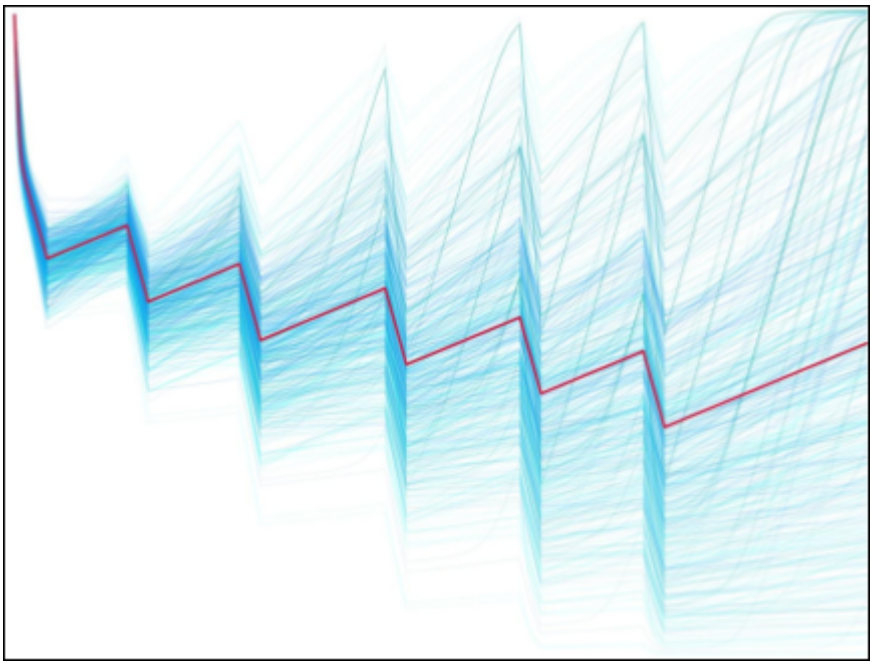
UNIVERSITY OF GOTHENBURG

DEPARTMENT OF PHYSICS

MODELLING CANCER DYNAMICS DURING THERAPY FROM LIQUID BIOPSY MEASUREMENTS



Simone Piccioni
Charlotte Preunkert

Degree project for Master of Science (120 hec) with a major in Physics with Specialization in Complex Adaptive Systems
2024, 120 HEC
Second Cycle

Master's Thesis 2024
Modelling cancer dynamics during therapy from liquid biopsy measurements

Supervisor: Eszter Lakatos
Examiner: Sviatlana Shashkova and Marija Cvijovic

Department of Physics
Division of Complex Adaptive Systems

University of Gothenburg
SE-412 96 Gothenburg

Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Computing Disease Evolution in Cancer

Chalmers University of Technology
SE-412 96 Gothenburg

Cover: Trajectories of a hierarchical model, the red one is generated with the mean of the posteriors.

Abstract

Liquid biopsy measurements make it possible to monitor the progression of solid cancers in almost real-time. By simulating the interaction between tumour growth and treatment, mathematical models can be used to predict the tumour evolution. In this thesis, multiple mathematical models were fitted to liquid biopsy measurements of neuroblastoma patients. The models ranged from well-known and relatively simple models, such as Logistic or Bertalanffy, to more complex ones, which take two sub-populations and resistance or regrowth into account. Different inference methods, including maximum likelihood estimation and hierarchical modelling, were compared. Additionally, we checked if the models were structurally identifiable using synthetic data.

We showed that multiple-population models fit the data significantly better compared to single-population ones. This highlights the complexity of the disease and could explain for the bad prognosis related to it. In particular, the Fast-Slow Exponential model offers a simple model that outperforms all tested 1D-models. This model assumes that there are two sub-populations, which both decay exponentially under treatment, but at different rates.

Keywords: mathematical modelling, neuroblastoma, liquid biopsy, MTM, multi-population models.

Acknowledgements

We want to thank our supervisor Eszter Lakatos for her help, guidance and kindness, that made all of this possible!
Special thanks go to the "*Math-Biology Group*" for the weekly meetings in which we received support and valuable feedback, and the occasional amazing lunches at Wijkanders.
And finally, we want to thank the doctors and researchers from Sahlgrenska University Hospital who provided us with the data, answered medical questions and even showed us around the laboratories.

Charlotte Preunkert and Simone Piccioni, Gothenburg, June 2024

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ABC-SMC	Approximate Bayesian Computation Sequential Monte Carlo
AIC	Akaike Information Criterion
cfDNA	cell-free DNA
COJEC	cisplatin [C], vincristine [O], carboplatin [J], etoposide [E], and cyclophosphamide [C]
ctDNA	circulating tumour DNA
IVP	Initial Value Problem
L-BFGS-B	Limited memory Broyden-Fletcher-Goldfarb-Shanno with Bounds
MLE	Maximum Likelihood Estimation
MTM	Mean Tumour Molecules
NUTS	No-U-Turn Sampler
ODE	Ordinary Differential Equation
PSO	Particle Swarm Optimisation
S-LS	Sensitive - Less Sensitive
SLSQP	Sequential Least Squares Quadratic Programming

Contents

1
Introduction

Liquid biopsy measurements are revolutionising many areas of medicine, including cancer detection and monitoring. They make it possible to monitor the progression of the disease in almost real-time while being minimally invasive. This opens the door for personalised treatments. The goal of personalised cancer therapy is to administer the right drug combination for the right person. This leads to a more effective treatment with less toxicity, as the exposure to ineffective treatments is reduced [25].
Mathematical models can simulate the interplay between tumour growth and treatment, enabling the prediction of tumour evolution, see [14], and the identification of optimal treatment, see [11].

In this chapter, we provide an overview of the medical background, the dataset, and the aims of this project.

1.1 Medical background

1.1.1 Liquid biopsy and MTM

A liquid biopsy is the sampling and analysis of non-solid biological tissue. It is minimally invasive and used for diagnosing and monitoring multiple diseases, such as cancer. It provides a methodology for obtaining tumour-derived information from bodily fluids. Many bodily fluids can be biopsied, but blood is most commonly used. From the blood, cell-free DNA (cfDNA) can be isolated [16].

cfDNA is mainly released by apoptosis (natural cell death), necrosis (cell death caused by external factors), and secretion (controlled release of substances by cells or tissues) [2].
Circulating tumour DNA (ctDNA) is a type of cfDNA, which originates from a tumour. Tumour-specific mutations can be found in ctDNA, see Figure 1.1.

Tumour burden within liquid biopsies can be quantified using MTM, which stands for **M**ean **T**umour **M**olecules. MTM counts the number of DNA copies carrying previously identified cancer-specific mutations present in the cfDNA per ml of plasma. It has been shown that MTM is a good measure of molecular disease burden [10].

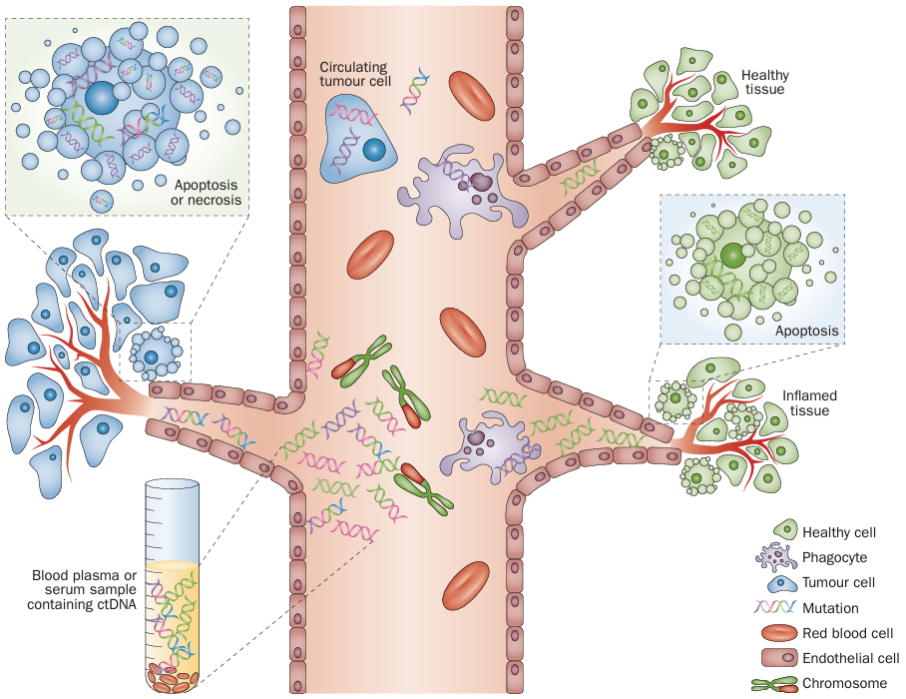


Figure 1.1: Release of cfDNA into the blood by healthy and tumour cells. Reproduced from [4] with permission from Springer Nature.

1.1.2 Neuroblastoma

This section is based on [15].
Neuroblastoma is a cancer, which originates from neural crest cell derivatives in the sympathetic nervous system. Most cases are diagnosed in children up to five years old. Neuroblastic tumours arise in the adrenal medulla or in the paraspinal sympathetic ganglia of the neck, chest, abdomen, or pelvis. The most common presentation of neuroblastoma is a painless abdominal mass.

At the time of diagnosis, the tumour is classified into different clinical prognostic categories based on, among others, age of the patient, tumour location and metastasis. Based on that, patients are divided into non-high-risk or high-risk neuroblastoma. Non-high-risk patients are usually treated by surgery, chemotherapy together with surgery or sometimes just monitored, without interventions. They have high survival rates.
High-risk patients currently have an overall five-years survival rate below 50%. New treatments, which decrease relapse and prevent disease progression are needed. Currently, most patients receive chemotherapy, then surgery, followed by consolidation therapy.

1.1.3 Rapid COJEC for high-risk neuroblastoma

Rapid COJEC is a standard chemotherapy schedule for high-risk neuroblastoma patients [13]. Patients undergo eight cycles with ten days between two cycles. Three different courses (A/B/C) are alternated, in which two or three out of the five

different drugs (cisplatin, vincristine, carboplatin, etoposide and cyclophosphamide) are given to the patient over two days. Table 1.1 shows the recommended rapid COJEC schedule.

Day	0	10	20	30	40	50	60	70
Course	A	B	C	B	A	B	C	B
Vincristine	↓	↓	↓	↓	↓	↓	↓	↓
Carboplatin	↓				↓			
Etoposide	↓↓		↓↓		↓↓		↓↓	
Cisplatin		←		←		←		←
Cyclophosphamide			↓↓				↓↓	

Table 1.1: Recommended rapid COJEC schedule, adapted from [13].
↓: drug is given on the first day of the cycle,
↓↓: drug is given on both the first and the second day of the cycle,
←: infusion of the drug over 24h starting on the first day of treatment 4h after the first drug is given.

Rapid COJEC aims to administer higher single doses over a substantially shorter treatment period, with shorter intervals between cycles, compared to standard schedules. This makes further treatment, like surgery, earlier possible and by that, possibly, improves survival [19].

1.2 Dataset

The dataset comes from Sahlgrenska University Hospital. It consists of 13 neuroblastoma patients. We excluded one patient from our analysis because it only contained data at a single time point. Eight patients were diagnosed with high-risk neuroblastoma, and the other four patients with non-high-risk neuroblastoma.
For each patient individually ten genomics locations were selected, on which mutations were monitored. The MTM is calculated as the sum of detected mutations on those ten genes per ml of plasma.

