UiO : Faculty of Medicine
University of Oslo

Master Thesis

A model-based cost-effectiveness analysis of liver transplantation for patients with nonresectable colorectal cancer: comparing deceased and live liver donation and their spillover effects.

Tung Duy Nguyen

Department of Health Management and Health Economics

The Faculty of Medicine

Thesis submitted as a part of the Master of Philosophy in Health Economics, Policy, and Management

Supervisor: **Emily Burger and Gudrun Bjørnelv**

UNIVERSITY OF OSLO

May 16th, 2021

Table of Contents

Abstract

Background: In Norway, the standard of care for patients with non-resectable colorectal cancer with liver-only metastasis (NCRLM) is palliative chemotherapy alone, which results in a dismal 5-year overall survival rate of 10 percent. The current two trials of offering liver transplantation from a deceased donor to NCRLM patients in Norway, namely SECA-I and SECA-II, each with a different set of criteria, have proven improvement on life year saved for the patients, increasing the 5-year overall survival rate to 60% and 83%, respectively. Sjule and Vinter discovered, however, that the inclusion of NCRLM on the current deceased liver waiting list in Norway comes at the expense of the existing Status Quo patients on the list. The inclusion of NCRLM patients can result in longer waiting times and a reduction in life year for Status Quo patients. Living donor liver transplantation has the potential to be offered to NCRLM patients with the same efficacy as liver transplantation from deceased donors as a result of recent advances in surgical technique. In addition, a previous study by Bjrnevol et al. demonstrated that offering NCRLM patients deceased liver transplantation was cost-effective for highly selected patients.

Objective: This study aims to build a model to assess cost-effectiveness of both deceased and living donor liver transplantation for NCRLM patients, from both perspective of NCRLM group only and the entire population where there is spillover effect.

Methods: All three patient groups, NCRLM, Status Quo, and donors, were simulated using a discrete event simulation (DES) model across three strategies for NCRLM patients: palliative care only, deceased liver transplantation, and living donor liver transplantation. DES was chosen since it is a flexible model that allows for interaction among groups of patients.

Results: The model results show that, the expected cost per life year gain of living donor transplantation was 655,643.0 NOK under NCRLM perspective and 1,038,151.9 NOK for entire population perspective. In all circumstance, the highest cost per QALY gain for deceased liver transplantation was 805,342.6 NOK

Conclusion: Living donor transplantation was no longer cost-effective when the cost per life year saved criterion was applied from a broader perspective, whereas deceased liver transplantation remained cost-effective in all circumstances. Spillover effect can alter the cost-effectiveness of a strategy.

Acknowledgements

Firstly, I am profoundly grateful to my supervisors, Emily Burger and Gudrun Bjørnelv, for your unwavering support, guidance, and expertise throughout the duration of this thesis project. And most importantly, thank you so much for your patience and for inspiring me to complete this challenging project. This would not have been possible without our in-depth conversations! I never felt I was on my own on this project!

Second, I would like to sincerely thank Dr. Svein Dueland and Dr. Pål-Dag Line for their insightful commentary regarding living donor transplantation in Norway. Furthermore, I also want to express my appreciation to the University of Oslo and all of the professors at the Department of Health Management and Health Economics, Faculty of Medicine, for providing all of the instructive lectures and courses for the Master of Philosophy in Health Economics, Policy, and Management program.

Finally, I would like to acknowledge my family, friends, and classmates who have supported me throughout the two years of this program. I would like to thank my close friend, Vo Hai, for imparting to me a wealth of R programming knowledge and providing constructive feedbacks on my modeling ideas. Especially, I would like to thank my wife, Trang Bui, and my 4-year-old son, Quan-Anh Nguyen, for being a tremendous source of motivation and laugh.

Tung NGUYEN,

Oslo, May 2022.

List of Tables and Figures

List of Abbreviations

AC	Alcoholic Cirrhosis
AIC	Akaike Information Criterion
ALF	Acute Liver Failure
BIC	Bayesian Information Criterion
CEA	Cost Effectiveness Analysis
CRC	Colorectal Cancer
CRLM	Colorectal cancer with liver-only metastasis
DES	Discrete Event Simulation
EUR	Euro
HCC	Hepatocellular Carcinoma
HTA	Health Tech Assessment
ICER	Incremental Cost Effectiveness Ratio
INMB	Incremental Net Monetary Benefit
LY	Life year
NCRLM	Non-resectable colorectal cancer with liver-only metastasis
NOK	Norwegian Krone
PBC	Primary Biliary Cirrhosis
PSA	Probability Sensitivity Analysis
PSC	Primary Sclerosing Cholangitis
QALY	Quality-adjusted life year
RCT	Randomized Control Trial
WHO	World Health Organization
WTP	Willingness to pay

1. Introduction

Colorectal cancer (CRC) is the third most common type of cancer worldwide and ranks second in terms of mortality [1]. Norway has seen a dramatic increase in CRC over the last 50 years, with more than 12 patients diagnosed daily between 2015 and 2019 [2, 3]. Over half of CRC patients are diagnosed with metastases, primarily to the liver, or will develop them within five years. The only curative option is liver resection, but only about 10 percent to 20 percent of patients meet the criteria for surgery, depending on selection criteria [4]. For patients with nonresectable colorectal cancer with liver-only metastases (NCRLM), who are not resection surgical candidates but who receive palliative chemotherapy, the 5-year overall survival rate is approximately 10 percent [5].

According to the available evidence, liver transplantation may be a viable option for patients with NCRLM. Since 2011, Norway has conducted two clinical trials, SECA-I and SECA-II. Compared to palliative chemotherapy alone, offering NCRLM patients a liver transplant increased 5-year overall survival rates from 10 percent to 60 percent and 83 percent, respectively, for selected patients meeting SECA-I and SECA-II study criteria [6, 7].

Though liver-transplantation has been determined to be cost-effective compared to palliative chemotherapy alone in patients with NCRLM [8], liver transplantation for NCRLM is not yet considered standard of care in Norway, due in part to the surgery's stringent selection criteria. The primary impediment to offering liver transplantation to NCRLM patients is graft scarcity, as evidenced by various transplantation centers worldwide. In addition, as Sjule and Viner discovered in their 2021 thesis, although the total life year of all patients increased, offering liver transplantation from deceased donors comes with the opportunity cost of life years lost for patients currently on the liver transplant waiting list. They also discovered that as the number of NCRLM patients on the waiting list increased, patient wait times increased proportionally [9]. This necessitates an alternative for minimizing the negative impact on the current waiting list for deceased livers while maintaining the NCRLM group's life-year gain from liver transplantation.

Under current advance of technique, the transplantation could be done with the use of small segmental auxiliary grafts from either deceased or living donors, namely RAPID (**R**esection **A**nd **P**artial L**i**ver Segment 2/3 Transplantation With **D**elayed Total Hepatectomy) technique. The initial results from transplantation studies for the NCRLM patients applying the technique were promising with a number of successful transplant, though there is lack of post-operative

monitoring [10-12]. This creates the possibility of offering NCRLM patients a liver transplant from a living donor. To truly consider live donor transplantation as a new intervention for NCRLM, it is necessary to access two of the three pillars of Norwegian priority setting, namely the anticipated health gain and the used resources. Due to limited clinical evidence for live liver donations for NCRLM patients, with only one ongoing clinical trial, an exploratory cost-effectiveness analysis of living donor liver transplantation for NCRLM patients could be performed as part of an Early Health Technology Assessment (HTA). The model could aid in determining whether the characteristics of live liver donation satisfy the Norwegian criteria for cost-effectiveness. The following research aim is therefore posed:

> *From the healthcare perspective, including spillover effects, is offering living donor liver transplantation to patients with nonresectable colorectal cancer with liver-only metastases (NCRLM) in Norway cost-effective compared with either transplantation from deceased liver only or the current standard of care involving palliative care with chemotherapy only?*

To answer the research question, it is necessary to estimate the costs and effects of each alternative strategy for NCRLM patients, but in a broad perspective. Spillover effect refers to the fact that the intervention aimed at NCRLM patients has an impact on the costs and health outcomes of the other patient groups as well. For illustration, there is a strong need to account for the spillover effect for the inclusion of NCRLM patients on the current waiting list for deceased organs in Norway, given the existence of evidence for life year loss [9]. Also, while offering transplantation from living donors, there must be health consequences and costs for the donors. Resultantly, all three strategies will be evaluated using a discrete event simulation model that takes queueing effects and resource constraints into account.

This dissertation is organized as follows: chapter 2 provides background information on the disease and treatment strategies; chapter 3 provides a theoretical framework; chapter 4 describes the model and data inputs; chapter 5 summarizes the findings; chapter 6 discusses the cost-effectiveness of three alternatives, study's limitation and pertinent ethical topic; and chapter 7 concludes and answers the research questions. For the sake of clarity, a technical model will be included as an appendix.

2. Background

2.1. Nonresectable colorectal cancer with liver only metastasis

Colorectal carcinoma is one of the most common types of cancer in developed countries. Colorectal cancer was diagnosed in 4295 patients in Norway in 2019, accounting for 12.3 percent of all cancer diagnoses. The malignance is also the second leading cause of cancer death in Norway, accounting for 14 percent of all deaths, placing Norway's colorectal cancer incidence and mortality among the highest in the world [13]. Liver metastases will develop in 50% of patients within five years of diagnosis. Only liver resection surgery is curative for colorectal cancer with liver-only metastasis (CRLM). However, only about 20% of CRLM are candidates for resection [4]. For nonresectable patients (NCRLM) the current standard of care is palliative care combined with chemotherapy, which results in a poor overall survival rate of about 2 years and a 5-year overall survival rate of less than 10 percent [7].

2.2. Liver transplantation

In recent years, liver transplantation has become the standard of care for patients with end-stage liver disease and certain types of malignant liver tumors [14]. Currently, approximately 100 patients receive a liver transplantation in Norway each year, including re-transplantation, with a 5-year overall survival rate of nearly 85 percent [15]. Hepatocellular Carcinoma (HCC) (13.7 percent), Primary Biliary Cirrhosis (PBC) (6.7 percent), Primary Sclerosing Cholangitis (PSC) (18 percent), Acute Liver Failure (ALF) (8.3 percent), and Alcoholic Cirrhosis (AC) (10.1 percent) are the five most common indications for liver transplantation in Norway, accounting for approximately 57 percent of transplants performed [16]. The remaining proportion is for the less frequent diseases such as Biliary Atresia and Post-Hepatitis B cirrhosis. It is notable that 5-year post-operation survival of patients from different diseased varies, ranging from 72 percent to 96 percent [15].

Norway is currently among the countries with the shortest waiting time, with a median of only 35 days [16], compared to up to 11.3 months in the United States [17]. Recently, the only source of liver donation in Norway is from deceased donors who are either circulatory or brain dead. Aside from domestic donors, Norway participates in Scandia Transplant, an organ exchange program with other Nordic countries. In a circumstance such that an organ is accessible but there isn't a compatible recipient within a country, the program permits a member to receive liver from other nations or vice versa, exporting liver to other countries [15]. In

addition, living donor liver transplantation is being explored as an alternative, with a handful of surgeries performed in Nordic regions in 2021, including 1 from Norway [18].

2.2.1. Deceased liver transplantation for NCRLM patients

Liver transplantation has been shown to be significantly more effective than chemotherapy alone in a group of NCRLM patients, with a significant increase in overall survival. Two trials were conducted in Norway, namely SECA-I and SECA-II [19]. The overall 5-year survival rate of the SECA-I was 56 percent which is significantly higher than the rate of 10 percent for palliative care alone [7]. However, neither of these two trials is randomized, but rather consists of patients who have been carefully chosen based on the study's criteria[19].

Using the knowledge gained from the SECA-I study regarding the predictive factors, the SECA-II study applied more stringent selection criteria for liver transplantation to NCRLM patients [19]. These adverse factors include a maximum tumor size of 55 cm, progressive disease on chemotherapy, a less than 24-month interval between primary tumor resection and transplantation, and a pretransplant carcinoembryonic antigen level greater than 80 µg/ml [14]. In comparison to SECA-I, two of the inclusion criteria in this experiment were that the patient had to have a 10 percent response to chemotherapy and that the duration from diagnosis to liver transplantation had to be more than one year, rather than merely 6 weeks as in SECA-I. The 5-year overall survival of patients in SECA-II trial was reported at 83 percent [6].

The use of a deceased liver for transplantation has some drawbacks. First and foremost, deceased graft is scarce. As a result, including NCRLM patients on the waiting list comes at the expense of those patients who are already on the list. Furthermore, the allocation of deceased organs is associated with a significant cost of organ harvesting and transportation [8].

2.2.2. Live donor liver transplantation

New techniques, such as the RAPID technique, for NCRLM patients is a procedure that may allow for the expansion of the liver donor pool [11]. The technique was first introduced by a Line and colleagues, an Oslo-based group of surgeons. RAPID refers to “Resection And Partial Liver Segment 2/3 Transplantation With Delayed Total Hepatectomy” [20]. The RAPID technique is based on the principle of transplanting a small auxiliary left liver graft and at the same time, ligating the right portal vein. At later stage, when the transplanted graft has grown sufficiently large, residual hepatectomy will follow.

By 2020, a clinical trial, namely DD-RAPID-Procedure (NCT02215889), had been implemented in Oslo. There were 3 out of 6 patients had been transplanted with the technique.

Except for one patient who died 40 days after transplantation from graft hepatic artery thrombosis and biliary sepsis, two other patients survived the surgery. Of these, one has survived disease-free for 5.5 years, while the other is still alive two years after transplantation but has lung and skeletal recurrence [11]. Overall, compare to overall survival of NCRLM patients with standard of care chemotherapy, use of RAPID liver donation is promising.

However, given the aforementioned disadvantages of deceased donor liver transplantation and the fact that several countries do not yet have a split policy for deceased organs, the RAPID technique from deceased donor appears to fall short of the goal of expanding the donor pool. This circumstance leads to an novel approach of using RAPID technique but with a living donor, namely LD-RAPID [12]. The consideration is now shifts from the life years lost by other listed patients, if NCRLM patients were included on the waiting list, to the risk toward the living donor. To minimize the trade-off for the additional years of life gained by transplanted patients, the LD-RAPID procedure was developed with the goal of minimal risk for the donors [11].

Currently, eight LD-RAPID procedures have been performed worldwide [10]. Although it is still too early to determine the effectiveness of the procedure, initial results are encouraging. No reports of donor morbidity or mortality have been reported. An ongoing trial of offering liver donor transplantation for NCRLM patients, i.e., 'The LIVER- T(W)O – HEAL study' (NCT 03488953), has been initiated in Germany [10].

3. Theoretical Framework

This section aims to provide fundamental characteristics of health economics and health economic evaluation. The overall objective is to define the theoretical framework, which includes the economic evaluation approach and the health economic modeling technique chosen to address the research question.

3.1. Health economic perspective

3.1.1. Health economics and priority setting

Health economics, like other branches of economics, is concerned with how to allocate scarce resources to satisfy unlimited desires. In Norway, the overarching goal of health policy is to maximize the general population's healthy years of life. Regulation §2 in "The prioritization regulation" issued by the Norway Ministry of Health and Care Services state that Norwegian citizens have the right to necessary health care and will be given prioritization according to 1)

the expected benefits of health care; 2) the severity of the condition; and 3) the less of resources used [21]. In which, the expected health benefits are defined as increased patient's life expectancy and/or quality of life as a result of survival, reduced function loss; physical or mental functional improvement; reduction or avoidance of pain, physical or mental discomfort. These factors are also used to define the severity of a condition, such as the increased risk of death posing a threat to survival. These principles are especially fundamental when it comes to economic evaluation of health care which involves several groups of patients. All three pillars are typically considered concurrently, so an intervention that yields greater effects or treats a more severe disease could be deemed cost-effective despite being resource intensive. In Norway, disease severity is a significant determinant of the willingness to pay threshold for a new treatment. [22].

3.1.2. Economic evaluation and cost-effectiveness analysis

Health economic evaluation is a technique that supports policymakers in determining the most efficient use of scarce health care resources. Drummond et al. define economic evaluation as "*the comparative analysis of alternative courses of action in terms of both their costs and consequences* " [23]. This definition implies that a comprehensive economic analysis must consider both the costs and consequences of alternative interventions.

Numerous economic theories, such as the 'extra-welfarist approach,' 'decision analyses,' and 'constrained optimization,' serve as underlying principles for economic evaluation [24]. These theories serve as a framework for economic evaluation, defining key components such as expected values of outcomes (with discounting of future values), payoffs and constrained resources, and trade-offs. Additionally, these theories define the boundaries of an economic evaluation by requiring it to have a distinct target population (i.e. countries, groups of patients), analytic perspective (i.e. societal, healthcare, or payer), and scope (relevant outcomes and time horizon) [23].

Cost-effectiveness analysis (CEA) is one of the most frequently used economic evaluation techniques. Costs are typically quantified in monetary terms, whereas consequences are typically quantified in natural terms such as life years gained, quality-adjusted life years (QALYs), or affected cases averted. Norway strongly recommends CEA as an approach for conducting health economic evaluations [25].

3.1.3. Components of economic evaluation

Alternative strategies

Economic evaluation, by definition, is the process of comparing alternative courses of action, or in other words, alternative strategies or interventions. They must be unambiguously identified, and ideally, all relevant alternatives should be included. If a critical strategy is omitted from the comparison, the comparison may be biased, and the preferred strategies may be misidentified. Importantly, the current treatment alternative needs to be included.

Outcomes

Depending on the decision-making process, outcomes can be measured in various ways, such as the number of cancers prevented, as in the case of evaluating a screening program, or total life years gained, as in the case of diseased treatment. Another widely used measure is Quality Adjusted Life Years (QALYs) which is able to capture both quantity and quality of life [23]. QALY is widely applied as a measure for outcomes of the alternatives in the CEA, and is the recommended measure of outcome in Norway

The concept was that each health state has a utility or weight associated with it, namely health-related quality of life (HRQoL). HRQoL, which ranges from 0 to 1, reflects the patient's physical and mental well-being in comparison to the perfect health state. After that, the QALYs are calculated by multiplying the total number of life years by the HRQoL associated with those years in a particular health state. HRQoL can be quantified in a variety of ways, either directly or indirectly. Certain direct methods are referred to as standard gambles or time trade-offs [23]. The indirect methods employ several multi-utility scales to measure HRQoL, such as EQ-5D, Short Form 6D, 15, etc. [23]. However, the researcher should be fully aware of the potential for bias in HRQoL measurement, as it is heavily influenced by patient preference.

Costs

The fundamental concept of cost is that the resources employed to execute each strategy are quantified in monetary terms. There are several cost items that can be associated with an intervention, including medical costs, patient time costs, and caregiver time costs. Which costs to include is determined by the analysis perspective. Additionally, costs are typically gathered from a variety of sources that document the cost at various points in time and in a variety of currencies. As a result, historical expenses should be inflated to current time and a common currency should be used.

Time horizon

The chosen horizon must be long enough to capture all the relevant costs and outcomes of the considered strategies since they are the criteria to compare the effectiveness among the strategies. In order make a valid comparison, the cost and benefit across the time horizon must be discounted to a single point in time.

Discounting

Discounting is based on the concept of time value, with people placing a higher premium on consumption today than on consumption in the uncertain future. Thus, the future value should be discounted to current time in order to make a sound comparison. Applying the same concept, costs collected in the past should be inflated to a value of today to correctly reflect the value of time.

There are two types of values for discounting: one-off values and over-a-period values. The following are the discounting formulas [26]:

Discount of one-off value at *time* back to the model initiation point with discounting rate *r*:

$$Discounted\ value = \frac{value}{(1+r)^{time}}$$

Discounting value incurred over a period of time, with *value per unit of time*, beginning at time A and ending at time B in the model, back to the model's initiation point at a discount rate of *r*:

$$Discounted\ value = \frac{value}{r}(e^{-r \times timeA} - e^{-r \times time\ B})$$

3.1.4. Cost-effectiveness

The objective of economic evaluation is to determine which strategy is deemed cost-effective and should be favored. As the strategies are mutually exclusive, the incremental cost-effectiveness ratio (ICER) is used to determine the optimal strategy. Basically, ICER of strategy A over strategy B is calculated as:

$$ICER_{A\ over\ B} = \frac{\Delta_{Cost}}{\Delta_{Benefit}} = \frac{Cost_A - Cost_B}{Benefit_A - Benefit_B}$$

Clearly, strategies with a higher cost and fewer benefits will be dominated. In contrast, a strategy with lower costs and greater returns will utterly dominate others. In the event of a positive ICER, a higher ICER will also be regarded as being weakly dominated. The remaining

strategies shall be deemed cost-effective. Nevertheless, it is typically contingent on the threshold of willingness to pay. If benefit is measured in QALYs, then a positive ICER can be interpreted as the cost per QALY gained by selecting A over B. Typically, each setting has its own willingness to pay (WTP) per QALY threshold. In Norway, the WTP threshold is determined by the severity of the disease, which is typically measured by absolute shortfall, or the loss of future healthy life years. The more severe the disease, the higher the WTP threshold. A WTP to pay of 850,000 NOK per healthy life year will be assigned to diseases with an absolute shortfall of more than 20 years, which are categorized as the most severe diseases group [22].

Once the WTP threshold is known, an alternative calculation can be used to define the optimal strategy which is incremental net monetary benefit (INMB), calculated as follow:

$$INMB = (Benefit_A - Benefit_B) \times WTP\ Threshold - (Cost_A - Cost_B)$$

In short, a positive INMB shows that the value of the additional benefits (Δ*Benefit * WTP Threshold)* is worth more than its costs (Δ*Cost)*, from the perspective that define the WTP threshold.

3.1.5. Early health technology assessment

Early health technology assessment is a subset of health technology assessment (HTA) that evaluates technologies in their infancy. The evaluation could consist of an early examination of the potential implications of an intervention, as well as its financial repercussions, in order to determine the intervention's potential cost-effectiveness [26]. Costs and outcomes of an intervention are related to the sequence of events or health states that patients would encounter. Due to the novelty of the technology, a lack of costs information and clinical effect evidence is to be expected. Consequently, a model for early HTA could utilize a variety of data sources, such as administrative data, expert opinions, observational studies, and non-randomized trials. HTA is typically conducted from the perspective of the decision maker or the health care system, but other stakeholders may be involved. Involvement of another stakeholder may prompt consideration of opportunity costs, which are sacrificed when the new intervention is implemented [26].

3.2. Decision-analytic modelling

Randomized controlled trials (RCT) are frequently used as the gold standard for establishing the efficacy of a new medical technology or method of treatment. However, single trials can have a small sample size and a short time horizon with surrogate endpoints, which do not

capture all relative costs and consequences relevant over the course of the intervention, especially in multiple settings. When a new treatment technology is in its infancy, normally, an RCT cannot be conducted. Thus, economic evaluation often requires mathematical simulation modeling, i.e., decision-analytic modelling, that enables the synthesis of evidence from multiple sources as well as extrapolation over a longer time horizon [23]. Decision-analytic modelling supports decision-making in health care and other societal policies by informing long-term consequences and costs of an intervention. Furthermore, these models are normally established with an explicit examination of uncertainties by several techniques of validation [23]. The model chosen is determined by the nature of the decision problem, its time scope, the unit of analysis, and whether interaction between individuals and resources is required.[27]. This section represents the two most common types of models to give the ground for the reasons of employing the discrete event simulation model to answer the research question

3.2.1. Common decision-analytic models

The most common models for economic evaluation in health care are decision tree and state-transition model. The decision-tree is a straightforward and effective approach for decision-analytic modeling. From left to right, the strategic mutually exclusive and collectively exhaustive choices are presented as pathways in the decision-tree [23]. An individual must adhere to a single path and follow it to its conclusion. The path is determined by the nodes. The initial node, typically in the form of a square, displays alternative strategies. There are multiple potential outcomes for a patient's chosen strategy. Typically, these outcomes are presented by a circular node. Multiple outcomes are possible after the initial outcome. Thus, a circular node may be followed by multiple circular nodes. When there is no other possible outcome, a pathway terminates with a terminal node, which is typically depicted as a triangle. The sum of the probabilities of all possible outcomes at one circular node for a given strategy is 1. Then, costs and benefits are associated with each potential course of action. However, the decision tree does not handle transition time and is impractical when there are numerous alternatives with numerous outcomes [27].

The state-transition model is structured around mutually exclusive health states that represent transition of a disease or a treatment process. Because it reflects recurring or changing events over time, it overcomes the decision-tree model's time-memory deficiency [23]. The states, their associated costs and utility, the cycle length, and the transition probabilities are critical components of the state-transition model. State-transition models are classified according to

their unit of analysis, which can be either a cohort, referred to as a Markov model, or a single individual, referred to as microsimulation [27]. However, the state-transition has its own shortcomings, such as a lack of memory for previous events or an inability to deal with competing risks [26].

Along with the aforementioned disadvantages, neither the decision-tree nor the state-transition model allow for interaction between individuals within the model or between individuals and occupied resources. Thus, in the circumstance of modelling for two groups of patients and scarce resources, such as organ donation from deceased livers donors, there is a need to seek for a more flexible model, such as discrete event simulation (DES).

3.2.2. Discrete event simulation

DES is an individual-based model that is highly recommended when the decision problem is characterized by resource constraints such as in case of organ transplantation. In this model, participants are referred to as entities, and they have attributes such as age, blood type, or disease type. They will experience different events in the model at discrete time intervals. The model track and record progression of the patients. Additionally, over time, the patients' attributes, or heterogeneous characteristics, could be updated [26]. Due to the fact that participants will occupy or consume limited resources along the way, they may be required to queue and wait for resources until they become available. This is a significant feature of DES because it enables the impacts and potential health losses associated with queueing. If a resource reaches its maximum capacity (for example, a surgery room) or is not yet available (for example, a liver for transplantation), the entities must form a queue and wait for the resource to become available [26]. Additionally, DES is one of the most flexible decision analytic models available because it can enable an event, as well as the time to that event, to be dependent on either or both an entity's attribute and the previous event. The simulation time can also be easily assigned in the DES.

3.2.3. Queuing theory

The fundamental premise of queueing theory is that resources are normally scarce and constrained. When a resource is constrained by a first-arriving entity, such as a surgery room, or is not yet available, such as a graft not yet ready for transplantation, those entities must enter a queue to wait for the resource to become available. The queue serves as a waiting area, and the waiting time is typically measured from the time the entity enters the queue to the time the resource serves the entity. In health care, the rule of "first in, first out" or "first come, first

served" is frequently followed, except when other rules are specified, such as an entity with a higher priority or an urgent need can bypass the queue and be served with the next available resource. The recorded waiting time can be used as an final output of the model for decision-making or as an input for other calculation [26].

Typically, the queue is modelled using a Poisson point process with three fundamental assumptions: 1) Patients enter the queue independently of one another; 2) there are no concurrent arrivals; and 3) the average time interval between arrivals is fixed and known [28].

While modeling the queue, it's worth noting how long it takes for the simulated queue to become stable. The situation stems from the fact that during the model's early years, the queue was insufficiently long, and thus the waiting time did not accurately reflect the situation. This period is called burn-in period. The model must be run several times to determine the point at which the waiting time becomes stable. From that time, the analytic observation period can begin [29].

