + X_{12(m+1)}} e^{-rx} dx \end{aligned}$$

For the “special case” of  $K = 0$  (implies  $D = 0$ ):

$$\begin{aligned} \mathbb{E} \left[ LY_d^{(Dialysis)} \mid (K = 0, D = 0) \right] &= \mathbb{E} \left[ \int_0^{X_{13}} e^{-rx} dx \right] = \mathbb{E} \left[ \frac{1}{r} (1 - e^{-rX_{13}}) \right] \\ &= \frac{1}{r} (1 - M_{X_{13}}(-r)) \end{aligned}$$

And for the “general case” of  $K \geq 1$ , we note that a simplification is possible since one summation occurs whether  $D$  equals 0 or 1:

$$\begin{aligned} \mathbb{E} \left[ LY_d^{(Dialysis)} \mid (K = k) \right] &= \mathbb{E} \left[ \sum_{m=0}^{k-1} \int_{\sum_{n=1}^m X_{12n} + X_{21n}}^{[\sum_{n=1}^m X_{12n} + X_{21n}] + X_{12(m+1)}} e^{-rx} dx \right] \\ &\quad + \Pr(D = 0 \mid K = k) \mathbb{E} \left[ \int_{\sum_{n=1}^k X_{12n} + X_{21n}}^{[\sum_{n=1}^k X_{12n} + X_{21n}] + X_{13}} e^{-rx} dx \right] \end{aligned}$$

We first consider the summation from  $m = 0$  to  $k - 1$ , noting crucially that all  $X_{12n}$  and  $X_{21n}$  are independent of each other and have identical MGFs equal to  $M_{X_{12}}(\cdot)$  and  $M_{X_{21}}(\cdot)$  respectively:

$$\begin{aligned}
& \mathbb{E} \left[ \sum_{m=0}^{k-1} \int_{\sum_{n=1}^m X_{12n} + X_{21n}}^{[\sum_{n=1}^m X_{12n} + X_{21n}] + X_{12(m+1)}} e^{-rx} dx \right] \\
&= \frac{1}{r} \mathbb{E} \left[ \sum_{m=0}^{k-1} e^{-r(\sum_{n=1}^m X_{12n} + X_{21n})} - e^{-r([\sum_{n=1}^m X_{12n} + X_{21n}] + X_{12(m+1)})} \right] \\
&= \frac{1}{r} \mathbb{E} \left[ \sum_{m=0}^{k-1} e^{-r(\sum_{n=1}^m X_{12n} + X_{21n})} (1 - e^{-rX_{12(m+1)}}) \right] \\
&= \frac{1}{r} \mathbb{E} \left[ \sum_{m=0}^{k-1} (1 - e^{-rX_{12(m+1)}}) \prod_{n=1}^m e^{-rX_{12n}} e^{-rX_{21n}} \right] \\
&= \frac{1}{r} \left( \sum_{m=0}^{k-1} (1 - M_{X_{12}}(-r)) \prod_{n=1}^m M_{X_{12}}(-r) M_{X_{21}}(-r) \right) \\
&= \frac{1 - M_{X_{12}}(-r)}{r} \sum_{m=0}^{k-1} (M_{X_{12}}(-r) M_{X_{21}}(-r))^m
\end{aligned}$$

And now the second component of the formula:

$$\begin{aligned}
& \mathbb{E} \left[ \int_{\sum_{n=1}^k X_{12n} + X_{21n}}^{[\sum_{n=1}^k X_{12n} + X_{21n}] + X_{13}} e^{-rx} dx \right] = \frac{1}{r} \mathbb{E} \left[ e^{-r(\sum_{n=1}^k X_{12n} + X_{21n})} - e^{-r([\sum_{n=1}^k X_{12n} + X_{21n}] + X_{13})} \right] \\
&= \frac{1}{r} \mathbb{E} \left[ e^{-r(\sum_{n=1}^k X_{12n} + X_{21n})} (1 - e^{-rX_{13}}) \right] \\
&= \frac{1}{r} \mathbb{E} \left[ (1 - e^{-rX_{13}}) \prod_{n=1}^k e^{-rX_{12n}} e^{-rX_{21n}} \right] \\
&= \frac{1}{r} \mathbb{E} \left[ (1 - M_{X_{13}}(-r)) \prod_{n=1}^k M_{X_{12}}(-r) M_{X_{21}}(-r) \right] \\
&= \frac{1 - M_{X_{13}}(-r)}{r} (M_{X_{12}}(-r) M_{X_{21}}(-r))^k
\end{aligned}$$

Bringing together the special and the general case we now have:

$$\begin{aligned}
& \mathbb{E} \left[ LY_d^{(Dialysis)} \right] \\
&= \Pr(K = 0, D = 0) \mathbb{E} \left[ LY_d^{(Dialysis)} \mid K = 0, D = 0 \right] \\
&+ \sum_{k=1}^{\infty} \Pr(K = k) \mathbb{E} \left[ LY_d^{(Dialysis)} \mid K = k \right] \\
&= \frac{p_{13}}{r} \left( 1 - M_{X_{13}}(-r) \right) \\
&+ \sum_{k=1}^{\infty} \left( p_{12}^k p_{21}^{k-1} (p_{21} p_{13} + p_{23}) \left( \frac{1 - M_{X_{12}}(-r)}{r} \sum_{m=0}^{k-1} \left( M_{X_{12}}(-r) M_{X_{21}}(-r) \right)^m \right. \right. \\
&\left. \left. + \frac{p_{21} p_{13}}{(p_{21} p_{13} + p_{23})} \frac{1 - M_{X_{13}}(-r)}{r} \left( M_{X_{12}}(-r) M_{X_{21}}(-r) \right)^k \right) \right) \\
&= \frac{p_{13}}{r} \left( 1 - M_{X_{13}}(-r) \right) \\
&+ \frac{1}{r} \left( (p_{21} p_{13} + p_{23}) \left( 1 \right. \right. \\
&\left. \left. - M_{X_{12}}(-r) \right) \sum_{k=1}^{\infty} \left( p_{12}^k p_{21}^{k-1} \sum_{m=0}^{k-1} \left( M_{X_{12}}(-r) M_{X_{21}}(-r) \right)^m \right) \right. \\
&\left. + p_{21} p_{13} \left( 1 - M_{X_{13}}(-r) \right) \sum_{k=1}^{\infty} p_{12}^k p_{21}^{k-1} \left( M_{X_{12}}(-r) M_{X_{21}}(-r) \right)^k \right)
\end{aligned}$$

We then use the following equation to simplify (assumes  $p, b < 1$ ):

$$\sum_{k=1}^{\infty} ap^k \left( \sum_{m=0}^{k-1} b^m \right) = \frac{ap}{(1-p)(1-bp)}$$

$$\sum_{k=1}^{\infty} \left( p_{12}^k p_{21}^{k-1} \sum_{m=0}^{k-1} \left( M_{X_{12}}(-r) M_{X_{21}}(-r) \right)^m \right) = \frac{ap}{(1-p)(1-bp)}$$

Where

$$a = p_{21}^{-1}$$

$$p = p_{12} p_{21}$$

$$b = M_{X_{12}}(-r) M_{X_{21}}(-r)$$

Therefore

$$\begin{aligned} & \sum_{k=1}^{\infty} \left( p_{12}^k p_{21}^{k-1} \sum_{m=0}^{k-1} \left( M_{X_{12}}(-r) M_{X_{21}}(-r) \right)^m \right) \\ &= \frac{p_{12}}{(1 - p_{12}p_{21}) \left( 1 - p_{12}p_{21} M_{X_{12}}(-r) M_{X_{21}}(-r) \right)} \end{aligned}$$

And the more recognisable infinite series:

$$\sum_{j=0}^{\infty} ar^j = \frac{a}{1-r}$$

To simplify

$$\sum_{k=1}^{\infty} p_{12}^k p_{21}^{k-1} \left( M_{X_{12}}(-r) M_{X_{21}}(-r) \right)^k$$

By setting

$$\begin{aligned} j &= k - 1 \\ a &= p_{12} M_{X_{12}}(-r) M_{X_{21}}(-r) \\ r &= p_{12} p_{21} M_{X_{12}}(-r) M_{X_{21}}(-r) \end{aligned}$$

Such that

$$\sum_{k=1}^{\infty} p_{12}^k p_{21}^{k-1} \left( M_{X_{12}}(-r) M_{X_{21}}(-r) \right)^k = \frac{p_{12} M_{X_{12}}(-r) M_{X_{21}}(-r)}{1 - p_{12} p_{21} M_{X_{12}}(-r) M_{X_{21}}(-r)}$$

So finally we have:

$$\begin{aligned} & \mathbb{E} \left[ LY_d^{(Dialysis)} \right] \\ &= \frac{1}{r} \left( p_{13} \left( 1 - M_{X_{13}}(-r) \right) + \frac{p_{12} (p_{21} p_{13} + p_{23}) \left( 1 - M_{X_{12}}(-r) \right)}{(1 - p_{12} p_{21}) \left( 1 - p_{12} p_{21} M_{X_{12}}(-r) M_{X_{21}}(-r) \right)} \right. \\ & \quad \left. + \frac{p_{12} p_{21} p_{13} \left( 1 - M_{X_{13}}(-r) \right) M_{X_{12}}(-r) M_{X_{21}}(-r)}{1 - p_{12} p_{21} M_{X_{12}}(-r) M_{X_{21}}(-r)} \right) \end{aligned}$$

### Appendix 3: Independent time-to-event distributions approach to competing risks

In this approach it is assumed that there are multiple competing risks, represented by time-to-event random variables  $X_1, X_2, \dots$ , and it is further assumed that these are independent. All but the earliest of these time-to-event variables are latent (not observed). Such a description of competing events often arises from evidence synthesis, where different sources provide time-to-event distributions.