First Treatment	10.07.2020	0				C160P1_TNFRSF11B_118924519		C160P2_KNG1_186725120		C160P3_ACSM6_95207285	
Timepoint	Date	Days	total cfDNA (ng/ml)	Loaded cfDNA (ng)	MTM/ml plasma						
C160 - PRET	10.07.2020	0	125,9210526	30,22	58452,03593	2986	1917	2840	1060	2727	1253
C160 - A2	20.07.2020	10	10,06329114	34,07	1989,914658	2132	1085	1298	351	1790	629
C160 - A5	19.08.2020	40	8,486842105	25,08	282,8947368	1461	125	765	46	1261	93
C160 - A6	31.08.2020	52	7,125								
C160 - A7	10.09.2020	62	7,06								
C160 - A8	21.09.2020	73	10,54	36,13	124,2745641	1533	64	839	35	1374	54
C160 - POST-HD	15.03.2021	248	8,84	30,09	0,587570621	1207	1	808	0	1107	0
C160 - POST-RT	27.05.2021	321	6,53	20,88	0	409	0	235	0	247	0
C160 - MT4	19.10.2021	466	6,75	21,6	0,3125	2094	0	493	0	707	0

Figure 1.2: A snapshot of the data available for patient C160.

The dataset contains measurements of cfDNA, MTM and the number of reads and mutations of the ten genes. For most patients, the dataset also includes basic information about when in the treatment the measurement was taken, like PRET standing for pre-treatment or POSTOP for after surgery. For some time points the cfDNA measurements are available, but no information about mutations. The

number of measurements range from 5 to 27.
In addition to that, the tumour volume was monitored. Those measurements were usually taken on different days than the liquid biopsy measurements.

For the eight high-risk patients, we had access to detailed information about the treatment, including which medication was given to the patient at a certain time point. For an example, see Figure 1.3. This information is lacking for the non-high-risk patients.

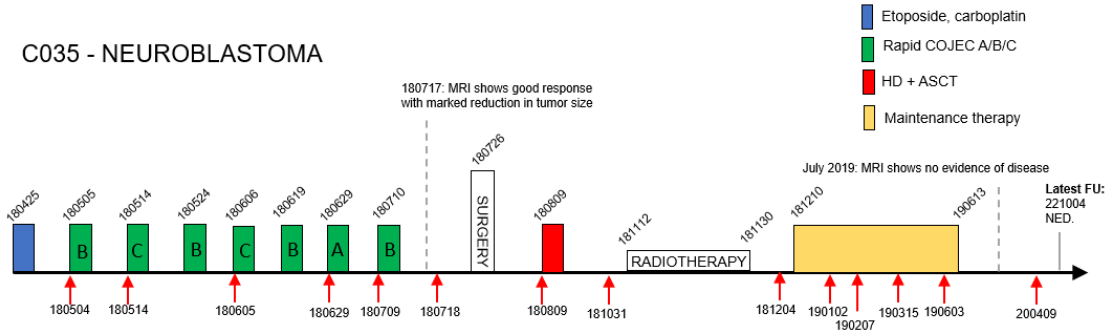


Figure 1.3: Treatment plan for high-risk patient C035. The red arrows mark when the liquid biopsy measurements were taken.

1.3 Aim and structure

Liquid biopsies are already used to monitor the progression of cancer, allowing also for personalised treatments. However a description of the underlying dynamic of the disease under therapy is still lacking.
This thesis aimed to formulate and test mathematical models that describe the cancer dynamics under treatment and take noise into account, to uncover this mechanism and make predictions. Patient-specific parameters were inferred for each model. Several parameter inference methods were tested. Then, the models were compared, taking both the model performance and the complexity of the model into account. The goal was to find at least one model, which fits most patients well and analyse it more in detail.

We focused on the high-risk patients since detailed treatment information is only available for them. Furthermore, we only modelled the dynamics during chemotherapy up until surgery. In fact, the dataset does not contain many measurements after surgery and ideally, most of the cancer is removed afterwards and the MTM values stay very small or are equal to zero.
We calculated the correlation coefficient ρ, between MTM and cfDNA levels and found that seven out of eight high-risk patients have $\rho > 0.95$ and only one has $\rho = 0.75$. Moreover, MTM is assumed to be less noisy compared to cfDNA, which is more sensible to biological fluctuations, or tumour volume measurements. For these reasons, we focused on the MTM values.

The thesis is organised as follows. Chapter 2 contains all models, both ODE and statistical ones, which we originally intended to test. In Chapter 3 the mathematical methods for optimisation, statistical inference and model selection are introduced. In Chapter 4 the implementations and results are discussed, followed by further modifications to some models of Chapter 2, with the subsequent analysis of the results. The results are summarised in Chapter 5 and further research questions are discussed.

2
Models

2.1 Binomial models

We first explored a simple statistical model to explain the observed mutated molecule counts without modelling dynamic interactions. An assumption that can be made is that the selected genomics locations either carry a mutation or not, resulting in a binomial distribution, for the number of mutations.

For a given patient, we labelled the different genomics locations used in the dataset to calculate the MTM values with $j = 1, 2, \ldots, 10$, and considered the random variables $X_j(t)$:

$$X_j(t) \sim \text{Bin}(n_j(t), p_j(t)). \tag{2.1}$$

$X_j(t)$ models the number of DNA copies that have a mutation on the jth genomics location, in the sample at time t. $n_j(t)$ is the total number of DNA copies used in the sample to check for mutations, i.e. number of DNA segments detected with a given genomics location, which are given in the dataset.
Different hypotheses on $p_j(t)$ are tested in the following sections.

2.1.1 Homogeneous binomials

The first hypothesis that we formulated assumes that for a given patient, the frequencies $p_j(t)$ are all equal, or equivalently $p_1(t) = p_2(t) = ... = p^*(t)$. This means that for each patient, all mutations occur with the same frequency, which can change over time.
To estimate $p^*(t)$, we calculated the empirical average rate at a given time t:

$$\hat{p}(t) = \frac{1}{N} \sum_{j=1}^{N} \frac{x_j(t)}{n_j(t)} \tag{2.2}$$

where N is the number of genomics locations.
For each genomics location j, we computed a rejection region with a significance level $\alpha = 5\%$, assuming that $p_j(t) = \hat{p}(t)$. This gave us the following rejection regions:

$$[0, l] \cup [u, 1] \tag{2.3}$$

with

$$\mathbb{P}(X_j(t)/n_j(t) < l) = 2.5\% \text{ and } \mathbb{P}(X_j(t)/n_j(t) > u) = 2.5\%. \tag{2.4}$$

We then checked how many mutation frequencies were inside or outside the rejection region, to see whether the hypothesis could be rejected or not.

2.1.2 Decaying binomials

The second hypothesis that we tested assumes different initial mutation frequencies for all the genomics locations, but a common exponentially decaying factor:

$$X_j(t) \sim \text{Bin}(n_j(t), c_j e^{-at}). \tag{2.5}$$

We used the first observation to estimate $c_j = x_j(t_1)/n_j(t_1)$, where the $x_j(t_1)$ is the actual number of mutated DNA copies at time t_1. The variable a, was estimated by maximising the log-likelihood of all the different genes combined, at all successive time steps $(t_2, t_3, \dots)$:

$$l(a|x) = \text{const.} + \sum_j \sum_{i=2,\dots} (n_j(t_i) - x_j(t_i)) \ln(1 - e^{-at_i} c_j) - x_j(t_i) a t_i. \tag{2.6}$$

We then computed confidence intervals at each time step to analyse the behaviour of each mutation frequency in time.
We approximated the variance to obtain the following expression for the confidence intervals:

$$\left[\hat{p}_i - 1.96 \sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{n_i}}, \hat{p}_i + 1.96 \sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{n_i}} \right], \tag{2.7}$$

where $\hat{p}_i = c_j e^{-at_i}$ for some genomics location j.

An interesting feature of this particular model is that calculating the expected value of the sum of all the different mutations:

$$\mathbb{E}\Big[\sum_j X_j(t) \Big] = e^{-at} \Big(\sum_j n_j(t) c_j \Big) \tag{2.8}$$

gives us a link with an exponential decaying ODE model, see Equation 2.9. In fact, the sum $\sum_j X_j(t)$ is directly proportional to the MTM value. This makes the exponential ODE a deterministic counterpart to this binomial model, with the extra assumption that the term $\sum_j n_j(t) c_j$ remains constant over time (this would be the case if the number of reads per genomics location is always approximately the same).

2.2 ODE models

Next, we explored ordinary differential equation (ODE) models that can capture the cancer dynamics over time.

2.2.1 Classical models

Simple population models are commonly used to simulate tumour growth/decay, e.g. see [14, 26].
The models we considered are:

- **Exponential**, which is the simplest non-trivial ODE model:

$$\dot{X} = rX, \tag{2.9}$$

  where the parameter r is usually called the growth/death rate (depending on the sign and the context).

- **Logistic**, which introduces the carrying capacity K:

$$\dot{X} = rX\left(1 - \frac{X}{K}\right), \tag{2.10}$$

  the parameter r is called the growth/death rate.

- **Bertalanffy**, which is a popular model for the growth of tumours:

$$\dot{X} = rX^{\frac{2}{3}} - \delta X. \tag{2.11}$$

  When X is assumed to be the tumour volume, the first term models a growth based on the tumour's surface. This comes from the assumption that on the inside of the tumour fewer resources are available, which makes it less favourable for reproduction. Thus, the growth of the tumour mass happens only on the surface, which is assumed to be proportional to $X^{2/3}$. The second term is a standard death term, proportional to the total tumour mass.
  The model's carrying capacity is $(r/\delta)^3$.
- **Gompertz**, which has been shown to describe the growth of various types of tumours, e.g. human breast cancer [17]:

$$\dot{X} = X\left(r - \delta \ln X\right). \tag{2.12}$$

  The model's carrying capacity is $e^{r/\delta}$.

2.2.2 Stairs models

Next, we modelled the effect and timing of treatment explicitly by introducing Stairs models, named after the stair-case-like dynamics of the treatment. The underlying assumption is that under treatment the cancer population decreases rapidly, while without treatment it grows.

Exponential Stairs

The Exponential Stairs model is a piece-wise exponential function. It assumes that the cancer is exponentially decreasing at a rate $\delta \geq 0$ when under the influence of a drug and otherwise grows at a rate $r \geq 0$. The time T for how long a drug influences the cancer can be treated as an additional parameter or fixed to a value based on knowledge about the drug.

$$\dot{X} = \begin{cases} rX & \text{if no treatment} \\ -\delta X & \text{if under treatment} \end{cases} \tag{2.13}$$

Bertalanffy Stairs

The Bertalanffy stairs model is an extension of the Bertalanffy model, see Equation 2.11, with additional exponential decay under treatment:

$$\dot{X} = \begin{cases} rX^{\frac{2}{3}} - \delta X & \text{if no treatment} \\ -(\delta + \delta_1)X & \text{if under treatment.} \end{cases} \tag{2.14}$$

In the absence of treatment, the model is identical to the Bertalanffy model, but when the patient is given a drug and afterwards, for a time T, the growth rate is turned off ($r = 0$), and the death rate is increased further by an additional parameter δ_1. Again, T can be treated as an additional parameter or be fixed.
During parameter estimation, in order to ensure that the carrying capacity is not below the first observation $y_{obs}(0)$, we considered the following constraint:

$$1.1 \cdot y_{obs}(0) < \left(\frac{r}{\delta}\right)^3. \tag{2.15}$$

The value 1.1 was arbitrarily selected.

2.2.3 S-LS (Sensitive - Less Sensitive)

Most solid tumours, such as neuroblastoma, likely represent heterogeneous populations of cells with varying chemosensitivities [7]. This heterogeneity can be modelled explicitly by including more than one type of cancer cells in the model.
A commonly used model incorporates the assumption that there are two different populations S and R, which respond differently to the treatment. S will die when the treatment is ongoing, while R is assumed to be totally resistant to it.
This can be written as an ODE system [11]:

$$\begin{cases} \dot{S} = r_S S(1 - \frac{S+R}{K}) - \delta S \\ \dot{R} = r_R R(1 - \frac{C \cdot S+R}{K}). \end{cases} \tag{2.16}$$

The parameter C models competition between the two populations and can be interpreted as how many more resources S is using compared to R.
The parameters r_S, and r_R are the growth-rates, while K is the overall carrying capacity and δ the death rate.

