



Operation Room Triage model

A collaboration of Erasmus MC researchers

Report created: 08 April, 2020, Last update model: 07/4/2020

Goal of this research

The aim of this model is to support the triage for semi-elective operation within the Erasmus MC by identifying disease with the greatest potential gain by surgery.

How to read this report

The first sections gives a summary of the research and includes the most important conclusions and gives an overview of important assumptions. In the second section we describe the model in more detail and provide an explanation of the files used to generate the results. All the code and data is provided in the appendix and can be used to replicate the results or modify the model for a different disease or surgery type. In the final section, Section 3, all model results are included.

Section 1: Summary of the research

Model structure

To generate the data that helps surgery triage, we make use of a cohort state-transition model, also often called a Markov model, with three health states. These health states are a pre-operation health state, *Preop*, a post-operation health state, *Postop*, and dead *Dead*. From the *Preop* health state patients, were patients in need for surgery are waiting for surgery, they can transition to the *Postop* health state at the time they are eligible for surgery, they can die, in which case they transition to *Dead* or they can remain in the *Preop* health state to wait for surgery. Individual in the *Postop* health state might die or stay in the *Postop* health state. *Dead* is an absorbing state. The state-transition diagram is shown below.

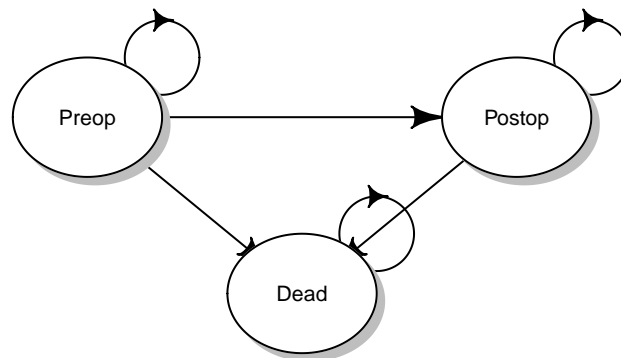


Figure 1: State-transition cohort model diagram.

We model a hypothetical cohort of representative patients over a lifetime horizon, up to a maximum age of 100 years, using weekly cycles. Since the average age per patient population is different, the total number of cycles is different per disease as the total number of cycles is defined by **100 years - average age in years of patients**.

The cohort is modeled over several strategies. The first strategy is immediate surgery, defined as surgery within 2 weeks, vs delayed surgery modelled. Delayed surgery is model in intervals of 10 week delays after 2 weeks up to a maximum of 1 year. The main outcome of each modeled strategy is the expected quality adjusted life years (QALYs). These QALYs are calculated using a discount rate of 1.5% as recommended by the [Dutch guidelines] (<https://english.zorginstituutnederland.nl/publications/reports/2016/06/16/guideline-for-economic-evaluations-in-healthcare>).

Most important assumptions

Model structure: - Comorbidities are not taken into account

- (Artificial) triangular distributions are used for parameter distribution, if the real distribution is not known.
- We assume that all operations are succesful (e.g. reoperation is not taken into account).
- The SARS-Cov2 virus causing the pandemic situation is not modeled.
- All patients are assumed to be not infected by COVID-19.

Related to Survival/treatment effect: - Before surgery the survival without surgery is applicable.

- After surgery the survival with surgery is applicable.

- Both survival with and without surgery is not age specific, unless specified.
- The survival after surgery is not affected by the delay of the surgery.
- The survival/effect of surgery is similar during an epidemic as compared to a normal setting (e.g. the difference of the health care system does not affect the patient outcomes).
- We don't include major complications (major bleeding etc.) of surgery because we assume that these complications are rare and equally distributed among the operations.
- The SARS-Cov-2 virus does not affect the survival.
- For cancer diseases the tumor doubling rate is used as an indication of when surgery becomes not effective anymore. We assume that after the tumor doubling time the surgery does not have an effect on the survival after surgery anymore and we apply the survival without surgery even after surgery.
- For some disease, surgery delay can cause an irreversable QoL decrease (e.g. getting blind).