The method for solving such problems involves mapping from this formulation to the pattern-mixture approach described above, i.e., to estimate the categorical distribution for which event occurs first, and for each of the possible events to derive the conditional distribution (and more importantly, its EMGF), given that it was the earliest event.

In summary (assuming there are only two competing risks), we need to calculate:

$$p = \Pr(D = 1) = \Pr(X_1 < X_2)$$

$$\mathbb{E}[X_1 \mid X_1 < X_2]$$

$$M_{X_1 \mid X_1 < X_2}(t) = \mathbb{E}[e^{tX_1} \mid X_1 < X_2]$$

$$M_{X_1 \mid X_1 < X_2}^j(t) = \mathbb{E}[X_1^j e^{tX_1} \mid X_1 < X_2]$$

These will require integrals involving conditional probability distributions such as  $f_{X_1 \mid X_1 < X_2}(x)$ , which can be estimated either by multiplying the probability density function of  $X_1$  by the survivor function of  $X_2$  or by multiplying the probability density function of  $X_2$  by the cumulative distribution function of  $X_1$ :

$$f_{X_1 \mid X_1 < X_2}(x) = \frac{f_{X_1}(x)(1 - F_{X_2}(x))}{\int_0^\infty f_{X_2}(x)F_{X_1}(x)dx} = \frac{f_{X_1}(x)(1 - F_{X_2}(x))}{\int_0^\infty f_{X_1}(x)(1 - F_{X_2}(x))dx}$$

We then use the law of the unconscious statistician for all necessary calculations:

$$\mathbb{E}[g(X_1) \mid X_1 < X_2] = \frac{\int_0^\infty g(x)f_{X_1}(x)(1 - F_{X_2}(x))dx}{\int_0^\infty f_{X_2}(x)F_{X_1}(x)dx}$$

Where  $g(x)$  is any function of interest, e.g.,  $g(x) = e^{-rx}$  to calculate  $M_{X_1 \mid X_1 < X_2}(-r)$ .



### Analytical results

If  $X_1$  and  $X_2$  are independent exponentially distributed random variables, with rate parameters  $\lambda_1$  and  $\lambda_2$  respectively, then the conditional distributions for each variable (conditional on it being the earlier time-to-event) are exponential, with rate parameters both equal to  $(\lambda_1 + \lambda_2)$ , and the probability that  $X_1$  is the earlier event is:

$$\mathbb{E}[\mathbf{1}_{X_1 < X_2}] = \frac{\lambda_1}{\lambda_1 + \lambda_2}$$

Similar results are obtained if  $X_1$  and  $X_2$  are Weibull with common shape parameter,  $k$ , but different scale parameters  $\lambda_1$  and  $\lambda_2$ . In this case the conditional distributions are both Weibull with equal scale and shape parameters. The scale parameter is

$$\lambda_{12} = \frac{\lambda_1 \lambda_2}{(\lambda_1^k + \lambda_2^k)^{\frac{1}{k}}}$$

And the shape parameter is  $k$ . The probability that  $X_1$  is the earlier event is

$$\mathbb{E}[\mathbf{1}_{X_1 < X_2}] = \frac{\lambda_1^k}{\lambda_1^k + \lambda_2^k}$$

### General case

In most cases it is unlikely that the conditional distributions will match known distributions with MGFs. In this case numerical integration is recommended.

Note that when there are more than two competing risks the complexity is not greatly increased:

$$p_i = \mathbb{E}[\mathbf{1}_{X_i = \min_j X_j}] = \int_0^\infty f_{X_i}(x) \prod_{j \neq i} (1 - F_{X_j}(x)) dx$$
$$f_{X_i | X_i = \min_j X_j}(x) = \frac{1}{p_i} f_{X_i}(x) \prod_{j \neq i} (1 - F_{X_j}(x))$$

Only 1-dimensional integrals need to be calculated, which can be efficiently estimated numerically.

When Kaplan–Meier estimators are involved, at least one of the time-to-event distributions is discrete, such that integrals involving  $f(x)$  are not appropriate, but summation or Riemann–Stieltjes integrals must be used.

## Appendix 4: Derivations of MGFs and EMGFs for certain time-to-event distributions

In this appendix we provide analytical (E)MGFs with derivations for a number of time-to-event distributions which may be encountered in health economic modelling. The distributions are:

- Exponential distribution – The hazard of the event is constant over time;
- Gamma distribution – The hazard function is either concave and increasing over time or convex and decreasing over time;
- Degenerate distribution – The event always happens after a particular length of time;
- Uniform distribution – The event is equally likely to occur at any time during a given window;
- Kaplan–Meier estimator – Empirical survival data (including censored observations) is used to estimate the survivor function.

Many of the derivations use the Gamma function (a generalisation of the factorial function beyond the natural/counting numbers), which has no closed form but is available in statistical packages and spreadsheet software:

$$\Gamma(z) = \int_0^{\infty} x^{z-1} e^{-x} dx$$

### Exponential random variable

MGF

$$\begin{aligned} X &\sim \text{Exp}(\lambda) \\ f_X(x) &= \lambda e^{-\lambda x} \\ M_X(t) &= \mathbb{E}[e^{tX}] \\ &= \int_0^{\infty} e^{tx} \lambda e^{-\lambda x} dx \\ &= \int_0^{\infty} \lambda e^{-(\lambda-t)x} dx \\ &= \left[ -\frac{\lambda}{\lambda-t} e^{-(\lambda-t)x} \right]_0^{\infty} \\ &= \frac{\lambda}{\lambda-t}, \quad t < \lambda \end{aligned}$$

EMGF

$$\begin{aligned}
 M_X^n(t) &= \mathbb{E}[X^n e^{tX}] \\
 &= \int_0^{\infty} x^n e^{tx} \lambda e^{-\lambda x} dx \\
 &= \lambda \int_0^{\infty} x^n e^{-(\lambda-t)x} dx \\
 &= \frac{\lambda}{(\lambda-t)^n} \int_0^{\infty} ((\lambda-t)x)^n e^{-(\lambda-t)x} dx
 \end{aligned}$$

Let  $n = z - 1$  and let  $u = (\lambda - t)x$  ( $t < \lambda$ ):

$$\begin{aligned}
 M_X^n(t) &= \frac{\lambda}{(\lambda-t)^{z-1}} \int_0^{\infty} \frac{u^{z-1} e^{-u} du}{\lambda-t} \\
 &= \frac{\lambda}{(\lambda-t)^z} \int_0^{\infty} u^{z-1} e^{-u} du \\
 &= \frac{\lambda}{(\lambda-t)^z} \Gamma(z) \\
 &= \frac{\lambda(z-1)!}{(\lambda-t)^z} \\
 &= \frac{\lambda n!}{(\lambda-t)^{n+1}}
 \end{aligned}$$

### Gamma random variable

MGF

$$\begin{aligned}
 X &\sim \Gamma(\alpha, \beta) \\
 f_X(x) &= \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \\
 M_X(t) &= \mathbb{E}[e^{tX}] \\
 &= \int_0^{\infty} e^{tx} \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} dx \\
 &= \frac{\beta^\alpha}{\Gamma(\alpha)} \int_0^{\infty} \underbrace{x^{\alpha-1} e^{-(\beta-t)x}}_{u=(\beta-t)x, t < \beta} dx \\
 &= \frac{\beta^\alpha}{\Gamma(\alpha)(\beta-t)^\alpha} \int_0^{\infty} u^{\alpha-1} e^{-u} du \\
 &= \frac{\beta^\alpha}{\Gamma(\alpha)(\beta-t)^\alpha} \Gamma(\alpha) \\
 &= \frac{\beta^\alpha}{(\beta-t)^\alpha} \\
 &= \left( \frac{\beta}{\beta-t} \right)^\alpha \\
 &= \left( 1 - \frac{t}{\beta} \right)^{-\alpha}
 \end{aligned}$$

EMGF

$$\begin{aligned}M_X^n(t) &= \mathbb{E}[X^n e^{tX}] \\&= \int_0^\infty \frac{x^n e^{tx} \beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} dx \\&= \frac{\beta^\alpha}{\Gamma(\alpha)} \int_0^\infty x^{n+\alpha-1} e^{-(\beta-t)x} dx \\&= \frac{\Gamma(n+\alpha) \beta^\alpha}{\Gamma(\alpha) (\beta-t)^{n+\alpha}} \\&= \frac{\Gamma(n+\alpha)}{\Gamma(\alpha) (\beta-t)^n} \left(1 - \frac{t}{\beta}\right)^{-\alpha}\end{aligned}$$

Again, assuming  $t < \beta$ .