The total tumour burden is assumed to be proportional to the sum of the two sub-populations:

$$Y = S + R. \tag{2.17}$$

The COJEC treatment consists of five different drugs, which makes resistance less likely. Therefore, instead of Model 2.16, we decided to assume that both subpopulations are sensitive to the chemotherapy, but to different degrees. For this reason, we added a death rate also for S_2 (what was previously known as R) that

satisfies the condition $\delta_2 << \delta_1$, where δ_1 is the death rate of the sensitive population S_1 (previously S).

To avoid the introduction of a new parameter and to merge all the parameters related to the advantage of one sub-population over the other, we assumed the following form for the death rate:

$$\delta_2 = \frac{\delta_1}{C}. \tag{2.18}$$

This gives the following ODE system:

$$\begin{cases} \dot{S}_1 = r_1 S_1 (1 - \frac{S_1+S_2}{K}) - \delta_1 S_1 \\ \dot{S}_2 = r_2 S_2 (1 - \frac{CS_1+S_2}{K}) - \frac{\delta_1}{C} S_2. \end{cases} \tag{2.19}$$

This model is called the Sensitive-Less Sensitive model or S-LS, which is a variation of the competitive Lotka-Volterra model with added removal.
To actively include the therapy in the model, when a patient receives a drug and for the time that this remains in the body, T, the growth rates are both set to zero, while δ_1 is set to a non-zero value. Instead, when a patient is not affected by the drug anymore, $\delta_1 = 0$ and the growth rates are set to their original values.

To get some general information about the dynamic, it is interesting to analyse the fixed points of the system, and their stability.
When treatment is on, Equation 2.19 reduces to

$$\begin{cases} \dot{S}_1 = -\delta_1 S_1 \\ \dot{S}_2 = -\frac{\delta_1}{C} S_2, \end{cases} \tag{2.20}$$

which gives two decaying exponentials, with the decaying rate of the less sensitive part smaller than the sensitive part.
When treatment is off, Equation 2.19 becomes:

$$\begin{cases} \dot{S}_1 = r_1 S_1 (1 - \frac{S_1+S_2}{K}) \\ \dot{S}_2 = r_2 S_2 (1 - \frac{CS_1+S_2}{K}). \end{cases} \tag{2.21}$$

The fixed points associated with it are $(0, 0)$, $(K, 0)$ and $(0, K)$. The stability analysis gives the following results:

Fixed Point	Stability
$(0, 0)$	Unstable, 2 positive eigenvalues
$(K, 0)$	Stable, 2 negative eigenvalues
$(0, K)$	Unstable, 1 negative eigenvalue, 1 equal to 0

The stability/instability of the fixed points follows from the choices of bounds for the parameters, see Table A.1. Figure 2.1 shows examples of the vector fields of Equation 2.21.

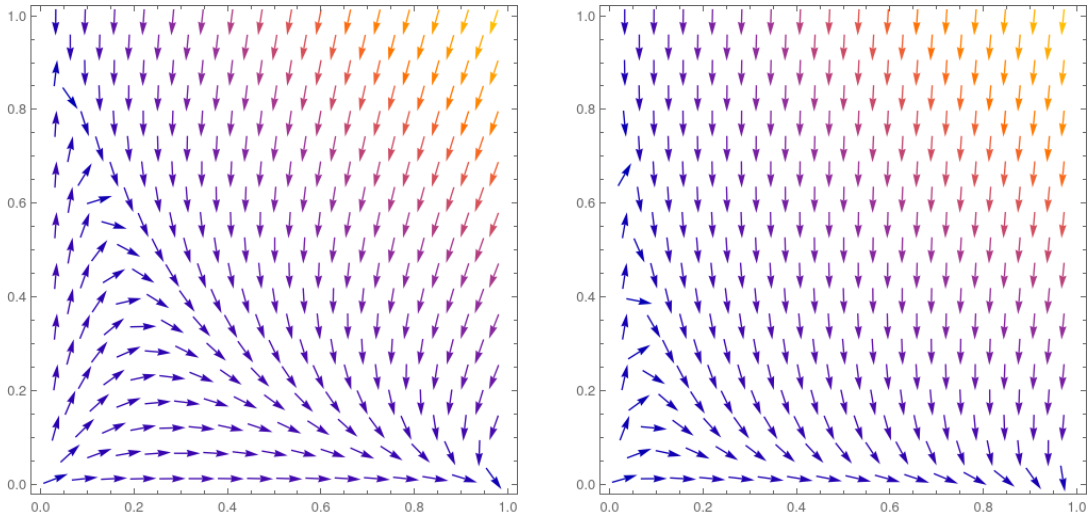


Figure 2.1: Vector field of Equation 2.21 non-dimensionalised ($x = S_1/K$ and $y = S_2/K$) with $r_1, r_2 = 1$, $C = 3$ on the left and $C = 10$ on the right.

In the special case $C = 1$, $S_1 + S_2 = K$ results in line of fixed points and the trajectories of the system are contained in a simpler one:

$$\begin{cases} \dot{S}_1 = r_1 S_1 \\ \dot{S}_2 = r_2 S_2, \end{cases} \tag{2.22}$$

which is a simple exponential growth/decay in both variables, until the line of fixed points is reached.
In general, we can say that if treatment is off and the initial sensitive population is non-zero, then the system will approach the equilibria $(K, 0)$.

When implementing this model, we decided to further reduce the number of parameters by considering the associated system in which time was reparametrised as $\tau = t/r_1$, and all the other parameters accordingly (we kept the same symbols), which is equivalent to just setting $r_1 = 1$. From now on r_2 is renamed as r.

2.2.4 S-LS*

Given the large number of parameters associated with the S-LS model, and the small dataset available, we decided to fix some parameters.
Following inference on the full S-LS model, we aimed to further reduce the number of parameters to be estimated from the measurements. Therefore, by looking at the mean of the estimated parameters, we fixed $K = 1.1 \cdot y_{obs}(0)$, where $y_{obs}(0)$ is the first observation. And we fixed $T = 2$ because of the nature of the COJEC therapy. The resulting model, given by Equation 2.19 with this choice of parameters was then called S-LS*, to tell it apart from the previous one.

3
Methods

3.1 Optimisation

To estimate the parameters of the models, functions, like a cost or likelihood function, had to be optimised. We used the optimisation algorithms L-BFGS-B, SLSQP and PSO, which are described in the following sections.

3.1.1 L-BFGS-B and grid search

The L-BFGS algorithm is based on a descent algorithm:

Algorithm 1 Descent algorithm

Choose a starting point x_0 and set $k = 0$
for $k = 1, \ldots,$ max_iters **do**
  Determine a descent direction d_k
  Determine a step length α_k, such that $f(x_k + \alpha_k d_k) < f(x_k)$
  Update $x_{k+1} = x_k + \alpha d_k$
  if termination criteria are satisfied **then** break
end for
return x_k

Newton's method, which is known for fast convergence, sets $d_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$. However, the computation of the inverse Hessian matrix $\nabla^2 f(x)$ can be computationally expensive. Quasi-Newton methods, like BFGS, instead compute an approximation to it. L-BFGS is a limited memory Quasi-Newton method, which represents the approximation to the Hessian obtained by the BFGS algorithm in outer product form and thus saves memory. L-BFGS-B is an extension of the L-BFGS algorithm, taking simple bounds on variables into account [9, 27].

We used the option `'l-bfgs-b'` available in the `scipy minimize` function as optimisation method [22] .

Newton's method, and thus also Quasi-Newton methods, do not guarantee global convergence. We observed that when minimising a complex objective function with potentially many local minima, the choice of initial parameters influenced the result. Therefore, we did the following:

1. Generate a grid of initial positions in the parameter space

2. For each initial position i, run L-BFGS-B and save the position x_i, to which L-BFGS-B converges
3. Choose the x_{i^*}, which has the lowest objective function value.

We called this method L-BFGS on a grid.

3.1.2 SLSQP for constrained optimisation

SLSQP stands for Sequential Least Squares Quadratics Programming and is a sequential quadratic programming (SQP) optimisation algorithm. SQP is one of the most effective methods for non-linearly constrained optimisation problems [9]. The algorithm optimises problems of the form:

$$\min_x f(x) \tag{3.1}$$
$$\text{s.t. } c_i(x) \geq 0,\ i \in \mathcal{I} \tag{3.2}$$
$$c_i(x) = 0,\ i \in \mathcal{E} \tag{3.3}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c_i : \mathbb{R}^n \to \mathbb{R}$.

Like L-BFGS-B it is based on a descent algorithm. To determine the descent direction, a quadratic approximation of the Lagrange function $L(x, \lambda) = f(x) - \sum_{j \in \mathcal{I} \cup \mathcal{E}} \lambda_k c_j(x)$ and a linear approximation of the constraints leads to the following optimisation problem:

$$\min_{d \in \mathbb{R}^n} \frac{1}{2} d^T B_k d + \nabla f(x_k) d \tag{3.4}$$
$$\text{s.t. } \nabla c_i(x_k) d + c_i(x_k) = 0,\ i \in \mathcal{E} \tag{3.5}$$
$$\nabla c_i(x_k) d + c_i(x_k) \geq 0,\ i \in \mathcal{I}, \tag{3.6}$$

where $\nabla f(x)$ and $\nabla c_i(x)$ are row vectors and $B = \nabla^2 L(x, \lambda)$, but in practice usually approximated by a Quasi-Newton method.
Algorithms to solve the linearised optimisation problem are described in [12]. We used `scipy`'s `minimize` function [22], which has the option to use `'slsqp'`. SLSQP is only a local optimisation algorithm, so we followed the same strategy as described in the previous section and considered multiple starting points within the constraints.

3.1.3 Particle Swarm Optimisation (PSO)

Particle Swarm Optimisation is a stochastic optimisation algorithm that was inspired by actual swarming mechanisms that can be found in nature, for example in bird flocks or fish schools [23].
The core actors of the algorithm are the *particles* (points in the parameter space), that together form a swarm of size N.

The algorithm is summarised in Algorithm 2, where PSO is implemented to find the global minimum of a function $f(x)$ over the subspace $[m_1, M_1] \times \cdots \times [m_n, M_n]$,

where n is the number of parameters.

Algorithm 2 PSO algorithm

for $i = 1, \dots, N; \quad j = 1, \dots, n$ **do**
Draw $r_{ij} \sim U([0,1])$
Initialise the ith particle's position x_i:

$$x_{ij} = m_j + r_{ij}(M_j - m_j)$$

Initialise the ith particle's velocity v_i:

$$v_{ij} = \frac{\alpha}{\Delta t}\left(- \frac{M_j - m_j}{2} + r_{ij}(M_j - m_j)\right)$$

end for
for $s = 1, \dots,$ max_iters **do**
for $i = 1, \dots, N$ **do**
if $f(x_i) < f(x_i^{pb})$ **then** $x_i^{pb} \leftarrow x_i$ ▷ particle's best
if $f(x_i) < f(x^{sb})$ **then** $x^{sb} \leftarrow x_i$ ▷ swarm's best
Draw $r_i, q_i \sim U([0,1])$
Update the velocity:

$$v_i \leftarrow w v_i + c_1 q_i \left(\frac{x_i^{pb} - x_i}{\Delta t} \right) + c_2 r_i \left(\frac{x^{sb} - x_i}{\Delta t} \right)$$

if $|v_i| > v_{max}$ **then** limit v_i i.e. $v_i \leftarrow v_i \frac{v_{max}}{|v_i|}$
Update the position:

$$x_i \leftarrow v_i \Delta t$$

Ensure that the positions are in the subspace
end for
if termination criteria are satisfied **then break**
end for
return x^{sb}

The parameters c_1, c_2 are usually set at the beginning of the algorithm and kept constant. They model the behaviour of the algorithm, namely, if we want to favour *exploration* or *exploitation*. If $c_1 > c_2$, the individual particle's history will have a bigger impact on the velocity update, which leads to more exploration. On the other hand with $c_1 < c_2$ the swarm's best value will have a bigger impact on the velocity, so all the particles will be more attracted around the same point, hence favouring exploitation.
An additional constraint for the choice of these parameters is $c_1 + c_2 < 4$, as discussed in [23].
The parameter w, commonly called inertia weight, allows us to also choose between a more exploratory or exploitative search. In fact, for $w > 1$, the velocity will be influenced more by its previous value, and this will favour more *exploration*, by hav-

ing less cohesion in the swarm. The opposite happens for $w < 1$ which will then favour *exploitation*.
A common approach is to keep $w > 1$ for the initial iterations to scan more space, and then turning it to $w < 1$ towards the last iterations, to find better approximations for the possible candidate.
Finally, v_{max} is set at the beginning, to ensure that the velocities do not become larger and larger.