3.2.4. Survival analysis

By and large, survival analysis is a collection of statistical procedures aimed at defining the 'time until an event occurs' [30] such as time to death, and time to transplantation, for example. The distinguishing feature of survival analysis is that it enables extrapolation of frequently censored data from clinical trials in the medical field [27]. That feature leads to a common goal of survival analysis is to estimate either survival or hazard function from the survival data. The survival function *S(t)* gives the probability that an individual will survive, or not fall, longer than a given time (t) [30]. In contrast, the hazard function *h(t)* focuses on falling probability by giving the potential hazard at time (t) given that the patient has lived up to that time.

The hazard function is commonly estimated by the Cox Proportional hazard function. The reason is that, as a robust model, the result from the Cox regression could close approximate the result for the parametric model. The Kaplan Meier method can be used to estimate the survival function. The method necessitates the collection of data on the subject throughout the observation period, as well as their status at the conclusion, which can be either death (event happened) or censored. Censorship implies that the model's participant's survival time cannot be determined precisely. It occurs when the observed survival time is less than the true survival time (left-censored) or greater than the true survival time (right-censored) [26]. For instance, right censored, which occurs frequently, refers to the situation in which the event does not happen before the observation period expires.

Extrapolation is a useful technique for dealing with censoring because it estimates the duration of complete survival that was not observed during the monitoring period. Several parametric survival models are used to aid in extrapolation, including Weibull, Gompertz, Log normal, Log logistic, and Exponential. Different models, which may produce different estimated survival times, making the selection of the model critical. The choice could be made based on the shape of the hazard function, the inspection of parametric model's the visual fit with the Kaplan-Meier curve, Akaike and Bayesian Information Criterion (AIC & BIC), and also the clinical relevant beyond the observed period [31]. Different models also give different shape of the hazard function, while Weibull and Gomperzt gives increasing or decreasing shape and Exponential gives constant hazard, Log normal and Log logistic has inverted U-shape hazard functions [30].

3.2.5. Common random values in DES

The term "common random values" refers to the fact that simulation in DES involves simulating a group of patients with a common set of attributes which is independent of intervention. Consequently, common random values are typically employed by the participant generator module. When a group of patients enters the model, they will be randomly assigned a set of attributes that will not change throughout their journey. In addition, if this group of patients is to be simulated across various interventions, the set of "common random values" could remain constant across simulations of every strategy [26].

3.2.6. Time-to-event in DES

After the best-fit model for the survival has been selected, the time-to-event could be randomly derived using commonly implemented standard distribution from a quantile functions formula [26]. The following are some frequently used distributions and their associated quantile function formulas to get the "time-to-event" [26]:

Table 1: Quantile Function Formulas for Deriving Event Times Using Commonly Implemented Standard Distributions (Caro et al., 2015)

Distribution	**Formula**
Exponential	$-\ln(random\ number)/\lambda$
Weibull	$[-\ln(random\ number)/\lambda]^{1/\gamma}$
Gompertz	$\{\ln[-\gamma \ln(random\ number)] - \lambda\}/\gamma$
Log-logistic	$\{\ln[-\gamma \ln(random\ number) + 1] - \lambda\}/\gamma$
Log-normal	$e^{\left[\frac{NomInv(random\ number)}{\gamma}\right]+\lambda}$

In which, the random number is randomly assigned a value between 0 and 1, λ is the hazard rate, or the scale parameter of the distribution and γ denotes the shape parameter. The time unit in the formula will be equal to the time unit used to calculate λ [26].

3.3. Uncertainty, sensitivity analysis and transparency

3.3.1. Uncertainty

There are various kinds of uncertainties that can lead to imprecision in economic evaluation and modeling. The four primary sources of uncertainty are stochastic (first order), parameters, heterogeneity, and structure. Stochastic refers to the frequent occurrence of uncertain outcomes of identical patients, especially in DES. The time-to-death of DES patients, for example, is derived from a quantile function with a random seed. Consequently, different patients would have varying times to death, which would contribute to the variance in the group's mean. This outcome variation is stochastic [26]. In addition, the variability between patients could also be a source of uncertainty, namely heterogeneity uncertainty. Parameter uncertainty refers to the estimation of input parameters in the model. Another source of uncertainty is the assumptions made through the conceptualization of a decision model, namely structural uncertainty.

3.3.2. Sensitivity analysis

A decision-analytic model should incorporate sensitivity analysis, a systematic examination of the uncertainties surrounding the primary outcome. A sensitivity analysis could be probabilistic or deterministic.[27]. Analyzing the quantitative relationship between changes in input parameter(s) and expected output, such as ICER, is the purpose of deterministic sensitivity analysis. A deterministic sensitivity analysis may be either one-way or multi-way, but is typically two-way. While one-way sensitivity analysis assesses the effect of changing one

parameter on the final outcome, two-way sensitivity analysis allows for the simultaneous modification of two parameters.[23]. On a larger scale, probability sensitivity analysis permits the assessment of a joint uncertainty across all parameters. Several samples of each parameter are taken using Monte Carlo simulation to generate the joint uncertainty, given the plausible standard error range and distribution type for each parameter [23].

3.3.3. Transparency

In numerous decision-making contexts involving health economic evaluation, model transparency is essential for gaining trust and reliability. Transparency means that the model should be replicable [32]. The model must be sufficiently explicable to non-technical audiences and should be accompanied by a technical appendix that allows the model to be replicated. Eliminating unnecessary states or occurrences would also increase transparency. With DES, the model's expression in plain language, with well-structured flowcharts, with fewer abbreviations and strange syntax, as well as the model's code made available, can aid in its comprehension [26].

3.3.4. Validation

Validation enables the modeler to demonstrate the accuracy of their model and to self-reflect on the impact of the model's assumptions. Validation as the procedure could be done with three following steps [26]: 1) Face validity, 2) internal validity, and 3) external validity. Face validity refers to the sound decision of model choice and its assumption for a specific problem. This assessment needs to be justified by clinical evidence and experience. One way to get face validity is to get support of multiple clinicians regarding choice of model design and inputs [26]. The following step is internal validation, which refers to the model's validity of mathematical calculation. This can be accomplished by conducting systematic checks on the model's output to determine whether it meets the expectation and provides a reliable logic. Finally, but certainly not least, is determining whether the output of the model is consistent with the real data provided by the real world. External validation is classified into three categories: dependent validation, independent validation, and predictive validation which respectively increases the independent level of real-world data benchmarking. In contrast to dependent validation, which compares model outputs to the observed data used to create the model, independent validation uses independent observed data as the comparator. Predictive validation is the most rigorous standard because it employs unpublished or ongoing independent clinical trials as comparator of the model's outputs [26].

4. Methods and Data

4.1. Analytic overview

This section describes the general analytic approach used to answer the research question. The information regarding the three alternatives, the population, choice of model, expected outcome and perspective will be given

4.1.1. The three strategies and population

Three strategies were developed around the NCRLM patients' alternatives, as summarized in Table 2. In this thesis, the criteria of SECA-II only will be applied for selection of NCRLM patients as potential candidates for liver transplantation. In the first strategy, patients with NCRLM receive only palliative chemotherapy, which is their current treatment alternative, whereas in strategies two and three, they are offered liver transplantation from deceased or living donors, respectively. The current wait-listed patients for deceased liver are referred to as Status Quo Patients. Among those three strategies, Strategy I reflects the situation of today that the NCRLM patients go on palliative chemotherapy only (see section 2.1.), only Status Quo Patient groups are included in the list of waiting for deceased liver, and the donors are living their normal lives without any liver resection. As reported in section 2.2.1., strategy II allows that NCRLM patients who meet SECA II selection criteria could also receive liver transplantation and be included on the Norwegian waiting list. Also, in strategy II, the live donors are not affected. Strategy III is motivated by current surgical advances (see section 2.2.2.). In this strategy, NCRLM patients are offered live donor liver transplantation which impose some cost and effects on health of the donors while the Status Quo patient group is not affected. Thus, there is no spillover effect from strategies 1 and 3 to Status Quo Patients. Similarly, only strategy 3 accounts for the effect on live liver donors. Hereafter the donor is also referred to as a group of patients.

Table 2: Three strategies and the populations affected by choice of strategy

	Strategy I	**Strategy II**	**Strategy III**
NCRLM Patients	Palliative care only ***(1)***	Deceased Liver Transplantation ***(3)***	Live Donor Liver Transplantation ***(4)***
Status Quo Patients	Deceased Liver Transplantation (Not affected by NCRLM patients) ***(2)***	Deceased Liver Transplantation (Include NCRLM patients) ***(3)***	Deceased Liver Transplantation (Not affected by NCRLM patients) ***(2)***
Live Liver Donors	Normal lives without donation ***(5)***	Normal lives without donation ***(5)***	Donation and care of liver ***(6)***

**(1), (2), (3), (4), (5), (6) are numbered for required simulation.*

4.1.2. Choice of model

As numbered in Table 2, seven simulations would need to be conducted for three groups of patients. Since there is no interaction among groups of patients in strategy I, groups (1), (2), and (5) in Table 2 are modelled independently. In strategy II, according to Vinter and Sjule (2021), the inclusion of NCRLM patients in the deceased liver waiting list come at cost of reduced total life year for Status Quo Patient. As a result, when modelling, strategy II needs to account for the spillover effect of the NCRLM's offering on the Status Quo patient. Therefore, the two group of patients in (3) are modeled together with queuing theory while (5) are model separately. In strategy III, the outcome of (2) could be applied for Status Quo Patients while there would be a significant effect on the donors as in (6) when they donate part of their liver. Indeed, both (5) and (6) required input from (4) for the fact that we simulate only a group of donors across three strategies and donor arrival depends on number of transplantation surgery simulated in (4). Details explanation follows in the description of the population at section 4.1.3. and the simulation model in section 4.2.

In order to account for the spillover effects which could be explained by queuing theory as reported in section 3.2.3., a DES would be chosen to answer the research question. The DES allows for interaction among patient groups with the queuing effect, precise timing of event, as well as the calculation of individual cost and effect based on their own trajectory. The main functions of DES have been reported in section 3.2.2.

4.1.3. The population

NCRLM patients

In each simulation, across three strategies, there was only one group of NCRLM patients with similar attributes that had been simulated. These attributes include median age, time of arrival, blood distribution, and the probabilities of dying either of NCRLM in palliative care or background mortality. In details, for simulation (1), (3), (4), I simulated the same group of NCRLM patients but different interventions, leading to different trajectories and outcomes. In this paper, NCRLM patients are selected based on SECA II study criteria (see section 2.2.1.). This selection gives roughly 2 patient per year in Norway, with interarrival between two patients of 180.5 days [33]. The specification of NCRLM's patients attribute would be reported as model inputs in each simulation. For the group of patients who meet SECA II criteria, the used median age was 59 based on expert opinions and SECA II trial.

Status Quo Patients

Currently, there are 12 patient types that are eligible for liver transplantation in Norway. Following Sjule and Vinter [9], this paper also accounted for the 5 most frequent diseases and group the rest as "Others". The five most frequent disease, together with their proportion in Norway and patient group's median age were note in Table 3, which was inherited from Sjule and Vinter, sourced in Nordic Liver Transplant Registry annual report 2019 [9, 15]. The interarrival time between of Status Quo Patients was about 3.55 days which gives roughly 101 patients per year [15].

Table 3: Sub-groups on most frequent liver disease for liver transplantation in Norway [8,14].

Liver Diseases	**Proportion**	**Median age**
Hepatocellular Carcinoma (HCC)	13.7 %	61
Primary Biliary Cirrhosis (PBC)	6.7 %	57
Primary Sclerosing Cholangitis (PSC)	18 %	43
Acute Liver Failure (ALF)	8.3 %	43
Alcoholic Cirrhosis (AC)	10.1 %	57
Others	43.2 %	57

The Live Donors

An consultation with Dr. Line and Dr. Dueland, the organ transplantation and oncology experts at Oslo University Hospital, was taken during the thesis process to gather opinion on how the donors could be modeled [34]. Accordingly, the number of transplantations taken in Strategy III was assumed to be the same as the case of Strategy II (n=2) to get a common ground for

effectiveness between strategy II and strategy III. In practice, if a patient could not get a chance to receive the graft from a live donor, he would probably get listed to wait for deceased liver. Thus, whenever a transplantation is taken, we have a donor arrival. In other words, the donor's arrival depends on simulation (4) in Strategy III as the time of the transplantation. Also, the attributes of donors are mainly taken by assumption as there was no evidence currently. The same group of donors, referred also as a group of patients, are modeled across the three strategies with simulation (6) of living their normal lives without donations and simulation (7) as in the case of donating part of their liver (see section 2.2.2).

4.1.4. Spillover effect

The definition of cost-effectiveness in section 3.1.4. only applies to the specified population. In other words, the conclusion regarding cost-effectiveness only applies to the patient population for which incremental costs and benefits were calculated. Assuming that a life year or QALY is treated equally across the entire population, it is possible to calculate the costs and effects for the entire population as the sum of the costs and effects for each participating group in the population. Therefore, it could be the case that deceased liver transplantation (strategy II) was cost-effective for the NCRLM group, with an ICER less than the WTP threshold when compared to chemotherapy alone (strategy I). When the entire population with the Status Quo groups are considered, the cost-effectiveness of deceased liver transplantation may change. The incremental effect for the entire population could even be negative if the intervention causes Status Quo Patients to lose an adequate amount of QALY. Under assumption that there was no spillover effect on the incremental cost of the Status Quo Patient, the incremental cost of both groups was solely determined by the incremental cost of NCRLM patients. Consequently, the new treatment may be cost-effective when only NCRLM patients are considered, but it was no longer cost-effective when the entire population was considered due to a negative ICER. This is an example of the primary consequence when the spillover effect is taken into account, as when there are multiple interacting patient groups in a population.

4.2. Generator of model's participants

Table 2 lists the seven patient trajectories that were simulated by the DES model for three distinct treatment strategies. This section describes participant generator, the first component of the model. The module generates distinct patient groups and deceased livers. When participants arrive at the model, their attributes are set as common random values. After the

population's ground is described in this section, in section 4.3, the model's schematic is presented.

4.2.1. Patients group generators

When the model simulates a group of patients, they are randomly assigned a set of attributes that was alternated between simulations. Thus, this collection of attributes is referred to as common random values because they remain constant across three simulations but change the next time a group of patients is generated. These frequently used values include blood distribution, background mortality (which varies according to age), and type of liver disease (for a group of Status Quo patients).

Patients' blood type distribution

The blood type distribution was determined using data from Norway's general population, which included 40 percent of those with blood type O, 49 percent of those with blood type A, 8 percent of those with blood type B, and 4 percent of those with blood type AB [35]. This was true for all patients included in the model, regardless of their patient's group. Thus, when a patient entered the model, they were randomly assigned a blood type based on the Norwegian blood distribution.

Age and background mortality

To simplify the model, each patient group was assigned a representative age. Background mortality was derived from survival analyses based on the age of the patient group.

The background mortality rate was used to calculate the time to death in the case that a transplant recipient can survive more than ten years after surgery based on their disease-specific post-transplantation survival. In other words, post-transplant mortality was only applied within ten years after surgery. Age-dependent background mortality was generated from survival analysis using a life Table that depicts the probability of dying at each age in Norway [36].

Patients' arrival

As described in Section 4.1.3, the average time interval between the arrivals of two NCRLM patients was 180.5 days , whereas the parameter for the arrivals of two Status Quo patients was 3.55 days [15]. The time interval between two patient arrivals follows an exponential distribution with a constant mean but varies after each event [28]. For the live donors, they arrive according to the assumed rate of transplantation which was dependent of average waiting

time of NCRLM patients in that scenario, according to expert opinion, as described in section 4.1.3. [14].

4.2.2. Deceased liver generators and matching principle

The time interval between the arrival of two livers in Norway was 3.18 days and follows an exponential distribution, as Sjule and Vinter extracted from Scandiatransplant data from 2015 to 2021 [18]. Additionally, the blood type of the liver was discretely distributed and randomly assigned upon arrival as follows: 44 percent of blood type O, 46.5 percent of blood type A, 7.5 percent of blood type B, and 2 percent of blood type AB [18, 28, 37]. The matching principle was based on the theory of queueing and the ABO blood group system.

Table 4: Parameters for common random values

Parameter [Source]	**Value**	**Distribution**
Interarrival time between NCRLM patients (in days) [9, 33]	180.5	Expo
Interarrival time between Status Quo patients (in days) [9, 18]	3.55	Expo
Patient blood type distribution [35]		
A	49%	Disc
B	8%	Disc
AB	4%	Disc
O	40%	Disc
Background mortality (Gamma, Lambda) [9, 36]		
Age 43	0.1143273 0.0017307	Gompertz
Age 57	0.1156375 0.009412	Gompertz
Age 59	0.1235331 0.0111042	Gompertz
Age 61	0.1267308 0.0133527	Gompertz
Interarrival time between liver (in days) [9, 18]	3.18	Expo
Liver blood type distribution [9, 18]		
A	46.5%	Disc
B	7.5%	Disc
AB	2%	Disc
O	44%	Disc

4.3. Simulation of patients' trajectories

After the patients were generated with the same random attributes, they were simulated to enter the three different strategies' trajectories. In this section, each trajectory, input parameters, outcome, and data analysis are described.

Patients were captured in ten year of arrival and follow until the end with a time horizon of up to 70 years to make sure that all death occur.

4.3.1. Strategy I – Simulation (1) - NCRLM patients in Palliative care only

4.3.1.1. Description and parameter inputs

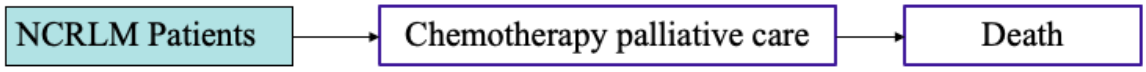

Figure 1: Strategy I - Simulation 1 - NCRLM patients in palliative care only

The first simulation in strategy I represents the current standard of care for NCRLM patients receiving only palliative care. Patients were admitted to the model and receive palliative care until death. The time-to-event of death was taken from a survival analysis conducted by Sjule and Vinter using data from the Nordic VII trial which investigated the overall survival of CRLM patients under chemotherapy only [33, 37, 38]. In simulation (1), the time-to-death of NCRLM patients follows a Gompertz distribution, resulting in an average survival time of 2.3 years per patient. Specific parameters are included in Table 5.

4.3.1.2. Data analysis and outcomes

In this trajectory of NCRLM patients with palliative care only, the time the patient enters the model, and the time of death were recorded to estimate the cost and health effects. This period was then utilized to calculate the cost and utility. From the 2019 paper by Bjørnelv et al., the costs, which included chemotherapy drugs, administrative expenses, hospitalization, and best supportive care were extracted [8]. Three months prior to a patient's death, the best supportive care was provided. Using the same HRQoL as used in Bjørnelv et al.'s paper [39, 40], the utility of life during this period was quantified. Details about costs and effects was also included in Table 5.

Table 5: Parameter for NCRLM patietns under palliative care only

Parameter [Source]	Value	Distribution /Unit
Survival time of NCRLM patients under palliative care (Gamma, Lambda) [9, 33]	0.2834202 0.2400557	Gompertz
HRQoL NCRLM patients under palliative care [39, 40]	0.82	
Cost palliative chemotherapy month 0 – 6 [39, 40]	381,798.6	NOK/year
Cost palliative chemotherapy month 7 – 12 [39, 40]	86,290.9	NOK/year
Cost palliative chemotherapy month 13 onwards [39, 40]	128,578.1	NOK/year
Cost best supportive care [39, 40]	934,881.2	NOK/year

4.3.2. Strategy I & III – Simulation (2) – Only Status Quo patients in the waiting list for deceased liver

4.3.2.1. Description and parameter inputs

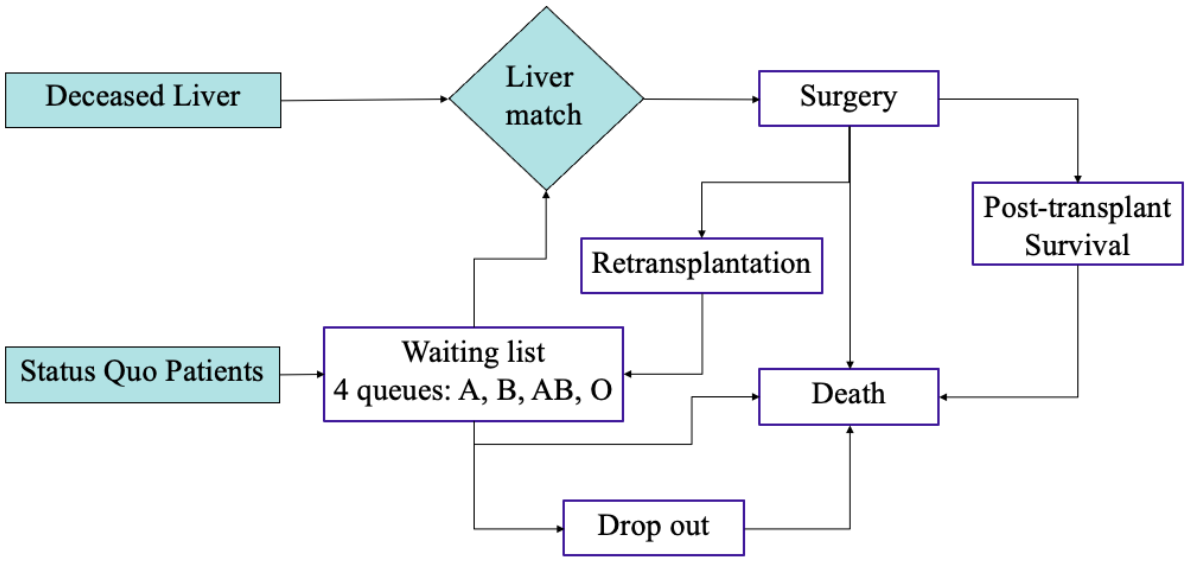

Figure 2: Strategy I & III - Simulation (2) - Only Status Quo in the waiting list for deceased liver

Figure 2 presents the trajectory of Status Quo Patients in the case that there are only them waiting for deceased liver. The events in the trajectory are as follow:

Status Quo Patient Arrival

They first arrived in the model, assigned with attributes of age, background mortality, blood type, disease type as described in section 4.1.3. and 4.2.1.

Liver arrival

The liver arrives in accordance with the liver generator module as described in section 4.2.2 using the *rexp* function in R to randomly determine the time between each arrival, but the average time was 3.18 days.

Priority

Prior to entering the queue, up to 10 percent of patients were given priority to be matched with a suitable liver as soon as one becomes available, per expert opinion [29]. In addition, a patient with a first transplantation failure who requires a second transplant will be returned to the queue and given priority. Those with a priority badge will be at the front of the line regardless of their arrival time.

Dropping out and death while waiting

When patients enter the queue, it was assumed that they would survive for a certain amount of time. In the absence of data, it was assumed that the survival of Status Quo Patients prior to transplantation was comparable to that of NCRLM patients receiving palliative care. This interval was adjusted to approximately 2,3 years and follows the Gompertz distribution. Therefore, if the prior transplant survival time was shorter than the liver transplant waiting time, the patient dies while waiting.

As assumed and rationalized by Sjule and Vinter, the drop-out rate was set at 6 percent, which means that approximately 6 percent of arrived patients would drop out and continue to live with prior-transplant survival until they die. According to expert opinion, patients withdraw because they no longer meet the requirements for liver transplantation [9].

Queuing and liver matching

As a matter of programming, queues are created in the R program for each of the liver blood types A, B, AB, and O. Based on ABO system, patients could be listed for more than a queue upon arrival. A patient with blood type A, for instance, could wait in queues for both liver types A and O, whereas a patient with blood type O could only wait for liver type O. When a liver arrives, the system sends a signal to the suitable waiting queue(s) and releases the first patients from the queue to undergo surgery. The queue operates according to the "first in, first out" principle, except for patients with assigned priority. When a patient leaves a queue, their name was also withdrawn from other queue that they are waiting, and other patients move forward

to wait for the next liver. If liver arrives but there was no recipient waiting, the liver just disappeared from the model and assumed to be exported to the other Nordic countries.

Surgery, failure, and re-transplantation

When undergoing transplantation surgery, there was a 2 percent chance of surgical failure leading directly to the death of patient [8, 16, 41]. Similarly, it was estimated that 13.4 percent of transplant recipients require a second transplantation after the first attempt failed. According to expert opinion, re-transplantation, if performed, would occur within 90 days of the initial surgery. Thus, a patient would face three possible outcomes during surgery: survival, death, or listing for re-transplantation. A patient would have only one opportunity for re-transplantation, meaning that there were only two possible outcomes for the second transplantation surgery: death or survival.

Once listed for re-transplantation, the patient was rolled back into the queue, marked with priority, and given a survival time such that approximately 74.9 percent of those who need it will receive a second transplantation, but no longer than 90 days [16]. Since queues are modeled in a complex manner, it was in fact difficult to obtain 74.9 percent with precision but approximately. However, this was the most accurate method for estimating the impact of re-transplantation on the waiting list. Regarding the queuing principle, R models time with a precision of one hundredth of a second. Thus, with a sample size of no more than a few thousand patients, the probability that a rolled-back patient for re-transplantation would enter the same queue at the same time as a first-time prioritized patient was extremely low. Therefore, a separate algorithm was not required to model this scenario.

Post-transplant survival

If a transplanted patient survives surgery, he was presumed to have a disease-specific post-transplant survival time of 10 years or less. If he survived beyond 10 years, he would be subject to the age-dependent Norwegian background mortality for the remainder of his life. In their paper, Sjule and Vinter included the parameters for survival time of six liver disease groups and the background mortality rate from their survival analyses [9] with data extracted from Fosby et al.[16] and Melum [15]. These parameters and their distribution were used to extract the time to death of post-transplant survival using the corresponding quantile function in Table 1. The parameters are summarized in Table 6.

4.3.2.2. Data analysis and outcomes

To track the timeline of Status Quo patients in the model, several model time points were recorded: the time of entering the model, the time of surgery, the time of second transplantation (if applicable), and the time of death. Thus, the waiting time, years of life before surgery, and years of life after surgery were easily extracted. Within the scope of a master's thesis project and due to a lack of resources, the costs and utility of Status Quo Patients were simplified to a yearly basis for time before and after surgery, based on the assumption that this will make them comparable to the NCRLM patients. See Table 6 for details.