### Degenerate distribution

This distribution has all its probability mass concentrated on a single point,  $a$ . Its expected value is  $a$  and as a constant random variable,  $\mathbb{E}[g(X)] = g(\mathbb{E}[X])$  for any function  $g(\cdot)$ .

MGF

$$\begin{aligned}X &\sim \delta(a) \\M_X(t) &= \mathbb{E}[e^{tX}] \\&= e^{ta}\end{aligned}$$

EMGF

$$M_X^n(t) = \mathbb{E}[X^n e^{tX}] = a^n e^{ta}$$

### Uniform distribution

MGF

$$\begin{aligned}X &\sim U(a, b) \\f_X(x) &= \begin{cases} (b-a)^{-1}, & a \leq x \leq b \\ 0, & \text{otherwise} \end{cases} \\M_X(t) &= \mathbb{E}[e^{tX}] \\&= \int_a^b \frac{e^{tx}}{b-a} dx \\&= \left[ \frac{e^{tx}}{t(b-a)} \right]_a^b \\&= \frac{e^{tb} - e^{ta}}{t(b-a)}\end{aligned}$$

EMGF

$$\begin{aligned}
 M_X^n(t) &= \mathbb{E}[X^n e^{tX}] \\
 &= \int_a^b \frac{x^n e^{tx}}{b-a} dx \\
 &= \int_0^b \frac{x^n e^{tx}}{b-a} dx - \int_0^a \frac{x^n e^{tx}}{b-a} dx
 \end{aligned}$$

Let  $n = z - 1$  and  $u = -tx$  and assume  $t < 0$ :

$$\int_0^b \frac{x^n e^{tx}}{b-a} dx - \int_0^a \frac{x^n e^{tx}}{b-a} dx = \frac{\gamma(z, -tb) - \gamma(z, -ta)}{(-t)^z (b-a)}$$

Where  $\gamma(z, \alpha) = \int_0^\alpha x^{z-1} e^{-x} dx$  (the lower incomplete gamma function).

### Kaplan–Meier estimator

Given a Kaplan–Meier curve (where  $d_i$  and  $n_i$  are the number of individuals dying/failing at time  $x_i$  and the number at risk just prior to time  $x_i$ ):

$$S(x) = \prod_{i: x_i < x} \left(1 - \frac{d_i}{n_i}\right)$$

The EMGF when  $S(x) = 0$  for some  $x$  is:

$$\begin{aligned}
 M_X^n(t) &= \int_0^\infty x^n e^{tx} dF(x) \\
 &= - \int_0^\infty x^n e^{tx} dS(x) \\
 &= \sum_i x_i^n e^{tx_i} (S(x_{i-1}) - S(x_i))
 \end{aligned}$$

Where we set  $S(x_0) = 1$ . Note that  $\int_a^b g(x) dF(x)$  is the Riemann–Stieltjes integral of  $g(x)$  with respect to  $F(x)$ .

This means that to calculate the (E)MGF for a Kaplan–Meier distribution we can calculate components of the (E)MGF alongside our calculation of the survival function, and eventually sum these.

If the Kaplan–Meier estimator is not a proper distribution function (i.e., survival does not tend to zero as time tends towards infinity), we can still obtain a relevant EMGF.

First, suppose that at  $x_i = \infty$  the survival curve drops to 0, meaning that we add a term

$\lim_{x_i \rightarrow \infty} x_i^n e^{tx_i} (S(x_{i-1}) - S(x_i)) = 0$  (if  $t < 0$ ), i.e., there is no adjustment needed if we

assume that a proportion never transitions, although this is unlikely to ever be a

realistic assumption. Alternatively we can extend the estimator with a parametric model and calculate a composite EMGF. If we define the parametric extension from a change point  $x^*$ :

$$M_X^n(t) = \sum_{i: x_i \leq x^*} x_i^n e^{tx_i} (S(x_{i-1}) - S(x_i)) \\ + S(x^*) \int_{x^*}^{\infty} x^n e^{tx} f(x - x^*) dx$$

## Appendix 5: Recommendations for numerical integration

Many of the distributions frequently encountered in health economic modelling do not have a closed form, or convergent series representations, and so it is necessary to employ numerical integration techniques (or to make structural changes to the model so that the distribution can be approximated by an exponential, Erlang, Coxian or phase-type distribution, as described by van Rosmalen et al.<sup>24</sup>). Well-established numerical methods can be employed, in particular Gaussian quadrature methods.

Gauss–Laguerre quadrature with a simple substitution of  $u = rx$  gives:

$$M_X(-r) = \int_0^{\infty} e^{-rx} f_X(x) dx = \int_0^{\infty} \frac{1}{r} e^{-u} f_X(u/r) du \approx \frac{1}{r} \sum_{i=1}^n w_i f_X(u_i/r)$$

Where  $u_i$  and  $w_i$  are the nodes and weights for  $n$ -node Gaussian quadrature.

However, when  $r$  is small this leads to evaluation of  $f_X$  at large values (where it is typically close to zero), and ultimately very poor numerical performance.

Gauss–Laguerre quadrature can still be appropriate if instead the exponential term is extracted:

$$M_X(-r) = \int_0^{\infty} e^{-rx} f_X(x) dx = \int_0^{\infty} e^{-x} e^{(1-r)x} f_X(x) dx \approx \sum_{i=1}^n w_i e^{(1-r)x_i} f_X(x_i)$$

Gauss–Legendre quadrature can also be used with two different approaches. The first uses the substitution  $x = (1 + u)/(1 - u)$ :

$$\begin{aligned} M_X(-r) &= \int_0^{\infty} e^{-rx} f_X(x) dx = \int_{-1}^1 e^{-r \frac{1+u}{1-u}} f_X \left( \frac{1+u}{1-u} \right) \frac{2}{(1-u)^2} du \\ &\approx 2 \sum_{i=1}^n w_i (1-u_i)^{-2} e^{-r \frac{1+u_i}{1-u_i}} f_X \left( \frac{1+u_i}{1-u_i} \right) \end{aligned}$$

The second approach is based on the quantile function and the substitution  $u = 2p - 1$ :

$$M_X(-r) = \int_0^1 e^{-rQ_X(p)} dp = \frac{1}{2} \int_{-1}^1 e^{-rQ_X \left( \frac{u+1}{2} \right)} du \approx \frac{1}{2} \sum_{i=1}^n w_i e^{-rQ_X \left( \frac{u_i+1}{2} \right)} \quad (48)$$

Of all the approaches, Equation (48) appears to have the most desirable numerical qualities, since it avoids excessive exploration of very low density areas. However, the quantile function may not be readily available for all distributions in all settings.

In some cases it may be advantageous to split an integral in the following manner:

$$\begin{aligned}
 \int_0^{\infty} f(x)dx &= \int_0^a f(x)dx + \int_a^{\infty} f(x)dx \\
 &= \frac{a}{2} \int_{-1}^1 f\left(\frac{a}{2}(u+1)\right) du + 2 \int_{-1}^1 f\left(a + \frac{1+u}{1-u}\right) \frac{du}{(1-u)^2} \\
 &\approx \frac{a}{2} \sum_{i=1}^n w_i f(a(u_i+1)/2) + 2 \sum_{i=1}^n \frac{w_i}{(1-u_i)^2} f\left(a + \frac{1+u_i}{1-u_i}\right)
 \end{aligned}$$

Where  $a$  is selected such that  $[0, a)$  covers the majority of the behaviour of  $f(x)$ , and may be informed by properties of the underlying random variables (e.g.,  $a = \mathbb{E}[X] + 2\sqrt{\text{Var}[X]}$ ).