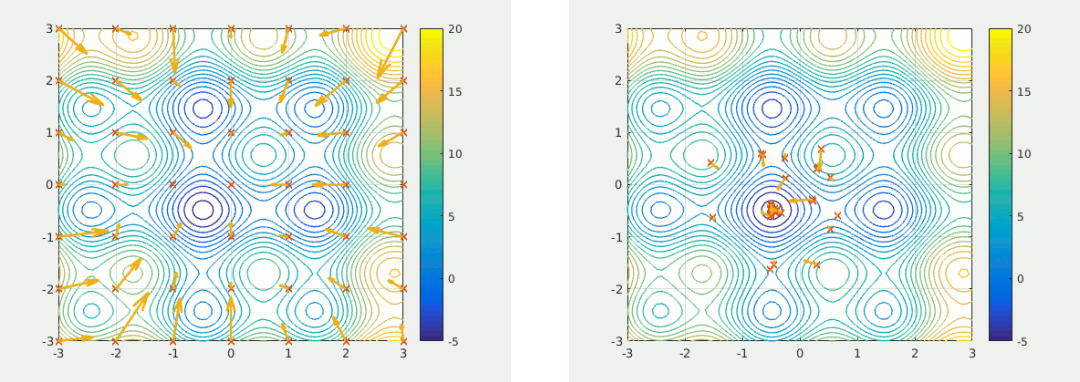


Figure 3.1: Example of PSO to find the global minimum of a function, from [24]. On the left, is the algorithm after a few iterations. On the right the algorithm after many iterations, when most of the swarm got closer to the minimum.

We used the Python implementation `PySwarms` [8].

3.2 Maximum likelihood estimation

The likelihood function $L(\theta|y)$ is the probability of observing data $y = \{y_j\}_{1 \leq j \leq n}$ assuming that θ is the actual parameter. For continuous probability distributions, maximising the likelihood function for one observation y_j is equivalent to maximising the probability density $f(\theta|y_j)$.

It is often assumed that the measurement errors are independent and normally distributed with mean $\mu = 0$ and standard deviation σ. In our dataset, the MTM measurements span multiple orders of magnitude. If the measured value is low, a measurement error of e.g. ten seems huge, whereas if the value is multiple hundreds of thousands, a similar measurement error is virtually negligible. Therefore, we do a log transformation on both the data and the predictions, adding a small constant ε to avoid $\ln(0)$, and assume:

$$\ln(y_{pred,j} + \varepsilon) - \ln(y_j + \varepsilon) \sim N(0, \sigma^2), \tag{3.7}$$

where $y_{pred,j}$ is generated by the model with parameter θ. Maximising the log-likelihood function l is equivalent to maximising the likelihood function since the logarithm is a strictly increasing function. Most optimisation algorithms aim to

find the minimum of a function, therefore the negative log-likelihood function is minimised. The negative log-likelihood function under assumption 3.7 is given by

$$-l = \frac{n}{2}\ln(2\pi\sigma^2) + \frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(\ln(y_{pred,j}+\varepsilon) - \ln(y_j+\varepsilon)\right)^2. \tag{3.8}$$

Since no information about the noise levels in our dataset is available, when estimating parameters by minimising the negative log-likelihood function, we treated the standard deviation σ as an additional parameter.

3.3 Bayesian inference

Bayesian inference is a common framework used in applications, especially in life sciences.
It is based on the Bayesian view of probability, in which the parameters of a model are treated as random variables with their own distributions.
The usual setup involves:

- proposing initial parameter distributions of the model $p_{\text{prior}}(\theta)$, called *priors*,
- collecting data x, and calculate the likelihood $L(\theta) = p(\theta|x)$,
- updating the priors by using Bayes' formula:

$$p_{\text{posterior}}(\theta) = \frac{L(\theta)p_{\text{prior}}(\theta)}{p(x)}. \tag{3.9}$$

The complicated step is the last one, which involves the term $p(x) = \int p(x|\theta)p_{\text{prior}}(\theta)d\theta$, which can be very difficult to evaluate, especially with many parameters.
Numerical methods are often used to overcome this.

Metropolis algorithm is summarised in Algorithm 3, see also [5]. It was one of the first such methods, and although nowadays it has been replaced with more efficient ones, the core idea still forms the basis of almost all approaches.

Algorithm 3 Metropolis Algorithm

Initialise $\theta^{(0)} \sim p_{\text{prior}}(\theta)$
for $n = 1, \ldots, N$ **do**
  Generate $\theta^* \sim q(\theta|\theta^{(n-1)})$
  Calculate $\alpha = \min\{1, \frac{L(\theta^*)q(\theta^*|\theta^{(n-1)})}{L(\theta^{(n-1)})q(\theta^{(n-1)}|\theta^*)}\}$
  Draw a random number $u \sim U([0,1])$
  Set $\theta^{(n)} = \begin{cases}\theta^* & \text{if } u \leq \alpha \\ \theta^{(n-1)} & u > 1-\alpha\end{cases}$
end for
return $\{\theta^{(i)}\}_{i=1,\ldots,N}$

In Algorithm 3, $q(\theta_2|\theta_1)$ is called transition probability or proposal distribution. It is responsible for proposing the next step θ_2, given the current one θ_1 (the particular

choice of q depends on the implementation).

After the burn-in phase, which disregards the first part of the returned sample, the algorithm returns $\{\theta^{(i)}\}_{i=n_0,\dots,N}$ which will be distributed according to $p_{\text{posterior}}(\theta)$, which avoids directly evaluating the intractable integral.
Usually, the algorithm is run several times, or even in parallel with each other, to check if there is actual convergence to a posterior distribution or not, with the results of the single runs referred to as *chains*.

More modern approaches combine Metropolis, with Hamiltonian dynamics, such as the Hamiltonian Monte-Carlo algorithm, see e.g. [5], to get more efficient samplers and have faster convergences.

Parallel to this, there are also methods to perform Bayesian inference that do not require to specify explicitly the likelihood, called Approximate Bayesian Computation, or ABC.
One very common implementation is ABC Sequential Monte Carlo or ABC-SMC, see [21]. While ABC has the advantage that it is applicable in cases where the likelihood cannot be defined, it usually requires much longer times to get convergent results, compared to other methods using likelihoods.
To run ABC-SMC we used the python library `pyABC` [20].

3.3.1 Hierarchical models

Hierarchical models, also referred to as mixed-effect or multi-level models, provide an alternative to estimating parameters for each patient separately. Some parameters, e.g. the death rate of a tumour, could be expected to be relatively similar for all patients. Therefore, it could be beneficial to estimate it for all patients together.

In the context of hierarchical modelling, each parameter can either be:

- **pooled**, i.e. exactly the same for each patient,
- **partially pooled**, i.e. it is assumed that for each patient the parameter is coming from the same distribution, which we call hyperdistribution and its parameters hyperparameters,
- **unpooled** or **free**, i.e. the parameter is completely independent from other patients.

Once it has been decided how to treat the parameters, prior distributions are chosen. Then, Bayesian methods, both with likelihood or likelihood-free methods, can be employed to obtain posteriors.
For our purposes, we assumed a Gaussian likelihood, and to perform inference, we used the method called NUTS, see [6], based on classical Hamiltonian Monte-Carlo techniques, with adaptive sampling strategies that make it even more efficient.
For the implementation, we used the python library `pyMC`, see [1].

3.3.2 Convergence metric $\hat{R}$

To evaluate whether Bayesian algorithms have converged to the posterior distributions, or whether more draws are still needed, it is useful to introduce some metric, see also [5].
If we obtained a sample $\{\theta_j^{(i)}\}_{j,i}$ where $j = 1, \ldots, m$ is the number of the chain, and $i = 1, \ldots, n$ is the number of the draw considered, we can calculate the variance in two different ways:

- Between the chains
$$B := \frac{n}{m-1} \sum_{j=1}^{m} \left(\bar{\theta}_j - \bar{\theta}\right)^2, \tag{3.10}$$
where $\bar{\theta}_j$ is the mean of θ calculated over the draws of the jth chain, while $\bar{\theta}$ is calculated over every draw of every chain. It essentially expresses the variance of the various means calculated over the different chains.
- Within the chains
$$W := \frac{1}{m} \sum_{j=1}^{m} \left[\frac{1}{n-1} \sum_{i=1}^{n} \left(\theta_j^{(i)} - \bar{\theta}_j\right)^2 \right], \tag{3.11}$$
which expresses the mean of the variances calculated over the different chains.

With these expressions, one can estimate σ^2, namely the true variance of θ, as:

$$\hat{\sigma}^2 = \frac{n-1}{n} W + \frac{1}{n} B, \tag{3.12}$$

which will approach σ^2 for $n \to \infty$.
For any finite n, W underestimates σ^2, since the chains might not have explored all the space yet, but will do so as $n \to \infty$. This motivates the introduction of:

$$\hat{R} = \sqrt{\frac{\hat{\sigma}^2}{W}}, \tag{3.13}$$

with $\hat{R} \geq 1$ for every finite n, but $\lim_{n\to\infty} \hat{R} = 1$.
Commonly used strategies are to accept the results of the inference if $\hat{R} < 1.1$ or $\hat{R} \leq 1.05$ – for this thesis, we decided to follow the latter.

3.4 Model selection with AIC

The likelihood function only reflects the conformity of the model to the observed data, but not the complexity of the model. Selecting the model with the highest likelihood will lead to choosing the most complex model, as a more complex model is more capable of adapting to the data [3].
The Akaike Information Criterion (AIC) is a widely used model selection tool, which takes L^*, the maximum value of the likelihood function, and k, the number of parameters of the model into account. It is computed as:

$$\mathrm{AIC} = 2k - 2\ln(L^*). \tag{3.14}$$

The lower the AIC, the better the model [14], thus between two models with the same likelihood, the one with more parameters will be penalised more.

The AIC works well when the number of observations n is big compared to k. If this is not the case, the AIC tends to not penalise more complex models enough [3].

3.5 Coefficient of determination R^2

In the context of linear regression, a common measure to assess the goodness of fit of the regression line to the data is the R^2 measure, aka coefficient of determination:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2}, \tag{3.15}$$

where y_i is the observed data, $\bar{y}_i$ is the empirical mean of the observed data, and $\hat{y}_i$ is the predicted data.
It takes values in $[0, 1]$, with 1 meaning that the model perfectly fits the data.

4

Implementation and results

4.1 Evaluation of statistical models

First, we evaluated how well the observed data can be explained by simple statistical models.
We did this by testing the hypotheses on the mutation frequencies, introduced in Section 2.1. Since the data spans multiple orders of magnitude, it was log-transformed using $\ln(x + \varepsilon)$, with $\varepsilon = 10^{-10}$ to avoid $\ln(0)$.

4.1.1 Homogeneous binomials

Homogeneous binomials assumes that the mutation frequency is the same at all previously picked genomics locations.