Related to quality of life: - The QoL without surgery is constant. This means that surgery delay is not effecting the QoL.

- The QoL after the surgery is not influenced by the time until surgery.
- The QoL is not age specific.
- No disutility of the surgery and during the time to recover from is included.
- For interventions which only aim to improve survival, the quality of life is constant (preoperative vs post-operative).
- *Add/check: Assumption we have with the QoL when they are equal /below/above*

Important conclusions and interpretaions

ADD SUMMARY OF THE RESULTS

Interpretation of

Secondary outcomes - derivative and area above the curve

Section 2: The model

In this section we describe some of the steps that are done to get the results. We refer to several functions that help to run the model. These files can be found in the appendix of this report but are not shown in full detail because in this report we like to mainly focus on the results. However, it is important to understand how the model is strucured and enourage those interested to understand the code, replicate the results or modify the model for you own needs.

The main file to run the model is the file called `main.R`. To generate update results of the model and this report this file needs to be executed.

To get this results the following steps describe the proces conceptually:

1. **Prepair the R environment for the simulation**
2. **Load the literature data into the environment** And modify it to be used for modeling a cohort using weekly cycles.

3. **Model a cohort over time** A model simulates a cohort of individuals up to an average age of 100 years while keeping track of the distribution of the cohort among the different health states *Preop*, *Postop*, *Dead*. For example, for the scenario of immediate surgery, surgery within 2 weeks, the cohort is modeled for two weeks using the survival rate before surgery. In these first two weeks of the model the individuals are at risk of dying according to the weekly probability of dying before surgery. If this happens they transition to *Dead* otherwise they stay in *Preop*. After 2 weeks, those in the *Preop* all get the surgery, and they make the transition to *Postop*. For the remaining cycles, they are at risk of dying according based on the survival rates after surgery.

3.1 **Calculate the QALYs** Based on the generated cohort trace, a matrix that described how the cohort is distributed among the three health states over time, the expected QALYs are calculated.

4. **Run all surgery scenarios** Repeat step 3 and 3.1 for a several times of delay for surgery. In our code we make use of a sequence of delay in steps up to 10 weeks up to a maximum of 1 year. In other words, we run the model for a scenario of surgery after 2, 12, 22, 32, 42 and 52 weeks.
5. **Incorporate parameter uncertainty** Using one parameter value for each model parameter in step 3 and 4 assumes we are certain about this values. However, this is not the case and our parameters come with some uncertainty. Some of this uncertainty is described by the literature, some is based on expert opinion. In order to capture parameter uncertainty on the model outcome we have to repeat steps 3 and 4 several times using different combinations of parameter values drawn from the distributions. Therefore, we repeat steps 3 and 4 100 times and calculate the average expected QALYS and the distribution around it.
6. **Repeat for all diseases** Step 3, 4 and 5 are now repeated for all diseases of interest.

In the paragraphs below we describe how these steps are incorporated in the code.

01 Required packages

To run the model we make use of some packages which can be downloaded from CRAN as well as a package in development, called **dampack**, which we download from Github. But running the following lines of code one can prepare the computer to replicated the results. In order to install the package from GitHub. The uncommended code about the code `p_load_gh("DARTH-git/dampack")` should be used to load the package into the environment. This should only be done once.

```
if (!require('pacman')) install.packages('pacman'); library(pacman) # use this package to conveniently
# load (install if required) packages from CRAN
p_load("here", "dplyr", "devtools", "scales", "ellipse", "ggplot2", "lazyeval", "igraph", "ggraph", "reprex")

# load (install if required) packages from GitHub
# install_github("DARTH-git/dampack", force = TRUE) #Uncomment if there is a newer version
p_load_gh("DARTH-git/dampack")
```

02 Required functions

In this section we load the functions needed for the model. The functions are stored in the *R* folder. The file **functions.R** contains some generic functions often used in decision modeling and epidemiological research. Like a function to convert probabilities to rates and adjust a annual rate to a weekly rate as we have in our cycle. The functions in this file are mainly provided by the DARTH workgroup.