Gaussian quadrature schemes have the advantage that they can be readily implemented in spreadsheet software, since the weights and quadrature points can be hard-coded (provided a constant number of nodes is used).

Users are recommended to check that satisfactory convergence has been achieved, noting that errors will accumulate with arithmetic operations, and that convergence may depend on the values of parameters (e.g., in a probabilistic sensitivity analysis).



## Appendix 6: R code listings for example

### MGF method

```
# Load necessary libraries
library(tidyverse)
library(flexsurv)

# [... Define parameters ...]

# Probability density and cumulative distribution functions
f_X1_treatment <- function(x) dweibull(x, scale = lambda_treatment, shape =
k1)
F_X1_treatment <- function(x) pweibull(x, scale = lambda_treatment, shape =
k1)
f_X1_control <- function(x) dweibull(x, scale = lambda_control, shape = k1)
F_X1_control <- function(x) pweibull(x, scale = lambda_control, shape = k1)
f_X2 <- function(x) dgomperz(x, rate = b2, shape = a2)
F_X2 <- function(x) pgomperz(x, rate = b2, shape = a2)

#' Convenience function for calculating discounted QALYs accrued between A and
B
#'
#' @param mgf_a_0  $M_A^{(0)}(-r)$ 
#' @param mgf_a_1  $M_A^{(1)}(-r)$ 
#' @param mgf_a_2  $M_A^{(2)}(-r)$ 
#' @param mgf_b_0  $M_B^{(0)}(-r)$ 
#' @param mgf_b_1  $M_B^{(1)}(-r)$ 
#' @param mgf_b_2  $M_B^{(2)}(-r)$ 
MGF_qaly <- function(mgf_a_0, mgf_a_1, mgf_a_2, mgf_b_0, mgf_b_1, mgf_b_2) {
  ((
    (u0*mgf_a_0 + u1*mgf_a_1 + u2*mgf_a_2)*r^2 +
    (u1*mgf_a_0 + 2*u2*mgf_a_1)*r +
    (2*u2*mgf_a_0)
  ) - (
    (u0*mgf_b_0 + u1*mgf_b_1 + u2*mgf_b_2)*r^2 +
    (u1*mgf_b_0 + 2*u2*mgf_b_1)*r +
    (2*u2*mgf_b_0)
  ))/r^3
}

# Calculate probability  $X1 < X2$  given receive treatment
p_treatment <- integrate(
  function(x) { f_X2(x)*F_X1_treatment(x) },
  lower = 0,
  upper = Inf,
  rel.tol = 1e-8
)$value
```

```

# Calculate probability X1 < X2 given receive control
p_control <- integrate(
  function(x) { f_X2(x)*F_X1_control(x) },
  lower = 0,
  upper = Inf,
  rel.tol = 1e-8
)$value

# Calculate EMGFs for all variables with j = 0, 1, 2
GM_X1_treatment <- map_dbl(
  0:2,
  ~ integrate(
    function(x) { x^(.) * exp(-r*x) * f_X1_treatment(x) * (1-F_X2(x)) /
p_treatment },
    lower = 0,
    upper = Inf,
    rel.tol = 1e-8
  )$value
)
GM_X1_control <- map_dbl(
  0:2,
  ~ integrate(
    function(x) { x^(.) * exp(-r*x) * f_X1_control(x) * (1-F_X2(x)) / p_control
},
    lower = 0,
    upper = Inf,
    rel.tol = 1e-8
  )$value
)
GM_X2_treatment <- map_dbl(
  0:2,
  ~ integrate(
    function(x) { x^(.) * exp(-r*x) * f_X2(x) * (1-F_X1_treatment(x)) / (1-
p_treatment) },
    lower = 0,
    upper = Inf,
    rel.tol = 1e-8
  )$value
)
GM_X2_control <- map_dbl(
  0:2,
  ~ integrate(
    function(x) { x^(.) * exp(-r*x) * f_X2(x) * (1-F_X1_control(x)) / (1-
p_control) },
    lower = 0,
    upper = Inf,
    rel.tol = 1e-8
  )$value
)

```

```

)
GM_X3 <- map_dbl(
  0:2,
  ~ integrate(
    function(x) { x^(.) * exp(-r*x) * dlnorm(x, meanlog = mu3, sdlog = sigma3)
  },
  lower = 0,
  upper = Inf,
  rel.tol = 1e-8
) $value
)

# Combine EMGFs to calculate costs and QALYs
MGF <- data.frame(
  arm = factor(c("Treatment", "Control")),
  cost_stable = c(
    c_treatment / r * (1 - p_treatment*GM_X1_treatment[1] - (1-
p_treatment)*GM_X2_treatment[1]),
    c_control / r * (1 - p_control*GM_X1_control[1] - (1-
p_control)*GM_X2_control[1])
  ),
  cost_progression = c(p_treatment, p_control) * c_progression *
  c(GM_X1_treatment[1], GM_X1_control[1]),
  cost_death = c_death * c(
    p_treatment * GM_X1_treatment[1] * GM_X3[1] + (1-p_treatment) *
GM_X2_treatment[1],
    p_control * GM_X1_control[1] * GM_X3[1] + (1-p_control) * GM_X2_control[1]
  ),
  cost_progressive = c(p_treatment, p_control) * c_pd / r *
  c(GM_X1_treatment[1], GM_X1_control[1]) * (1 - GM_X3[1]),
  QALY_stable = v_sd * c(
    p_treatment * MGF_qaly(1, 0, 0, GM_X1_treatment[1], GM_X1_treatment[2],
GM_X1_treatment[3]) +
    (1-p_treatment) * MGF_qaly(1, 0, 0, GM_X2_treatment[1],
GM_X2_treatment[2], GM_X2_treatment[3]),
    p_control * MGF_qaly(1, 0, 0, GM_X1_control[1], GM_X1_control[2],
GM_X1_control[3]) +
    (1-p_control) * MGF_qaly(1, 0, 0, GM_X2_control[1], GM_X2_control[2],
GM_X2_control[3])
  ),
  QALY_progressive = v_pd * c(p_treatment, p_control) * c(
    MGF_qaly(
      GM_X1_treatment[1], GM_X1_treatment[2], GM_X1_treatment[3],
      GM_X1_treatment[1]*GM_X3[1],
      GM_X1_treatment[2]*GM_X3[1] + GM_X1_treatment[1]*GM_X3[2],
      GM_X1_treatment[3]*GM_X3[1] + 2*GM_X1_treatment[2]*GM_X3[2] +
GM_X1_treatment[1]*GM_X3[3]
    ),

```

```

    MGF_qaly(
      GM_X1_control[1], GM_X1_control[2], GM_X1_control[3],
      GM_X1_control[1]*GM_X3[1],
      GM_X1_control[2]*GM_X3[1] + GM_X1_control[1]*GM_X3[2],
      GM_X1_control[3]*GM_X3[1] + 2*GM_X1_control[2]*GM_X3[2] +
      GM_X1_control[1]*GM_X3[3]
    )
  )
) %>% transmute(
  arm = arm,
  cost = cost_stable + cost_progression + cost_death + cost_progressive,
  QALY = QALY_stable + QALY_progressive,
  NMB = QALY * threshold - cost
)

```

## Discrete event simulation

```

# Load necessary libraries
library(dplyr)
library(flexsurv)

# [... Define parameters ...]

# Convenience function to calculate discounted QALYs
qaly <- function(a, b) {
  (exp(-r*a)*((a^2*u2+a*u1+u0)*r^2 + (2*a*u2+u1)*r + 2*u2)-exp(-
r*b)*((b^2*u2+b*u1+u0)*r^2 + (2*b*u2+u1)*r + 2*u2))/r^3
}

DES <- data.frame(
  # Generate TTE random variables

  iter = seq(1, n_DES),
  X1.treatment = rweibull(n = n_DES, scale = lambda_treatment, shape = k1),
  X1.control = rweibull(n = n_DES, scale = lambda_control, shape = k1),
  X2 = rgompertz(n = n_DES, rate = b2, shape = a2),
  X3 = rlnorm(n = n_DES, meanlog = mu3, sdlog = sigma3)
) %>% mutate(
  # Calculate path through model
  progressed.treatment = (X1.treatment < X2),
  progressed.control = (X1.control < X2),

  # Calculate time in Stable state
  LY_stable.treatment = pmin(X1.treatment, X2),
  LY_stable.control = pmin(X1.control, X2),

  # Calculate total time
  LY.treatment = if_else(progressed.treatment, X1.treatment + X3, X2),