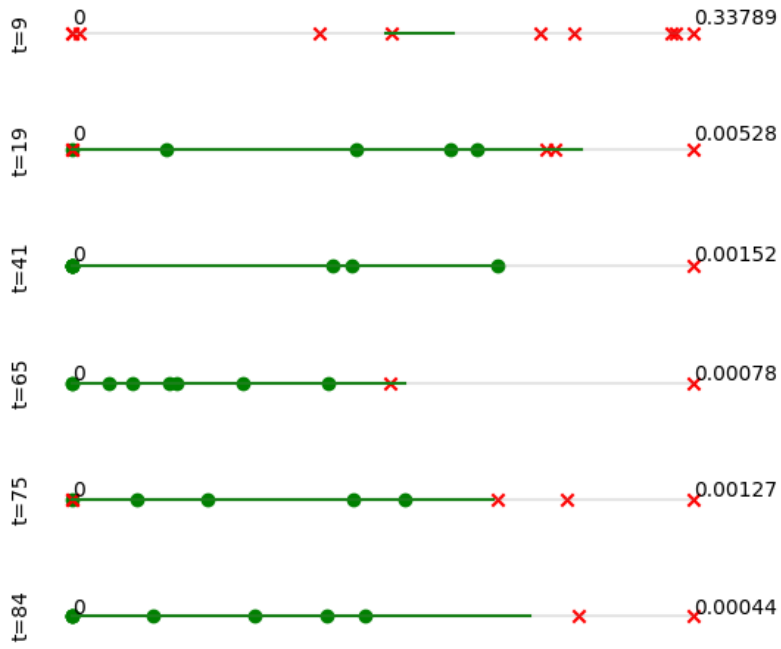


Figure 4.1: Patient C035 - The acceptance region for each mutation frequency is coloured in green, note that most of them overlap. The crosses (rejected) and circles (accepted) are the actual values for $p_j(t)$.

The results of this test all follow the same general behaviour for all patients. Namely, for the first time step only a few or no mutation frequency values at all are in the acceptance region. Hence the homogeneous hypothesis does not work at the beginning, where the decaying rate of the mutations differ from each other.
As time goes on, and the patient goes through the therapy, the number of mutations drops significantly to values close to or identically 0. Thus, the average $\hat{p}$ also goes to 0. This, combined with the discrete nature of the binomial distribution, makes the left rejection region disappear completely, leading to more and more accepted values. This is shown for one patient in Figure 4.1.

Overall we can say that the hypothesis is too strong, especially for the first time steps. Some mutations might have an advantage, and hence a bigger mutation frequency associated with them.
For later time steps, the hypothesis cannot be rejected for most mutation frequencies, but it is not informative to predict the dynamic, since the average value $\hat{p} \approx 0$.

4.1.2 Decaying binomials

The assumption behind the decaying binomial model is that all mutation frequencies decay exponentially at the same rate. Due to different starting points, this allows for initial heterogeneity.
The hypothesis was first tested on all patients, over all measurements, without getting good results, an example can be seen in Figure 4.2 for patient C189.

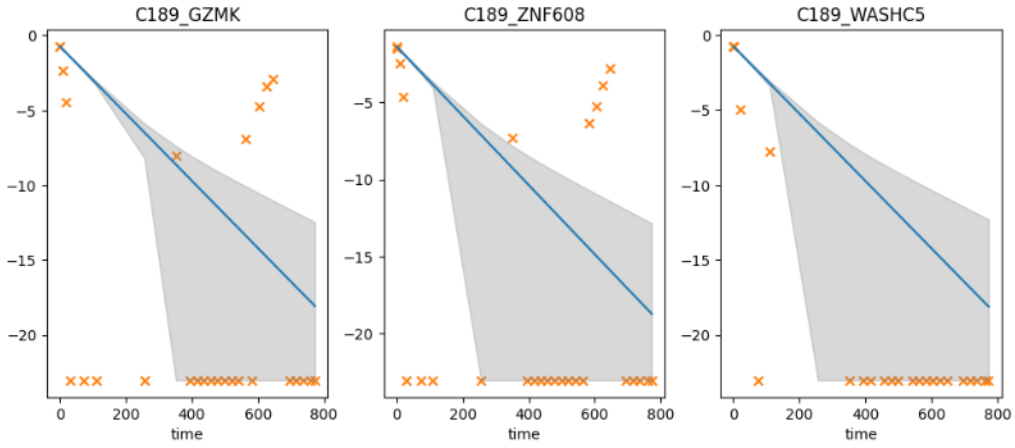


Figure 4.2: Patient C189 - Estimated value of the mutation frequency (blue line) with confidence interval (grey area) and the actual data (orange crosses) in log-scale $\ln(p(t) + \varepsilon)$.

This is not surprising, since for some patients the dataset covers quite a long time interval, which includes re-growth, in contrast to the hypothesis.
We then tested the hypothesis only during the chemotherapy of the high-risk patients.

For a few patients, see for example Figure 4.3, this model worked better than for

others, where almost all measurements were outside the interval.

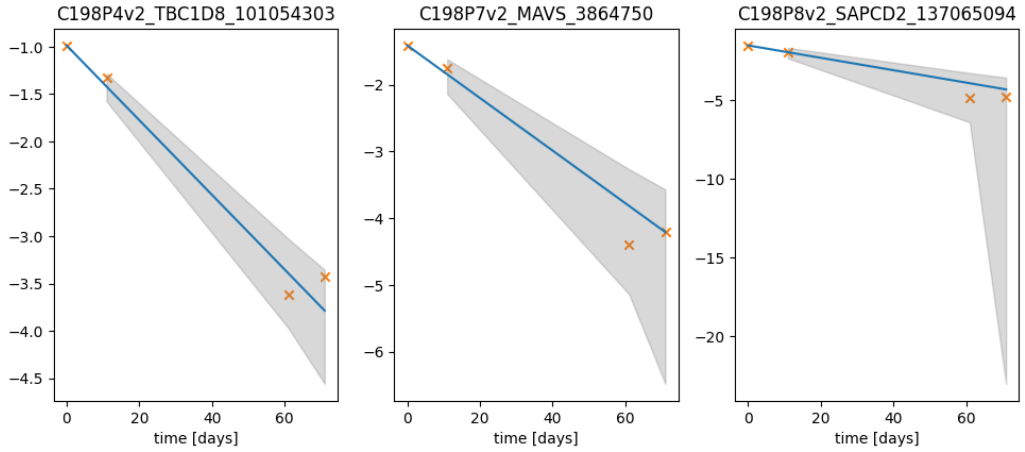


Figure 4.3: Patient C198 - Estimated value of the mutation frequency (blue line) with confidence interval (grey area) and the actual data (orange crosses) in log-scale $\ln(p(t) + \varepsilon)$. Out of ten genomics locations, for six the hypothesis can be accepted, the other four have at least one outlier.

In general, for no patient, the hypothesis holds for all the mutation frequencies. The decay deviates from an exponential one and additionally, there is too much heterogeneity between mutations at different genomics locations. For most patients, the mutation frequencies seem to slow down towards the end of the treatment and occasionally there are big fluctuations.
For these reasons, we had to discard this hypothesis.

4.2 Parameter estimation of ODE models

Next, the models of Section 2.2 are fitted to the MTM values. We assumed that MTM is directly proportional to the amount of tumour.
We focused mainly on high-risk patients, where the MTM spans multiple orders of magnitude. Therefore, we log-transformed it $\ln(x + \varepsilon)$, where ε is an arbitrary constant to avoid $\ln(0)$. We set $\varepsilon = 10^{-2}$, since this is one order of magnitude less than the lowest non-zero value in the dataset.

To avoid being too biased by the initial observations $y_i(0)$ in the initial value problem (IVP) for the classical and Stairs models, we treated the starting point as an additional parameter to optimise in the range $[0.8 \cdot y_i(0), 1.2 \cdot y_i(0)]$. Otherwise the IVP was started at $y_i(0)$ to prevent the introduction of additional parameters. The first measurement was then excluded in the calculation of likelihoods and AICs.

4.2.1 Optimising T

For the models which actively include the treatment, the duration of the treatment effects can be represented by the parameter T. When optimising T with a gradient-

based method one should be cautious, because of the time discretisation dt of the ODE. This can lead to flat parts in the likelihood function.
A gradient-based optimiser will get stuck and cannot move to the minimum. This can be prevented by setting the step size of the discrete derivative with respect to T to a value bigger than dt.
We explored that the same measurements can be obtained by, for example, a bigger T and a smaller death rate, but also by a smaller T, bigger death rate and bigger growth rate. This causes identifiability issues if no intermediate measurements or additional information are available.

We decided to fix T for this reason. According to the rapid COJEC schedule drugs are given on two consecutive days, see Table 1.1. Moreover, the drugs are very toxic and therefore should not stay in the body for too long. Therefore, we set $T = 2$, which was also supported by the clinicians who generated the dataset.

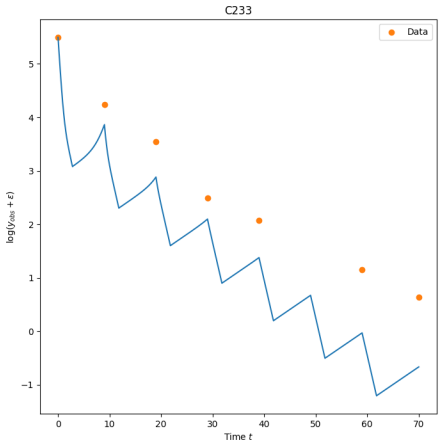
4.2.2 ABC-SMC

One of our first approaches to infer the parameters of the ODEs was to use ABC-SMC. An advantage of this method is that no assumptions on the noise distributions have to be made. After implementing and testing it for the S-LS, Bertalanffy and Exponential model, we decided not to continue with this method anymore. This method is time intensive, it is difficult to find a suitable setting that makes the algorithm converge, and finally the models with the estimated parameters, especially S-LS and Exponential, did not fit the data well. For S-LS the posterior distributions were often flat, see Figure 4.4 (b), and consequently the mean, which was used as the estimator for the parameters, was not a good choice. This could mean that for the S-LS model, a wide range of parameters can fit the data, which was also the inspiration behind fixing an additional parameter to derive the simpler S-LS* model.

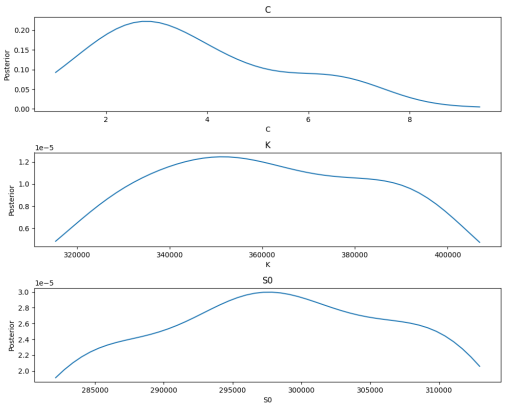
4.2.3 Exploring optimisation methods

Compated to ABC-SMC, likelihood-based methods offer significant speed-up. Since in our models the likelihood, see Section 3.2, can be established, we explored different optimisation methods to maximise it. The S-LS model has the most parameters, thus the parameter estimation is the most complex.
We subsequently tested PSO, L-BFGS-B and a combination of both optimisation methods to minimise the log-likelihood function of synthetic data generated based on the S-LS model.

A synthetic dataset of ten patients with eight measurements each was generated. The rapid COJEC schedule consists of eight treatments with a distance of ten days from each other. However, in our dataset, those distances were not always met. To make the generated dataset more realistic, the distance between two treatments was randomly set to 10 days with a probability of 40% and to 9, 11 or 12 days with a probability of 20% each.
Similar to our actual dataset, measurements were taken on the first day of each treat-



(a) Model, where the parameters were estimated by the mean of their posterior distribution.



(b) Posterior distribution of the parameters C, K and S_0.

Figure 4.4: ABC-SMC on patient C233 with the S-LS model.

ment cycle and a first observation of MTM was randomly chosen between $3 \cdot 10^4$ and 10^6 (approximately the range of the first observation of the high-risk neuroblastoma patients in our dataset). Then a set of parameters within the bounds, see Table A.1, was randomly chosen and measurement values calculated following the dynamics of the S-LS model.
As an additional constraint, the parameter set was accepted if at least two measurements were below 250 and the second measurement was above 30. Those conditions are fulfilled by all patients in our real dataset and prevent us from including dynamics which are different from the ones in our dataset.