Table 6: Parameter inputs for simulation (2)- Only Status Quo patient in the waiting list

Parameter [Source]	**Value**	**Distribution /Unit**
Survival time of Status Quo patients before transplantation (Gamma, Lambda) [estimated]	1.9~2.2 0.04~0.06	Gompertz
Survival time of Status Quo patients after transplantation (Gamma, Lambda) [9, 15, 16]		
HCC	0.838474089 0.119512298	Weibull
PBC	0.780242352 0.072519412	Weibull
PSC	0.928784349 0.039010673	Weibull
AC	1.129434878 0.038154942	Weibull
ALF	0.449819045 0.171966601	Weibull
Others [estimated]	0.449819045 0.171966601	Weibull
Dropout rate [9]	6%	Disc
Priority rate [29]	10%	Disc
Death in surgery [8, 16, 41]	2%	Disc
Re-transplantation rate [8, 16, 41]	13.4%	Disc
Probability of getting a retransplantation [16]	74.9%	Disc
HRQoL Status Quo Patients prior transplant[estimated]	0.4	
HRQoL Status Quo Patients post-transplant [estimated]	0.82	
Cost of care Status Quo Patients prior transplant [estimated]	128578.1	NOK/Year
Cost of care Status Quo Patients post-transplant [estimated]	179657.6	NOK/Year
Cost of first time transplantation [39, 40]	1,284,448.9	NOK
Cost of retransplantation [39, 40]	278,302.2	NOK

4.3.3. Strategy II – Simulation (3) – Both Status Quo and NCRLM patients in the waiting list

4.3.3.1. Description and parameter inputs

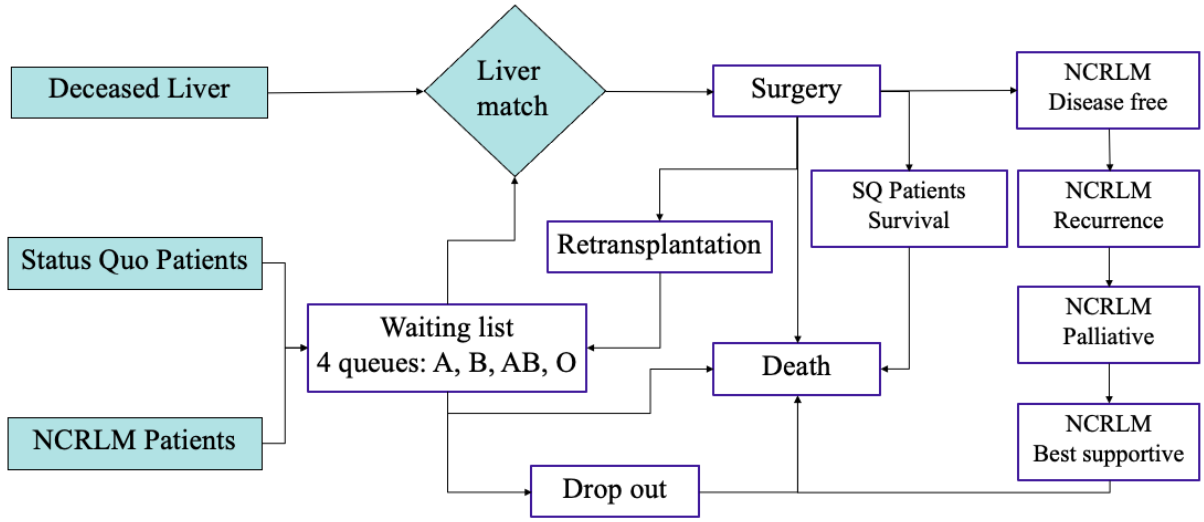

Figure 3: Strategy II - Simulation (3) -Both Status Quo and NCRLM patients are included in the waiting list

Status Quo Patients and Deceased Liver

As described in section 4.3.2, the trajectory and input parameters of Status Quo patients, as well as liver arrival, were identical to simulation (2) of Strategy I & III. In this simulation, the only difference was the addition of NCRLM patients, which might have a spillover effect on the Status Quo patients. All the parameters were used from Table 6.

NCRLM Patients – trajectory and parameter inputs

The trajectory of NCRLM patients in this simulation is essentially identical to that of Status Quo patients, with the exception of four stages in the post-transplant survival period [8]. These four stages consist of disease-free, cancer recurrence, palliative care, and the best supportive care period prior to death. Similar to Status Quo patients, if NCRLM patients survive 10 years after transplantation, they die at the same background rate as Norwegians of the same age. The post-transplant time-to-death parameters of NCRLM patients with SECA II criteria were provided by Sjule and Vinter [37] with primary data from Dueland et al. [6].

The arrival of NCRLM corresponds to simulations (1). If they do not drop out, they were added to the waiting list alongside Status Quo patients. Those who drops out, approximately 10 percent of the listed patients, will continue their lives with palliative care. There are a number of reasons for dropping out, including developing metastasis on a side other than the liver or

no longer meeting the surgical requirements [9]. Due to the severity of the disease, 10 percent of NCRLM patients was given priority on the waiting list, according to the opinion of experts [29]. Patients with NCRLM would also pass away while awaiting a graft if their survival time was shorter than the graft's waiting period. At surgery, NCRLM patients has the same probabilities of re-transplantation or death as Status Quo patients [16].

4.3.3.2. Data analysis and outcomes

After the simulation, the data for Status Quo patients and NCRLM patients were separated. The Status Quo Patients data set was handled similarly to simulation (2) (section 4.3.2.2.).

Due to resource limitations, after consulting with an expert [29], it was assumed that one-third of the post-transplant survival period for NCRLM patients would be disease-free, and a similar amount of time would be allocated for palliative care. Extracted from the two papers of Bjørnelv et al., the costs scheme for NCRLM patients was broken down on a quarterly basis which was quite specific [8, 42]. For instance, the cost of first-year recurrence was high due to the possibility of multiple resection surgeries, whereas the cost of subsequent years was primarily comprised of follow-up hospitalization and drugs. Using the EU inflation rate for the corresponding period and the EUR–NOK exchange rate for 2022, the price was converted to Norwegian krone. The utility and HRQoL were extracted from the same source. See Table 7 for details.

Table 7: Parameter for simulation (3) - NCRLM receiving deceased liver transplantation

Parameter [Source]	**Value**	**Distribution /Unit**
Survival time of NCRLM patients prior transplantation (Gamma, Lambda) [9, 33]	0.2834202 0.2400557	Gompertz
Survival time of NCRLM patients post-transplant (Gamma, Lambda) [9, 33] [6]	0.0478487 0.106535	Gompertz
Dropout rate [9]	10%	Disc
Priority rate [29]	10%	Disc
Death in surgery [8, 16, 41]	2%	Disc
Re-transplantation rate [8, 16, 41]	13.4%	Disc
Probability of getting a retransplantation [16]	74.9%	Disc
HRQoL NCRLM Patients prior transplant [39, 40]	0.82	
HRQoL NCRLM Patients post-transplant disease-free month 1-3 [39, 40]	0.64	
HRQoL NCRLM Patients post-transplant disease-free month 3 onwards [39, 40]	0.71	
HRQoL NCRLM Patients post-transplant recurrence 1st year [39, 40]	0.82	
HRQoL NCRLM Patients post-transplant recurrence 2nd year onwards [39, 40]	0.85	
HRQoL NCRLM Patients post-transplant palliative care [39, 40]	0.82	
Cost of care NCRLM Patients prior transplant [39, 40]	381798.6.1	NOK/Year
Cost of first time transplantation [39, 40]	1,284,448.9	NOK
Cost of retransplantation [39, 40]	278,302.2	NOK
Cost of care post survival disease free month 1-3 [39, 40]	279,553.6	NOK/Year
Cost of care post survival disease free month 4 onward [39, 40]	242,796.1	NOK/Year
Cost of care post survival recurrence month 1-12 [39, 40]	308,882.6	NOK/Year
Cost of care post survival recurrence month 12 onwards [39, 40]	167,973.5	NOK/Year
Cost of care post survival palliative month 0-6 [39, 40]	381,798.6	NOK/Year
Cost of care post survival palliative month 7-12 [39, 40]	86,290.9	NOK/Year
Cost of care post survival palliative month 12 onwards [39, 40]	128,578.1	NOK/Year

4.3.4. Strategy III – Simulation (4) – NCRLM patients to receive liver transplantation from live donors

4.3.4.1. Description and parameter inputs

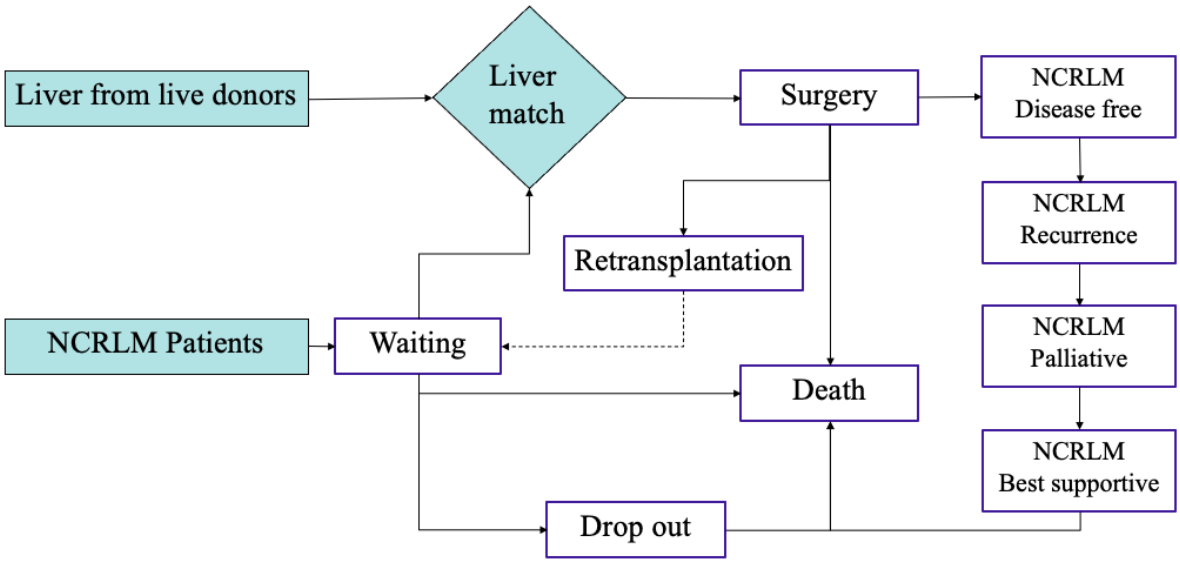

Figure 4: Strategy III - Simulation (4) - NCRLM patients to receive liver from live donors

The case of receiving liver from living donors was simulated based on expert opinions [34] due to the lack of actual data. The trajectory of NCRLM patients in this instance was comparable to their trajectory in the case of transplantation from a deceased donor, except that there was no queue for the graft. Since there was no queue, no priority was given.

According to oncology specialists, NCRLM patients enter the model, wait for an average of two to three weeks, and if they do not drop out or die, they may undergo surgery [34]. The dropout rates were identical to simulation (3). Nonetheless, because the waiting period was shorter than the current waiting period in Norway, which has a median of 40 days [18], a lower death rate while waiting could be anticipated. In fact, all other NCRLM-specific parameters were imitated from strategy II in order to assume that the efficacy of liver transplantation from a living donor was identical to that from a deceased donor. Therefore, the post-transplant survival was just the same as simulation (3).

In the case of re-transplantation, experts assert that patients should be placed on a waiting list for deceased livers since it was not likely they will get a re-transplantation from another living donors. At a rate of two patients per year and 13.4 percent of patients undergoing re-transplantation, only 0.26 NCRLM patients would be placed on the waiting list each year, or 2.6 patients in a time span of 10 years. In the scenario where Norway has a surplus of

approximately 12 deceased livers per year and participates in the Scandia Transplant - an organ facilitating program [18], an additional 0.26 patients per year was expected to have a minimal effect on the waiting list. Due to this, the queue of Status Quo patients in strategy III was maintained at the same level as in simulation (2), and additional NCRLM patients included for the purpose of re-transplantation had no effect on the queue. In this scenario, the time, success rate, and cost of re-transplantation were recorded to fully account for cost and effectiveness of the NCRLM patient population only.

4.3.4.2. Data analysis and outcomes

It was recorded when NCRLM patients enter the model, undergo surgery, receive re-transplantation (if applicable), and die. Prior- and post-transplant costs and utilities for NCRLM patients were identical to strategy II, but transplantation was more expensive [8, 42]. Since the RAPID technique (see section 2.2.2) necessitates two transplantation surgeries and one liver harvesting surgery from a living donor, the cost of a living donor liver transplant consists of one transplantation surgery, as with a deceased donor, and two hepatic resection surgeries. Most of the parameters for this simulation was the same as reported in Table 7 except the waiting time and transplantation cost as can be seen in Table 8.

Table 8: Cost of transplantation for living donor liver transplantation

Parameter [Source]	**Value**	**Unit**
Cost of first time transplantation [39, 40]	1,553,360.9	NOK
Cost of retransplantation [39, 40]	263,545.2	NOK

4.3.5. Strategy I & II – Simulation (5) – Live donors in their normal lives

4.3.5.1. Description and parameter inputs

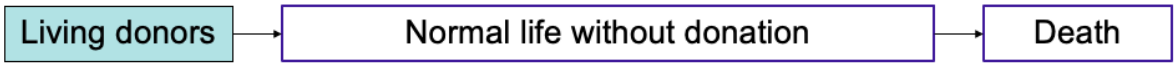

Figure 5: Strategy I & II - Simulation (5) - Living donors in the normal lives

As described in section 4.1.3, whenever simulation (4) results in a transplantation surgery, a donor arrives. Thus, the donor arrivals base on the timeframe for surgery in simulation (4) in accordance with arrival of NCRLM patients in the ten-year period. The donors were also monitored until their demise. For model simplification, all donors were given a median age of 43 and a mortality corresponding to that age.

4.3.5.2. Data analysis and outcomes

Using mortality of the median age, the life years of the donors were pulled and recorded. It was assumed that there was no cost for the donor in that case and the donors have a health life such that the total quality-adjusted life years were equal to the total life years.

4.3.6. Strategy III – Simulation (6) – Live donors in the case of donation

4.3.6.1. Description and parameter inputs

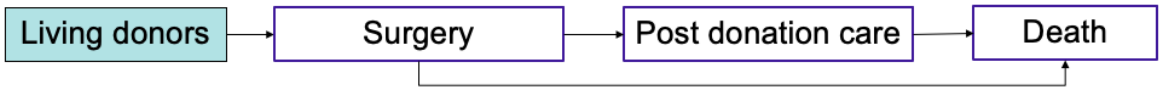

Figure 6: Strategy III - Simulation (6) - Donors in the case of liver donation

Right after arrival, donors would undergo an open hepatic resection to donate a portion of their liver, primarily the left lateral portion [34]. According to experts, there was a 0.02 percent chance that a donor will die during surgery, which was an extremely low risk [34]. However, if a simulated population was large enough, the number of life years lost due to surgical deaths can be observed. Otherwise, the donation has no effect on the donor's life expectancy, but there was a slight reduction in quality of life immediately after the donation.

4.3.6.2. Data analysis and outcomes

The cost of liver resection was included in the cost of transplantation for a patient with NCRLM. Donors were only associated with the cost of care. It was assumed that the cost of care will be greatest in the first year and then decrease over the subsequent nine years due to the low cost of follow-up. The QALY was assumed to decrease in the first year with HRQoL equal to 0.82, but from the second year on, the donor was assumed to live a normal, healthy life. This assumption was based on an initial finding that liver donation has no long-term effect on the quality of life of the donors [43], but it was reasonable to expect a reduction in quality of life within five weeks of surgery [44]. The quality of life should gradually return to normal after five weeks. All parameter inputs can be found in Table 9.

Table 9: Parameter inputs - simulation (6) - Donors in donation case

Parameter [Source]	**Value**	**Unit**
Probability of dying in surgery [34]	0.02%	
HRQoL donors first year after donation [estimated]	0.82	
HRQoL donors second year onwards [estimated]	1	
Cost of care first year after donation [estimated]	279,553.6	NOK/year
Cost of care year 2 to year 10 [estimated]	41,047.5	NOK/year

4.4. Analytic assumption

4.4.1. Simulation software

R was used to program the DES model, which is a powerful programming language that enables the effect of queuing in discrete event simulation models. However, except for an open package named ‘simmer' that enables discrete event simulation in R, there was no pre-built package or model for health economic evaluation using DES with queuing effect in R to my knowledge. As a result, the model was largely constructed from scratch, with some valuable input from Koffijberg et al. [45] regarding the patient's trajectory.

Some survival analyses to get the input of background mortality was also conducted using Excel and Stata.

4.4.2. Time horizon, burn-in period and discounting

In all simulations, a time horizon was selected to ensure that all incremental health effects and costs were accounted for, which often requires a lifetime and can last up to seventy years if the donor lives a normal life. However, the observed population include the ones who arrive within a timespan of ten-year programmatic period. As the model required a burn-in period in order for the queue and waiting time become stable, the model for strategy II needed to run several times to determine burn-in period. Figure 7 shows us the curve of average waiting time. Accordingly, ten-year programmatic period starts from beginning of year 7 until the end of year 16, model time, would be used to captured patients’ arrivals, for all three strategies. Hereby year 7 is called and used as model start, upon which the analytic time horizon begins. All the costs and effects in the model was discounted to the model starting time - the beginning of programmatic period. Discounting method was adherent to Norwegian guideline [25] using the formulars reported in section 3.1.3.

4.4.3. Willingness to pay threshold

In Norway, there is currently no indication of the willingness to pay threshold per life year. The WTP threshold per QALY is determined by illness severity. An important pillar for defining disease severity is absolute shortfall, or future life-year loss. The average age of patients with NCRLM was 59, with a two-year life expectancy after diagnosis [33]. Considering that the average life expectancy in Norway is 83 years [46], the NCRLM's expected life year loss is then 22 years. Therefore, the highest threshold per QALY in Norway

for diseases with an absolute shortfall of over 20 years, which is 850,000NOK per year [22], was applied to assess cost-effectiveness of three strategies in this paper.

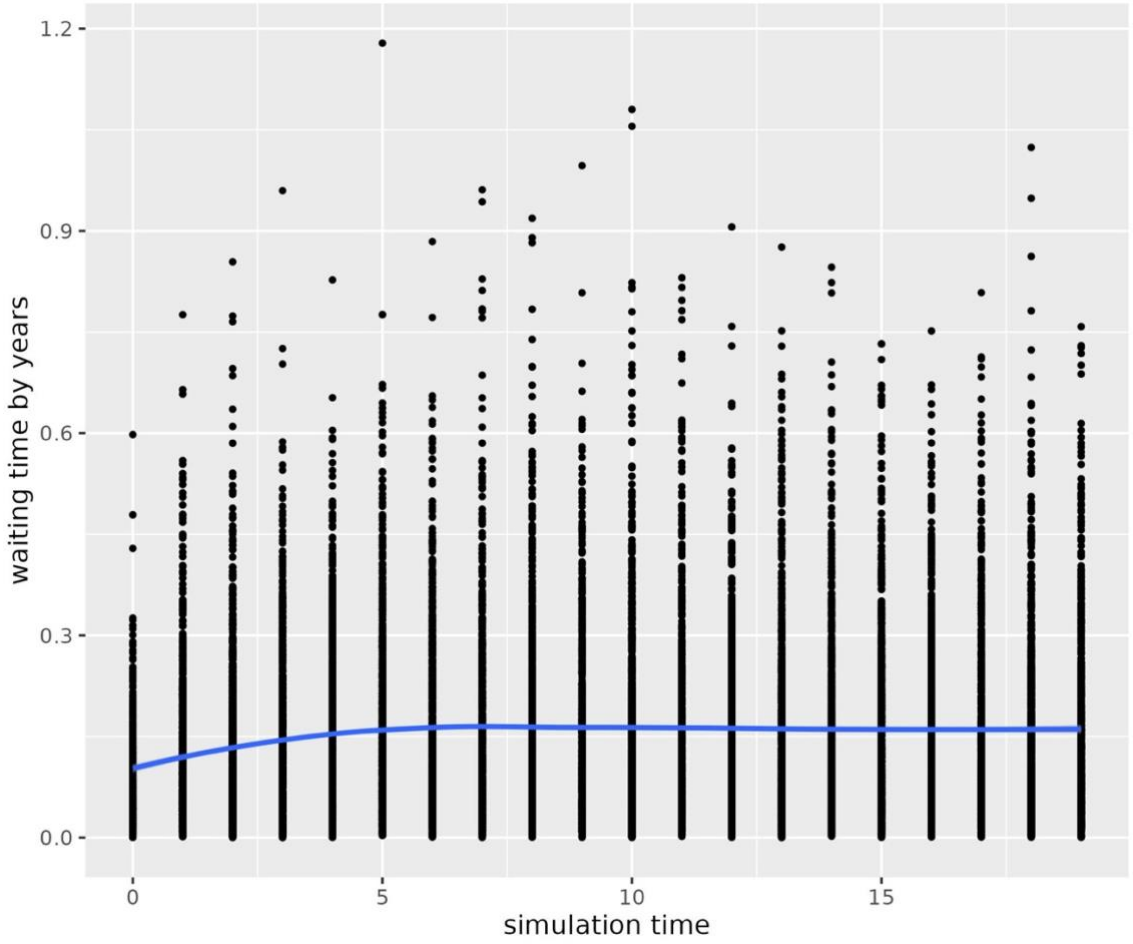

Figure 7: Average waiting time of Status Quo Patients in simulation (3) over 20 years of the model

4.4.4. Numbers of simulation

There were a significant number of randomly generated values in the model, including arrival time, blood or disease distributions, and, to a large extent, time-to-event values. This randomness may introduce stochastic uncertainty into the model. To ensure that the common random values were maintained across the three strategies, each simulation in the model was conducted using the same seed. In other words, in each simulation, the same group of patients, such as the NCRLM group, was simulated using three distinct strategies with identical arrival times and other shared attributes, such as blood distribution and background mortality survival time.

To reduce stochastic noise, the results of multiple simulations must be averaged to use as final model outcomes. The simulation (3) of strategy II, as the most complex simulation with queuing effect and the participation of NCRLM patients has been selected as the benchmark, with the outcome of total life year of Status Quo patients. 1500 iterations of the model were performed, and the average of Status Quo Patient's total life years over the number of iterations was used to create a curve. Stability was achieved when an additional model replication with the same input values, but distinct random number sequences, does not produce a statistically significant difference in the final average result. Figure 8 demonstrates that after 1000 simulations, the average total life year in simulation (3) became stable with minimal variation. Therefore, each strategy was simulated with 1000 replications.

The outcome of costs, QALY and LY were also discounted in each simulation and recorded. According to Norwegian guidelines for HTA, a discount rate of 4 percent was applied [25]. The discounting formulas was used as prescribed in section 3.1.4. The final outcomes were the average values of 1000 simulations. This also applies to the discounted values. The value per patient was determined by dividing the average results by the average number of patients in that group across all simulations.

4.4.5. Economic perspective and expected outcome

In each strategy, costs and effects were calculated for each group of patients. This analysis was performed using the healthcare perspective, i.e., including all costs associated directly with each intervention, from the time they enter the model until they die. Costs were measured in 2022 Norwegian Kroner (NOK) and effects were measured in total life years (LY) and Quality-adjusted Life Year (QALY). All the values were discounted to the model analytic start time.

ICERs were calculated based on the discounted total value of cost, LY, and QALY of each strategy for all three groups of patients. The ratios were used as main criteria to compare strategies. The ICERs were presented both as cost per LY saved and cost per QALY gained. In order to demonstrate the impact of including spillover effects to non-NCRLM patients, I first present the ICER when calculated based on the outcomes of NCRLM patients only, which estimates the cost-effectiveness of the intervention toward that group of patients. Second, I present the ICER for all three strategies when spillover effects were accounted for, meaning, when including costs and effects to all three groups of patients.

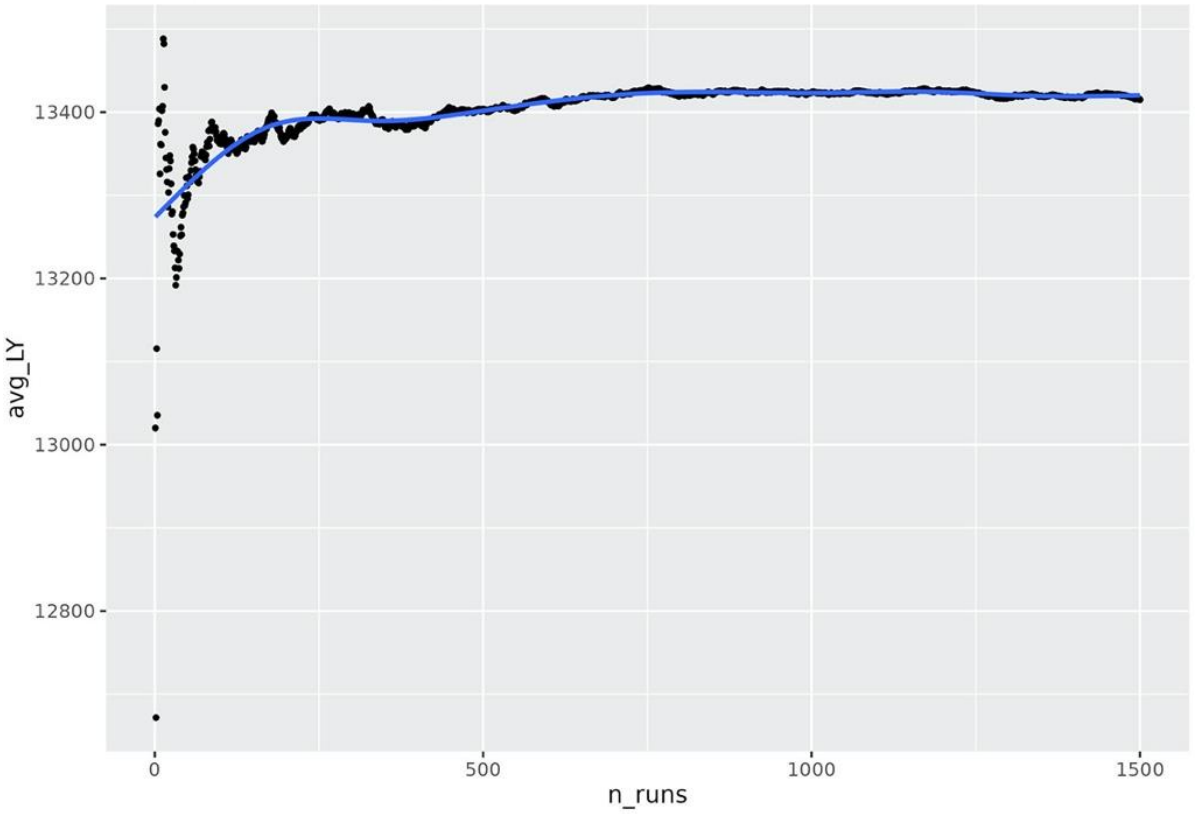

Figure 8: Average total LY of Status Quo Patients over the number of model simulations

4.5. Model validation

4.5.1. Internal Validity

Internal validity was determined by determining whether the outcomes were as anticipated and in line with the model's inputs. All outputs were examined, including the number of patients in the model, blood distribution, background mortality, and disease distribution among Status Quo patients. In particular, the liver matching module in simulations (2) and (3) has been examined carefully to ensure that it complies with the ABO system and that the queueing theory principle has been accurately modeled.

4.5.2. External Validity

The objective of external validity is to ensure that the patient's trajectory and its outcome were logic and reflective of the observed pathways of that group of Norwegian patients. As shown in Table 4, the waiting time was used as a benchmark for reflection. The models output values of waiting time by blood type correspond to the range summarized in Norway for the five-year period 2016-2021[9, 18]. However, the median value was less than what was reported in

Norway because the model does not account for a minimum waiting time, for the purpose of queue simplification. Therefore, the waiting time after a patient enters a model was as short as a few minutes which was not practical. This was a constraint imposed by the model on the waiting time. However, the estimated waiting time, which was different just in days from the observation, should not have a significant impact on the total life years per patient group as the main outcome.

Table 10: Validation of model waiting time

Waiting time by blood type	**Median *(mean)* in the model** (days)	**Median *(range)* in Norway** (days) **[9, 18]**
A	27.6 ***(39.4)***	32 ***(26-42)***
B	39.1 **(53.6)**	48 ***(40-63)***
AB	18.5 **(27.9)**	21 ***(14-26)***
O	75.7 **(89.0)**	92 ***(66-111)***

In addition, it was anticipated that if the number of NCRLM patients in the model increases, the waiting time for Status Quo patients will increase while the total life year will decrease. This projection has been supported as model outcome depicted in Figures 8 and 9 when increase the number of NCRLM patients per year in the model.