```

```

LY.control = if_else(progressed.control, X1.control + X3, X2),

# Calculate QALYs
QALY_stable.treatment = v_sd * qaly(0, LY_stable.treatment),
QALY_stable.control = v_sd * qaly(0, LY_stable.control),
QALY_progressive.treatment = if_else(progressed.treatment, v_pd *
qaly(X1.treatment, X1.treatment+X3), 0),
QALY_progressive.control = if_else(progressed.control, v_pd *
qaly(X1.control, X1.control+X3), 0),
QALY.treatment = QALY_stable.treatment + QALY_progressive.treatment,
QALY.control = QALY_stable.control + QALY_progressive.control,

# Calculate cost components
cost_stable.treatment = c_treatment / r * (1 - exp(-r *
LY_stable.treatment)),
cost_stable.control = c_control / r * (1 - exp(-r * LY_stable.control)),
cost_progression.treatment = if_else(progressed.treatment,
c_progression*exp(-r*X1.treatment), 0),
cost_progression.control = if_else(progressed.control, c_progression*exp(-
r*X1.control), 0),
cost_death.treatment = c_death * exp(-r*LY.treatment),
cost_death.control = c_death * exp(-r*LY.control),
cost_progressive.treatment = if_else(progressed.treatment, c_pd / r * (exp(-
r*X1.treatment) - exp(-r*(X1.treatment+X3))), 0),
cost_progressive.control = if_else(progressed.control, c_pd / r * (exp(-
r*X1.control) - exp(-r*(X1.control+X3))), 0),

# Calculate total costs
cost.treatment = cost_stable.treatment + cost_progression.treatment +
cost_death.treatment + cost_progressive.treatment,
cost.control = cost_stable.control + cost_progression.control +
cost_death.control + cost_progressive.control
)

```

## Markov cohort simulation

```

# Load necessary libraries
library(heemod)
library(tidyverse)

# Function to prepare Markov model for given cycle length and tunnel
# state limit
prep_MM <- function(.cycle_length, .state_time_limit) {
  lst(
    cycle_length = .cycle_length,
    state_time_limit = .state_time_limit,

    par_mod = define_parameters(

```

```

# Model parameters
dr_annual = 0.035,
cycle_length = .cycle_length,
dr = rescale_discount_rate(dr_annual, 1, cycle_length),
model_years = (model_time - 1) * cycle_length,
u0 = 0.95,
u1_abs = 0.002,
u1 = -u1_abs,
u2_abs = 0.0005,
u2 = -u2_abs,
v_sd = 0.9,
v_pd = 0.6,
c_treatment = 480,
c_control = 200,
c_progression = 3000,
c_death = 5000,
c_pd = 1000,
lambda_control = 1.5,
hr_treatment = 0.56,
k1 = 2,
lambda_treatment = lambda_control * hr_treatment ^ (-1 / k1),
a2 = 0.4,
b2 = 0.1,
mu3 = 0,
sigma3 = 1,

# Cumulative hazard of progression
CumHaz_progress_control_now = ((markov_cycle - 1) * cycle_length /
lambda_control) ^
    k1,
CumHaz_progress_control_next = (markov_cycle * cycle_length /
lambda_control) ^
    k1,
CumHaz_progress_control_incr = CumHaz_progress_control_next -
CumHaz_progress_control_now,
CumHaz_progress_treatment_now = ((markov_cycle - 1) * cycle_length /
lambda_treatment) ^
    k1,
CumHaz_progress_treatment_next = (markov_cycle * cycle_length /
lambda_treatment) ^
    k1,
CumHaz_progress_treatment_incr = CumHaz_progress_treatment_next -
CumHaz_progress_treatment_now,

# Cumulative hazard of death
CumHaz_death_stable_now = b2 / a2 * (exp(a2 * (markov_cycle - 1) *
cycle_length) - 1),

```

```

    CumHaz_death_stable_next = b2 / a2 * (exp(a2 * markov_cycle *
cycle_length) - 1),
    CumHaz_death_stable_incr = CumHaz_death_stable_next -
CumHaz_death_stable_now,
    CumHaz_sum_control = CumHaz_progress_control_incr +
CumHaz_death_stable_incr,
    CumHaz_sum_treatment = CumHaz_progress_treatment_incr +
CumHaz_death_stable_incr,

# Transition probabilities
p_remain_stable_control = exp(-CumHaz_sum_control),
p_remain_stable_treatment = exp(-CumHaz_sum_treatment),
p_progress_control = if_else(
  CumHaz_sum_control > 0,
  CumHaz_progress_control_incr / CumHaz_sum_control *
  (1 - exp(-CumHaz_sum_control)),
  0
),
p_progress_treatment = if_else(
  CumHaz_sum_treatment > 0,
  CumHaz_progress_treatment_incr / CumHaz_sum_treatment *
  (1 - exp(-CumHaz_sum_treatment)),
  0
),
p_death_stable_control = if_else(
  CumHaz_sum_control > 0,
  CumHaz_death_stable_incr / CumHaz_sum_control *
  (1 - exp(-CumHaz_sum_control)),
  0
),
p_death_stable_treatment = if_else(
  CumHaz_sum_treatment > 0,
  CumHaz_death_stable_incr / CumHaz_sum_treatment *
  (1 - exp(-CumHaz_sum_treatment)),
  0
),
p_death_progressive = define_survival(
  distribution = "lnorm",

  meanlog = mu3,
  sdlog = sigma3
) %>% compute_surv(time = state_time, cycle_length = cycle_length)
),

# Transition matrix in control arm
mat_control = define_transition(
  state_names = c("stable", "progressive", "death"),

```

```

    p_remain_stable_control, p_progress_control, p_death_stable_control,
    0, C, p_death_progressive,
    0, 0, 1
  ),

  # Transition matrix in treatment arm
  mat_treatment = define_transition(
    state_names = c("stable", "progressive", "death"),

    p_remain_stable_treatment, p_progress_treatment,
    p_death_stable_treatment,
    0, C, p_death_progressive,
    0, 0, 1
  ),

  # Stable state payoffs
  state_stable = define_state(
    cost_undisc = cycle_length * dispatch_strategy(control = c_control,
                                                    treatment = c_treatment),

    QALY_undisc = cycle_length * v_sd *
      (u0 + u1 * model_years + u2 * model_years ^ 2),
    cost = discount(cost_undisc, r = dr),
    QALY = discount(QALY_undisc, r = dr)
  ),

  # Progressive state payoffs
  state_progressive = define_state(
    cost_undisc = cycle_length * c_pd + if_else(state_time == 1,
c_progression, 0),
    QALY_undisc = cycle_length * v_pd *
      (u0 + u1 * model_years + u2 * model_years ^ 2),
    cost = discount(cost_undisc, r = dr),
    QALY = discount(QALY_undisc, r = dr)
  ),

  # Death state payoffs
  state_death = define_state(
    cost_undisc = if_else(state_time == 1, c_death, 0),
    QALY_undisc = 0,
    cost = discount(cost_undisc, r = dr),
    QALY = discount(QALY_undisc, r = dr)
  ),

  # Description of control arm
  strat_control = define_strategy(
    transition = mat_control,

    stable = state_stable,

```



```

    progressive = state_progressive,
    death = state_death
  ),

  # Description of treatment arm
  strat_treatment = define_strategy(
    transition = mat_treatment,

    stable = state_stable,
    progressive = state_progressive,
    death = state_death
  ),

  # Model characteristics
  time_horizon = 20,
  cycles = ceiling(time_horizon / .cycle_length)
)
}

# Function to run a Markov model which has been prepared
do_MM <- function(prepare) {
  res_mod <- run_model(
    parameters = prepare$par_mod,

    control = prepare$strat_control,
    treatment = prepare$strat_treatment,

    init = c(1, 0, 0),

    cycles = prepare$cycles,

    cost = cost,
    effect = QALY,

    state_time_limit = c(
      progressive = min(prepare$state_time_limit, prepare$cycles),
      death = 1
    ),

    method = "life-table"
  )

  res_mod_summary <- summary(res_mod, threshold = 20000)

  return(
    res_mod_summary$res_values %>%
      select(arm = .strategy_names, cost, QALY) %>%
      mutate(NMB = QALY*20000 - cost)
  )
}

```

```
}
```

```
prep <- prep_MM(1/12, 24)
```

```
results <- do_MM(prep)
```