Method	Time [minutes]	Average max likelihood
PSO $w = 1$	13	3.91
PSO $w = 1.1$ and $w = 0.9$	17	1.45
PSO $w = 1.1$ + L-BFGS 20	28	25.8
L-BFGS 20	15	25.8
L-BFGS 70	58	25.8
L-BFGS 128	105	25.8

Table 4.1: Runtime and average of the different optimisation methods on the same ten synthetic patients.

The L-BFGS optimisation with starting points on a grid with two nodes in each direction results in $2^7 = 128$ starting positions for S-LS when including T and σ as parameters. It managed to find values of the objective function, which are closer than 10^{-4} to the actual maximum. With an average runtime of approximately five seconds per starting point, it takes a long time to run. Moreover, this method scales

exponentially with the number of parameters to be estimated, which does not make it very versatile.
A first attempt to reduce the optimisation time was to only randomly choose 70 or 20 of the grid nodes (L-BFGS 70/20). This method managed to find similarly low values for the objective function as the original L-BFGS but took significantly less time. Moreover, it does not directly depend on the number of parameters, which makes it easily scalable.
One potential pitfall of this method is that the *"exploration"* over the entire space is mainly done in the random initialisation part, since L-BFGS only uses local information about the function to find a local minimum.

An alternative is PSO, which can search the parameter space in a more explorative way and does not require differentiability. Moreover, it does not scale with the number of parameters.
We tested PSO with 80 particles, 300 iterations and $w = 1$. It was the fastest of the tested optimisation methods, but also did not get close to the maximum of the function.
To improve the results, we tried to first explore the parameter space by setting $w = 1.1$ in 300 iterations and then ran 100 iterations with $w = 0.9$ to get close to a potential minimum. This approach failed. Therefore, we ultimately chose to merge PSO and L-BFGS into Algorithm 4.

Algorithm 4 PSO+L-BFGS

Run PSO with 80 particles for 300 iterations, with $w = 1.1$ and $c_1 = c_2 = 0.9$
From the 80 particles, select the 20 best-scoring ones: $\{x_i\}_{i=1...20}$
for $i = 1, \ldots, 20$ **do**
    Run L-BFGS with starting position x_i, save best position and score: y_i,b_i
end for
$i_{best} = \text{argmax}_i \{b_i\}_i$
return $y_{i_{best}}$

This algorithm usually performed as well as L-BFGS-B on a grid, as can be seen for example in Table 4.1. It does not directly depend on the number of parameters, which makes it scalable. And finally has both an *"exploratory"* and an *"exploitative"* nature, given by PSO and L-BFGS-B respectively.
For these reasons, we decided to use this optimiser as the standard one for this thesis.

Neither PSO nor L-BFGS-B can handle constraints, but in the Bertalanffy Stairs model we want to force biological constraints onto the model. In this case, SLSQP, see Section 3.1.2, with 70 starting points was used.

4.3 ODE models comparison

We proceeded to fit the models to the dataset by optimising the log-likelihood with a combination of PSO and LBFGS, see Algorithm 4. The results are summarised in

Figure 4.5, the left picture shows the comparison of all classical models, and on the right are all the other models, which have some treatment mechanism.
It can be clearly seen that overall S-LS/S-LS* are the best models at capturing the real data, with a $R^2 = 0.971$.
At the same time, the Bertalanffy model is also very good at fitting the general dynamic with a $R^2 = 0.951$.

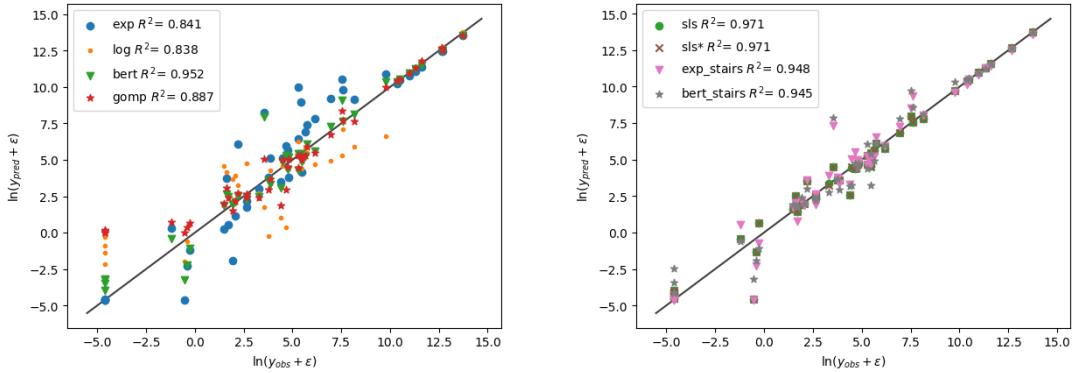


Figure 4.5: On the x-axis are the real observations in log-scale, and on the y-axis are all the predicted observations in log-scale for all the patients. For the S-LS/S-LS* models the first predicted point is identical to the observed one, by construction. This could lead to a higher R^2 score.

To better compare the results of the different ODE models, we calculated the respective AIC scores, see Equation 3.14.

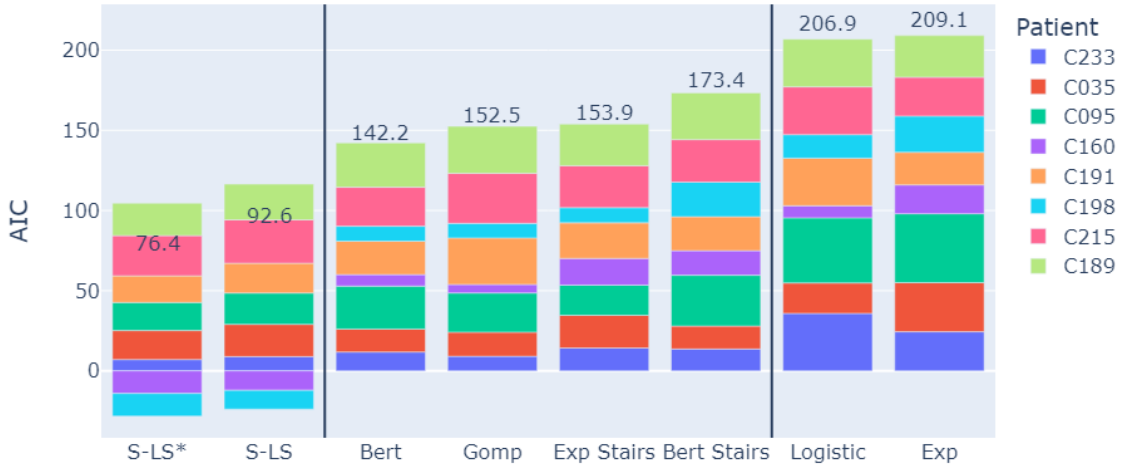


Figure 4.6: AIC scores for all models and patients. Note that some scores are negative. The numbers on the bars are the sum over all patients.

From Figure 4.5 and Figure 4.6 the models can be grouped into three main categories:

- The first category contains *S-LS** and *S-LS*, which overall are the best-performing models. On one hand, this was expected, given the fact that those are the most

complex models we considered. On the other hand, the AIC scores have not been too much penalised by the bigger number of parameters. What can be noticed is that S-LS, despite being more flexible than S-LS*, has the same R^2 score, but a higher AIC, which could hint at an excessive number of parameters. However, it is important to remember that the AIC is not a good metric when the sample size is so small. To draw a sound conclusion more measurements are needed.
Despite these good results, it is important to mention that, for multiple patients, the obtained dynamics looked quite unrealistic, as if the model/optimiser were overfitting the data, see Figure 4.7.

- The mid category contains *Bertalanffy*, *Gompertz* and the *Stairs* models. We can note that there is a difference between the ranking obtained in terms of R^2 and the AIC rank. Specifically, in the Bertalanffy Stairs model, despite having a relatively high R^2, the increased number of parameters is penalised in the AIC rank, making it worse than classical models. In some sense, we can say that the extra number of parameters in this model is not "worth it". But again, a bigger sample size is needed to draw conclusions.
  Moreover, for some patients, the Stairs dynamics have lots of oscillations, especially for the Exponential Stairs. This goes against the doctors' expectations. Overall Bertalanffy, despite not having any treatment mechanism, seems to capture the overall general dynamic.

- The last category contains the *Logistic* and the *Exponential* models, which can be seen both from the AIC scores and from the R^2 that are not good at capturing the data. One reason might be the simplicity of these models.

In Figure 4.7, an example of the dynamics of some models can be seen.
Note that the drug effect in the Stairs models is modelled by an exponential decay and thus is linear in the log-scale. Contrary to that, the treatment effect of the S-LS and S-LS* models is given by the sum of two different exponential decays, with a slower and faster rate.

Summarising, we can say that: on one side some models are just not good enough to fit the data, and on another side, we have not enough data to draw conclusions.

4.4 Exploring identifiability on a synthetic dataset

Synthetically generated data gives us the chance to check whether the parameter estimation methods can recover the original parameter under ideal conditions, i.e. if a parameter is structurally identifiable.

For each of the models in the best two groups of Figure 4.6, a dataset of 15 synthetic patients was generated as explained in Section 4.2.3. Then the parameters were estimated with SLSQP for Bertalanffy Stairs and PSO + L-BFGS for the other

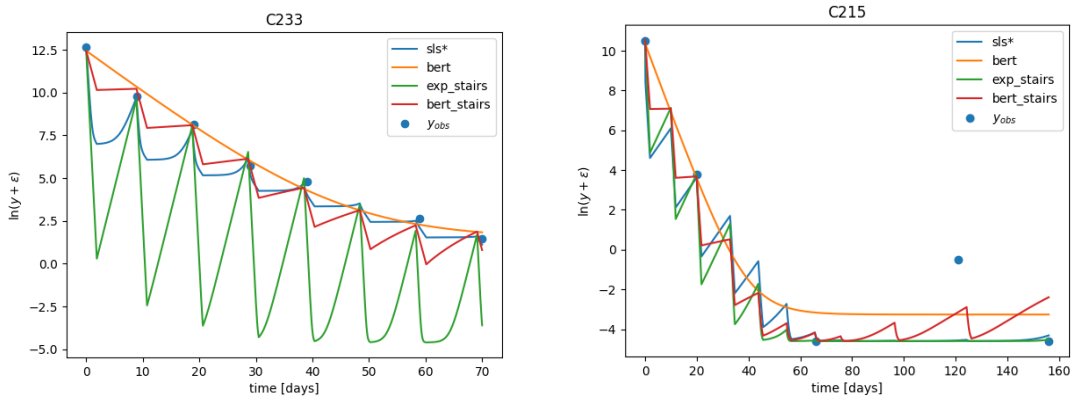


Figure 4.7: Example of dynamics for a few models of two patients. On the x-axis, we have time expressed in days, while on the y-axis the log-transformed predictions of models and the measured MTM.

models. The relative error

$$\frac{|p_{\text{est}} - p_{\text{orig}}|}{p_{\text{orig}} + 0.01}, \tag{4.1}$$

where p_{est} is the estimated parameter and p_{orig} the original parameter, was computed for each parameter. The average relative error of each model is displayed in Figure 4.8.

For both S-LS and S-LS* only the first parameter, S_0, is identifiable, all other parameters have an average relative error of at least 9%. Note that the negative log-likelihood is nevertheless very close to the actual minimum in the estimated parameters. There are two options for why this is possible. First, eight measurements, which are always taken on the first day of a cycle, may not provide enough information to estimate the parameters correctly. Second, multiple different sets of parameters could lead to approximately or exactly the same dynamics and are not distinguishable for us. These options are further investigated in Section 4.4.2.

The parameters of all other models are identifiable.