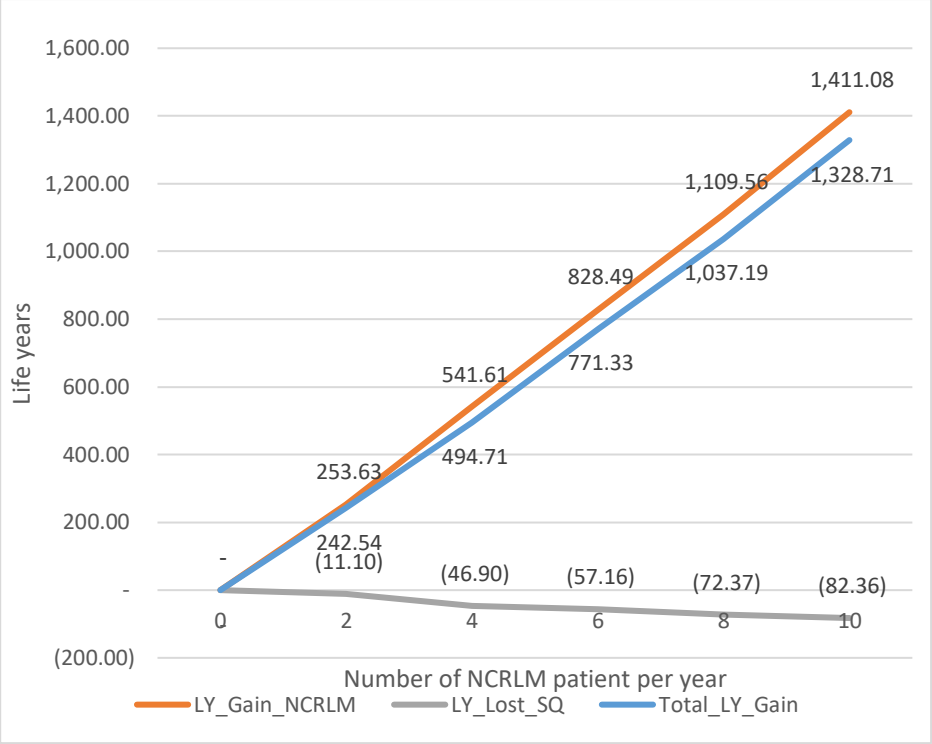

Figure 9: Life year variation on the number of NCRLM patients per year in the model

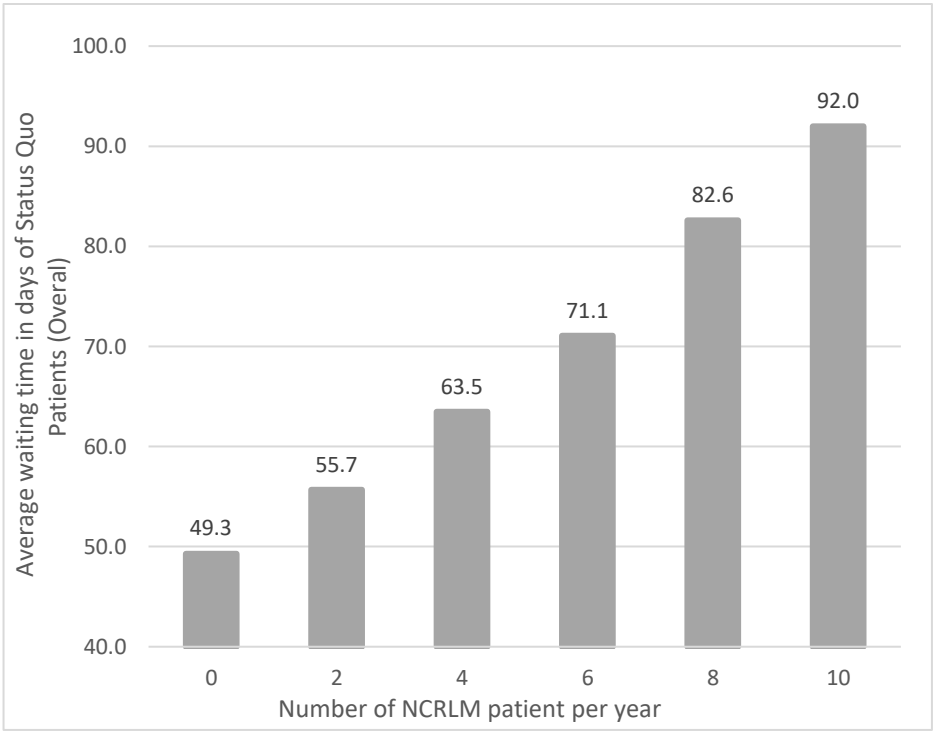

Figure 10: Average waiting time in days of Status Quo Patients by number of NCRLM patients per year

5. Results

5.1. Summarized outcomes by patient groups

5.1.1. NCRLM patients costs and health effects

For strategy I, the life expectancy of NCRLM patients was projected to be 2.33 years with the current standard of care consisting of chemotherapy and palliative care (strategy I). It was evident that liver transplants, whether from deceased or living donors, produced significantly better outcomes for NCRLM patients. For instance, the average life expectancy of NCRLM patients undergoing deceased liver transplantation (Strategy II) and live liver donation (Strategy III) was 14.82 and 15.37 years, respectively, resulting in an increase in QALYs. Compared to strategy I, the QALYs of NCRLM patients treated with strategies II and III have increased to 11.74 and 12.17, respectively.

As shown in Table 11, the expected cost per NCRLM patient in strategies II and III was higher with 2,037,666.9 and 2,173,068.47 NOK respectively, compared to 349,876.5 NOK for strategy I.

5.1.2. Status Quo Patients costs and health effects

In the model, there were only two scenarios for Status Quo patients: waiting in line for a liver without NCRLM patients or with NCRLM patients, as in simulations (2) and (3), respectively. As shown in Table 11, the outcomes for Status Quo patients under Strategies I and III were identical for the case of queuing without NCRLM patients, with the gain of 13.06 life years which was equivalent to 10.35 QALY per patient.

When switching from strategy I/III to strategy II with 2 NCRLM patients per year, there was a loss of approximately 11.1 years of life for the entire group of 1029 Status Quo patients over the 10-year programmatic period, or 0.01 years for a single patient. In addition, the waiting period increased from 49.3 days to 55.7 days.

Due to the inclusion of two NCRLM patients, strategy II reduced the cost for the entire group of Status Quo patients, from 1,171,639,161 to 1,169,526,911. Costs of care decreased because there was a decrease in total life years. With a longer waiting period, some Status Quo patients might die before they were matched with a suitable liver; consequently, some transplantation costs were also eliminated.

5.1.3. Live donors costs and health effects

In strategies I and II, donors live normal lives without donating, whereas in strategy III, donors donate a portion of their liver which cause a negligible amount of LY was lost. The loss was only 0.18 years for the entire group.

As the only cost incurred to the live donor was in strategy III, which were associated cost of care after the donation, I found these patients incurred a cost of 282,157.35 for the entire group of 18 patients over the entire monitoring period.

Table 11: Summary output for each group of patients

Group	Outcomes	Strategy I		Strategy II		Strategy III	
		Entire group	*Per patient*	**Entire group**	*Per patient*	**Entire group**	*Per patient*
NCRLM patients (20*)	Life year	**47.26**	*2.33*	**300.90**	*14.82*	**311.95**	*15.37*
	Discounted Life year	**23.09**	*1.14*	**107.02**	*5.27*	**111.21**	*5.48*
	Waiting Time (days)	**#N/A**	*#N/A*	**1091.35**	*54.6*	**#N/A**	*#N/A*
	Total Cost (NOK)	**13,818,685**	*680,723.40*	**104,658,954**	*5,155,613.50*	**114,552,949**	*5,643,002.41*
	Discounted Total Cost (NOK)	**7,102,502**	*349,876.95*	**41,364,638**	*2,037,666.90*	**44,113,290**	*2,173,068.47*
	QALYs	**39.04**	*1.92*	**238.29**	*11.74*	**246.97**	*12.17*
	Discounted QALYs	**19.17**	*0.94*	**67.55**	*3.33*	**70.30**	*3.46*
Status Quo Patients (1029*)	Life year	**13,439.10**	*13.06*	**13,428**	*13.05*	**13,439.10**	*13.06*
	Discounted Life year	**4,558.13**	*4.43*	**4,554.82**	*4.43*	**4,558.13**	*4.43*
	Waiting Time (year)	**50,705.8**	*49.3*	**57,243.0**	*55.7*	**50,705.8**	*49.3*
	Total Cost (NOK)	**1,171,639,161**	*1,138,508.56*	**1,169,526,911**	*1,136,456.04*	**1,171,639,161**	*1,138,508.56*
	Discounted Total Cost (NOK)	**962,175,458**	*934,967.89*	**959,780,254**	*932,640.42*	**962,175,458**	*934,967.89*
	QALYs	**10,656.02**	*10.35*	**10,639.92**	*10.34*	**10,656.02**	*10.35*
	Discounted QALYs	**3,576.80**	*3.48*	**3,567.99**	*2.30*	**3,576.80**	*3.48*
Live Donors (18*)	Life year	**578.92**	*31.98*	**578.92**	*31.98*	**578.74**	*31.97*
	Discounted Life year	**163.70**	*9.04*	**163.70**	*9.04*	**163.65**	*9.04*
	Total Cost (NOK)	**-**	-	**-**	-	**5,107,048**	*282,157.35*
	Discounted Total Cost (NOK)	**-**	-	**-**	-	**2,598,992**	*143,590.72*
	QALYs	**578.92**	*31.98*	**578.92**	*31.98*	**575.49**	*31.80*
	Discounted QALYs	**163.70**	*9.04*	**163.70**	*9.04*	**156.05**	*8.62*

(*) refers to group average number of patients over the 10-year programmatic period

Table 12: Summary output for entire population

Group	Outcomes	Strategy I	Strategy II	Strategy III
Entire Population	Life year	14,065.28	14,307.82	14,329.79
	Discounted Life year	4,744.92	4,825.54	4,832.99
	Total Cost	1,185,457,846	1,274,185,865	1,291,299,158
	Discounted Total Cost	969,277,960	1,001,144,892	1,008,887,740
	QALYs	11,273.98	11,457.13	11,478.48
	Discounted QALYs	3,759.67	3,799.24	3,803.15

Table 13: Incremental Cost-Effectiveness among strategies, for NCRLM patients only and entire population

	Cost (NOK)	Incremental Cost (NOK)	QALY	Incremental QALY	ICER (NOK/QALY)	LY	Incremental LY	ICER (NOK/LY)
NCRLM Only								
Strategy I	7,102,502	-	19.17	-	-	23.09	-	-
Strategy II	41,364,638	34,262,136	67.55	48.38	708,236.5	107.02	83.93	408,223.5
Strategy III	44,113,290	2,748,652	70.30	2.76	997,656.0	111.21	4.19	655,643.0
Entire Population								
Strategy I	969,277,960	-	3,759.67	-	-	4,744.92	-	-
Strategy II	1,001,144,892	31,866,932	3,799.24	39.57	805,342.6	4,825.54	80.61	395,303.5
Strategy III	1,008,887,740	7,742,848	3,803.15	3.92	1,976,673.6	4,832.99	7.46	1,038,151.9

5.2. Cost-effectiveness for CRLM patients only

According to section 5.1.1, after discounting, strategy I was the least expensive and least effective, followed by strategies II and III (Table 13).

5.2.1. Cost per life-year saved

When estimating the cost-effectiveness using life years as outcome measure, the ICER of strategy II was 408,223.5 NOK per LY saved while the ICER of strategy III was 655,643 NOK per LY saved. Given the Norwegian WTP-threshold benchmarks for a very severe disease, live liver donation (strategy III) would be considered a cost-effective treatment strategy (Table 13).

5.2.2. Cost per QALY gained

When estimating the cost-effectiveness using QALYs as outcome measure, the ICER of strategy II increased to 708,236.5 which was still lower than the WTP threshold per Norwegian guideline for the most severe disease. Thus, strategy II was still considered cost-effective. The ICER of strategy III increased to 997,656.0. Since this was higher than the most severe willingness-to-pay for one QALY in Norway, strategy III would not be considered cost-effective.

5.3. Cost-effectiveness for entire population with spillover effects

The outcomes of the model presented in Table 4 demonstrate that once the intervention for NCRLM patients was modified, the outcomes for the other groups, Status Quo patients and donors, were also subjected to change. The total costs and effects are displayed in Table 12, which indicates that a loss of life years for one group of patients could be offset by a gain of life years for another group. The principle of compensation under entire population perspective also applies to costs and QALYs.

5.3.1. Cost per life-year saved

First, when a broader perspective of entire population was applied, Strategy II was still cost effective. The cost per life-year saved by switching from strategy I to strategy II was even lower, at 395,303.5. However, the ICER of 1,038,151.9 NOK/LY of strategy III was higher than the highest WTP threshold in Norway. Therefore, when a broader perspective applied with spillover effect, every cost and life-year gain count, strategy III was no longer cost effective.

5.3.2. Cost per QALY gained

Strategy II had an ICER of 805,342.6 and was thus cost-effective when the QALYs were used as the outcome measure and analyses were applied to the entire population (including spillover effects). Using the discounted QALY, the incremental effect from strategy II to strategy III was quite low, while the cost increases, resulting in a very high ICER of 1,976,736,6 NOK/QALY. The more expansive the perspective, the less likely it is that strategy III will be cost effective.

6. Discussion

6.1. Discussion of results

In this study, I developed a DES model to estimate the costs and effects of three alternative strategies around treatments of NCRLM patients. Strategy I reflects the current standard of care, in which NCRLM patients receive only palliative chemotherapy, while strategy II and III reflect the alternative treatments where NCRLM-patients receive deceased donor liver transplantation or living donor liver transplantation, respectively. I estimated outcomes for NCRLM-patients, but also, for entire population when including outcomes for Status Quo patients, who were affected if NCRLM-patients were added to the waiting list for deceased liver transplantation (strategy II), and living donors, who were affected if NCRLM-patients were offered a living donor liver transplantation (strategy III). Costs and effects were estimated for each patient subgroup and then totaled for the population as a whole. As a measure of effectiveness, either LY or QALY was used to calculate the ICER for the subset of NCRLM patients and for the entire population. The following section details the paper's most important findings.

6.1.1. Inclusion of NCRLM patients on the waiting list comes at the opportunity cost of Status Quo patients

In strategy II, NCRLM patients were added to the waiting list alongside Status Quo patients, resulting in an increase of waiting time from an average of 49.3 days to 55.7 days and a loss of 11,1 life years for all Status Quo patients. The total life year gain for the entire population, however, was positive at 242.5 years. This finding also supports the findings of Sjule and Vinter in their paper, in which they evaluate the opportunity cost of including NCRLM patients on the waiting list for a deceased liver under a variety of scenarios [9]. When an increasing number of NCRLM patients were included in the model's validation, this finding is also bolstered. As shown in Figures 8 and 9, as the number of NCRLM patients on the waiting list per year increases, the average waiting time for a liver increases while the total number of life

years lost decreases for Status Quo patients. As waiting time increases, there was a greater likelihood that Status Quo patients would die while waiting for a liver transplant.

Since Norway has a surplus of approximately 12 deceased livers per year and also participates in the Nordic organ exchange program, which is facilitated by Scandia Transplant, the loss of life years associated with including 2 NCRLM patients was minimal in comparison to the life years gained.

6.1.2. Liver transplants from living donors to NCRLM patients incur expenses and health risks for the donors.

When switching NCRLM patients from strategy I of palliative chemotherapy only to strategy III of liver transplantation from a living donor, it was anticipated that 311.95 life years for the entire group, or 15.37 years per patient, would be gained. These results, when discounted, were 111.21 and 5.48 years, respectively (Table 11). It appears that there was no loss of life years for the donor after discounting (life expectancy remained at 9.04 years in both cases, per Table 11). In fact, the entire group of donors lost 0.18 years, which was equivalent to 0.05 years when discounted. After discounting, the incremental effects were negligible because the time in the model can span the lifetime of patients, which can be several decades from the starting time until they die. This number makes sense given that the probability of death during open hepatic resection for donation was 0.02 percent [34]. The model has simulated 18,000 donors over 1000 simulations, with an average of 18 donors per simulation. Thus, the number of simulated donors who died during surgery was only 3,6. When the efficacy of each strategy was summed across all patient groups, the loss of the donor group may be negligible and has little effect on the change in total discounted life for the entire population.

The cost of the donor's open hepatic surgery was actually accounted for as the cost of organ harvesting for NCRLM patients. In strategy III, the donor incurred only the cost of post-donation care, which was 2,598,992 NOK for the entire donor pool. This cost has a significant impact on the ICER, as the incremental cost between strategies II and III was 7,742,848 NOK (Table 13).

6.1.3. The impact of outcome measurement and spillover effect on the cost-effectiveness of strategies

First, the liver diseases that can be treated with a transplant were quite severe. Consequently, HRQoL was relatively low, making QALY a much more comprehensive outcome measurement than the life year alone. Consequently, ICERs utilizing incremental QALYs as

the denominator were higher than ICERs utilizing LY as the primary outcome. As in the case of NCRLM with the same WTP threshold, strategy III may no longer be cost effective if ICER with QALY was used instead of LY under the same WTP threshold. The HRQoLs in this model were largely estimated, with the exception of NCRLM patients who have HRQoLs data from reputable sources. Therefore, ICER of LY, which is more certain, should be added as a reference for cost-effectiveness.

The ICERs were calculated for both the NCRLM patient population and the entire population. As stated in section 5., the broader the perspective, the less likely it is that strategy III will be cost-effective. First, there was a significant life year gain from strategy I to strategy II, both when looking at the NCRLM group alone and when looking at the entire population, as the life year loss of Status Quo patients was minimal and there was no loss for the donor. Also, the cost increased significantly due to the availability of liver transplantation for NCRLM patients. Therefore, when a border perspective was applied, the change in ICER between strategies I and II was not significant and strategy II was cost effective in both scenarios with ICERs of less than 850,000 NOK/QALY.

When spillover effects were applied to the ICER of strategy III over strategy II, the cost increased significantly while the gained effect was small. The cost increased for three reasons: increased cost of Status Quo patient (due to the addition of more life years of Status Quo patients in strategy II); the cost of care for the donors after transplantation; and most importantly, the cost of liver transplantation from a living donor was considerably higher than the cost of liver transplantation from a deceased donor. Approximately 22 life years (and a similar amount of QALY) were gained by switching from strategy II to strategy III. After discounting, however, the incremental effect was negligible, with only 3.92 QALYs and 7.46 Lys, respectively. Therefore, the ICER of strategy III over strategy II was extremely high, making it not cost-effective, even with the highest WTP threshold in Norway.

6.2. Contribution and Limitations

6.2.1. Model strength

The DES model in R has been constructed in a way that it can simulate all three strategies' scenarios while also allowing patient groups to interact. As described in section 4.4.2, the spillover effect on Status Quo patients has been validated such that the inclusion of NCRLM causes an increase in waiting time and loss of life years for Status Quo Patients. To accurately reflect the queueing effect in strategy II when NCRLM patients were added to the waiting list,

the model also accounted for the burn-in period, which was 7 years. The spillover effect on the donor has been associated with an increase in cost and a reduction in QALY for the donor, as reported in Table 11. The model is novel because it made it possible to account for the spillover effect in transplantation. To the best of my knowledge, this type of model has never been created before, and no economic evaluation in health care model in R has ever been performed for queueing effect. The outcomes were recorded precisely for each patient group and across three strategies, including both QALYs and life years. The specific outcomes have demonstrated the model's transparency.

6.2.2. Data collection and uncertainty

First, due to a lack of resources and the complexity of the model, there were a few structural and input assumption limitations. There was no minimum waiting time modeled with the intent of simplifying the model. Therefore, in rare cases the simulated wait time was as short as a few minutes. This structural element reduces the model's median waiting time as reported in Table 4. Due to the fact that post-transplant survival was independent of waiting time, the effect of a few days shorter waiting time on the final outcome of life year or QALY was supposed to be insignificant [47].

Second, there were also some parameter uncertainties. Due to a lack of data, the survival time of the Status Quo patients prior to surgery was not precisely estimated, but rather based on the similar indicator of NCRLM patients. This could lead to an underestimation of the risk of the Status Quo patient dying while on the waiting list. Consequently, this would lead to a further underestimation of the spillover effect of including NCRLM patients on the waiting list for deceased livers in strategy II. In addition, both the cost and utility of Status Quo patients prior to or after transplantation were estimated based on the indicator of NCRLM patients. The ICER of strategy II of 805,342.6 NOK/QALY was close to the maximum willingness to pay of 850,000 NOK/QALY. Thus, there was possibility that the uncertainty could affect the cost-effectiveness of strategy II. However, the estimation of the spillover effect as well as the opportunity cost of offering deceased liver transplantation to NCRLM patients (strategy II) were believed to be robust, because the critical time-to-event and cost data were obtained from reputable sources [8, 15, 16, 18, 33, 37, 42].

Moreover, the majority of strategy III's parameters were based on either assumption from deceased liver transplantation cases or expert opinion. In actuality, the offer of liver transplantation from a living donor was expected to have the same efficacy as transplantation

from deceased donors, as all parameters of efficacy were identical. These parameters include post-transplantation survival, dropout rate, surgical mortality rate, and re-transplantation rate. However, the total life year was slightly increased, which may be explained by the fact that in the model, the waiting time in strategy III was a maximum of 21 days (3 weeks), whereas in strategy II, the average waiting time was approximately 54 days. Strategy III reduces the likelihood that an NCRLM patient will die while waiting, resulting in a greater total life year gained. For details, see Table 11. This estimation of living donor effectiveness was acceptable given the lack of data regarding liver transplants from living donors in Norway.

Concerning the estimation of cost for Strategy III, the cost of living donor transplant surgery was replicated primarily from deceased transplantation plus the cost of two liver resection surgeries, one for organ harvesting and the other for the second RAPID technique surgery on the patient. This estimation made the cost of transplantation from a living donor more expensive than from a deceased donor. However, since the cost of deceased donor transplantation includes a portion of expensive organ facilitation, this estimate may be inaccurate. Indeed, the cost of facilitating deceased organs was substantial due to the cost of having a team on call 24 hours a day, seven days a week, along with an airplane, whenever a deceased organ becomes available. Not only is the standby cost allocated to deceased livers, but also to all deceased organs, including kidneys, hearts, and lungs. This makes it difficult to allocate exactly stand-by cost to each deceased liver transplanted [29]. In contrast, this expense could be eliminated in the case of living donor transplantation, as the donor and recipient would travel to the same hospital for the operation. Therefore, living donor liver transplantation may be less expensive than liver transplantation from a deceased donor. In this case, with a lower transplantation cost, given that the cost of post-donation care was relatively comparable, Strategy III could become cost-effective with cost per QALY gain less than the current WTP threshold, at least for NCRLM only, since Strategy II was cost effective (Table 13).

Besides, some important parameters of this model were still deterministic such as post-transplantation survival parameters. For now, with a small sample and a short observation period, these parameters were pulled from extrapolation and assumed that if a patient could survival up to ten years after the transplantation, he or she could get the background mortality. Thus, it would be beneficial to perform probabilistic sensitivity analysis (PSA) on this parameter. However, within the scope of a master thesis and limitation of computational capacity, the condition was not favorable to perform a PSA for this model.

6.3. Ethical aspects of live donor transplantation in Norway

From strategy II to strategy III, a small loss of life years was observed for the donor, amounting to 0.18 life years for a sample of 18 patients (0.01 year per patient). After discounting, this loss for the group was 0.05 years, which was negligible compared to the incremental of 7.46 discounted life years gain from strategy II to strategy III. As discussed in section 6.1.2, the majority of the spillover effect toward the donor was on the cost side, not the health loss side. Due to the small impact on the final ICER caused by the extremely low probability of death during donation, the decision maker would be unable to carefully consider the actual loss of the donor. The average number of life years lost does not accurately reflect the loss for the donor; when the loss occurs due to death in surgery, it can be devastating for the donor and their family.

In reality, a donor's life year may be significantly more valuable than a patient's life year; in contrast, a QALY gain for a sick patient may be different to a health person. However, either QALY or LY was treated the same in a cost-effectiveness analysis framework. In this model, when QALY was applied, the donor lost 7.65 QALYs, which decreased the incremental QALY of the entire population while the QALY for both NCRLM patients and Status Quo Patients increased. This impact of QALY loss for the donor has had a substantial impact on the cost per QALY gained of strategy III compared to strategy II.

In addition, although the clinician opinion report a mortality rate of the donor was low in this instance, another study found that the mortality and morbidity rates of donors could be as high as 0.2 percent and 24.2 percent, respectively [48]. The mortality rate of 0.2 percent among donors is ten times greater than the parameter used in the model of this paper. This rate could make strategy III strongly dominated from the perspective of the entire population. In the case of NCRLM patients receiving a liver transplant from a living donor, there is thus a need for further research into the effects on the donor.

6.4. Future research prospect

With the limitation noted in section 6.2., these estimations of cost and effectiveness for each strategy could result in imprecise cost-effectiveness estimates for three strategies from the perspective of the entire population with a specific WTP threshold. Thus, the most significant contribution of this paper was the construction of the model. Future estimates of cost-effectiveness are therefore anticipated to incorporate improved parameter inputs. In addition, a

probabilistic sensitivity analysis of the transplantation's efficacy is required to improve robustness of the conclusion.

7. Conclusion

In conclusion, this study developed a model to estimate the cost-effectiveness of three alternatives for three group within a population: NCRLM patients, Status Quo patients, who were waiting for deceased liver transplantation, and living liver donors. The three options centered on NCRLM patients were palliative care only, liver transplantation from deceased donors, and living donor transplantation, namely as strategy I, II and III in the model. The framework has successfully accounted for the spillover effect from the perspective of the entire population, providing the decision maker with a comprehensive view of the costs and effects of various strategies. One strategy may be cost-effective for one group of patients, but when its effects on other groups were considered, it may no longer be cost-effective, as in the case of live liver donation (strategy III) with an ICER using life years in this paper. The paper also differentiates the impact of cost per QALY gained versus cost per LY gained on decision making.

The initial results of this model also support the prior findings of Bjørnelv et al [8, 42] such that liver transplantation from deceased donor was cost-effective for highly selected patients (as in SECA-II criteria), either from the NCRLM group of patients only or from entire population perspective. The model also reflects the findings of Sjule and Vinter [9] that the inclusion of NCRLM patients in the waiting list come at the opportunity cost of increased waiting time and some life year loss for the Status Quo patients. Therefore, though require further research incorporate future empirical studies, offering deceased liver transplantation seems to be cost-effectiveness considering both the direct patient consequences and spillover effects under assumptions of a comprehensive QALY metric.