## Markov microsimulation

```
# Load necessary libraries
```

```
library(tidyverse)
```

```
library(flexsurv)
```

```
# [... Define parameters ...]
```

```
# Multinomial sampling [Krijkamp et al. 2018]
```

```
samplelev <- function(probs, m) {
```

```
  d <- dim(probs)
```

```
  n <- d[1]
```

```
  k <- d[2]
```

```
  lev <- dimnames(probs)[[2]]
```

```
  if (!length(lev))
```

```
    lev <- 1:k
```

```
  ran <- matrix(lev[1], ncol = m, nrow = n)
```

```
  U <- t(probs)
```

```
  for (i in 2:k) {
```

```
    U[i, ] <- U[i, ] + U[i - 1, ]
```

```
  }
```

```
  if (any((U[k, ] - 1) > 1e-05))
```

```
    stop("error in multinom: probabilities do not sum to 1")
```

```
  for (j in 1:m) {
```

```
    un <- rep(runif(n), rep(k, n))
```

```
    ran[, j] <- lev[1 + colSums(un > U)]
```

```
  }
```

```
  ran
```

```
}
```

```
# Function to run the microsimulation [adapted from Krijkamp et al. 2018]
```

```
# Modifications:
```

```
# - Remove function arguments and rely on bindings being provided in  
# environment (certain variables/functions were already treated this  
# way)
```

```
# - Call Probs with 'dur' to allow sojourn time-dependent transition  
# probabilities
```

```
# - Call Costs with 'dur' to allow sojourn time-dependent costs
```

```
# - Call Effs with 't' to allow wall time-dependent utility
```

```
# - Calculate 'dur' as time in current state (not time in any disease  
# state)
```

```
MicroSim <- function(TR.out = TRUE, TS.out = TRUE, Trt = FALSE, seed = 1) {
```

```

# TR.out: should the output include a Microsimulation trace? (default is
TRUE)
# TS.out: should the output include a matrix of transitions between states?
(default is TRUE)
# Trt: are the n.i individuals receiving treatment? (scalar with a
Boolean value, default is FALSE)
# seed: starting seed number for random number generator (default is 1)

# Calculate the cost and QALY discount rates
v.dwc <- 1 / (1 + d.c) ^ (0:n.t)
v.dwe <- 1 / (1 + d.e) ^ (0:n.t)

# Create the matrix capturing the state name/costs/health outcomes for all
individuals at each time point
m.M <- m.C <- m.E <- matrix(
  nrow = n.i,
  ncol = n.t + 1,
  dimnames = list(paste("ind", 1:n.i, sep = " "),
                  paste("cycle", 0:n.t, sep = " "))
)

# Initial health state
m.M[, 1] <- v.M_1

# Set the seed for every individual for the random number generator
set.seed(seed)

# create the dur variable that stores the number of cycles the individual
has occupied the current state
# all individuals spend one cycle in the starting state
dur <- rep(1, n.i)

# estimate costs and QALYs per individual for the initial health state
m.C[, 1] <- Costs(m.M[, 1], dur, Trt)
m.E[, 1] <- Efs (m.M[, 1], 0, Trt)

for (t in 1:n.t) {
  # calculate the transition probabilities at cycle t
  m.p <- Probs(m.M[, t], dur, Trt)

  # sample the next health state and store that state in matrix m.M
  m.M[, t + 1] <- samplev(prob = m.p, m = 1)

  # Increment dur if stayed in the same state, or set to 1 if moved state
  remained <- m.M[, t + 1] == m.M[, t]
  dur[remained] <- dur[remained] + 1
  dur[!remained] <- 1
}

```

```

# estimate costs and QALYs per individual during cycle t + 1 conditional
on treatment
m.C[, t + 1] <- Costs(m.M[, t + 1], dur, Trt)
m.E[, t + 1] <- Efs(m.M[, t + 1], t, Trt)

# display the progress of the simulation
cat('\r', paste(round(t / n.t * 100), "% done", sep = " "))

}

# total (discounted) cost and QALYs per individual
tc <- m.C %>% v.dwc
te <- m.E %>% v.dwe

# average (discounted) cost and QALYs
tc_hat <- mean(tc)
te_hat <- mean(te)

if (TS.out == TRUE) {
  # create a matrix of transitions across states
  TS <- paste(m.M, cbind(m.M[, -1], NA), sep = "->")
  TS <- matrix(TS, nrow = n.i)

  # name the rows and columns
  rownames(TS) <- paste("Ind", 1:n.i, sep = " ")
  colnames(TS) <- paste("Cycle", 0:n.t, sep = " ")
} else {
  TS <- NULL
}

if (TR.out == TRUE) {
  TR <- t(apply(m.M, 2, function(x)
    table(factor(
      x, levels = v.n, ordered = TRUE
    ))))
  # create a distribution trace
  TR <- TR / n.i

  # name the rows and columns
  rownames(TR) <- paste("Cycle", 0:n.t, sep = " ")
  colnames(TR) <- v.n
} else {
  TR <- NULL
}

results <- list(
  m.M = m.M,
  m.C = m.C,
  m.E = m.E,

```

```

    tc = tc,
    te = te,
    tc_hat = tc_hat,
    te_hat = te_hat,
    TS = TS,
    TR = TR
  )
  return(results) # return the results
}

# Function to calculate transition probabilities for all individuals
Probs <- function(M_it, dur, Trt) {
  # M_it: health state occupied by individual i at cycle t (character
variable)
  # dur: the duration spent in the current state
  # Trt: whether patient is receiving Treatment (as opposed to Control)

  # create vector of state transition probabilities
  m.p.it <- matrix(NA, n.s, n.i)
  # assign names to the vector
  rownames(m.p.it) <- v.n

  # Calculate delta cumulative hazards
  # Stable to Progressive
  H.SP <-
  pweibull(
    q = (dur[M_it == "Stable"]-1)*c1,
    scale = Trt*lambda_treatment + (1-Trt)*lambda_control,
    shape = k1,
    lower.tail = FALSE,
    log.p = TRUE) -
  pweibull(
    q = dur[M_it == "Stable"]*c1,
    scale = Trt*lambda_treatment + (1-Trt)*lambda_control,
    shape = k1,
    lower.tail = FALSE,
    log.p = TRUE)
  # Stable to Dead
  H.SD <-
  pgompertz(
    q = (dur[M_it == "Stable"]-1)*c1,
    shape = a2,
    rate = b2,
    lower.tail = FALSE,
    log.p = TRUE) -
  pgompertz(
    q = dur[M_it == "Stable"]*c1,

```

```

    shape = a2,
    rate = b2,
    lower.tail = FALSE,
    log.p = TRUE)
# Progressive to Dead
H.PD <-
  plnorm(
    q = (dur[M_it == "Progressive"]-1)*c1,
    meanlog = mu3,
    sdlog = sigma3,
    lower.tail = FALSE,
    log.p = TRUE) -
  plnorm(
    q = dur[M_it == "Progressive"]*c1,
    meanlog = mu3,
    sdlog = sigma3,
    lower.tail = FALSE,
    log.p = TRUE)

# Probabilities leaving Stable
H.S <- H.SP + H.SD
p.SS <- exp(-H.S)
p.SP <- H.SP/H.S * (1 - exp(-H.S))
p.SD <- H.SD/H.S * (1 - exp(-H.S))

# Probabilities leaving progressive
p.PP <- exp(-H.PD)
p.PD <- 1 - exp(-H.PD)

# update the v.p with the appropriate probabilities
m.p.it[, M_it == "Stable"] <- rbind(p.SS, p.SP, p.SD)
m.p.it[, M_it == "Progressive"] <- rbind(0, p.PP, p.PD)
m.p.it[, M_it == "Dead"] <- c(0, 0, 1)
ifelse(colSums(m.p.it) == 1,
       return(t(m.p.it)),
       print("Probabilities do not sum to 1")) # return the transition
probabilities or produce an error
}

# Function to calculate costs across all individuals
Costs <- function(M_it, dur, Trt = FALSE) {
  # M_it: health state occupied by individual i at cycle t (character
variable)
  # dur:   the duration spent in the current state
  # Trt:   is the individual being treated? (default is FALSE)

  c.it <- rep(0, length(M_it))
  if (any(M_it == "Stable"))