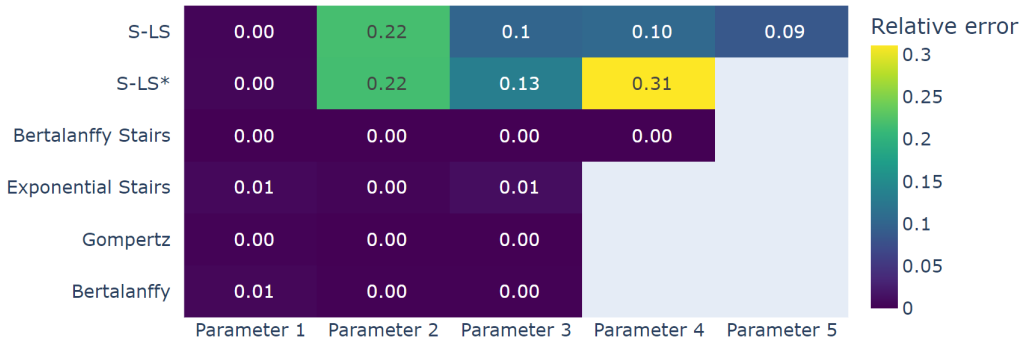


Figure 4.8: Average relative error in parameter estimation on a synthetic dataset ($n = 15$) without noise.

4.4.1 Identifiability in data with noise

This section investigates how robust the identifiable models are when noise is gradually added. The same synthetic dataset from the identifiability analysis in Section 4.4 was used, but disturbed according to Equation 4.2 for $\sigma \in \{0, 0.01, 0.05, 0.1, 0.5\}$.

$$\hat{y}_{obs} = \exp(\ln(y_{obs}) + \xi) = y_{obs} \cdot \exp(\xi), \quad \xi \sim N(0, \sigma^2) \tag{4.2}$$

Optimisation for the models with only three parameters (plus σ) is done by L-BFGS on a grid with $3^{3+1} = 81$ starting points. The results are plotted in Figure 4.9. Without any noise, the estimated parameters are equal to the original parameters for all models. Already slight disturbances of the observations ($\sigma = 0.01$) have a negative effect on the quality of the parameter estimation. None of the parameters is close to the original ones anymore at $\sigma = 0.5$.

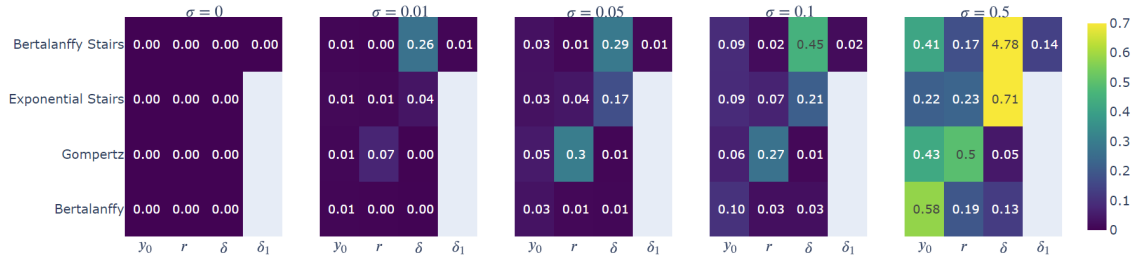


Figure 4.9: Average relative error, see Equation 4.1, of the parameters in the Bertalanffy, Gompertz and Stair models over 15 synthetic patients.

4.4.2 Non-identifiability of S-LS*

The previous experiments on synthetic data have shown that while S-LS and S-LS* achieve good R^2 and AIC scores, the parameters, which the optimisation algorithm finds, are often not close to the real parameters. In this section we want to find out if identifiability can be improved by taking more measurements.

We generated a dataset of 30 synthetic patients as described in Section 4.2.3, optimised the log-likelihood function with Algorithm 4 and computed the relative error, see Equation 4.1, for each parameter and simulation. We noticed that the average relative error over all simulations was high, but only because of a few outliers. Therefore, we decided not to take the average relative error as a metric of how well the model is doing, but the percentage of simulations with a relative error lower than 0.05.

The first strategy was to gradually add points in the middle of two treatments starting from the beginning.
The results are summarised in Figure 4.10, left. The parameters S_0 and r_0 are estimated correctly in most of the cases and by adding more points the estimation gets better. The parameters δ and C are not estimated correctly in the majority of the cases, no matter how many extra points were added. At the same time, the optimised parameters, sometimes after adding one extra point, led to predictions, which

look the same as the original dynamics, see Figure 4.10, right. This means, that no matter how many extra points we add, it is unlikely that the original parameters are estimated, meaning that the model is non-identifiable.
While δ and C on their own are not estimated well, $\frac{\delta}{C}$, the death rate of the less sensitive cells, achieves a high score. In Equation 2.21 the multiplication of S_1 by C further gets multiplied by r_2 and $\frac{1}{K}$, which are very small numbers. The impact of C here seems to be small compared to the impact that it can have on the death rate of the less sensitive cells. That is why in the next section, we fixed C in the logistic part of the function.

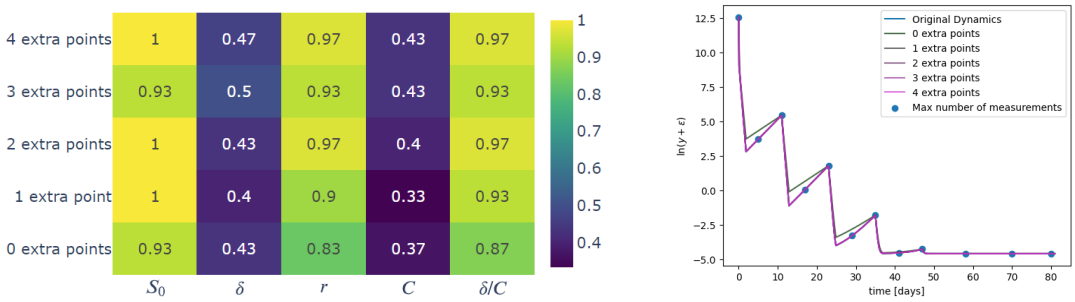


Figure 4.10: Proportion of simulations with a relative error of less than 0.05 in the parameter estimation for the S-LS* model and an example of the dynamics and how more measurements can make the prediction better.

4.5 Updates to S-LS model

The S-LS and S-LS* models have good AIC scores, but the problem of having non-identifiable parameters and over-fitted or biologically unrealistic dynamics still remains. In this section, we analyse a few adjustments that can be made to overcome these problems to some extent.

After discussing with the doctors and researchers who collected the data, we decided to make some changes to the S-LS model to make it more realistic.
The changes we made were based on the following facts:

- From a medical perspective it is very unlikely that oscillations of any kind happen between one treatment and the other, for this reason, we limited the regrowth for both populations to a maximum 0.01.

- Instead of assuming that the therapy at the beginning is very close to the carrying capacity, which is unrealistic for many cases, we considered an upper bound of $K = 10^6$, as suggested by the doctors.

- After analysing the effects of changing the competitive factor $C > 1$ only in the logistic part of the equation (the part when the treatment is off), we figured out that the other parameters can always fit the data equally well. For this reason, we decided to fix it to an arbitrary value, 3.

This led to the following ODE:

$$\begin{cases} \dot{S}_1 = rS_1(1 - \frac{S_1+S_2}{10^6}) - \delta_1 S_1 \\ \dot{S}_2 = \frac{1}{C}(rS_2(1 - \frac{3 \cdot S_1+S_2}{10^6}) - \delta_1 S_2). \end{cases} \tag{4.3}$$

The two equations look very much alike, except for the coefficient in front of the second equation, that makes the S_2 dynamic slower compared to S_1. And by the choice of the competitive factor, inside the logistic term, we still have that the only stable fixed point is $(K, 0)$.

After running several simulations of this model, we decided to completely remove the growth part, as we saw that, we the new choice of parameters, the growth-effects were negligible.
This led us to our final model, which we called *Fast-Slow (FS) Exponential*:

$$\begin{cases} \dot{S}_1 = -\delta S_1 \\ \dot{S}_2 = -\frac{\delta}{C} S_2 \end{cases} \tag{4.4}$$

We considered two versions of this model, depending on whether we had information about the treatment. If the injection times were known, then δ was set to zero two days after the treatment, which translates into having constant values for S_1 and S_2 between the end of the effect of the drug and the next shot. If there is no information about the treatment, then δ is kept constant for all the times steps.
As for S-LS/S-LS*, in the IVP, S_1 is initialised as $S_1(0) = S_0 \cdot y_{\text{obs}}(0)$ where S_0 is an additional parameter that represents the fraction of sensitive cells in the initial population, while $S_2(0) = (1 - S_0) \cdot y_{\text{obs}}(0)$. The ranges for the parameters can be found in the appendix in Table A.1.

It is interesting to compute the tipping point, where $S_1 = S_2$:

$$T_{\text{switch}} = \frac{1}{\delta} \frac{C}{C-1} \ln\left(\frac{S_0}{1-S_0}\right).$$

In fact, before T_{switch}, the dynamic will be dominated by the fast decaying exponential S_1, while afterwards by the slow-decaying S_2. If the treatment times are known, the previous formula has to be modified to take the flat parts into account.

A limitation of the Fast-Slow Exponential, compared to S-LS or the Stairs models, is that it cannot model regrowth, neither under treatment nor afterwards.

4.6 Evaluation of Fast-Slow Exponential

4.6.1 Fitting to dataset

We fitted the Fast-Slow Exponential model, see Equation 4.4, to our dataset, by maximising the log-likelihood function with Algorithm 4.
The results were surprisingly good, given the relative simplicity of the model. In

MODEL	AIC
S-LS*	76.4
FS Exponential	89.7
S-LS	92.6
FS Exp - No Treatment	98.0
...	
Exponential Stairs	153.9

Table 4.2: AIC model comparison, the patient-wise comparison can be found in Figure 4.14.

fact, an overall AIC score of 89.7 was achieved, which places the model between S-LS* and S-LS as can be seen from Table 4.2. Also, in terms of R^2, the model achieved a total score of $R^2 = 0.975$. As for the S-LS/S-LS* models, the first predicted point is identical to the first observed one, by construction.

We also tried to fit the Fast-Slow Exponential without considering the treatment, which performed worse than the standard Fast-Slow Exponential, as it can be seen in Table 4.2. For this reason, we did not analyse it any further for high-risk patients.

In Figure 4.11, we can see an example of the fitted model for two patients. For both of them, in the first time steps the dynamic is dominated by the fast-decaying population. This goes on until it reaches comparable levels to the slow decaying population in a neighbourhood of T_{switch}. Afterwards, the sensitive population drops even further and the less sensitive population takes over.

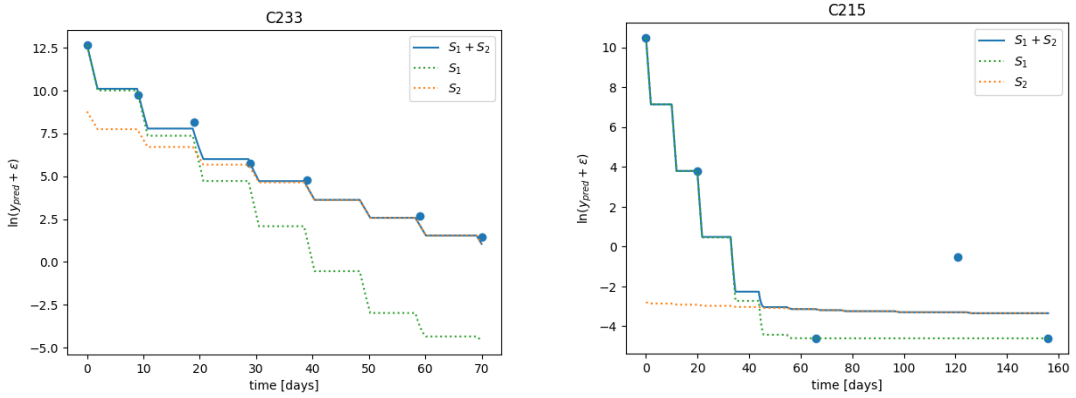


Figure 4.11: Example of Fast-Slow Exponential model on two patients of the dataset.