8. Reference

1. WHO. *Cancer* 2020 [cited 2021; Available from: https://www.who.int/en/news-room/fact-sheets/detail/cancer.
2. Oyeyemi, S.O., et al., *Exploring geographical differences in the incidence of colorectal cancer in the Norwegian Women and Cancer Study: a population-based prospective study.* Clin Epidemiol, 2019. **11**: p. 669-682.
3. Cancer Registry of Norway, *Cancer in Norway 2019*. 2019: Oslo.
4. Quan, D., et al., *The role of liver resection for colorectal cancer metastases in an era of multimodality treatment: A systematic review.* Surgery, 2012. **151**(6): p. 860-870.
5. Masi, G., et al., *Randomized trial of two induction chemotherapy regimens in metastatic colorectal cancer: an updated analysis.* Journal of the National Cancer Institute, 2011. **103**(1): p. 21-30.
6. Dueland, S., et al., *Survival Following Liver Transplantation for Patients With Nonresectable Liver-only Colorectal Metastases.* Ann Surg, 2020. **271**(2): p. 212-218.
7. Dueland, S., et al., *Chemotherapy or Liver Transplantation for Nonresectable Liver Metastases From Colorectal Cancer?* Ann Surg, 2015. **261**(5): p. 956-960.
8. Bjørnelv, G.M.W., et al., *Cost-effectiveness of liver transplantation in patients with colorectal metastases confined to the liver.* British Journal of Surgery, 2018. **106**(1): p. 132-141.
9. Sjule, H.M. and C.A. Vinter, *The true opportunity costs of offering liver transplantation to patients with colorectal liver metastases: A discrete event simulation analysis*, in *Department of Health Management and Health Economics*. 2021, University of Oslo: Oslo.
10. Rauchfuß, F., et al., *Living donor liver transplantation with two-stage hepatectomy for patients with isolated, irresectable colorectal liver—the LIVER-T(W)O-HEAL study.* World Journal of Surgical Oncology, 2019. **17**(1): p. 11.
11. Nadalin, S., et al., *RAPID procedure for colorectal cancer liver metastasis.* International Journal of Surgery, 2020. **82**: p. 93-96.
12. Königsrainer, A., et al., *Paradigm shift in the management of irresectable colorectal liver metastases: living donor auxiliary partial orthotopic liver transplantation in combination with two-stage hepatectomy (LD-RAPID).* Annals of Surgery, 2019. **270**(2): p. 327-332.
13. Cancer registry of Norway, *Cancer in Norway - Cancer incidence, mortality, survival and prevalence in Norway*. 2019, Cancer registry of Norway: Oslo.
14. Smedman, T.M., et al., *Liver transplantation for unresectable colorectal liver metastases in patients and donors with extended criteria (SECA-II arm D study).* BJS Open, 2020. **4**(3): p. 467-477.
15. Melum, E., *The Nordic Liver Transplant Registry (NLTR) - Annual report 2020*, t.N.L.T.R. (NLTR), Editor. 2021, Scandia Transplant

16. Fosby, B., et al., *Liver transplantation in the Nordic countries - An intention to treat and post-transplant analysis from The Nordic Liver Transplant Registry 1982-2013.* Scand J Gastroenterol, 2015. **50**(6): p. 797-808.

17. Kim, W.R., et al., *OPTN/SRTR 2016 Annual Data Report: Liver.* American Journal of Transplantation, 2018. **18**(S1): p. 172-253.

18. Scandiatransplant. *Scandiatransplant figures*. 2021; Available from: http://www.scandiatransplant.org/data/scandiatransplant-figures.

19. Dueland, S., et al., *Selection criteria related to long-term survival following liver transplantation for colorectal liver metastasis.* American Journal of Transplantation, 2020. **20**(2): p. 530-537.

20. Line, P.-D., et al., *A novel concept for partial liver transplantation in nonresectable colorectal liver metastases: the RAPID concept.* Annals of surgery, 2015. **262**(1): p. e5-e9.

21. omsorgsdepartementet, H.-o., *Forskrift om prioritering av helsetjenester, rett til nødvendig helsehjelp fra spesialisthelsetjenesten, rett til behandling i utlandet og om klagenemnd (prioriteringsforskriften)*. 2000, Helse-og omsorgsdepartementet Oslo.

22. Magnussen, J. *Severity of illness and priority setting in Norway* 2015; Available from: https://www.regjeringen.no/no/dokumenter/pa-ramme-alvor/id2460080/?fbclid=IwAR0Eqd75UKwjy1tCmoOxMjkJCgDOtabo3HBl3r_QpmbN_vzVpn_yLlVduZk.

23. Drummond, M.F., et al., *Methods for the economic evaluation of health care programmes*. 2015: Oxford university press.

24. Pedersen, K., *The Health Benefits, Resource Use and Cost-Effectiveness of Current and Future Cervical Cancer Screening Policies in Norway*, in *Department of Health Management and Health Economics - Institute of Health and Society - Faculty of Medicine*. 2018, University of Oslo: Oslo.

25. Legemiddelverk, S., *Guidelines for the submission of documentation for single technology assessment (STA) of pharmaceuticals*. 2020.

26. Caro, J.J., et al., *Discrete event simulation for health technology assessment*. 2015: CRC press.

27. Briggs, A., M. Sculpher, and K. Claxton, *Decision modelling for health economic evaluation*. 2006: Oup Oxford.

28. Medhi, J., *Stochastic models in queueing theory*. 2002: Elsevier.

29. Burger, E. and G. Bjørnelv, *Master Thesis Supervision Meeting March 11th 2022*, T. Nguyen, Editor. 2022.

30. Kleinbaum, D.G. and M. Klein, *Survival analysis: a self-learning text*. Vol. 3. 2012: Springer.

31. Latimer, N.R., *Survival analysis for economic evaluations alongside clinical trials—extrapolation with patient-level data: inconsistencies, limitations, and a practical guide.* Medical Decision Making, 2013. **33**(6): p. 743-754.

32. Briggs, A.H., et al., *Model parameter estimation and uncertainty: a report of the ISPOR-SMDM Modeling Good Research Practices Task Force-6.* Value in Health, 2012. **15**(6): p. 835-842.

33. Dueland, S., et al., *Chemotherapy or liver transplantation for nonresectable liver metastases from colorectal cancer?* Ann Surg, 2015. **261**(5): p. 956-60.

34. Dueland, S. and P.-D. Line, *Consultation regarding Live Liver Transplantation in Norway* in *HMM4501 - Master Thesis at Faculty of Medicine, Uniersity of Oslo*, Tung Nguyen, Emily Burger, and Gudrun Bjørnelv, Editors. 2022.

35. Norsk Helseinformatikk. *Blodtyper*. 2021 [cited 2021; Available from: https://nhi.no/kroppen-var/organer/blodtyper/

.

36. Statistisk sentralbyrå (Statistics Norway). *07902: Dødelighetstabeller, etter kjønn og alder 1966 - 2021*. 2021; Available from: https://www.ssb.no/statbank/table/07902.

37. Vinter, C.A., *The true opportunity costs of offering liver transplantation to patients with colorectal liver metastases: A discrete event simulation analysis*. 2021.

38. Tveit, K.M., et al., *Phase III trial of cetuximab with continuous or intermittent fluorouracil, leucovorin, and oxaliplatin (Nordic FLOX) versus FLOX alone in first-line treatment of metastatic colorectal cancer: the NORDIC-VII study.* Journal of clinical oncology, 2012. **30**(15): p. 1755-1762.

39. Ratcliffe, J., et al., *Assessing health-related quality of life pre [ndash] and post [ndash] liver transplantation: A prospective multicenter study.* Liver transplantation, 2002. **8**(3): p. 263-270.

40. Färkkilä, N., et al., *Health-related quality of life in colorectal cancer.* Colorectal Disease, 2013. **15**(5): p. e215-e222.

41. Bruns, H., et al., *Prediction of postoperative mortality in liver transplantation in the era of MELD-based liver allocation: a multivariate analysis.* PLoS One, 2014. **9**(6): p. e98782.

42. Bjørnelv, G.M.W., et al., *Cost-effectiveness of liver transplantation versus last-resort systemic therapy for colorectal liver metastases.* British Journal of Surgery, 2022.

43. Janik, M.K., et al., *Long-Term Health-Related Quality of Life in Living Liver Donors.* Annals of transplantation, 2019. **24**: p. 45-51.

44. Fretland, Å.A., et al., *Quality of life from a randomized trial of laparoscopic or open liver resection for colorectal liver metastases.* British Journal of Surgery, 2019. **106**(10): p. 1372-1380.

45. H. Koffijberg, M. van de Ven, and K. Degeling. *Discrete Event Simulation in R to Support Healthcare Decision Making - Using simmer for Discrete Event Simulation in R*. 2021 [cited 2022; Available from: https://github.com/koendegeling/SMDM_DESinR.

46. Statistics Norway. *05375: Life expectancy, remaining years, by age, contents, year and sex*. 2022 [cited 2022; Available from: https://www.ssb.no/en/statbank/table/05375/tableViewLayout1/.

47. Yao, F.Y., et al., *A follow-up analysis of the pattern and predictors of dropout from the waiting list for liver transplantation in patients with hepatocellular carcinoma: implications for the current organ allocation policy.* Liver Transplantation, 2003. **9**(7): p. 684-692.

48. Cheah, Y.L., et al., *Incidence of death and potentially life-threatening near-miss events in living donor hepatic lobectomy: A world-wide survey.* Liver Transplantation, 2013. **19**(5): p. 499-506.

Appendix

R Code (added for transparency – to be cleaned for future replication)

This code includes 7 modules:

I. **Module I. Common Values**
II. **Module II. Functions**
III. **Module III. Trajectories**
IV. **Module IV. Common random values**
V. **Module V. To define burn-in period**
VI. **Module VI. To define number of simulations**
VII. **Module VII. Simulation, data analysis, and capture of outcomes**

```
###Common values####
#rm(list=ls())
### In this page we will create a value that use commonly across the
model###
#### Load packages, otherwise you can install by removing the hash below;
# install.packages('simmer')
# install.packages('simmer.plot')
# install.packages('flexsurv')
# install.packages('data.table')
# install.packages('tibble')
# install.packages('tidyverse')
# install.packages('DiagrammeR')
# install.packages('dplyr')

library(simmer)
library(simmer.plot)
library(flexsurv)
library(data.table)
library(tibble)
library(tidyverse)
library(DiagrammeR)

### I. Parameters ------------------------ ------------------------
------------------------ ------------------------

# This section defines the parameters that are used to define the discrete
event simulation.
# These are the parameters that define and are used in the trajectory. The
prefix of the
# parameter name indicates what type of parameter it is:
#   c_    cost
#   d_    distribution parameter
#   p_    probability
#   u_    health utility value
#   a_    arrival
## 1.0 Background mortality parameter ``
Background_df <- tibble(
  age = c(43,57,59,61),
  ##survival background
  death_shape_bg = c(0.1143273, 0.1156375, 0.1235331, 0.1267308 ),
  death_rate_bg = c(0.0017307, 0.009412, 0.0111042, 0.0133527 ),
)


get_background_data = function(type, data) {
  Background_df[Background_df$age == type, paste0(data), drop = TRUE]}

    get_background_data(43,"death_shape_bg")

mean(rgompertz(1000,0.1143273,0.0017307))

mean(rllogis(1000,0.7288876,2.7190940))

# Test the tible data set

mean(rgompertz(1000, get_background_data(59,"death_shape_bg"),
get_background_data(59,"death_rate_bg")))
```

```
## 1.1 NCRLM patients parameters----------

#> These parameters are used across the model
#> palliative survival parameter


#> Arrival parameters
a_avg_NCRLM_arrival <- 180.5
a_NCRLM_rate_arrival <- 365/a_avg_NCRLM_arrival

#> NCRLM Blood discrete distribution (no mention in the paper)
p_NCRLM_bl_A <- 0.49
p_NCRLM_bl_B <- 0.08
p_NCRLM_bl_O <- 0.39
p_NCRLM_bl_AB <- 0.04


#> Parameters of the Gompertz distribution for palliative care and
background mortality

d_NCRLM_death_shape_pal <- 0.2834202
d_NCRLM_death_rate_pal  <- 0.2400557
mean(rgompertz(1000, d_NCRLM_death_shape_pal, d_NCRLM_death_rate_pal))

mean(rgompertz(1000,0.1255331,0.0031042))

d_NCRLM_death_shape_bg <- 0.01
d_NCRLM_death_rate_bg <- 0.05
mean(sort(rgompertz(1000, 0.008487, 0.066535)))

# Parameters for the post transplantation survival process: parameters of
the Weibull distribution that is used to
# model the time from the surgery to the start of the recovery process
(d_),
d_surival_shape_NCRLM_post_sur <- 0.008487
d_surival_scale_NCRLM_post_sur <- 0.066535

#plot(sort(fn_rweibull(1000, 0.838474089,0.119512298)))
#> For strategy 2 survival post surgery

d_surival_shape_NCRLM_post_sur_str2  <- 0.5
d_surival_rate_NCRLM_post_sur_str2  <- 0.0005

mean(rgompertz(1000,d_surival_shape_NCRLM_post_sur_str2
,d_surival_rate_NCRLM_post_sur_str2 ))
mean(rgompertz(1000,0.1143273 ,0.0017307 ))
#> Dropout rate of first arrive NCRLM str2
p_drop_NCRLM_str2 <- 0.1

#> #> Priority rate of first arrive NCRLM str2
p_priority_NCRLM_str2 <- 0.1

#> #> Probability of re-transplantation
str2_p_deceased_retrans_NCRLM <- 0.09

#> Time surviving while waiting for retransplantation - maximum 90days
```

```
str2_retrans_death_shape <- 236.7204
str2_retrans_death_rate <- 7.69*10^-7

mean(rgompertz(1000, str2_retrans_death_shape, str2_retrans_death_rate))
mean(rgompertz(1000, 236.7204, 7.69*10^-7))

#> Probability of die while surgery
#>
str2_p_deceased_die <- 0.02


#> NCRLM Cost
c_NCRLM_cost_palliative_year <-816
c_NCRLM_survival_care <- 200  #For use in str2&3

#> One-off cost deceased donor strategy 2
NCRLM_c_1st_surgery_str2 <- 3500
NCRLM_c_2nd_surgery_str2 <- 5000

#> One-off cost live donor strategy 3
c_NCRLM_1st_surgery_str3 <- 3500
c_NCRLM_2nd_surgery_str3 <- 5000

#> NCRLM Utility
u_NCRLM_palliative <- 0.82
u_NCRLM_after_surgery <- 0.8    #For use in str2&3

# Parameters for the probability of drop out from the waiting list

p_drop_out_NCRLM_str3 <- 0.1

# Parameters for the waiting to surgery: parameters of the Weibull
distribution that is used to
# model the waiting time if not drop out.
d_sur_waiting_shape_NCRML_Str3 <- 5.54767

d_sur_waiting_scale_NCRML_Str3 <- 2.987

#> test time patient can die while waiting for transplantation

mean(rgompertz(1000, d_sur_waiting_shape_NCRML_Str3,
d_sur_waiting_scale_NCRML_Str3))

# Parameters for the probability of dying at surgery

p_die_surgery_NCRLM_str3 <- 0.02
p_die_surgery_NCRLM_str3_retrans <- 0.02*0.749 + (1-0.749)



### Re transplantation
# Parameters for the probability of re-transplantation (survive for 90
days or being re-tranplanted)

p_retranplantation_NCRLM_str3 <- 0.09

# Parameters for time dying after surgery failure and wait for
retransplantaion
```

```
d_death_shape_retrans_NCRML_Str3 <-  9.5
d_death_rate_retrans_NCRML_Str3 <-  0.7

#> test time patient can die while waiting for restranplantation
mean(rgompertz(1000, d_death_shape_retrans_NCRML_Str3,
d_death_rate_retrans_NCRML_Str3))

#> waiting time for second transplantation - Weibull distribution
d_waiting_shape_retrans_NCRML_Str3 <- 2.5
d_waiting_scale_retrans_NCRML_Str3<- 0.2

mean(rweibull(1000,d_waiting_shape_retrans_NCRML_Str3,d_waiting_scale_retrans_NCRML_Str


## 1.2 SQ patients parameters ------------------
#> These parameters are used in strategy 2 only

##> Arrival parameters
SQ_Patients_avg_day_arrival <- 3.55/365
SQ_Patients_arrival_rate <- 1/SQ_Patients_avg_day_arrival
mean(rexp(1000,rate = SQ_Patients_arrival_rate ))*365

#> SQ_Patients Blood distribution

p_Status_Quo_bl_A <- 0.49
p_Status_Quo_bl_B <- 0.08
p_Status_Quo_bl_O <- 0.39
p_Status_Quo_bl_AB <- 0.04

#> SQ_Patients disease distribution
p_Status_Quo_HCC_1 <- 0.137
p_Status_Quo_PBC_2 <- 0.067
p_Status_Quo_PSC_3 <- 0.18
p_Status_Quo_AC_4 <- 0.083
p_Status_Quo_ALF_5 <- 0.066
p_Status_Quo_Other_6 <-0.467


#> Table to look up parameter for 6 types of Status quo patients
# define fn_rweibull()

fn_rweibull = function(n, shape, scale){
  (-log(runif(n))/ scale)^(1/shape)
}

#> ##FIXME fill in with exact parameter here
SQ_df <- tibble(
  type = 1:6,
  ##survival before surgery
  death_shape1 = seq(1.9, 2.2, length.out =  6),
  death_rate1 = seq(0.04, 0.06, length.out =  6),
  ## survival after surgery
  death_shape2 = c(0.838474089, 0.780242352, 0.928784349, 1.129434878,
0.449819045, 0.449819045 ),
  death_scale2 = c(0.119512298, 0.072519412,  0.039010673, 0.038154942,
0.171966601 ,  0.171966601 ),
  drop_rate= seq(0.058,0.062, length.out = 6),
  priority_rate= seq(0.08,0.12, length.out = 6)
```

```
)

mean(rgompertz(1000,1.9,0.04))

# input data  (death_shape, rate) of StatusQuo patients

get_SQ_data = function(type, data) {
  SQ_df[SQ_df$type == type, paste0(data), drop = TRUE]
}

#> Probability of re-transplantation
##NOTE these are now the same as NCRLM
str2_p_deceased_retrans <- 0.134

#> Time surviving while waiting for retransplantation - maximum 90days

str2_retrans_death_shape <- 10.3
str2_retrans_death_rate <- 3.39

mean(rgompertz(1000, str2_retrans_death_shape, str2_retrans_death_rate))

#> Probability of die while surgery
str2_p_deceased_die <- 0.02

#>Define health economics parameters use in strategy 2
#FIXME here now assumne that cost and QALY are the same for all 6 types of
SQ patients.

#> Health utility values for the time before surgery (or for the life time
if the individual
#> is not eligible for surgery) and the time after surgery for those who
received transplantation.

SQ_u_prior_surgery <- 0.4
SQ_u_after_surgery <- 0.82

#> Continuous cost:
SQ_c_before_trans_care <- 128578.1
SQ_c_survival_care <- 179657.6

#> One-off cost
SQ_c_1st_surgery <- 3500
SQ_c_2nd_surgery <- 500


## 1.3 Deceased Liver Parameters.------------

#> Liver arrival

Avg_Liver_arrival<- 3.18/365
d_liver_arrival_rate <- 1/Avg_Liver_arrival

Liver_first_day_arrival <- 35/365
#exponentially distributed
mean(rexp(1000,rate = d_liver_arrival_rate))*365
distribution = from(start_time = Liver_first_day_arrival, dist =
function() rexp(1000,rate = d_liver_arrival_rate))
```

```
#> blood distribution
p_Liver_bl_A <- 0.465
p_Liver_bl_B <- 0.075
p_Liver_bl_O <- 0.44
p_Liver_bl_AB <-0.02



## 1.4 Donor parameters --------------------
#> background survival parameter

d_death_shape_donor_bg <- 0.1143273
d_death_rate_donor_bg  <- 0.0017307

mean(rgompertz(1000, d_death_shape_donor_bg, d_death_rate_donor_bg))

#> post donation survival parameters

d_death_shape_donor_sur <- 0.1143273
d_death_rate_donor_sur  <- 0.0017307

mean(rgompertz(1000, d_death_shape_donor_sur, d_death_rate_donor_sur))

#> probability of dying in surgery
p_die_surgery_donor <- 0.03/100

#> cost of post donation survival donor care per year
inflation_rate <- 0.02
Exchange_rate_EUR_to_NOK <- 9.67
c_donor_care_1st_year   <-
6810*4*(1+inflation_rate)^(2022-2019)*Exchange_rate_EUR_to_NOK
c_donor_care_2_10_year  <-
10*4*(1+inflation_rate)^(2022-2019)*Exchange_rate_EUR_to_NOK


#> utility of donor pos donation
u_donor_care_1_year <- 0.82

## 1.5 Health Economics Parameters --------------

discount_rate <- 0.04

### Parameter table

##Discount to point in time (year of start running the model)

time_of_discounting <- 7

inflation_rate <- 0.02
Exchange_rate_EUR_to_NOK <- 9.67


## Str1 - Cost

C_Prior_Palliative_chemotherapy_month_0_6 <-
9119*4*(1+inflation_rate)^(2022-2018)*Exchange_rate_EUR_to_NOK
C_Prior_Palliative_chemotherapy_month_7_12 <-
2061*4*(1+inflation_rate)^(2022-2018)*Exchange_rate_EUR_to_NOK
```

```
C_Prior_Palliative_chemotherapy_month_13_onward <-
3071*4*(1+inflation_rate)^(2022-2018)*Exchange_rate_EUR_to_NOK
C_best_supportive_care <-
22329*4*(1+inflation_rate)^(2022-2018)*Exchange_rate_EUR_to_NOK


## Str2 - Cost NCRLM

C_waiting <-
9119*4*(1+inflation_rate)^(2022-2018)*Exchange_rate_EUR_to_NOK
C_str2_1st_transplantation <-
125167*(1+inflation_rate)^(2022-2019)*Exchange_rate_EUR_to_NOK
C_str2_2nd_transplantation <-
27120*(1+inflation_rate)^(2022-2019)*Exchange_rate_EUR_to_NOK

C_Post_survival_disease_free_month_1_3  <-
6810*4*(1+inflation_rate)^(2022-2019)*Exchange_rate_EUR_to_NOK
C_Post_survival_disease_free_month_4_onwards <-
5915*4*(1+inflation_rate)^(2022-2019)*Exchange_rate_EUR_to_NOK
C_Post_survival_recurrence_month_1_12 <- 7525 *
4*(1+inflation_rate)^(2022-2019)*Exchange_rate_EUR_to_NOK
C_Post_survival_recurrence_month_13_onwards <-
3946*4*(1+inflation_rate)^(2022-2019)*Exchange_rate_EUR_to_NOK

C_Post_Surival_Palliative_chemotherapy_month_0_6 <-
10340*4*(1+inflation_rate)^(2022-2018)*Exchange_rate_EUR_to_NOK
C_Post_Surival_Palliative_chemotherapy_month_7_12  <-
3281*4*(1+inflation_rate)^(2022-2018)*Exchange_rate_EUR_to_NOK
C_Post_Surival_Palliative_chemotherapy_month_13_onward <-
4291*4*(1+inflation_rate)^(2022-2018)*Exchange_rate_EUR_to_NOK


## Str2 - Utility

U_Disease_free_cycle_1 <- 0.64
U_Disease_free_cycle_2_and_onward  <- 0.71
U_Recurrence_first_year  <-   0.82
U_Recurrence_year_2_and_onward <-   0.85
U_Palliative <-  0.82


## Str3 - Cost NCRLM

C_waiting <-
9119*4*(1+inflation_rate)^(2022-2018)*Exchange_rate_EUR_to_NOK
C_str3_1st_transplantation <-
98047*(1+inflation_rate)^(2022-2018)*Exchange_rate_EUR_to_NOK +
                              25682*2*(1+inflation_rate)^(2022-2019)*Exchange_rate_EU
C_str3_2nd_transplantation <-
25682*(1+inflation_rate)^(2022-2019)*Exchange_rate_EUR_to_NOK
```

```
### Module II. Functions  ------------------------
------------------------ ------------------------ ------------------------
--------------------
getwd()
## 2.1 continuous discount for continuous cost
#> t0: time model start
#> t1: time the cost start
#> t2: time the cost end
#> Thus, cost is accumulated as cost/year * (t2-t1) then this amount is
continuously
#> discounted to t1 then discretely discounted to t0

fn_discount_continuous <- function(amount, time_of_discount = NULL,
cost_start = NULL, cost_end = NULL, rate = discount_rate, timefactor = 1)
{
  if( !is.null(time_of_discount) & !is.null(start)) {
    ## we discount continuous cost/QALY to the time its start-> then
discretely discount it back to t0 - the
    #> model starting time.
    # amount is value per unit of time, here, yearly
    disc   <- amount/(rate)*(exp(-rate*cost_start)-exp(-rate*cost_end))/
(1+rate)^(cost_start - time_of_discount);
  } else {
    stop('Time argument(s) not specified correctly for
fn_discount_continuous() function')
  }
  out <- c(Discounted = disc)
  return(out)
}


## 2.2 Discrete discount for one-off cost
#>Simple discrete discount it back to t0

fn_discount_discrete <- function(amount, time_of_event = NULL,rate =
discount_rate) {
  if( !is.null(start)) {
    disc   <- amount/(1+rate)^(time_of_event);
  } else {
    stop('Time argument(s) not specified correctly for
fn_discount_discrete() function')
  }
  out <- c(Discounted = disc)
  return(out)
}

## 2.3 remove queue funcion for strategy 2

remove_queue = function(patient, organ) {
  a = c('organ_a', "organ_b", 'organ_ab', "organ_o")
  a = a[a != organ]
  paste0(patient, a)
}


## 2.4 NCRLM strategies 3 functions
```

```
#> drop out function
fn_drop_out_NCRLM_str3 <- function() {

  out <- if(runif(1) < p_drop_out_NCRLM_str3) {1} else {0}

  return(out)

}

#> die in surgery function

fn_die_surgery_NCRLM_str3 <- function() {

  out <- if(runif(1) < p_die_surgery_NCRLM_str3) {1} else {0}

  return(out)

}

fn_die_surgery_donor <- function() {

  out <- if(runif(1) < p_die_surgery_donor) {1} else {0}

  return(out)

}

#> retransplantation functions
fn_re_transplantation_NCRLM_str3 <- function() {

  out <- if(runif(1) < p_retranplantation_NCRLM_str3) {1} else {0}

  return(out)

}

fn_die_surgery_NCRLM_str3_retrans <- function() {

  out <- if(runif(1) < p_die_surgery_NCRLM_str3_retrans) {1} else {0}

  return(out)

}
```

```
### Module III. Trajectory ------------------------
------------------------ ------------------------ ------------------------
--------------------
getwd()
## 3.1 Strategy 1 - Palliative care only for NCRLM Patients
#################

#> Trajectory for end of cancer
Str1_NCRLM_trj_main = function(sim){

  Str1_NCRLM_trj_end =  trajectory() %>%
    set_attribute(
      keys   = c("TimeOfDeath","TimeToDeath"),
      values = function() c(now(sim), now(sim) - get_attribute(sim,
"TimeOfReferral"))
    )

  Str1_NCRLM_trj_main <- trajectory() %>%

    # Referral:
    # - recording the time of referral
    # - setting the moment at which the individual should be transferred
to the Str1_NCRLM_trj_end
    #   trajectory, regardless of where in the pathway they are at that
moment, based on the
    #   background mortality, using the renege_in() function.
    set_attribute(keys = "TimeOfReferral", values = function() now(.env =
sim)) %>%
    renege_in(t = function()rgompertz(1, d_NCRLM_death_shape_pal,
d_NCRLM_death_rate_pal), out = Str1_NCRLM_trj_end) %>%
    seize(resource = "Palliative") %>%
    wait()

  Str1_NCRLM_trj_main
}


####### 3.2 Strategy 2 - deceased donor for NCRLM patients - queuing with
Status Quo patients####

## 3.2.1 Liver signal trajectory------------------------

organ_signal = trajectory() %>%
  set_attribute(
    keys   = c("Time organ arrive"),
    values = function()now(sim)) %>%
  send(signals = function() {
    x = get_attribute(sim, "organ_blood_type")
    case_when(x == 1 ~ "organ_a",
              x == 2 ~ "organ_b",
              x == 3 ~ "organ_ab",
              x == 4 ~ "organ_o")})

### 3.2.1a Remove "clone" in other queue when 1 clone goes to surgery

remove_queue = function(patient, organ) {
  a = c('organ_a', "organ_b", 'organ_ab', "organ_o")
```

```
  a = a[a != organ]
  paste0(patient, a)
}


### 3.2.2 NCRLM patient trajectory--------------
# when patients go to death, memorize their time of death and time to
death
end_trj = trajectory("end_trajectory") %>%
  set_attribute(
    keys   = c("TimeOfDeath","TimeToDeath"),
    values = function() c(now(sim), now(sim) - get_attribute(sim,
"TimeOfReferral"))
  )

## Trajectory for NCRLM queue

organ_queue_trj = function(organ){
  trajectory() %>%
    # set death time of normal patients (can customize for different blood
types)
    ##FIXME how to set death time ?
    send(signals = function() paste0(get_name(sim), organ),
         delay = function() get_attribute(sim, "ncrlm_mortality")) %>%
    renege_if(signal = function() paste0(get_name(sim), organ),
              out = trajectory() %>%
                #leave(prob =1)
                join(end_trj)
    ) %>%
    seize(paste0("queue_", organ), 1) %>%
    trap(organ) %>%
    wait() %>%
    untrap(organ) %>%
    # send signal to remove "clone" in others queue
    send(function() remove_queue(patient = get_name(sim), organ)) %>%
    # invalidate death time before surgery
    renege_abort() %>%
    # surgery state

    branch(
      #FIXME prob fail/succeed/ death surgery here
      option = function() sample(c(1,2,3), size = 1, prob =
c(str2_p_deceased_retrans_NCRLM,
                                                              1 -
str2_p_deceased_retrans_NCRLM - str2_p_deceased_die,
                                                              str2_p_deceased_die)),
      continue = c(TRUE, TRUE, TRUE),

      #FIXME fail: rollback to queue, only one times?
      trajectory() %>%
        release(paste0("queue_", organ),1) %>%
        #FIXME death within 90 day distribution
        renege_in(t = function() rgompertz(1, str2_retrans_death_shape,
str2_retrans_death_rate),
                  out = end_trj) %>%

        set_attribute("surgery_time_fail", values = function() now(sim))
%>%
```

```
        set_prioritization(values = c(1, NA, NA)) %>%
        # Rollback to trap, second times came here will go next
        rollback(11, times = 1),

      # succeed:
      trajectory() %>%
        release(paste0("queue_", organ),1) %>%
        set_attribute(keys = "surgery_time_succeed", values = function()
now(sim)) %>%
        #FIXME update new time to death
        timeout(function()
        {
          x = rgompertz(1,d_surival_shape_NCRLM_post_sur_str2
,d_surival_rate_NCRLM_post_sur_str2)
          if (x >= 10) get_attribute(sim, "bg_mortality") else x

        }
          ),

      # death while surgery
      trajectory() %>%
        set_attribute(keys = "die in surgery", values = 1) %>%
        release(paste0("queue_", organ), 1)
    )


}

multiple_queue = function(organ){
  n = length(organ)
  list_trajectory = map(organ, ~ organ_queue_trj(.x))
  trajectory() %>%
    clone(n = n,list_trajectory)
}

# NCRLM patient trajectory

NCRLM_patient = trajectory("case2") %>%
  # put blood type here

  # set_priority for several patient base on probability
  branch(option = function() sample(c(0,1), size = 1,
                                    prob = c(1-p_priority_NCRLM_str2,
p_priority_NCRLM_str2)),
         continue = TRUE,
         trajectory() %>%
           set_prioritization(values = c(1,NA,NA))) %>%

  set_attribute(keys = "TimeOfReferral", values = function() now(.env =
sim)) %>%
  #FIXME add leave() p_drop_NCRLM_str2 for drop out patient, live for a
while and death
  #FIXME here the drop and the survival if retrans is kept as same as
Status Quo patients


  leave(prob = p_drop_NCRLM_str2, out = trajectory() %>%
```

```
          set_attribute(keys = "drop_out", values = 1) %>%
          timeout(t = function() get_attribute(sim, "ncrlm_mortality"))
%>%
          join(end_trj)) %>%
  #FIXME dua vao dataframe
  ## depends on blood type, queue in different queue, once match in a
queue, release other queue

  branch(option = function() get_attribute(sim, "blood_type"),
         continue = rep(TRUE,4),
         # blood type a
         multiple_queue(c("organ_a", "organ_o")),
         # type b
         multiple_queue(c("organ_b","organ_o")),
         # type ab
         multiple_queue(c("organ_a", "organ_b", "organ_ab", "organ_o")),
         # type o
         multiple_queue('organ_o')
  ) %>%
  join(end_trj)


###3.2.3 Status quo patient trajectory ----------------------
## Status Quo patients queue for organ



SQ_organ_queue_trj = function(organ){

  trajectory() %>%
    # set death time of normal patients (can customize for different blood
types)
    send(signals = function() paste0(get_name(sim), organ),
         delay = function() get_attribute(sim, "sq_mortality")) %>%
    renege_if(signal = function() paste0(get_name(sim), organ),
              out = trajectory() %>%
                join(end_trj)
    ) %>%
    seize(paste0("queue_", organ), 1) %>%
    trap(organ) %>%
    wait() %>%
    untrap(organ) %>%
    send(function() remove_queue(patient = get_name(sim), organ)) %>%

    # turn off the death clock if patients can survive until the surgery
    renege_abort() %>%

    # surgery state
    branch(

      # prob fail/succeed/ death surgery here
      option = function() sample(c(1,2,3), size = 1, prob =
c(str2_p_deceased_retrans,
                                                              1 -
str2_p_deceased_retrans - str2_p_deceased_die,
                                                              str2_p_deceased_die)),
      continue = c(TRUE, TRUE, TRUE),
```

```
      #FIXME fail: set new survival time, rollback to queue, only one
times?
      trajectory() %>%
        release(paste0("queue_", organ),1) %>%
        #FIXME death within 90 day rgompezt distribution
        renege_in(t = function() rgompertz(1, str2_retrans_death_shape,
str2_retrans_death_rate), out = end_trj) %>%

        set_attribute("surgery_time_fail", values = function() now(sim))
%>%

        set_prioritization(values = c(1, NA, NA)) %>%

        # Rollback to trap, second times came here will go to success
automatically
        rollback(11, times = 1),

      # succeed:
      trajectory() %>%
        release(paste0("queue_", organ),1) %>%
        set_attribute(keys = "surgery_time_succeed", values = function()
now(sim)) %>%
        #FIXME update new time to death
        timeout(function()
          {x =  fn_rweibull(1, get_SQ_data(get_attribute(sim, "SQ_type"),
"death_shape2"),
                            get_SQ_data(get_attribute(sim, "SQ_type"),
"death_scale2"))
          if( x >=10 ) get_attribute(sim, "bg_mortality") else x
        }
        ),
      # death while surgery
      trajectory() %>%
        set_attribute(keys = "die in surgery", values = 1) %>%
        release(paste0("queue_", organ), 1)
    )


}

SQ_multiple_queue = function(organ){
  n = length(organ)
  list_trajectory = map(organ, ~ SQ_organ_queue_trj(.x))
  trajectory() %>%
    clone(n = n,list_trajectory)
}



# Status Quo patients main trajectory

Status_Quo_patient = trajectory("case2") %>%

  # set_priority for several patient base on probability

  branch(option = function() sample(c(0,1), size = 1, prob = c(1-
get_SQ_data(get_attribute(sim, "SQ_type"), "priority_rate"),
```

```
                                                                    get_SQ_data(get_attribut
"SQ_type"), "priority_rate"))),
         continue = TRUE,
         trajectory() %>%
           set_prioritization(values = c(1,NA,NA))) %>%


  set_attribute(keys = "TimeOfReferral", values = function() now(.env =
sim)) %>%

  # branch: waiting for matching organ or drop out
  leave(prob = function() get_SQ_data(get_attribute(sim, 'SQ_type'),
"drop_rate"), out = trajectory() %>%
          set_attribute(keys = "drop_out", values = 1) %>%
          timeout(function() get_attribute(sim, "sq_dropout_mortality"))
%>%
          join(end_trj)) %>%

  ##if not drop out, patient receive a death-clock before surgery, if they
can survive until surgery, they can have it

  ## depends on blood, people will que in different que foe organ type
  # blood type a
  branch(option = function() get_attribute(sim, "blood_type"),
         continue = rep(TRUE,4),
         # blood type a
         SQ_multiple_queue(c("organ_a", "organ_o")),
         # type b
         SQ_multiple_queue(c("organ_b","organ_o")),
         # type ab
         SQ_multiple_queue(c("organ_a", "organ_b", "organ_ab",
"organ_o")),
         # type o
         SQ_multiple_queue('organ_o')) %>%
  join(end_trj)



####### 3.3  Strategy 3 - live donor for NCRLM Patients #################

### Trajectory of NCRLM patients
------------------------------------------------------------------------------

str3_trj_end_CRLM <- trajectory() %>%
  set_attribute(
    keys   = c("TimeOfDeath","TimeToDeath"),
    values = function() c(now(sim), now(sim) - get_attribute(sim,
"TimeOfReferral"))
  )

##4.2 Trajectory if patient go for re transplantation (skip until read
4.3)
str3_trj_retransplantation <- trajectory() %>%
  set_attribute(
    keys = c("Retransplantation"),
    values = 1
  )%>%
  renege_abort() %>%
```

```
  renege_in(rgompertz(1, d_death_shape_retrans_NCRML_Str3,
d_death_rate_retrans_NCRML_Str3), out = str3_trj_end_CRLM) %>%
  #waiting for surgery
  seize(resource = "Waiting_retrans") %>%
  timeout(task = function() rweibull(1,
d_waiting_shape_retrans_NCRML_Str3, d_waiting_scale_retrans_NCRML_Str3))
%>%
  release(resource = "Waiting_retrans") %>%

  #taking second surgery, death or survive only

  seize(resource = "Surgery_retrans") %>%
  set_attribute(keys = "TimeOf2ndSurgery", values = function() now(.env =
sim)) %>%
  release(resource = "Surgery_retrans") %>%
  # Surgery outcome: die/survive
  branch(option = function() fn_die_surgery_NCRLM_str3_retrans(), continue
= c(F),

         # 1) die
         trajectory() %>%
           set_attribute(
             keys = "die in 2nd surgery",
             values = 1
           ) %>%
           join(str3_trj_end_CRLM)
  ) %>%
  renege_abort() %>%
  ##set in new survival of successful surgery
  renege_in(t=function()
  {
    x = rgompertz(1,d_surival_shape_NCRLM_post_sur_str2
,d_surival_rate_NCRLM_post_sur_str2)
    if (x >= 10) get_attribute(sim, "bg_mortality") else x

  }

              #rweibull(1,d_surival_shape_NCRLM_post_sur,
                       #d_surival_scale_NCRLM_post_sur)
    ,out = str3_trj_end_CRLM) %>%
  wait()

##4.3 The main trajectory of patients from beginning.

# The main trajectory implements the general pathway of patients.

str3_trj_main_CRLM <- trajectory() %>%

  # Referral or starting time with Palliative survival.

  set_attribute(keys = "TimeOfReferral", values = function() now(.env =
sim)) %>%
  renege_in(t = function()rgompertz(1, d_NCRLM_death_shape_pal,
d_NCRLM_death_rate_pal),
            out = str3_trj_end_CRLM) %>%


  ##Either dropout or continue waiting for surgery
```

```
  branch(option = function() fn_drop_out_NCRLM_str3(), continue = c(F),

         # 0) drop_out
         trajectory() %>%
           set_attribute(keys = "Drop Out", values = 1) %>%
           wait()
         ##NOTE + CHECKME
         ##Here can turnoff the general clock and set in new survival time
for drop out patients only
  ) %>%
  # Other go Waiting for surgery with time in weibull distribution

  seize(resource = "Waiting") %>%
  timeout(task = function() rgompertz(1, d_sur_waiting_shape_NCRML_Str3,
d_sur_waiting_scale_NCRML_Str3)) %>%
  release(resource = "Waiting") %>%

  # Surgery:
  # - also record the time of surgery to calculate the time between
referral and surgery, as
  #   well as to track whether the individual has received surgery
  seize(resource = "Surgery") %>%
  set_attribute(keys = "TimeOfSurgery", values = function() now(.env =
sim)) %>%
  release(resource = "Surgery") %>%

  branch(option = function() fn_die_surgery_NCRLM_str3(), continue = c(F),

         # 0) die
         trajectory() %>%
           set_attribute(
             keys = "die in surgery",
             values = 1
           ) %>%
           join(str3_trj_end_CRLM)
  ) %>%
  branch(option = function() fn_re_transplantation_NCRLM_str3(), continue
= c(F),

         # 0) retransplantation
         trajectory() %>%
           join(str3_trj_retransplantation)
         ##NOTE
         ##here we define a new pathway of patient with re transplantation
- there is no queue so
         ##no need to roll them back.,
  ) %>%
  # New survival time
  renege_abort() %>%
  renege_in(t=function()
  {x = rgompertz(1,d_surival_shape_NCRLM_post_sur_str2
,d_surival_rate_NCRLM_post_sur_str2)
    if (x >= 10) get_attribute(sim, "bg_mortality") else x

  },
  out = str3_trj_end_CRLM) %>%
  ##FIXME - double check with the survival curve after 10 years?
```

```
  wait()

### Trajectory Donor normal
life-------------------------------------------------------------------------------

#> Donor trajectory in normal case - recording time of referral, time of
death, time to death as background mortality.
str3_donor_trj_end_bg =  trajectory() %>%
  set_attribute(
    keys   = c("TimeOfDeath","TimeToDeath"),
    values = function() c(now(sim), now(sim) - get_attribute(sim,
"TimeOfReferral"))
  ) %>%
  set_attribute(keys = "Death of background", values = 1)


str3_donor_trj_main_bg <- trajectory() %>%

  set_attribute(keys = "TimeOfReferral", values = function() now(.env =
sim)) %>%
  renege_in(t = function() get_attribute(sim, "renege_in_time"),
            out = str3_donor_trj_end_bg ) %>%
  wait()

###Trajectory Donor in case of
donation---------------------------------------------------------------------------

#> Donor trajectory in donation case

str3_donor_trj_end_sur =  trajectory() %>%
  set_attribute(
    keys   = c("TimeOfDeath","TimeToDeath"),
    values = function() c(now(sim), now(sim) - get_attribute(sim,
"TimeOfReferral"))
  )

str3_donor_trj_main_sur <- trajectory() %>%

  # Referral:
  # - recording the time of referral
  # - setting the moment at which the donor should be transferred to the
str3_donor_trj_end_bg
  #   trajectory, regardless of where in the pathway they are at that
moment, based on the
  #   background mortality, using the renege_in() function.
  set_attribute(keys = "TimeOfReferral", values = function() now(.env =
sim)) %>%
  renege_in(t = function() get_attribute(sim, "renege_in_time"),
            out = str3_donor_trj_end_bg ) %>%


  branch(option = function() fn_die_surgery_donor(), continue = c(F),

         # 0) die
         trajectory() %>%
           set_attribute(
             keys = "die in surgery",
             values = 1
```

```
        ) %>%
        join(str3_donor_trj_end_sur)
) %>%
wait()
```

```
###Module IV. Common Randome values####
getwd()
# I. Common random parameters ------------------------
------------------------ ------------------------ ------------------------

## Here we add background mortality according to age:


## 1.1 NCRLM_input_df ----------------------------------
##FIXME randomly assign age from 40~71 (median 59.1) - from age get
background mortality - if survival after surgery > 10 years
## then they will die with background mortality (fix easily with table
data analysis)

##FIXME try to keep the death of palliative care the same across the three
strategies - for drop out patients,
## they will die with the pre-determined time.

NCRLM_arrival_time <- cumsum(rexp(n = 300, a_NCRLM_rate_arrival))
NCRLM_arrival_time <- NCRLM_arrival_time[NCRLM_arrival_time < 20]