```

```

    c.it[M_it == "Stable"] <- c1 * (Trt*c_treatment + (1-Trt)*c_control)
  if (any(M_it == "Progressive"))
    c.it[M_it == "Progressive"] <- c_progression * (dur[M_it == "Progressive"]
== 1) + c1 * c_pd
  if (any(M_it == "Dead"))
    c.it[M_it == "Dead"] <- c_death * (dur[M_it == "Dead"] == 1)

  return(c.it)
}

# Function to calculate QALYs across all individuals
Effs <- function(M_it, t, Trt = FALSE) {
  # M_it: health state occupied by individual i at cycle t (character
variable)
  # t: the current cycle
  # Trt: is the individual treated? (default is FALSE)

  u.baseline <- u0 + u1 * ((t + 0.5) * c1) + u2 * ((t + 0.5) * c1) ** 2

  u.it <- rep(0, length(M_it))
  u.it[M_it == "Stable"] <- u.baseline * v_sd
  u.it[M_it == "Progressive"] <- u.baseline * v_pd
  u.it[M_it == "Dead"] <- 0

  # calculate the QALYs during cycle t
  QALYs <- u.it * c1
  return(QALYs)
}

# Function factory
# Returns a function with all necessary parameters in a suitable
# environment
prep_MarkovMicrosim <- function(params, n_microsim, cycle_length) {
  e <- new.env(parent = globalenv())

  e$n.i <- n_microsim
  e$n.t <- floor(20 / cycle_length)
  e$c1 <- cycle_length
  e$v.n <- c("Stable", "Progressive", "Dead")
  e$n.s <- length(e$v.n)
  e$v.M_1 <- rep("Stable", e$n.i)
  e$d.c <- e$d.e <- (1 + params$dr) ** cycle_length - 1
  e$v.Trt <- c("Control", "Treatment")

  # Copy over parameters
  e <- list2env(params, e)

  l_MicroSim <- MicroSim

```

```

e$Probs <- Probs
e$Costs <- Costs
e$Effs <- Effs

environment(l_MicroSim) <- e
environment(e$Probs) <- e
environment(e$Costs) <- e
environment(e$Effs) <- e

return(l_MicroSim)
}

# Perform the microsimulation given a function from the function
# factory and combine results for control and treatment arms
do_MarkovMicrosim <- function(microsim) {
  control <- microsim(TR.out = FALSE, TS.out = FALSE, Trt = FALSE, seed = 1)
  treatment <- microsim(TR.out = FALSE, TS.out = FALSE, Trt = TRUE, seed = 1)
  return(
    bind_rows(
      tibble(arm = "Control", costs = as.vector(control$tc), QALYs =
as.vector(control$te)),
      tibble(arm = "Treatment", costs = as.vector(treatment$tc), QALYs =
as.vector(treatment$te))
    )
  )
}

results <- do_MarkovMicrosim(prepare_MarkovMicrosim(meanParams, 100000, 1/12))

```



## Appendix 7: Detailed description of capabilities of different methods

### Sojourn time-dependent transitions

Transitions between states can depend on the length of time in the current state (excluding case when individuals start in a given state and cannot return to it). Can use non-exponential time-to-event distributions.

MGF method

**Fully supported.** Any distribution for which  $\int_0^{\infty} x^j e^{-rx} dx$  is finite and amenable to analytic or numerical evaluation can be incorporated into the method.

Markov cohort without tunnel states

**Not supported.** There is no memory component in these models, so transition probabilities cannot depend on the length of time in the current state.

Markov cohort with tunnel states

**Partially supported.** Tunnel states provide a memory component allowing transition probabilities to vary, but the extent to which transitions can depend on sojourn time is limited by the number of tunnel states.

Markov microsimulation

**Fully supported.** Within a Markov microsimulation a full memory can be represented in the state and transition probabilities can be calculated accordingly.

Discrete event simulation

**Fully supported.** Fundamental property of the method.

### Wall time-dependent transitions

Transitions between states can depend on the length of time since the model start (excluding case when individuals start in a given state and cannot return to it). Typical example is age-related other cause mortality.

MGF method

**Not supported (at present).** This capability has not yet been incorporated into the method.

Markov cohort with/without tunnel states

**Fully supported.** Transition probabilities are updated for each cycle. Tunnel states are not required.

Markov microsimulation

**Fully supported.** Transition probabilities are updated for each cycle.

Discrete event simulation

**Fully supported.** Transitions can be modelled by events whose time origin is the model start. Also, new events can be sampled conditionally on the current time elapsed (or any other aspect of an individual's history).

### **Kaplan–Meier survival**

Directly use Kaplan–Meier survival curves, without fitting a parametric model or piecewise exponential model. Particularly valuable when parametric models give a poor fit.

MGF method

**Fully supported.** Described in the section *Methods for evaluating MGFs*.

Markov cohort with/without tunnel states and Markov microsimulation

**Partially supported.** The use of the time cycle means that Kaplan–Meier survival curves can only be represented up to a certain fidelity.

Discrete event simulation

**Fully supported.** Given a Kaplan–Meier curve, its inverse (the quantile function) can be obtained, and a uniform random variable between 0 and 1 can be sampled in order to sample a random time-to-event in accordance with the Kaplan–Meier curve.

### **Sojourn time-dependent payoffs**

Payoffs depend on the time in the current state, e.g., costs are initially high within a state but then diminish.

MGF method

**Partially supported.** The method supports one-off payoffs, polynomial payoffs and exponential payoffs (and any linear combination of these). It does not currently support, e.g., piecewise constant payoffs.

Markov cohort without tunnel states

**Not supported.** Without tunnel states there is no memory of how long an individual has been in a state.

Markov cohort with tunnel states

**Partially supported.** Dependency is limited by the number of tunnel states.

Markov microsimulation

**Fully supported.**

Discrete event simulation

**Fully supported.**

### **Wall time-dependent payoffs**

Payoffs depend on the time since the model start, e.g., health state utility values decline with age.

MGF method

**Partially supported.** The method supports one-off payoffs, polynomial payoffs and exponential payoffs (and any linear combination of these). It does not currently support, e.g., piecewise constant payoffs.

Markov cohort with/without tunnel states

**Fully supported.**

Markov microsimulation

**Fully supported.**

Discrete event simulation

**Fully supported.**

### **Monte Carlo variation**

There is random error introduced into results, which must either be minimised with a large number of simulations, or accounted for within statistical analyses.

MGF method

**No** (unless a Monte Carlo algorithm is used for numerical integration, although this is unlikely to be necessary as deterministic numerical integration algorithms are well suited to one-dimensional integrals).

Markov cohort with/without tunnel states

**No.**

Markov microsimulation

**Yes.**

Discrete event simulation

**Yes.**

### **Convergence behaviour**

The global truncation error as a function of  $n$ , which is the number of quadrature nodes for the MGF method, the cycle length for the Markov cohort simulations, and the number of simulations for the microsimulation methods. In general one would prefer an algorithm with exponential convergence, i.e.,  $\mathcal{O}(c^{-n})$  for some  $c > 1$  to an algorithm

with polynomial convergence  $\mathcal{O}(n^{-b})$  for some  $b$ . One would also prefer an algorithm with error  $\mathcal{O}(n^{-b})$  to an algorithm with error  $\mathcal{O}(n^{-a})$  if  $b > a$ .

#### MGF method

Behaviour depends on the integrands. If numerical integration is not required then there is no error from the method (except for rounding errors). If all numerical integration is conducted on  $[-1,1]$  and all functions are analytic in this domain, exponential convergence will be obtained,<sup>25</sup> i.e.,  $\mathcal{O}(c^{-n})$  for some  $c > 1$ ; otherwise, if a function is not analytic (e.g., Weibull with shape parameter  $\gamma < 1$ ) it is challenging to derive a general error bound, since the  $2n^{\text{th}}$  derivative of the function must be obtained<sup>26, 27</sup>

#### Markov cohort without tunnel states

$$\mathcal{O}(n^{-1})$$

The global truncation error is  $\mathcal{O}(n^{-1})$  since it is  $\mathcal{O}(h)$  (where  $h$  is the step size) and  $h \propto n^{-1}$ .

#### Markov cohort with tunnel states

$$\mathcal{O}\left(n^{-\frac{1}{2}}\right)$$

As for a Markov cohort without tunnel states the global truncation error is  $\mathcal{O}(h)$  however to halve the step size it is necessary to double the number of cycles and double the number of tunnel states, i.e.,  $h \propto n^{-\frac{1}{2}}$ .

#### Markov microsimulation

$$\mathcal{O}\left(n^{-\frac{1}{2}}\right)$$

There are two sources of error in a Markov microsimulation. First is the error inherent in a Markov model without tunnel states (tunnel states are not needed in a Markov microsimulation), and additionally there is Monte Carlo error. Monte Carlo error is the greater of these and is  $\mathcal{O}(n^{-1/2})$ .

#### Discrete event simulation

$$\mathcal{O}\left(n^{-\frac{1}{2}}\right)$$

Discrete event simulation only includes Monte Carlo error, which is the same as for Markov microsimulation.

## Appendix 8: Assumption of independence of random variables

It is assumed in the method outlined in this paper that the time-to-event random variables in a model are independent. This is important because even though  $\mathbb{E}_{X,Y}[X + Y] = \mathbb{E}_X[X] + \mathbb{E}_Y[Y]$  it is not generally true that  $M_{X+Y}(t) = M_X(t)M_Y(t)$  – it is only true if  $X$  and  $Y$  are independent.