The files **model.R** and **functions_PSA.R** includes functions specifically written for this research. The aim of these functions is to efficiently program and to make our code easier to read and more transparent. In section

XX we explain the model specific function in more detail, but we will first provide some information about the data used by these function.

```
source("../R/functions.R")      # Load general functions useful for state-transition models
# Model specific functions
source("../R/model.R")          # Code of the main model
source("../R/functions_PSA.R")  # Load the PSA function
```

03 Input model parameters

03.1 Load parameter data

We import the the parameters found in the literature for the different populations and different interventions. These parameters are stored in the `Model parameters.xlsx` file in the *data* folder and a list of reference can be found **ADD REFERENCE TO LITERATURE**

Moreover, we import the the age- and gender-specific mortality rates in 2018, downloaded from the CBS.

03.2 Data handling

This imported data requires some data handling before it can be used in our model, like adjusting the survival rates and annual probabilities to weekly probabilities. As well as translating tumor doubling time from time in days in a weekly unit.

All model parameters have a mean estimate as well as a description. The data describes the type of distribution as well as the parameters required to reconstruct the distribution. As described in the data from the parameters, some of these distributions come from the literature, while others are (artefically) informed using expert opinion.

03.3 Prepare

03.3.1 Make PSA dataset

We make sure of a probabilistic sensitivity analysis to demonstrate the robustness of the model outcomes. In other words, how sensitive are our model outcomes due to the uncertainty of the input parameters. A PSA requires a PSA dataset which contains a set of parameter values drawn from the distribution for each PSA iteration.

To generate this PSA dataset we use the function `make_psa_df` as described in the `functions_PSA.R` file in the *R* folder. Based on the parameter values and there corresponding distribution the function generate a dataset with parameter values of each iteration of the PSA. This PSA dataset is called `param_psa`, which stands of parameters for the psa.

The following table shows the distributions of the parameters in the PSA.

Table 1: The parameters with assumed distributions.

Population	Param	Intervention	Estimate, 95%CI
Women with resectable mammacarcinoma	Surv_no_tx	Surgical resection	0.00086 (0.00069 - 0.00103)
Women with resectable mammacarcinoma	Surv_tx	Surgical resection	0.00049 (0.00049 - 0.00050)
Women with resectable mammacarcinoma	QoL_no_tx	Surgical resection	0.80 (0.76 - 0.84)
Women with resectable mammacarcinoma	QoL_tx	Surgical resection	0.80 (0.76 - 0.84)
Women with resectable mammacarcinoma	Age	Surgical resection	59.22 (47.22 - 69.56)
Women with resectable mammacarcinoma	Time_noeff_Surv	Surgical resection	21.67 (18.31 - 24.83)

Population	Param	Intervention	Estimate, 95%CI
Patients with end-stage renal disease	Surv_no_tx	Living donor renal transplant	0.00137 (0.00105 - 0.00183)
Patients with end-stage renal disease	Surv_tx	Living donor renal transplant	0.00074 (0.00069 - 0.00081)
Patients with end-stage renal disease	Tx_eff	Living donor renal transplant	0.55 (0.42 - 0.68)
Patients with end-stage renal disease	QoL_no_tx	Living donor renal transplant	0.69 (0.62 - 0.77)
Patients with end-stage renal disease	QoL_tx	Living donor renal transplant	0.69 (0.62 - 0.77)
Patients with end-stage renal disease	Age	Living donor renal transplant	50.46 (41.17 - 58.07)