NCRLM_input_df <- tibble(time = NCRLM_arrival_time,
                         blood_type = sample(c(1,2,3,4), size =
length(NCRLM_arrival_time),
                                             replace = TRUE,
                                             prob = c(p_NCRLM_bl_A,
                                                      p_NCRLM_bl_B,
                                                      p_NCRLM_bl_AB,
                                                      p_NCRLM_bl_O)),
                         preemptible = 1,
                         ncrlm_mortality =
rgompertz(length(NCRLM_arrival_time), d_NCRLM_death_shape_pal,
d_NCRLM_death_rate_pal),
                         # bg_mortality: bg_mortality for after-surgery
patients live longer than 10 years.
                         ) %>%
  mutate(age = 59) %>%
  left_join(Background_df) %>%
  mutate(.keep= "unused", bg_mortality = rgompertz(n(), death_shape_bg,
death_rate_bg))


## 1.2  SQ_input_df ----------------------------------
SQ_arrival_time <- cumsum(rexp(n = 4500, SQ_Patients_arrival_rate))
SQ_arrival_time <- SQ_arrival_time[SQ_arrival_time < 20]

SQ_input_df <- tibble(
  time = SQ_arrival_time,
  blood_type = sample(
    c(1, 2, 3, 4),
    size = length(SQ_arrival_time),
    replace = TRUE,
    prob = c(
      p_Status_Quo_bl_A,
      p_Status_Quo_bl_B,
      p_Status_Quo_bl_AB,
      p_Status_Quo_bl_O
    )
  ),
```

```
  type = sample(
    1:6,
    size = length(SQ_arrival_time),
    replace = TRUE,
    prob = c(
      p_Status_Quo_HCC_1,
      p_Status_Quo_PBC_2,
      p_Status_Quo_PSC_3,
      p_Status_Quo_AC_4,
      p_Status_Quo_ALF_5,
      p_Status_Quo_Other_6
    )
  ),
  preemptible = 1
) %>%
  mutate(age = case_when(type %in% c(2,4,6) ~ 57,
                         type == 1 ~ 61,
                         TRUE ~ 43)) %>%
  left_join(SQ_df, by = "type") %>%
  left_join(Background_df, by= "age") %>%
  mutate(sq_mortality = fn_rweibull(n(), death_shape2, death_scale2),
         sq_dropout_mortality = rgompertz(n(), death_shape1, death_rate1),
         bg_mortality = rgompertz(n(), death_shape_bg, death_rate_bg),
         SQ_type = type)



## 1.3 Deceased Liver input ------------

# Common random way of getting liver arrival data

organ_arrival_time = cumsum(rexp(7500,rate = d_liver_arrival_rate))
organ_arrival_time = organ_arrival_time[organ_arrival_time < 30]

organ_input_df = tibble(
  time = organ_arrival_time,
  organ_blood_type = sample(1:4,
                            size = length(organ_arrival_time),
                            replace = TRUE,
                            prob=c(p_Liver_bl_A,
                                   p_Liver_bl_B,
                                   p_Liver_bl_AB,
                                   p_Liver_bl_O ))
)
```

```
## Module V. Waiting time to define Burn-in period
#> run the simulation
str2b_waiting = local({
  library(here)
  source(here("Final_I.CommonValues2103.R"), local= TRUE)
  source(here("Final_II.Functions2103.R"), local =TRUE)
  source(here("FInal_III.Trajectory2103.R"), local =TRUE)
  source(here('Final_IV.CommonRandomValues2103.R'), local = TRUE)
  time_horizon <- 60

  str2b = function(i){

    source(here('Final_IV.CommonRandomValues2103.R'), local = TRUE)
    sim <<- simmer(paste0(i)) %>%
      # set capacity = 1 because when organ available (signal), only 1
patient will be release
      # not set preemptive = TRUE
      add_resource("queue_organ_a", capacity = 1, preemptive = TRUE) %>%
      add_resource("queue_organ_b", capacity = 1, preemptive = TRUE) %>%
      add_resource("queue_organ_ab", capacity = 1, preemptive = TRUE) %>%
      add_resource("queue_organ_o", capacity = 1, preemptive = TRUE) %>%
      ## Deceased liver transplant, all organ signal to one generator
      add_dataframe("organ", mon = 2,  organ_signal,
                    data = organ_input_df,
                    time = "absolute",
                    col_attributes = c("organ_blood_type")) %>%
      ## Other patients
      add_dataframe("SQ_patient", mon = 2, Status_Quo_patient,
                    data = SQ_input_df,
                    time = "absolute",
                    col_preemptible = "preemptible",
                    col_attributes = c("blood_type", "SQ_type",
"sq_mortality","bg_mortality",
                                       "sq_dropout_mortality")) %>%
      ## NCRLM patients
      add_dataframe("NCRLMpatient", mon = 2,  NCRLM_patient, data =
NCRLM_input_df,
                    col_preemptible = "preemptible",time = "absolute",
                    col_attributes = c("blood_type", "ncrlm_mortality",
"bg_mortality"))

    sim %>% reset() %>% simmer::run(until = time_horizon)
    #> Extract result for both patients--------

    df_str2B_both <- get_mon_attributes(sim) %>%
      tibble() %>%
      group_by(name, key) %>%
      mutate( a = row_number()) %>%
      ungroup() %>%
      mutate(key = case_when(key == "surgery_time_fail"~ paste0(key, a),
                             TRUE ~ key)) %>%
      pivot_wider(id_cols = name, names_from = key, values_fn = max)

    # get out NCRLM patients data frame and do data analysis
    df_str2B_NCRLM_both <- df_str2B_both %>% filter(str_detect(name,
"NCRLM"))

    str2b_NCRLM_wait = df_str2B_NCRLM_both %>%
```

```
      select(TimeOfReferral, surgery_time_succeed, surgery_time_fail1,
TimeOfDeath, blood_type) %>%
      # remove people who die before surgery
      filter(!if_all(c(surgery_time_succeed, surgery_time_fail1),.fns =
is.na)) %>%
      transmute(time_in = TimeOfReferral,
                time_out = case_when(!is.na(surgery_time_fail1) ~
surgery_time_fail1,
                                     is.na(surgery_time_succeed) ~
TimeOfDeath,
                                     TRUE ~ surgery_time_succeed),
                blood_type)

    str2b_NCRLM_wait
  }
})

  # run simulation 100 times (parallel)
  # input mc.cores =  how many core your pc has

  library(parallel)

str2b_NCRLM_wait = mclapply(1:1000, function(i) str2b_waiting(i), mc.cores
= 1)
saveRDS(str2b_NCRLM_wait, here::here("check-data/wait-time-1000-
runs.rds"))

# alternative if parallel is not working
#str2b_NCRLM_wait = map(1:10, ~str2b(.x))


# bind multiple list-dataframe to one data frame

str2b_NCRLM_wait = data.table::rbindlist(str2b_NCRLM_wait,idcol = 'sim')
%>%
  tibble()

# average waiting time

str2b_NCRLM_wait_summarize = str2b_NCRLM_wait %>%
  group_by(sim, year = floor(time_in) ) %>%
  summarise(wait_time = mean(time_out - time_in))

str2b_NCRLM_wait_summarize %>%
  ggplot(aes(year, wait_time))+
  geom_point(size = 0.8)+
  geom_smooth(method = "loess")+
  labs(y = "waiting time by years",
       x = "simulation time")

ggsave(here::here("check-data/wait-time.jpg"))

### wait time  by blood type
blood_type = c("A", "B", "AB", "O")

 str2b_NCRLM_wait %>%
   mutate(wait_time = time_out - time_in) %>%
   mutate(blood_type = .env$blood_type[blood_type]) %>%
```

```
  group_by(sim, blood_type,year = floor(time_in)) %>%
  summarise(wait_time = mean(wait_time)) %>%
  ggplot(aes(year, wait_time))+
  geom_point()+
  geom_smooth()+
  facet_wrap(~blood_type)

ggsave(here::here("check-data/wait-time-by-blood-type.jpg"))

## average waiting time
avg_str2b_NCRLM_wait = function(start_year){

  str2b_NCRLM_wait %>%
    filter(time_in > start_year, time_in < 10 + start_year ) %>%
    mutate(wait_time = time_out -time_in) %>%
    group_by(blood_type) %>%
    # compute waitime median, 10% and 90$ quantile
    summarise(median_wait_time = median(wait_time),
              mean_wait_time = mean(wait_time),
              first_quantile_wait_time = quantile(wait_time, 0.25),
              third_quantile_wait_time = quantile(wait_time, 0.75),
              min_wait_time = min(wait_time),
              max_wait_time = max(wait_time)) %>%
  # convert year to days
  mutate(across(-blood_type, .fns = ~.x * 365.25))
}

avg_str2b_NCRLM_wait(7)
```

```
##Module VI. To define the number of simulation

library(parallel)
library(tidyverse)

str2b = local({
  library(here)
  source(here("Final_I.CommonValues2103.R"), local= TRUE)
  source(here("FInal_III.Trajectory2103.R"), local =TRUE)
  source(here('Final_IV.CommonRandomValues2103.R'), local = TRUE)
  time_horizon <- 60

  # let global assigment stop here
  sim = NA


  f = function(i){
    message(paste0("run sim", i))
    # link to the file of commonrandom in your pcs
    source(here('Final_IV.CommonRandomValues2103.R'),local = TRUE)

    sim <<- simmer()
      # set capacity = 1 because when organ available (signal), only 1
patient will be release
      # not set preemptive = TRUE
    sim %>%
      add_resource("queue_organ_a", capacity = 1, preemptive = TRUE) %>%
      add_resource("queue_organ_b", capacity = 1, preemptive = TRUE) %>%
      add_resource("queue_organ_ab", capacity = 1, preemptive = TRUE) %>%
      add_resource("queue_organ_o", capacity = 1, preemptive = TRUE) %>%
      ## Deceased liver transplant, all organ signal to one generator
      add_dataframe("organ", mon = 2,  organ_signal,
                    data = organ_input_df,
                    time = "absolute",
                    col_attributes = c("organ_blood_type")) %>%
      ## Other patients
      add_dataframe("SQ_patient", mon = 2, Status_Quo_patient,
                    data = SQ_input_df,
                    time = "absolute",
                    col_preemptible = "preemptible",
                    col_attributes = c("blood_type", "SQ_type",
"sq_mortality","bg_mortality",
                                       "sq_dropout_mortality")) %>%
      ## NCRLM patients
      add_dataframe("NCRLMpatient", mon = 2,  NCRLM_patient, data =
NCRLM_input_df,
                    col_preemptible = "preemptible",time = "absolute",
                    col_attributes = c("blood_type", "ncrlm_mortality",
"bg_mortality"))

    sim %>% reset() %>% simmer::run(until = time_horizon)
    #> Extract result for both patients--------

    df_str2B_both <- get_mon_attributes(sim) %>%
      tibble() %>%
      group_by(name, key) %>%
      mutate( a = row_number()) %>%
      ungroup() %>%
```

```
      mutate(key = case_when(key == "surgery_time_fail"~ paste0(key, a),
                             TRUE ~ key)) %>%
      pivot_wider(id_cols = name, names_from = key, values_fn = max)

    # get out NCRLM patients data frame and do data analysis
    df_str2B_SQ <- df_str2B_both %>% filter(str_detect(name, "SQ"))

    return(sum(df_str2B_SQ$TimeToDeath, na.rm= TRUE))
  }
})
str2b_SQ_LY = mclapply(1:1500, function(i) str2b(i), mc.cores = 1)
### average time to death of SQ ~ n runs
str2b_SQ_LY = tibble(n_runs = seq_along(str2b_SQ_LY),
                     avg_LY = cummean(str2b_SQ_LY))
saveRDS(str2b_SQ_LY, here("check-data/str2b_SQ_LY.rds"))
str2b_SQ_LY |>
  ggplot(aes(n_runs, avg_LY))+
  geom_point()+
  geom_smooth(se = FALSE)

str2a = local({
  library(here)
  source(here("Final_I.CommonValues2103.R"), local= TRUE)
  source(here("FInal_III.Trajectory2103.R"), local =TRUE)
  source(here('Final_IV.CommonRandomValues2103.R'), local = TRUE)
  time_horizon <- 60

  # let global assigment stop here
  sim = NA


  f = function(i){
    message(paste0("run sim", i))
    # link to the file of commonrandom in your pcs
    source(here('Final_IV.CommonRandomValues2103.R'),local = TRUE)

    sim <<- simmer()
    sim %>%
      # set capacity = 1 because when organ available (signal), only 1
patient will be release
      # not set preemptive = TRUE
      add_resource("queue_organ_a", capacity = 1, preemptive = TRUE) %>%
      add_resource("queue_organ_b", capacity = 1, preemptive = TRUE) %>%
      add_resource("queue_organ_ab", capacity = 1, preemptive = TRUE) %>%
      add_resource("queue_organ_o", capacity = 1, preemptive = TRUE) %>%
      ## Deceased liver transplant, all organ signal to one generator
      add_dataframe("organ", mon = 2,  organ_signal,
                    data = organ_input_df,
                    time = "absolute",
                    col_attributes = c("organ_blood_type")) %>%
      ## Other patients
      add_dataframe("SQ_patient", mon = 2, Status_Quo_patient,
                    data = SQ_input_df,
                    time = "absolute",
                    col_preemptible = "preemptible",
                    col_attributes = c("blood_type", "SQ_type",
"sq_mortality","bg_mortality",
                                       "sq_dropout_mortality"))
```

```
    #> Run simulation
    #>
    time_horizon <- 60
    sim %>% reset() %>%  run(until = time_horizon) %>% invisible()


    ## get data for SQ only case and perform data analysis.

    df_str2A_SQ = get_mon_attributes(sim) %>%
      tibble() %>%
      group_by(name, key) %>%
      mutate( a = row_number()) %>%
      ungroup() %>%
      mutate(key = case_when(key == "surgery_time_fail"~ paste0(key, a),
                             TRUE ~ key)) %>%
      pivot_wider(id_cols = name, names_from = key, values_fn = max)


    df_str2A_SQ <- df_str2A_SQ %>% filter(str_detect(name, "SQ"))

     return(sum(df_str2A_SQ$TimeToDeath, na.rm = TRUE))
  }
})

str2a_SQ_LY = mclapply(1:1500, function(i) str2b(i), mc.cores = 1)
### average time to death of SQ ~ n runs
tibble(x=seq_along(str2a_SQ_LY),
       y = cummean(str2a_SQ_LY)) |>
  ggplot(aes(x,y))+
  geom_point()+
  geom_smooth(se = FALSE)

ggsave(here("check-data/str2a_SQ_LY_1500runs.jpg"))
```

```
##Module VII. Run simulation accross three strategies 1000 times

sim_str1 = local({
  library(here)
  source(here("Final_I.CommonValues2103.R"), local= TRUE)
  source(here("FInal_III.Trajectory2103.R"), local =TRUE)
  source(here('Final_II.Functions2103.R'), local = TRUE)
  time_horizon <- 60
  sim_str1 = NA

  f = function(year){
    source(here('Final_IV.CommonRandomValues2103.R'), local = TRUE)

    sim_str1 <<- simmer() %>%
      add_resource(name = "Palliative", capacity = Inf) %>%
      ## NCRLM patients
      add_dataframe("NCRLMpatient", mon = 2,
Str1_NCRLM_trj_main(sim_str1), data = NCRLM_input_df,
                    col_preemptible = "preemptible",time = "absolute",
                    col_attributes = c("blood_type"))


    # Running the simulation until all events have occurred (no right
censored)
    sim_str1 %>% reset() %>% run(until=30);

    df_str1 <- get_mon_attributes(sim_str1) %>%
      pivot_wider(id_cols = name, names_from = key, values_from = value)

    df_str1 <- df_str1 %>% filter(TimeOfReferral>= year, TimeOfReferral <=
year +10)

    df_str1 <- df_str1 %>%
      ## compute output
      mutate(LY = if_else(is.na(TimeToDeath), 30- TimeOfReferral,
TimeToDeath)) %>%
      mutate(
        Cost_Best_supportive = C_best_supportive_care * (if_else(LY >=
0.25,0.25,LY)),
        dCost_Best_supportive= fn_discount_discrete(Cost_Best_supportive,
time_of_event = TimeOfDeath - time_of_discounting),
        Cost_0_6_Pal_Care = C_Prior_Palliative_chemotherapy_month_0_6 *
(if_else(LY >= 0.75,0.5,

if_else(LY<=0.25,0,LY - 0.25))),
        dCost_0_6_Pal_Care =
fn_discount_continuous(C_Prior_Palliative_chemotherapy_month_0_6,time_of_discount
= time_of_discounting,
                                                    cost_start =
TimeOfReferral, cost_end = (if_else(LY >= 0.75, TimeOfReferral + 0.5,

if_else(LY<=0.25,TimeOfReferral,TimeOfDeath - 0.25)))),
        Cost_7_12_Pal_Care = C_Prior_Palliative_chemotherapy_month_7_12 *
(if_else(LY >= 1.25,0.5,

if_else(LY<=0.5,0,LY - 0.75))),
```

```
        dCost_7_12_Pal_Care =
fn_discount_continuous(C_Prior_Palliative_chemotherapy_month_7_12
,time_of_discount = time_of_discounting,
                                                        cost_start =
TimeOfReferral+0.5, cost_end = (if_else(LY >= 1.25, TimeOfReferral + 1,

if_else(LY<=0.5,TimeOfReferral+0.5,TimeOfDeath - 0.25)))),
        Cost_13_Pal_Care = C_Prior_Palliative_chemotherapy_month_13_onward
* (if_else(LY >= 1.25, LY - 0.25,0)),
        dCost_13_Pal_Care =
fn_discount_continuous(C_Prior_Palliative_chemotherapy_month_13_onward,time_of_discount
=  time_of_discounting,
                                                      cost_start =
TimeOfReferral+1, cost_end = (if_else(LY >= 1.25, TimeOfDeath - 0.25,
TimeOfReferral+1))),
        dLY = fn_discount_continuous(1,time_of_discount =
time_of_discounting,
                                     cost_start = TimeOfReferral,
                                     cost_end = TimeOfDeath),
        total_cost = Cost_Best_supportive + Cost_0_6_Pal_Care +
Cost_7_12_Pal_Care + Cost_13_Pal_Care,
        d_total_cost = dCost_Best_supportive + dCost_0_6_Pal_Care +
dCost_7_12_Pal_Care + dCost_13_Pal_Care
      ) %>%
      mutate(

        QALY = LY * u_NCRLM_palliative,
        dQALY = fn_discount_continuous(u_NCRLM_palliative,time_of_discount
= time_of_discounting,
                                       cost_start = TimeOfReferral,
cost_end = TimeOfDeath)
      )


    df_str1 %>%
      summarise(n_patients = n(), across(c(LY, dLY, total_cost:dQALY),
sum))

  }

})

## strategy 1 - run 1000

library(parallel)
sim_str1_rs = mclapply(1:1000, function(i) sim_str1(year = 7), mc.cores =
1)
options(scipen = 5)
sim_str1_rs = data.table::rbindlist(sim_str1_rs)
saveRDS(sim_str1_rs, file = here::here("run-and-result/sim_str1_rs.rds"))

Output_str1_avg<- sim_str1_rs %>%
  ## calculate mean
  summarise(across(c(n_patients,LY,dLY,total_cost,d_total_cost,QALY,dQALY),
mean, na.rm = TRUE))


str_2a = local({
```

```
  library(here)
  source(here("Final_I.CommonValues2103.R"), local= TRUE)
  source(here("FInal_III.Trajectory2103.R"), local =TRUE)
  source(here('Final_II.Functions2103.R'), local = TRUE)
  time_horizon <- 60
  sim = NA

  f = function(year){
    source(here('Final_IV.CommonRandomValues2103.R'), local = TRUE)
    sim <<- simmer()
    sim %>%
      # set capacity = 1 because when organ available (signal), only 1
patient will be release
      # not set preemptive = TRUE
      add_resource("queue_organ_a", capacity = 1, preemptive = TRUE) %>%
      add_resource("queue_organ_b", capacity = 1, preemptive = TRUE) %>%
      add_resource("queue_organ_ab", capacity = 1, preemptive = TRUE) %>%
      add_resource("queue_organ_o", capacity = 1, preemptive = TRUE) %>%
      ## Deceased liver transplant, all organ signal to one generator
      add_dataframe("organ", mon = 2,  organ_signal,
                    data = organ_input_df,
                    time = "absolute",
                    col_attributes = c("organ_blood_type")) %>%
      ## Other patients
      add_dataframe("SQ_patient", mon = 2, Status_Quo_patient,
                    data = SQ_input_df,
                    time = "absolute",
                    col_preemptible = "preemptible",
                    col_attributes = c("blood_type", "SQ_type",
"sq_mortality", "bg_mortality","sq_dropout_mortality"))

    # Run simulation
    sim %>% reset() %>%  simmer::run(until = time_horizon)


    # get data for SQ only case and perform data analysis.


    df_str2A_SQ_Only_All = get_mon_attributes(sim) %>%
      tibble() %>%
      group_by(name, key) %>%
      mutate( a = row_number()) %>%
      ungroup() %>%
      mutate(key = case_when(key == "surgery_time_fail"~ paste0(key, a),
                             TRUE ~ key)) %>%
      pivot_wider(id_cols = name, names_from = key, values_fn = max)


    df_str2A_SQ_Only_SQ <- df_str2A_SQ_Only_All %>%
filter(str_detect(name, "SQ")) %>%
      filter(TimeOfReferral>= year, TimeOfReferral <= year +10)
    #> Calculate Life year in time horizon
    df_str2A_SQ_Only_SQ <- df_str2A_SQ_Only_SQ %>%
      mutate(LY = if_else(is.na(TimeToDeath), time_horizon-
TimeOfReferral, TimeToDeath)) %>%
      mutate(LY_before_sur = if_else(is.na(surgery_time_succeed),
                                     if_else(is.na(TimeOfDeath),
```

```
                                          time_horizon -
TimeOfReferral,
                                          TimeOfDeath - TimeOfReferral
                                    ),
                                    surgery_time_succeed-TimeOfReferral))
%>%
      mutate(LY_after_sur = LY - LY_before_sur,
             dLY = fn_discount_continuous(1, time_of_discount =
time_of_discounting,
                                          cost_start = TimeOfReferral,
                                          cost_end = TimeOfReferral + LY)
      ) %>%
      mutate (waiting_time = case_when(!is.na(drop_out) ~ NA_real_,
                                       !is.na(surgery_time_fail1) ~
surgery_time_fail1 - TimeOfReferral,
                                       !is.na(surgery_time_succeed) ~
surgery_time_succeed - TimeOfReferral,
                                       TRUE ~ TimeOfDeath - TimeOfReferral
      ))



    #> Calculate cost and QALYs SQ patients in case 1 group only
    df_str2A_SQ_Only_SQ <- df_str2A_SQ_Only_SQ  %>%
      ##compute cost
      ##FIXME - case of die in surgery
      mutate(
        total_cost = SQ_c_before_trans_care * LY_before_sur +
SQ_c_survival_care * LY_after_sur +
          if_else(is.na(surgery_time_succeed),0,
                  if_else(is.na(surgery_time_fail1),
C_str2_1st_transplantation,
                          C_str2_1st_transplantation +
C_str2_2nd_transplantation)),
        d_total_cost = fn_discount_continuous(SQ_c_before_trans_care,
time_of_discount = time_of_discounting,
                                              cost_start = TimeOfReferral,
                                              cost_end = TimeOfReferral +
LY_before_sur) +
          if_else(is.na(surgery_time_succeed), 0,
fn_discount_continuous(SQ_c_survival_care, time_of_discount =
time_of_discounting,
                                                                    cost_start
= TimeOfReferral + LY_before_sur,
                                                                    cost_end
= TimeOfReferral + LY)) +
          if_else(is.na(surgery_time_succeed),0,
                  if_else(is.na(surgery_time_fail1),
                          fn_discount_discrete(C_str2_1st_transplantation,
                                               time_of_event =
surgery_time_succeed - time_of_discounting),

                          fn_discount_discrete(C_str2_1st_transplantation,
                                               time_of_event =
surgery_time_fail1 - time_of_discounting) +
                            fn_discount_discrete(C_str2_2nd_transplantation,
                                                 time_of_event =
surgery_time_succeed - time_of_discounting))),
```

```
        QALY = LY_before_sur * SQ_u_prior_surgery + LY_after_sur *
SQ_u_after_surgery,
        dQALY = fn_discount_continuous(SQ_u_prior_surgery,
time_of_discount = time_of_discounting,
                                       cost_start = TimeOfReferral,
                                       cost_end = TimeOfReferral +
LY_before_sur) +
          if_else(is.na(surgery_time_succeed),0,
fn_discount_continuous(SQ_u_after_surgery, time_of_discount =
time_of_discounting,
                                                                    cost_start
= TimeOfReferral + LY_before_sur,
                                                                    cost_end
= TimeOfReferral + LY))
      )


    df_str2A_SQ_Only_SQ %>%
      summarise(n_patients = n(), across(c(LY, dLY, waiting_time,
total_cost:dQALY), sum, na.rm = TRUE))
  }
})

## mclapply(1:n, ..) indicate the number of simulation

library(parallel)
sim_str2a_rs = mclapply(1:1000, function(i) str_2a(year = 7), mc.cores =
1)
options(scipen = 5)
sim_str2a_rs = data.table::rbindlist(sim_str2a_rs)
saveRDS(sim_str2a_rs, here::here("run-and-result/sim_str2a_rs.rds"))

Output_str2a_avg<- sim_str2a_rs %>%
  ## calculate mean
  summarise(across(c(n_patients,LY,dLY,waiting_time,
total_cost,d_total_cost,QALY,dQALY), mean, na.rm = TRUE))


str_2b = local({

  library(here)
  source(here("Final_I.CommonValues2103.R"), local= TRUE)
  source(here("FInal_III.Trajectory2103.R"), local =TRUE)
  source(here('Final_II.Functions2103.R'), local = TRUE)
  time_horizon <- 60

  f = function(year) {

    source(here('Final_IV.CommonRandomValues2103.R'), local = TRUE)
    sim <<- simmer()
    # set capacity = 1 because when organ available (signal), only 1
patient will be release
    # not set preemptive = TRUE
    sim %>%
      add_resource("queue_organ_a", capacity = 1, preemptive = TRUE) %>%
      add_resource("queue_organ_b", capacity = 1, preemptive = TRUE) %>%
      add_resource("queue_organ_ab", capacity = 1, preemptive = TRUE) %>%
```

```
      add_resource("queue_organ_o", capacity = 1, preemptive = TRUE) %>%
      ## Deceased liver transplant, all organ signal to one generator
      add_dataframe("organ", mon = 2,  organ_signal,
                    data = organ_input_df,
                    time = "absolute",
                    col_attributes = c("organ_blood_type")) %>%
      ## Other patients
      add_dataframe("SQ_patient", mon = 2, Status_Quo_patient,
                    data = SQ_input_df,
                    time = "absolute",
                    col_preemptible = "preemptible",
                    col_attributes = c("blood_type", "SQ_type",
"sq_mortality", "bg_mortality",
                                       "sq_dropout_mortality")) %>%
      ## NCRLM patients
      add_dataframe("NCRLMpatient", mon = 2,  NCRLM_patient, data =
NCRLM_input_df,
                    col_preemptible = "preemptible",time = "absolute",
                    col_attributes = c("blood_type", "ncrlm_mortality",
'bg_mortality'))
    sim %>% reset() %>% simmer::run(until = time_horizon)

    df_str2B_both <- get_mon_attributes(sim) %>%
      tibble() %>%
      group_by(name, key) %>%
      mutate( a = row_number()) %>%
      ungroup() %>%
      mutate(key = case_when(key == "surgery_time_fail"~ paste0(key, a),
                             TRUE ~ key)) %>%
      pivot_wider(id_cols = name, names_from = key, values_fn = max)


    df_str2B_both <- df_str2B_both %>%
      filter(TimeOfReferral >= year, TimeOfReferral <= year +10)

    ### NCRLM
    df_str2B_NCRLM_both <- df_str2B_both %>% filter(str_detect(name,
"NCRLM"))

    #> Calculate Life year in time horizon
    df_str2B_NCRLM_both <- df_str2B_NCRLM_both %>%
      mutate(LY = if_else(is.na(TimeToDeath), time_horizon-
TimeOfReferral, TimeToDeath)) %>%
      mutate(dLY = fn_discount_continuous(1, time_of_discount =
time_of_discounting,
                                          cost_start = TimeOfReferral,
                                          cost_end = TimeOfDeath)) %>%
      mutate (waiting_time = case_when(!is.na(drop_out) ~ NA_real_,
                                       !is.na(surgery_time_fail1) ~
surgery_time_fail1 - TimeOfReferral,
                                       !is.na(surgery_time_succeed) ~
surgery_time_succeed - TimeOfReferral,
                                       TRUE ~ TimeOfDeath - TimeOfReferral
      )) %>%
      mutate(LY_before_sur = if_else(is.na(surgery_time_succeed),
                                     if_else(is.na(TimeOfDeath),
                                             time_horizon -
TimeOfReferral,
```

```
                                          TimeOfDeath - TimeOfReferral
                                      ),
                                      surgery_time_succeed-TimeOfReferral))
%>%

      mutate(LY_after_sur = LY - LY_before_sur) %>%
      mutate(disease_free_until = if_else(is.na(surgery_time_succeed),
NA_real_ , (surgery_time_succeed + LY_after_sur*0.33))) %>%
      mutate(palliative_start =
               if_else(is.na(surgery_time_succeed), NA_real_,TimeOfDeath -
LY_after_sur*0.33)) %>%
      mutate(
        disease_free = disease_free_until - surgery_time_succeed,
        recurrence = palliative_start - disease_free_until,
        palliative = TimeOfDeath - palliative_start
      )

    df_str2B_NCRLM_both <- df_str2B_NCRLM_both  %>%

      ##FIXME toFIXCOSTcompute cost

      mutate(
        Cost_Best_supportive = C_best_supportive_care * (if_else(LY >=
0.25,0.25,LY)),
        dCost_Best_supportive= fn_discount_discrete(Cost_Best_supportive,
time_of_event = TimeOfDeath - time_of_discounting),
        Cost_0_6_Pal_Care = C_Prior_Palliative_chemotherapy_month_0_6 *
(if_else(LY >= 0.75,0.5,

if_else(LY<=0.25,0,LY - 0.25))),
        dCost_0_6_Pal_Care =
fn_discount_continuous(C_Prior_Palliative_chemotherapy_month_0_6,time_of_discount
= time_of_discounting,
                                                    cost_start =
TimeOfReferral, cost_end = (if_else(LY >= 0.75, TimeOfReferral + 0.5,

if_else(LY<=0.25,TimeOfReferral,TimeOfDeath - 0.25)))),
        Cost_7_12_Pal_Care = C_Prior_Palliative_chemotherapy_month_7_12 *
(if_else(LY >= 1.25,0.5,

if_else(LY<=0.5,0,LY - 0.75))),
        dCost_7_12_Pal_Care =
fn_discount_continuous(C_Prior_Palliative_chemotherapy_month_7_12
,time_of_discount = time_of_discounting,
                                                     cost_start =
TimeOfReferral+0.5, cost_end = (if_else(LY >= 1.25, TimeOfReferral + 1,

if_else(LY<=0.5,TimeOfReferral+0.5,TimeOfDeath - 0.25)))),
        Cost_13_Pal_Care = C_Prior_Palliative_chemotherapy_month_13_onward
* (if_else(LY >= 1.25, LY - 0.25,0)),
        dCost_13_Pal_Care =
fn_discount_continuous(C_Prior_Palliative_chemotherapy_month_13_onward,time_of_discount
=  time_of_discounting,
                                                   cost_start =
TimeOfReferral+1, cost_end = (if_else(LY >= 1.25, TimeOfDeath - 0.25,
TimeOfReferral+1))),

      ) %>%
```

```
      mutate(
        Cost_waiting = LY_before_sur * C_waiting,
        d_cost_waiting = fn_discount_continuous(C_waiting,
time_of_discount = time_of_discounting,
                                                cost_start =
TimeOfReferral, cost_end = surgery_time_succeed
        ),
        Cost_transplantation = C_str2_1st_transplantation,
        d_Cost_transplantation =
fn_discount_discrete(Cost_transplantation, time_of_event =
                                                             if_else(is.na(surgery_time_fail
surgery_time_succeed, surgery_time_fail1)
        ),
        Cost_retransplantation = if_else(is.null(surgery_time_fail1),0,
                                         C_str2_2nd_transplantation),
        d_Cost_retransplantation =
fn_discount_discrete(Cost_retransplantation,
                                                             time_of_event =
surgery_time_succeed),
        Cost_diseasefree_0_3 =  C_Post_survival_disease_free_month_1_3 *
if_else(disease_free >=3, 0.25, disease_free),
        d_Cost_diseasefree_0_3 =
fn_discount_continuous(C_Post_survival_disease_free_month_1_3,
time_of_discount = time_of_discounting,
                                                             cost_start =
surgery_time_succeed,
                                                             cost_end =
surgery_time_succeed + if_else(disease_free >=3, 0.25, disease_free)),
        Cost_diseasefree_4_onwards =
C_Post_survival_disease_free_month_4_onwards * if_else(disease_free >=3,
disease_free - 0.25, 0),
        d_Cost_diseasefree_4_onwards =
fn_discount_continuous(C_Post_survival_disease_free_month_4_onwards,
time_of_discount = time_of_discounting,
                                                                  cost_start =
surgery_time_succeed + 0.25,
                                                                  cost_end =
disease_free_until),
        Cost_recurrence_1_12 = C_Post_survival_recurrence_month_1_12 *
if_else(recurrence >=12, 1, recurrence),
        d_Cost_recurrence_1_12 =