There are two types of dependence we are likely to consider incorporating into a model:

- $X$  and  $Y$  represent competing events in a single state (e.g.,  $X$  represents time to receiving a kidney transplant and  $Y$  represents death on dialysis) but one (the greater) is always a latent variable since the random variable taking the lesser value represents the event which takes place while the competing event does not take place;
- $X$  and  $Y$  represent events which can occur sequentially (e.g.,  $X$  represents time to receiving a kidney transplant and  $Y$  represents time to graft failure).

The first thing we note is that if  $X$  and  $Y$  are conditionally independent given another random variable,  $Z$ , then we are able to proceed as before and take the expectation over  $Z$ .

For example, consider:

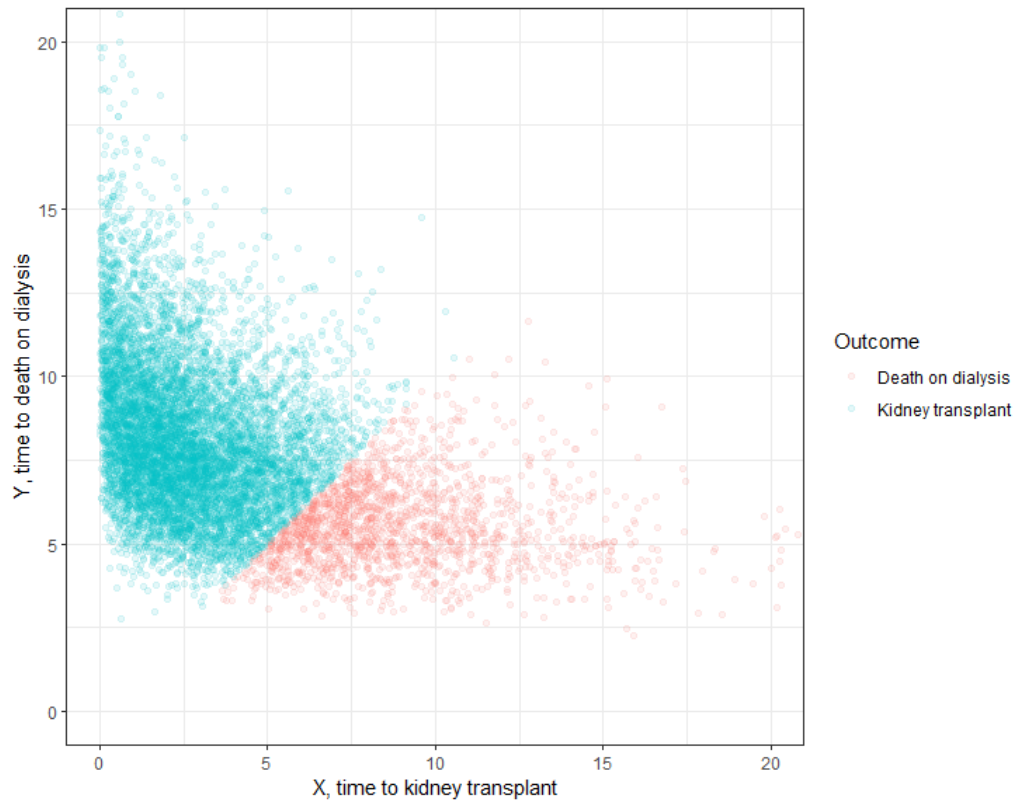
- $X$  is the time to receiving a kidney transplant;
- $Y$  is the time to graft failure;
- $Z$  is a measure of immunological risk (which makes it harder to obtain a kidney transplant and increases the risk of graft failure).

We would partition the population according to values of  $Z$ , evaluate the model substituting the conditional distributions for  $X$  and  $Y$  given  $Z$  and treating them as independent, then take a weighted average of the results according to the probabilities of the different values of  $Z$ .

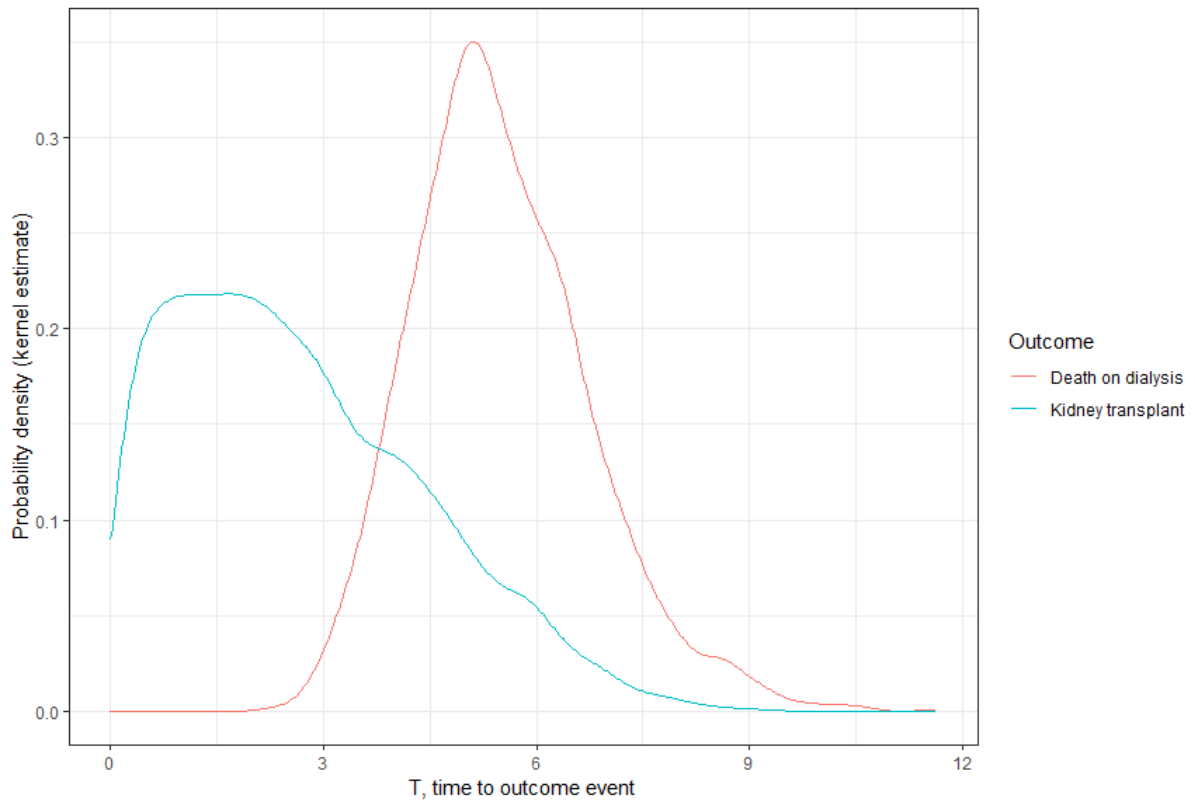
<b>Z, immunological risk</b>	<b>Pr(Z)</b>	<b>Costs (£)</b>	<b>QALYs</b>
Low	30%	60,000	20.0
Moderate	50%	80,000	14.0
High	20%	120,000	10.0
<b>Weighted average</b>		<b>82,000</b>	<b>15.0</b>

If  $X$  and  $Y$  represent competing events then it does not necessarily make sense to talk about them being dependent, since only one is ever observed so it is not possible to deduce their joint probability distribution. What can be deduced though is the joint probability distribution of  $D$  (a categorical random variable denoting which event takes place) and  $T$  (the time-to-event variable for the event which takes place). The pattern-mixture approach (as used in statistical flowgraph modelling) represents this joint distribution by modelling  $\Pr(D)$  and  $f(T | D)$ , since  $f(D, T) = f(T | D) \Pr(D)$ . There is no restriction that this should be equivalent to an alternative formulation with  $X$  and  $Y$  as independent random variables and setting  $D = \operatorname{argmin}\{X, Y\}$  and  $T = \min\{X, Y\}$ .

For example, suppose that  $X$  and  $Y$  represent the competing events of receiving a kidney transplant and dying on dialysis. Suppose that there is some negative correlation in the latent time-to-event variables, such that an individual with a short time to kidney transplantation would tend to have (in the absence of the transplantation) lived for a long time on dialysis and vice versa, as shown in the figure below.



This can be represented in the pattern-mixture approach by calculating the proportion of patients with each outcome and then the appropriate distribution for the time spent on dialysis (prior to transplantation or death) is estimated separately for each outcome, as shown in the figure below.



In summary, we conclude that the restriction that TTE variables in the model be independent is not a significant restriction as conditional independence can be exploited, by conditioning on some covariate (like immunological risk in the example above) or on the branch taken in the model (e.g., death on dialysis versus kidney transplant in the next example).