03.3.2 Make the transition probability matrix

A key component of Markov model is a structure called the transition probability matrix that described the probabilities of transitioning from one health state to another and to remain in a health state. The functions `make_m_trans` and `trans_operation` coded in the `model.R` file in the *R* folder generate this structure. The rows of the matrix describe the health state the individual in the cohort started from, while the columns describe where the individual transitions to. An example is shown below. In this unrealistic example, those in the *Preop* health state have a 20% chance of dying every week and an 80% probability to stay in the *Preop* state, while after surgery, the group has a 5% chance of dying every week.

```
##           Preop Postop Dead
## Preop    0.8   0.00 0.20
## Postop   0.0   0.95 0.05
## Dead     0.0   0.00 1.00
```

The transition probabilities in our model are depending on the cycle in which the cohort is in. For example, for the scenario in which surgery is delayed to 12 weeks, the cohort can not transition from *Preop* to *Postop* therefore, the probability to transition from *Preop* to *Postop* until week 12 should be 0. At week 12, the cohort is at risk of dying and the remaining cohort will get the surgery and transition to the *Postop* health state. The “switching on and off” of the transition from *Preop* to *Postop* is done by the `trans_operation` function.

This means that the transition matrix is different at different cycles. Therefore, we make a transition probability matrix for every cycle. In more technical terms, we add an extra dimension to the 3x3 matrix, a dimension of the number of cycles. This results in an array of transition probabilities. While running the model the transition probability matrix corresponding to the cycle is used to model what happens to the cohort.

In the `make_m_trans.R` - Describe how we deal with the tumor doubling rate & how the CBS data is used in addition to the disease specific data

04 Run the analysis

Based on the conceptual algorithm as described in the beginning of this paragraph and the function defined we run the model by executing the `main.R` file. At the end of the file the results are stored in the *output* folder. These results are used to generate the figures as plotted in section 3.

The files are: - `res_psa.Rdata`: a dataframe including the results of the PSA

- `psa_pooled.Rdata`: a dataframe including a summary of the PSA data
- `input_psa.Rdata` : The input data to make the PSA dataframe
- `psa_parameters.RData` : The parameter inputs for the PSA run

05 Calculate model outcomes

The main outcome of the model is the expected QALYs at each time point of surgery. These values are informative, but it might be hard to interpret the effect of surgery delay. Therefore, we calculate the derivative at each time interval using the `calculateDerivative` function from the `model.R` file. This function calculated how many QALYs on average are lost per week of surgery delay within the 10 weeks interval. The more QALYs are on average lost within a week of delay the more a patient can benefit from early surgery. This could be a patient population with indication for surgery that might need to get priority for surgery. While a value of zero indicates that waiting for 10 more weeks within that time interval does not result in an expected loss in QALYs. This indicates that this might be a population with surgery indication can wait a little longer. The function also creates figures of these results, which are stored in the *figures* folder

Section 3: Disease specific results

In this section we describe the disease specific results in descending order of urgency.

Main results

The following table presents the main results of the model. The results are ordered in descending urgency.

	QALY, delay = 2 weeks	QALY, delay = 12 weeks	QALY, delay = 22 weeks
Patients with end-stage renal disease	10.69 (9.32 - 12.21)	10.63 (9.25 - 12.16)	10.55 (9.20 - 12.11)
Women with resectable mammary carcinoma	12.69 (11.05 - 14.27)	12.64 (11.01 - 14.23)	12.60 (10.98 - 14.21)

Impact on remaining QALYs

The following figures represent the impact on the remaining QALYs by delaying surgery. The first figure represents the population for whom the most urgency is required, because delaying this surgery results in the fastest drop in remaining QALYs.