fn_discount_continuous(C_Post_survival_recurrence_month_1_12,
time_of_discount = time_of_discounting,
                                                             cost_start =
disease_free_until,
                                                             cost_end =
disease_free_until + if_else(recurrence >=12, 1, recurrence)),
        Cost_recurrence_13_onward =
C_Post_survival_recurrence_month_13_onwards * if_else(recurrence >=12,
recurrence - 1, 0),
        d_Cost_recurrence_13_onward =
fn_discount_continuous(C_Post_survival_recurrence_month_13_onwards,
time_of_discount = time_of_discounting,
                                                               cost_start =
disease_free_until+ if_else(recurrence >=12, 1, recurrence),
                                                               cost_end =
palliative_start),
```

```
        Cost_palliative_0_6 =
C_Post_Surival_Palliative_chemotherapy_month_0_6 * if_else(palliative >=
0.75, 0.5,

if_else(palliative <=0.25, 0, palliative - 0.25)
        ),

        d_Cost_palliative_0_6 = if_else(palliative >= 0.25,
fn_discount_continuous(C_Post_Surival_Palliative_chemotherapy_month_0_6,
time_of_discount = time_of_discounting,

cost_start = palliative_start,

cost_end = if_else(palliative >= 0.5, palliative_start + 0.5,
palliative_start + palliative

)),0)
      ) %>%

      mutate(
        Cost_palliative_7_12 =
C_Post_Surival_Palliative_chemotherapy_month_7_12 * if_else(palliative >=
1.25, 0.5,

if_else(palliative <= 0.75,0, palliative - 0.75)
        ),
        d_Cost_palliative_7_12 = if_else(palliative >= 0.75,
fn_discount_continuous(C_Post_Surival_Palliative_chemotherapy_month_7_12 ,
time_of_discount = time_of_discounting,

cost_start = palliative_start + 0.5,

cost_end = if_else(palliative >= 1.25, palliative_start + 1,
palliative_start + palliative

)),0
        ),
        Cost_palliative_13_onward =
C_Post_Surival_Palliative_chemotherapy_month_13_onward *
if_else(palliative >= 1.25, palliative - 1.25, 0 ),
        d_Cost_palliative_13_onward = if_else(palliative >= 1.25,
fn_discount_continuous(C_Post_Surival_Palliative_chemotherapy_month_13_onward
, time_of_discount = time_of_discounting,

cost_start = palliative_start + 1,

cost_end = palliative_start + palliative - 0.25
        ),0)
      ) %>%

      mutate(
        total_cost =  if_else(is.na(surgery_time_succeed),
                              Cost_Best_supportive + Cost_0_6_Pal_Care +
Cost_7_12_Pal_Care + Cost_13_Pal_Care,
                              Cost_Best_supportive + Cost_waiting +
Cost_transplantation + Cost_retransplantation +  Cost_diseasefree_0_3
```

```
                              + Cost_diseasefree_4_onwards +
Cost_recurrence_1_12 + Cost_recurrence_13_onward + Cost_palliative_0_6 +
Cost_palliative_7_12 +
                                Cost_palliative_13_onward),

        d_total_cost = if_else(is.na(surgery_time_succeed),
                               dCost_Best_supportive + dCost_0_6_Pal_Care
+ dCost_7_12_Pal_Care + dCost_13_Pal_Care,
                               dCost_Best_supportive + d_cost_waiting +
d_Cost_transplantation  + d_Cost_retransplantation +
d_Cost_diseasefree_0_3
                               + d_Cost_diseasefree_4_onwards +
d_Cost_recurrence_1_12 + d_Cost_recurrence_13_onward +
d_Cost_palliative_0_6 +  d_Cost_palliative_7_12 +
                                 d_Cost_palliative_13_onward),

        QALY = if_else(is.na(surgery_time_succeed),
                       LY * u_NCRLM_palliative,
                       LY_before_sur * u_NCRLM_palliative +
                         if_else(disease_free >= 0.25,
U_Disease_free_cycle_1 *0.25 + (disease_free -
0.25)*U_Disease_free_cycle_2_and_onward,
                                 disease_free * U_Disease_free_cycle_1 ) +
                          if_else(recurrence >= 1, U_Recurrence_first_year
+ (recurrence - 1)* U_Recurrence_year_2_and_onward, recurrence*
U_Recurrence_first_year) +
                          palliative* U_Palliative),

        dQALY = if_else(is.na(surgery_time_succeed),
                        fn_discount_continuous(u_NCRLM_palliative,time_of_discount
= time_of_discounting,
                                               cost_start =
TimeOfReferral, cost_end = TimeOfDeath),

                        fn_discount_continuous(u_NCRLM_palliative,time_of_discount
= time_of_discounting,
                                               cost_start =
TimeOfReferral, cost_end = TimeOfReferral + LY_before_sur) +

                          if_else(disease_free >= 0.25,
                                  fn_discount_continuous(U_Disease_free_cycle_1,
time_of_discount = time_of_discounting,
                                                         cost_start =
surgery_time_succeed, cost_end = surgery_time_succeed + 0.25) +
                                    fn_discount_continuous(U_Disease_free_cycle_2_and_o
time_of_discount = time_of_discounting,
                                                           cost_start =
surgery_time_succeed+0.25, cost_end = surgery_time_succeed +
disease_free),
                                  fn_discount_continuous(U_Disease_free_cycle_1,
time_of_discount = time_of_discounting,
                                                         cost_start =
surgery_time_succeed, cost_end = surgery_time_succeed + 0.25)) +
                          if_else( recurrence >= 1,
                                   fn_discount_continuous(U_Recurrence_first_year,
time_of_discount = time_of_discounting,
                                                          cost_start =
disease_free_until, cost_end = disease_free_until +1) +
```

```
                                      fn_discount_continuous(U_Recurrence_year_2_and_onw
time_of_discount = time_of_discounting,
                                                             cost_start =
disease_free_until +1, cost_end = palliative_start),
                                    fn_discount_continuous(U_Recurrence_first_year,
time_of_discount = time_of_discounting,
                                                           cost_start =
disease_free_until, cost_end = palliative_start)
                            )+
                            fn_discount_continuous(U_Palliative,
time_of_discount = time_of_discounting,
                                                   cost_start =
palliative_start, cost_end = TimeOfDeath)

        )
      )

    Output_str2B_NCRLM_both <- df_str2B_NCRLM_both %>%
      ## calculate mean
      summarise(n_patients = n(),
across(c(LY,dLY,waiting_time,total_cost,d_total_cost,QALY,dQALY), sum,
na.rm= TRUE)) %>%
      mutate(id = "NCRLM")


    ## SQ result

    df_str2B_SQ_both <- df_str2B_both %>% filter(str_detect(name, "SQ"))


    #> Calculate Life year in time horizon
    df_str2B_SQ_both <- df_str2B_SQ_both %>%
      mutate(LY = if_else(is.na(TimeToDeath), time_horizon-
TimeOfReferral, TimeToDeath)) %>%
      mutate(dLY = fn_discount_continuous(1, time_of_discount =
time_of_discounting,
                                          cost_start = TimeOfReferral,
                                          cost_end = TimeOfReferral + LY))
%>%
      mutate (waiting_time = case_when(!is.na(drop_out) ~ NA_real_,
                                       !is.na(surgery_time_fail1) ~
surgery_time_fail1 - TimeOfReferral,
                                       !is.na(surgery_time_succeed) ~
surgery_time_succeed - TimeOfReferral,
                                       TRUE ~ TimeOfDeath - TimeOfReferral
      )) %>%
      mutate(LY_before_sur = if_else(is.na(surgery_time_succeed),
                                     if_else(is.na(TimeOfDeath),
                                             time_horizon -
TimeOfReferral,
                                             TimeOfDeath - TimeOfReferral
                                     ),
                                     surgery_time_succeed-TimeOfReferral))
%>%
      mutate(LY_after_sur = LY - LY_before_sur)

    #> Total life year SQ_Both
    Total_LY_SQ_str2B_Both <- sum(df_str2B_SQ_both$LY, na.rm = FALSE)
```

```
    #> Calculate cost and QALYs SQ patients both

    df_str2B_SQ_both <- df_str2B_SQ_both %>%
      ##compute cost
      ##FIXME - case of die in surgery
      mutate(
        total_cost = SQ_c_before_trans_care * LY_before_sur +
SQ_c_survival_care * LY_after_sur +
          if_else(is.na(surgery_time_succeed),0,
                  if_else(is.na(surgery_time_fail1),
C_str2_1st_transplantation,
                          C_str2_1st_transplantation +
C_str2_2nd_transplantation)),
        d_total_cost = fn_discount_continuous(SQ_c_before_trans_care,
time_of_discount = time_of_discounting,
                                              cost_start = TimeOfReferral,
                                              cost_end = TimeOfReferral +
LY_before_sur) +
          if_else(is.na(surgery_time_succeed), 0,
fn_discount_continuous(SQ_c_survival_care, time_of_discount =
time_of_discounting,
                                                                    cost_start
= TimeOfReferral + LY_before_sur,
                                                                    cost_end
= TimeOfReferral + LY)) +
          if_else(is.na(surgery_time_succeed),0,
                  if_else(is.na(surgery_time_fail1),
                          fn_discount_discrete(C_str2_1st_transplantation,
                                               time_of_event =
surgery_time_succeed - time_of_discounting),

                          fn_discount_discrete(C_str2_1st_transplantation,
                                               time_of_event =
surgery_time_fail1 - time_of_discounting) +
                            fn_discount_discrete(C_str2_2nd_transplantation,
                                                 time_of_event =
surgery_time_succeed - time_of_discounting))),

        QALY = LY_before_sur * SQ_u_prior_surgery + LY_after_sur *
SQ_u_after_surgery,
        dQALY = fn_discount_continuous(SQ_u_prior_surgery,
time_of_discount = time_of_discounting,
                                       cost_start = TimeOfReferral,
                                       cost_end = TimeOfReferral +
LY_before_sur) +
          if_else(is.na(surgery_time_succeed),0,
fn_discount_continuous(SQ_u_after_surgery, time_of_discount =
time_of_discounting,
                                                                  cost_start
= TimeOfReferral + LY_before_sur,
                                                                  cost_end
= TimeOfReferral + LY))
      )


    Output_str2B_SQ_both <- df_str2B_SQ_both %>%
```

```
      ## calculate mean
      summarise(n_patients = n(),
across(c(LY,dLY,waiting_time,total_cost,d_total_cost,QALY,dQALY), sum,
na.rm = TRUE)) %>%
      mutate(id = "SQ")

    return(rbind(Output_str2B_NCRLM_both, Output_str2B_SQ_both))

  }
})

library(parallel)
sim_str2b_rs = mclapply(1:1000, function(i) str_2b(year = 7), mc.cores =
1)
options(scipen = 5)
sim_str2b_rs = data.table::rbindlist(sim_str2b_rs)
saveRDS(sim_str2b_rs, here::here("run-and-result/sim_str2b_rs.rds"))



Output_str2b_2_NCRLM_avg_0805_2 <- sim_str2b_rs %>%
  group_by(id) %>%
  ## calculate mean
  summarise(across(c(n_patients,LY,dLY,waiting_time,
total_cost,d_total_cost,QALY,dQALY), mean, na.rm= TRUE))


str3 = local({
  library(here)
  source(here("Final_I.CommonValues2103.R"), local= TRUE)
  source(here("Final_II.Functions2103.R"), local =TRUE)
  source(here("FInal_III.Trajectory2103.R"), local =TRUE)
  source(here('Final_IV.CommonRandomValues2103.R'), local = TRUE)
  time_horizon <- 70
  sim = NA


  f = function(year){

    source(here('Final_IV.CommonRandomValues2103.R'), local = TRUE)


    sim <<- simmer()
    sim %>%
      add_resource(name = "Waiting", capacity = Inf) %>%
      add_resource(name = "Surgery", capacity = Inf) %>%
      add_resource(name = "Surgery_retrans", capacity = Inf) %>%
      add_resource(name = "Waiting_retrans", capacity = Inf) %>%
      add_dataframe("NCRLMpatient", mon = 2,  str3_trj_main_CRLM, data =
NCRLM_input_df,
                    col_preemptible = "preemptible",time = "absolute",
                    col_attributes = c("blood_type", "ncrlm_mortality",
'bg_mortality'))
    # add_resource(name = "Waiting", capacity = Inf) %>%
    # add_resource(name = "Surgery", capacity = Inf) %>%
    # add_resource(name = "Surgery_retrans", capacity = Inf) %>%
    # add_resource(name = "Waiting_retrans", capacity = Inf) %>%
```

```
    # add_dataframe("Str3_NCRLM", mon = 2,  str3_trj_main_CRLM, data =
NCRLM_input_df,
    #               col_preemptible = "preemptible",time = "absolute",
    #               col_attributes = c("blood_type"))

    # Running the simulation until all events have occurred
    sim %>% reset() %>% run(until=70);


    # Summarizing the recorded attributes using the custom function
    df_str3_NCRLM <- get_mon_attributes(sim) %>%
      pivot_wider(id_cols = name, names_from = key, values_from = value)

    df_str3_NCRLM = df_str3_NCRLM %>%
      filter(TimeOfReferral >= year, TimeOfReferral <= year +10)

    ## compute time frame
    # In case TimeOf2ndSurgery is not exists
    if(is.null(df_str3_NCRLM$TimeOf2ndSurgery))
df_str3_NCRLM$TimeOf2ndSurgery = NA

    df_str3_NCRLM = df_str3_NCRLM %>%
      mutate(LY = if_else(is.na(TimeToDeath), 70 - TimeOfReferral,
                          TimeToDeath)) %>%
      mutate(LY_before_sur = case_when(!is.na(TimeOfSurgery) ~
TimeOfSurgery - TimeOfReferral,
                                       !is.na(TimeOf2ndSurgery) ~
TimeOf2ndSurgery - TimeOfReferral,
                                       !is.na(TimeToDeath) ~ TimeToDeath,
                                       # truong hop cuoi tat ca 3 cai tren
deu NA
                                       TRUE ~ 70 - TimeOfReferral)) %>%
      mutate(LY_after_sur = LY - LY_before_sur) %>%
      mutate(dLY = fn_discount_continuous(1, time_of_discount =
time_of_discounting,
                                          cost_start = TimeOfReferral,
                                          cost_end = TimeOfReferral + LY))

    # Add sample Cost

    df_str3_NCRLM <- df_str3_NCRLM  %>%
      mutate(disease_free_until = if_else(is.na(TimeOfSurgery),
NA_real_,TimeOfReferral + LY_before_sur + LY_after_sur*0.33)) %>%
      mutate(palliative_start =
               if_else(is.na(TimeOfSurgery), NA_real_,TimeOfDeath -
LY_after_sur*0.33)) %>%
      mutate(
        disease_free = disease_free_until - TimeOfReferral -
LY_before_sur,
        recurrence = palliative_start - disease_free_until,
        palliative = TimeOfDeath - palliative_start
      ) %>%
      mutate(
        Cost_Best_supportive = C_best_supportive_care * (if_else(LY >=
0.25,0.25,LY)),
        dCost_Best_supportive= fn_discount_discrete(Cost_Best_supportive,
time_of_event = TimeOfDeath - time_of_discounting),
```

```
        Cost_0_6_Pal_Care = C_Prior_Palliative_chemotherapy_month_0_6 *
(if_else(LY >= 0.75,0.5,

if_else(LY<=0.25,0,LY - 0.25))),
        dCost_0_6_Pal_Care =
fn_discount_continuous(C_Prior_Palliative_chemotherapy_month_0_6,time_of_discount
= time_of_discounting,
                                                    cost_start =
TimeOfReferral, cost_end = (if_else(LY >= 0.75, TimeOfReferral + 0.5,

if_else(LY<=0.25,TimeOfReferral,TimeOfDeath - 0.25)))),
        Cost_7_12_Pal_Care = C_Prior_Palliative_chemotherapy_month_7_12 *
(if_else(LY >= 1.25,0.5,

if_else(LY<=0.5,0,LY - 0.75))),
        dCost_7_12_Pal_Care =
fn_discount_continuous(C_Prior_Palliative_chemotherapy_month_7_12
,time_of_discount = time_of_discounting,
                                                     cost_start =
TimeOfReferral+0.5, cost_end = (if_else(LY >= 1.25, TimeOfReferral + 1,

if_else(LY<=0.5,TimeOfReferral+0.5,TimeOfDeath - 0.25)))),
        Cost_13_Pal_Care = C_Prior_Palliative_chemotherapy_month_13_onward
* (if_else(LY >= 1.25, LY - 0.25,0)),
        dCost_13_Pal_Care =
fn_discount_continuous(C_Prior_Palliative_chemotherapy_month_13_onward,time_of_discount
=  time_of_discounting,
                                                   cost_start =
TimeOfReferral+1, cost_end = (if_else(LY >= 1.25, TimeOfDeath - 0.25,
TimeOfReferral+1))),

      ) %>%
      mutate(
        Cost_waiting = LY_before_sur * C_waiting,
        d_cost_waiting = fn_discount_continuous(C_waiting,
time_of_discount = time_of_discounting,
                                                cost_start =
TimeOfReferral, cost_end = TimeOfReferral + LY_before_sur),

        Cost_transplantation = C_str3_1st_transplantation,
        d_Cost_transplantation =
fn_discount_discrete(Cost_transplantation, time_of_event = TimeOfSurgery
        ),

        Cost_retransplantation = if_else(is.null(TimeOf2ndSurgery),0,
                                         C_str3_2nd_transplantation),
        d_Cost_retransplantation =
fn_discount_discrete(Cost_retransplantation,
                                                       time_of_event =
TimeOf2ndSurgery),

        Cost_diseasefree_0_3 =  C_Post_survival_disease_free_month_1_3 *
if_else(disease_free >=3, 0.25, disease_free),
        d_Cost_diseasefree_0_3 =
fn_discount_continuous(C_Post_survival_disease_free_month_1_3,
time_of_discount = time_of_discounting,
                                                       cost_start =
TimeOfReferral + LY_before_sur,
```

```
                                                              cost_end =
TimeOfReferral + LY_before_sur + if_else(disease_free >=3, 0.25,
disease_free)),
        Cost_diseasefree_4_onwards =
C_Post_survival_disease_free_month_4_onwards * if_else(disease_free >=3,
disease_free - 0.25, 0),
        d_Cost_diseasefree_4_onwards =
fn_discount_continuous(C_Post_survival_disease_free_month_4_onwards,
time_of_discount = time_of_discounting,
                                                               cost_start =
TimeOfReferral + LY_before_sur + 0.25,
                                                               cost_end =
disease_free_until),
        Cost_recurrence_1_12 = C_Post_survival_recurrence_month_1_12 *
if_else(recurrence >=12, 1, recurrence),
        d_Cost_recurrence_1_12 =
fn_discount_continuous(C_Post_survival_recurrence_month_1_12,
time_of_discount = time_of_discounting,
                                                              cost_start =
disease_free_until,
                                                              cost_end =
disease_free_until + if_else(recurrence >=12, 1, recurrence)),
        Cost_recurrence_13_onward =
C_Post_survival_recurrence_month_13_onwards * if_else(recurrence >=12,
recurrence - 1, 0),
        d_Cost_recurrence_13_onward =
fn_discount_continuous(C_Post_survival_recurrence_month_13_onwards,
time_of_discount = time_of_discounting,
                                                                 cost_start =
disease_free_until+ if_else(recurrence >=12, 1, recurrence),
                                                               cost_end =
palliative_start),
        Cost_palliative_0_6 =
C_Post_Surival_Palliative_chemotherapy_month_0_6 * if_else(palliative >=
0.75, 0.5,

if_else(palliative <=0.25, 0, palliative - 0.25)
        ),

        d_Cost_palliative_0_6 = if_else(palliative >= 0.25,
fn_discount_continuous(C_Post_Surival_Palliative_chemotherapy_month_0_6,
time_of_discount = time_of_discounting,

cost_start = palliative_start,

cost_end = if_else(palliative >= 0.5, palliative_start + 0.5,
palliative_start + palliative

)),0),
        Cost_palliative_7_12 =
C_Post_Surival_Palliative_chemotherapy_month_7_12 * if_else(palliative >=
1.25, 0.5,

if_else(palliative <= 0.75,0, palliative - 0.75)
        ),
        d_Cost_palliative_7_12 = if_else(palliative >= 0.75,
fn_discount_continuous(C_Post_Surival_Palliative_chemotherapy_month_7_12 ,
time_of_discount = time_of_discounting,
```

```
cost_start = palliative_start + 0.5,

cost_end = if_else(palliative >= 1.25, palliative_start + 1,
palliative_start + palliative

)),0),
        Cost_palliative_13_onward =
C_Post_Surival_Palliative_chemotherapy_month_13_onward *
if_else(palliative >= 1.25, palliative - 1.25, 0 ),
        d_Cost_palliative_13_onward = if_else(palliative >= 1.25,
fn_discount_continuous(C_Post_Surival_Palliative_chemotherapy_month_13_onward
, time_of_discount = time_of_discounting,

cost_start = palliative_start + 1,

cost_end = palliative_start + palliative - 0.25
        ),0),


      )

    df_str3_NCRLM <- df_str3_NCRLM  %>%
      rowwise() %>%
      mutate(
        total_cost =  if_else(is.na(TimeOfSurgery),
                              sum(Cost_Best_supportive , Cost_0_6_Pal_Care
, Cost_7_12_Pal_Care , Cost_13_Pal_Care, na.rm = TRUE),
                              sum(Cost_Best_supportive , Cost_waiting ,
Cost_transplantation , Cost_retransplantation ,  Cost_diseasefree_0_3
                                  , Cost_diseasefree_4_onwards ,
Cost_recurrence_1_12 , Cost_recurrence_13_onward , Cost_palliative_0_6 ,
Cost_palliative_7_12 ,
                                 Cost_palliative_13_onward, na.rm =
TRUE)),

        d_total_cost = if_else(is.na(TimeOfSurgery),
                               sum(dCost_Best_supportive ,
dCost_0_6_Pal_Care , dCost_7_12_Pal_Care , dCost_13_Pal_Care, na.rm =
TRUE),
                               sum(dCost_Best_supportive , d_cost_waiting
, d_Cost_transplantation  , d_Cost_retransplantation ,
d_Cost_diseasefree_0_3
                                   , d_Cost_diseasefree_4_onwards ,
d_Cost_recurrence_1_12 , d_Cost_recurrence_13_onward ,
d_Cost_palliative_0_6 ,  d_Cost_palliative_7_12 ,
                                   d_Cost_palliative_13_onward, na.rm =
TRUE)),

      ) %>%
      ungroup() %>%
      mutate(
        QALY = if_else(is.na(TimeOfSurgery),
                       LY * u_NCRLM_palliative,
                       LY_before_sur * u_NCRLM_palliative +
                         if_else(disease_free >= 0.25,
U_Disease_free_cycle_1 *0.25 + (disease_free -
0.25)*U_Disease_free_cycle_2_and_onward,
```

```
                                disease_free * U_Disease_free_cycle_1 ) +
                           if_else(recurrence >= 1, U_Recurrence_first_year
+ (recurrence - 1)* U_Recurrence_year_2_and_onward, recurrence*
U_Recurrence_first_year) +
                           palliative* U_Palliative),

        dQALY = if_else(is.na(TimeOfSurgery),
                        fn_discount_continuous(u_NCRLM_palliative,time_of_discount
= time_of_discounting,
                                               cost_start =
TimeOfReferral, cost_end = TimeOfReferral + LY),

                        fn_discount_continuous(u_NCRLM_palliative,time_of_discount
= time_of_discounting,
                                               cost_start =
TimeOfReferral, cost_end = TimeOfReferral + LY_before_sur) +

                          if_else(disease_free >= 0.25,
                                  fn_discount_continuous(U_Disease_free_cycle_1,
time_of_discount = time_of_discounting,
                                                         cost_start =
TimeOfReferral + LY_before_sur, cost_end = TimeOfReferral + LY_before_sur
+ 0.25) +
                                    fn_discount_continuous(U_Disease_free_cycle_2_and_o
time_of_discount = time_of_discounting,
                                                           cost_start =
TimeOfReferral + LY_before_sur+0.25, cost_end = TimeOfReferral +
LY_before_sur + disease_free),
                                  fn_discount_continuous(U_Disease_free_cycle_1,
time_of_discount = time_of_discounting,
                                                         cost_start =
TimeOfReferral + LY_before_sur, cost_end = TimeOfReferral + LY_before_sur
+ 0.25)) +
                          if_else( recurrence >= 1,
                                   fn_discount_continuous(U_Recurrence_first_year,
time_of_discount = time_of_discounting,
                                                          cost_start =
disease_free_until, cost_end = disease_free_until +1) +
                                     fn_discount_continuous(U_Recurrence_year_2_and_onw
time_of_discount = time_of_discounting,
                                                            cost_start =
disease_free_until +1, cost_end = palliative_start),
                                   fn_discount_continuous(U_Recurrence_first_year,
time_of_discount = time_of_discounting,
                                                          cost_start =
disease_free_until, cost_end = palliative_start)
                           )+
                           fn_discount_continuous(U_Palliative,
time_of_discount = time_of_discounting,
                                                  cost_start =
palliative_start, cost_end = TimeOfReferral + LY)

        )
      )


    Output_str3_NCRLM <- df_str3_NCRLM %>%
      ## calculate mean
```

```
      summarise(n_patients = n(), across(c(LY,dLY,total_cost:dQALY), sum))
%>%
      mutate(id = "NCRLM")



    ###3.2 Donor normal case simulation------------

    # donor data
    donor_distribution <- sort(df_str3_NCRLM$TimeOfSurgery)
    donor_distribution = tibble(
      referal = donor_distribution,
      renege_in_time = rgompertz(
        length(donor_distribution),
        d_death_shape_donor_bg,
        d_death_rate_donor_bg
      )
    )



    sim <<- simmer()

    sim %>%
      add_dataframe(name_prefix = "Donor_sur", trajectory =
str3_donor_trj_main_bg,
                    data= donor_distribution,
                    time = "absolute",
                    col_time = "referal",col_attributes =
"renege_in_time",
                    mon = 2)


    # Running the simulation until all events have occurred aka 50 years

    sim %>% reset() %>% run(until=70);

    df_str3_donor_bg <- get_mon_attributes(sim) %>%
      pivot_wider(id_cols = name, names_from = key, values_from = value)

    #> Calculate Life year in time horizon

    ifelse(is.na(df_str3_donor_bg$TimeOfDeath),
           df_str3_donor_bg <- df_str3_donor_bg %>%
             mutate(LY = 70-TimeOfReferral),
           df_str3_donor_bg <- df_str3_donor_bg %>%
             mutate(LY = if_else(is.na(df_str3_donor_bg$TimeOfDeath),
                                 70-TimeOfReferral,
                                 TimeToDeath))
    )

    df_str3_donor_bg <- df_str3_donor_bg %>%
      mutate(dLY = fn_discount_continuous(1, time_of_discount =
time_of_discounting,
                                          cost_start = TimeOfReferral,
                                          cost_end = TimeOfReferral + LY))
%>%
      mutate(
```

```
        total_cost = 0,
        d_total_cost = 0,
        QALY = LY,
        dQALY = dLY
      )



    Output_str3_donor_bg<- df_str3_donor_bg%>%
      ## calculate mean
      summarise(n_patients =n(),
across(c(LY,dLY,total_cost,d_total_cost,QALY,dQALY), sum)) %>%
      mutate(id = "donor_bg")

    ### 3.3 Donor donation case simulation-----------


    sim <<- simmer()
    sim %>%
      add_dataframe(name_prefix = "Donor_sur", trajectory =
str3_donor_trj_main_sur,
                    data= donor_distribution, col_time = "referal",
                    time = "absolute",
                    col_attributes = "renege_in_time",
                    mon = 2)



    # Running the simulation until all events have occurred
    sim %>% reset() %>% run(until=70);
    ##FIXME - There will be the case all donor live until simulation end -
then what we can do to
    #> have a Life Year column?

    df_str3_donor_sur <- get_mon_attributes(sim) %>%
      pivot_wider(id_cols = name, names_from = key, values_from = value)

    #> Calculate Life year in time horizon
    #>
    ifelse(is.null(df_str3_donor_sur$TimeOfDeath),
           df_str3_donor_sur <- df_str3_donor_sur %>%
             mutate(LY = 70-TimeOfReferral),
           df_str3_donor_sur <- df_str3_donor_sur %>%
             mutate(LY = if_else(is.na(df_str3_donor_sur$TimeOfDeath),
                                 70-TimeOfReferral,
                                 TimeToDeath))
    )

    #> Calculation of cost
    df_str3_donor_sur <- df_str3_donor_sur %>%
      mutate(dLY = fn_discount_continuous(1, time_of_discount =
time_of_discounting,
                                          cost_start = TimeOfReferral,
                                          cost_end = TimeOfReferral + LY),
             total_cost = if_else(LY>=10, c_donor_care_1st_year +
9*c_donor_care_2_10_year,
                                  if_else(LY>=1,c_donor_care_1st_year +
(LY-1)*c_donor_care_2_10_year,
```

```
                                          LY*c_donor_care_1st_year)),
            d_total_cost = if_else(LY>=10,
fn_discount_continuous(c_donor_care_1st_year, time_of_discount =
time_of_discounting,
                                                                  cost_start
= TimeOfReferral, cost_end = TimeOfReferral+1)
                                      +
fn_discount_continuous(c_donor_care_2_10_year,time_of_discount =
time_of_discounting,
                                                          cost_start =
TimeOfReferral+1,
                                                          cost_end =
TimeOfReferral+10),
                                    if_else(LY>=1,fn_discount_continuous(c_donor_care_1
time_of_discount = time_of_discounting,
                                                               cost_start
= TimeOfReferral, cost_end = TimeOfReferral+1) +
                                             fn_discount_continuous(c_donor_care_2_10_
= time_of_discounting,
                                                                 cost_start
= TimeOfReferral+1,
                                                                 cost_end
= TimeOfReferral+LY),
                                           fn_discount_continuous(c_donor_care_1st_yea
time_of_discount = time_of_discounting,
                                                                  cost_start
= TimeOfReferral, cost_end = TimeOfReferral+LY))),
            QALY = if_else(LY >= 1, u_donor_care_1_year + LY-1,
u_donor_care_1_year*LY),
            dQALY = if_else(LY >= 1,
fn_discount_continuous(u_donor_care_1_year, time_of_discount =
time_of_discounting,
                                                          cost_start =
TimeOfReferral, cost_end = TimeOfReferral+1)+
                                fn_discount_continuous(1,time_of_discount =
time_of_discounting,
                                                         cost_start =
TimeOfReferral+1,
                                                         cost_end =
TimeOfReferral+LY),
                               fn_discount_continuous(u_donor_care_1_year,
time_of_discount = time_of_discounting,
                                                       cost_start =
TimeOfReferral, cost_end = TimeOfReferral+LY))
      )




    Output_str3_donor_sur <- df_str3_donor_sur %>%
      ## calculate mean
      summarise(n_patients = n(),
across(c(LY,dLY,total_cost,d_total_cost,QALY,dQALY), sum)) %>%
      mutate(id= "donor_sur")
```

```
    return(rbind(Output_str3_NCRLM, Output_str3_donor_bg,
Output_str3_donor_sur))
  }


})


library(parallel)
sim_str3_rs = mclapply(1:1000, function(i) str3(year = 7), mc.cores = 1)

options(scipen = 5)
sim_str3_rs = data.table::rbindlist(sim_str3_rs)
saveRDS(sim_str3_rs, here::here("run-and-result/sim_str3_rs.rds"))

Output_str3_avg_0805<- sim_str3_rs %>%
  group_by(id) %>%
  ## calculate mean
  summarise(across(c(n_patients,LY,dLY,total_cost,d_total_cost,QALY,dQALY),
mean))
```