Urgency trend

The following figures present the impact of delay on the slope of the line, the derivative. That means that if the line goes down, the urgency goes up (more QALYs are lost per week). This can be used to seek when an intervention becomes especially critical to perform.

```
all_images <- vector(mode = "character", length = length(order_populations))
order_populations <- as.vector(order_populations)

for(i in 1: length(order_populations)){
  p <- order_populations[i]
  p <- gsub(' ', "_", p)
  all_images[i] <- paste("../figures/", p, "_derivatives.png", sep = "")
}

knitr::include_graphics(all_images)
```

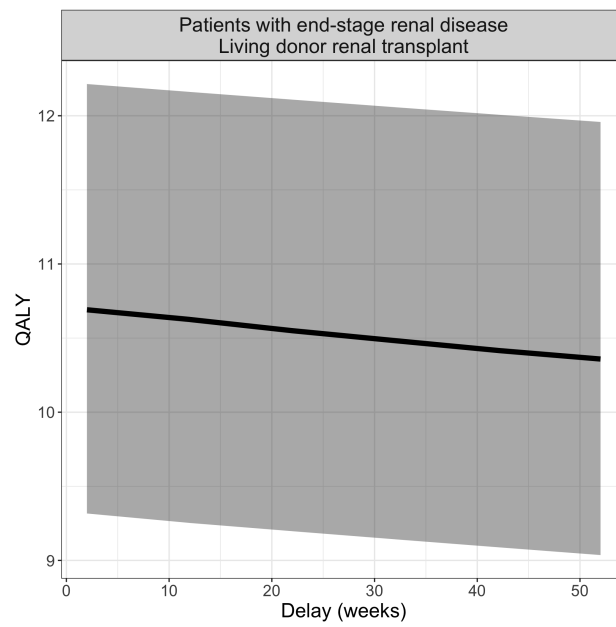


Figure 2: Dummy figure

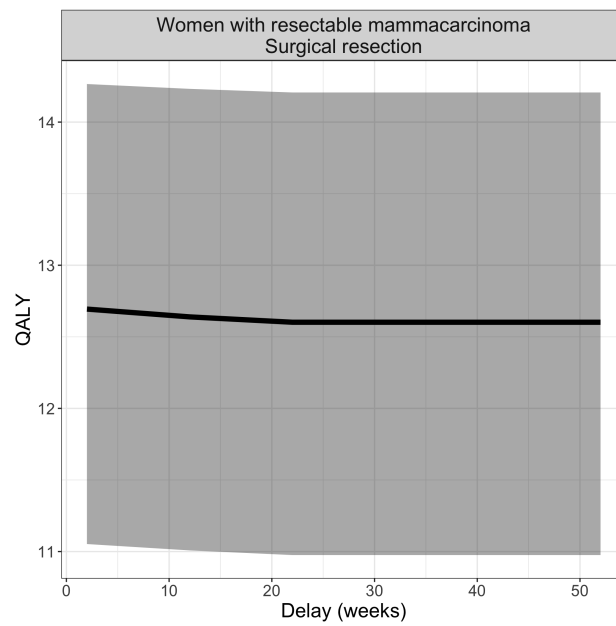
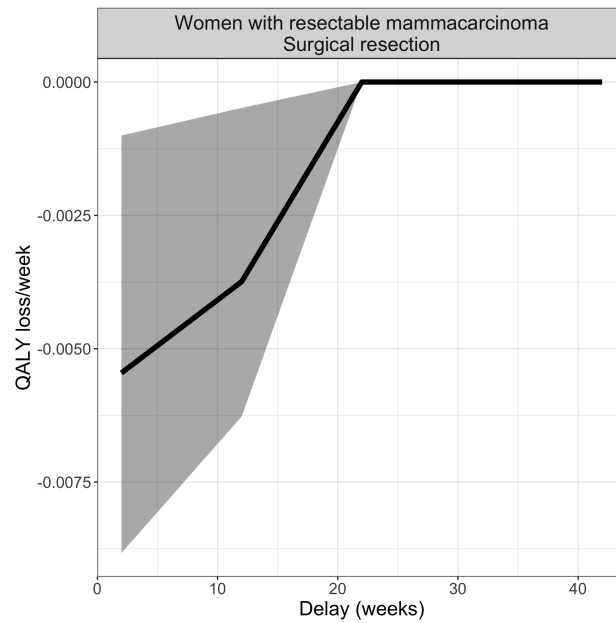
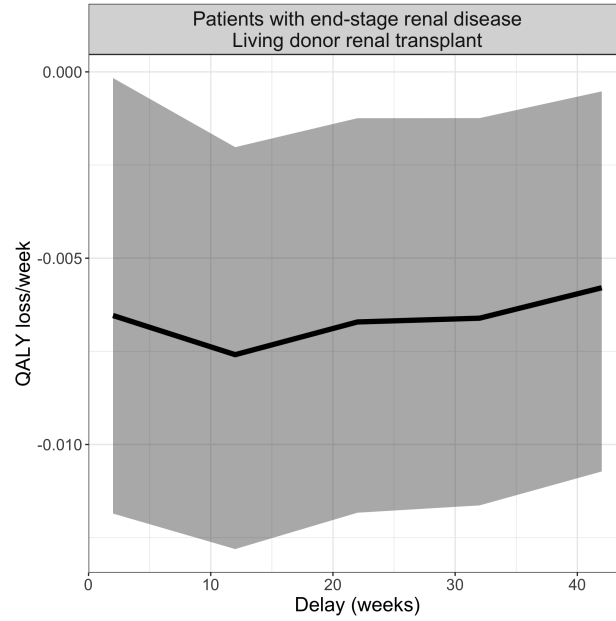
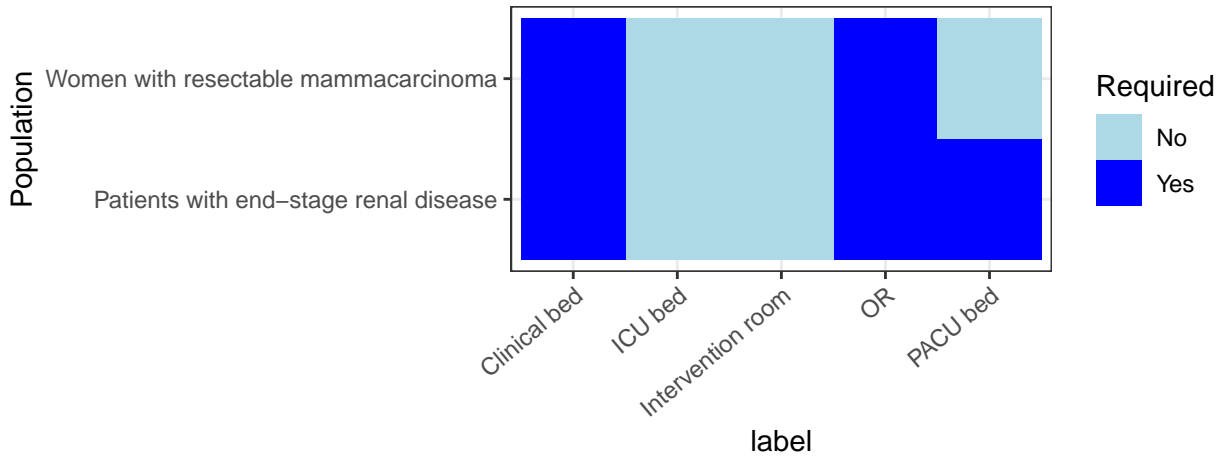


Figure 3: Dummy figure



Decision tool

Finally, we present here the requirement of specific capacity to facilitate an intervention. We have ordered the figure in descending order of urgency. Therefore, this tool can be used to choose what interventions need to be prioritized, and what capacity is needed.



Acknowledgement

For this research we made use of the template developed by the Decision Analysis in R for Technologies in Health (DARTH) workgroup: <http://darthworkgroup.com>.

The notation of our code is based on the following provided framework and coding convention: Alarid-Escudero, F., Krijkamp, E.M., Pechlivanoglou, P. et al. A Need for Change! A Coding Framework for Improving Transparency in Decision Modeling. *PharmacoEconomics* 37, 1329–1339 (2019). <https://doi.org/10.1007/s40273-019-00837-x>.

We would like to acknowledge XXX names of people that helped with this research (and are not authors of the document).

Funding/rewards that need to in included?