4.6.2 Identifiability

One of the main weaknesses of the S-LS/S-LS* model is its non-identifiability. We studied if FS Exponential, a simplified version of S-LS, has the same problems. Therefore, we generated a synthetic dataset according to Section 4.2.3 and gradually

added noise. Figure 4.12 shows the average relative error of the parameter estimation. Without noise all parameters are identifiable. Once the noise is added, the error increases, but especially the error in S_0 remains low. Note that $S_0 \in [0.92, 1]$, so the relative error cannot be as high as for other parameters.

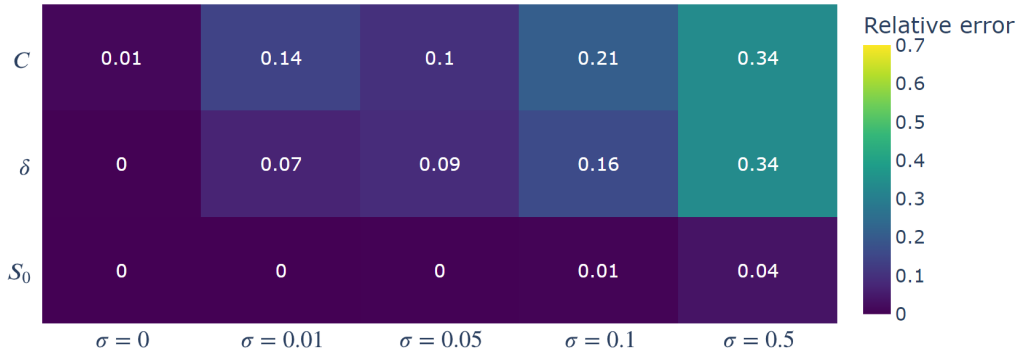


Figure 4.12: Average relative error of parameter estimation in the Fast-Slow Exponential model under noise.

Given that without noise, the parameters are identifiable, the next question is, how many measurements are needed to identify the parameters. The model has three parameters: S_0, δ, and C, additionally an initial value $y(0)$ is required, so at least four measurements are needed.
We selected measurement points based on the following principles. The four measurements should not all be on the flat part between treatments and the doctors want to know the dynamics as soon as possible. Therefore, we simulated a measurement on the first two days of the first two cycles, resulting in four measurements in total.
On a synthetic dataset of 50 randomly generated parameter sets without noise, the average relative error over all simulations and parameters was around 10^{-3}. This means that all parameters of the model are identifiable with only four measurements, even if they are all before T_{switch}. It also means, that if this model was true and there was a way to not have any noise in the data, we could make reliable and precise predictions after the treatment in the second cycle.

4.6.3 Predictions with noise

In practice, the assumption that there is no noise does not hold. There will always be fluctuations in the MTM values due to biological reasons, measurement errors or not monitoring the exact time of treatments and measurements.
In addition to the model parameters, the standard deviation σ was optimised for each patient. It was between 0.01 (lower bound, C160) and 1.69 (C215), with a mean of 0.69. We decided to set the noise level to $\sigma = 1$ in this simulation, which is greater than the estimated σ for six out of eight patients.

In this section we wanted to find out if, and how well we can predict later time-points after having added noise to synthetically generated data. As shown in Figure 4.12

already at a noise level of $\sigma = 0.5$ the errors in the parameter estimation are huge. So, we cannot expect that with $\sigma = 1$, the estimated parameters will be similar to the original parameters. That still does not mean that the resulting dynamics are completely different. Instead of computing and comparing relative errors of the parameters, therefore, we computed the average distance of the log-transformed original dynamics and predictions in 200 evenly spaced points.

On the same synthetic dataset of 50 patients as in the previous section, we simulated disturbed measurements according to Equation 4.2 with $\sigma = 1$ for every day until four days after the last treatment. For the parameter estimation, a subset of those measurements is used. By not generating new noisy measurements for each parameter estimation, the results are comparable.

To get an idea of how much the noise influences the results, measurements on the first day ($m = 8$), the first two days ($m = 16$) and the first three days ($m = 24$) of each cycle were selected and the parameters estimated based on those measurements. Those simulations were later used as a comparison with the predictive strategies. The results are in Table 4.3 under "Whole" and an example is in Figure 4.13. As expected, the more measurements were used, the lower the average error.

Two strategies for the predictions were considered. The first strategy is called "Half" and only considered measurements in the first half of the treatment, always on the first two days of a cycle ($m = 8$) and then adding measurements on the 3nd and 4th ($m = 16$) and on the 5th and 6th ($m = 24$) day of the cycle. The second strategy "First" simulated a measurement every day until $m \in \{8, 16, 24\}$ measurements were reached.

m	**Whole**	**Half**	**First**
8	0.50	0.91	2.83
16	0.33	0.55	1.64
24	0.27	0.44	0.89

Table 4.3: Average distance between prediction and original dynamics in 200 evenly spaced points. m is the number of measurements, "Whole" means that the measurements were spread over the whole interval, "Half" means, that all measurements were taken during the first half of the treatment, "First" means daily measurements on the first m days.

Table 4.3 shows that for all methods the average error decreased when more measurements were used. This trend did not hold for all simulations, see the predictions with 24 measurements in Figure 4.13.
Furthermore, not only the number of measurements but also when they were taken, determined how well the prediction matched the original dynamics. The later on the last measurement was taken, the better the prediction.

Assuming that the FS Exponential model and log-scale normally distributed noise with $\sigma \approx 1$ is true, the doctors should be cautious when trying to predict the

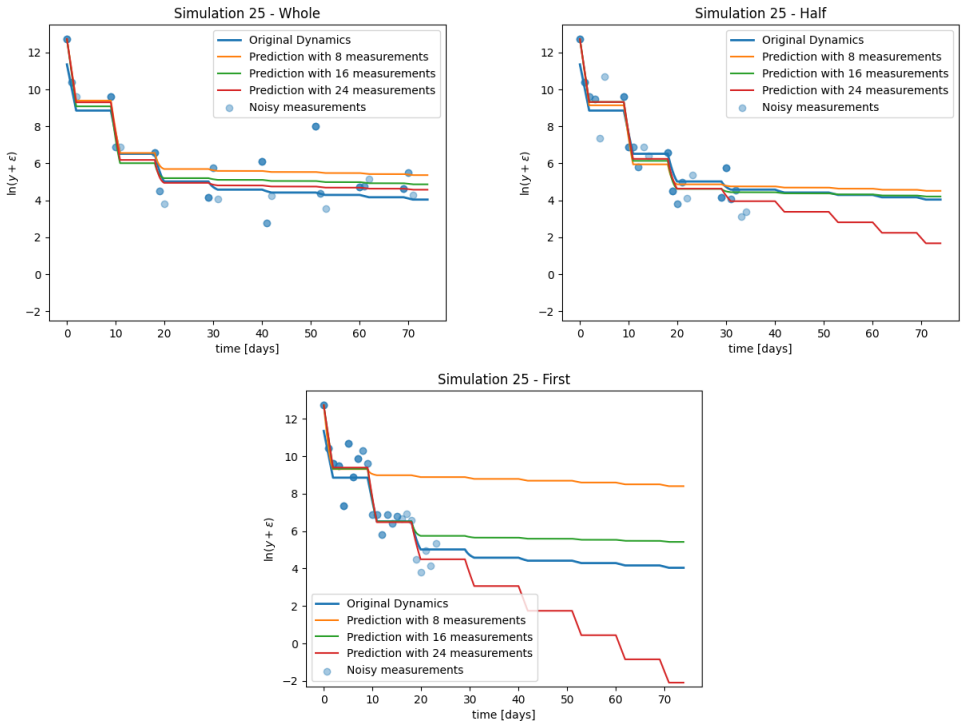


Figure 4.13: Example of the predictions with m noisy measurements with a synthetic dataset.

dynamics only with a few measurements because the expected errors are high. Even if plenty of measurements are available early on, they should wait at least until the 4th cycle before drawing conclusions as that lowers the error significantly.

4.6.4 Hierarchical modelling

Hierarchical modelling allows to combine the information across different patients, but it could fail when the model is non-identifiable. In fact, the convergence becomes unlikely or multi-modal posteriors are generated, which would not be informative, as discussed in [18]. Given that S-LS* is non-identifiable, see Section 4.4.2, we decided to focus only on FS Exponential which is identifiable, see Section 4.6.2.
We did this by allowing patient-specific parameters which were modelled by different prior distributions. Their means and variances are set by hyper-prior distributions, which are shared between all patients.
The scheme used for the hierarchical model can be seen in Figure 4.14, while the details for the prior distributions can be found in Table 4.4.
In order to speed up the sampling process, we increased the time discretisation dt, from 0.1 to 0.4. We tested that for this model it would still generate the same solutions.
We then ran the NUTS algorithm, see Section 3.3.1, over 2 chains with 15000 draws each, of which 1000 are used for the burn-in phase. Due to limited computational resources, we ran the algorithm with these settings multiple times instead of using more chains, getting the same final results.

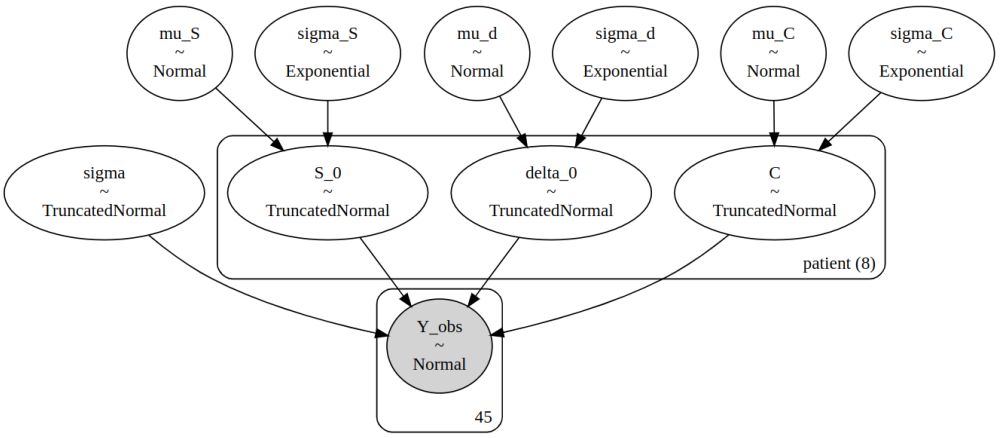


Figure 4.14: Hierarchical model implementation for FS Exponential model: Patient-specific parameters (S_0, δ_0, C), whose mean and variance are set by the hyper-prior parameters (μ_S, σ_S, μ_d, σ_d, μ_C, σ_C) and the pooled noise parameter σ.

Name	Distribution	Type
μ_S	$N(0.98, 0.01^2)$	Hyper
μ_d	$N(2, 0.5)$	Hyper
μ_C	$N(10, 1)$	Hyper
σ_S	$\text{Exp}(\frac{1}{0.05})$	Hyper
σ_d	$\text{Exp}(\frac{1}{0.5})$	Hyper
σ_C	$\text{Exp}(\frac{1}{2})$	Hyper
S_0	$\text{TrN}(\mu_S, \sigma_S^2)$ on $[0.85, 1]$	Partially-pooled
δ_0	$\text{TrN}(\mu_d, \sigma_d^2)$ on $[1, 5]$	Partially pooled
C	$\text{TrN}(\mu_C, \sigma_C^2)$ on $[1, 55]$	Partially-pooled
σ	$\text{TrN}(0.65, 0.5^2)$ on $[0.01, 2.5]$	Pooled

Table 4.4: Parameters of the prior distributions. $\text{TrN}(\mu, \sigma^2)$ on $[a, b]$ stands for truncated normal distribution with mean μ, standard deviation σ, a lower bound of a and upper bound of b.

In the final result that we obtained, $\hat{R}$ were in the acceptance range for all estimates, see Section 3.3.2, the posterior distributions looked very much alike across all runs. One example of posteriors can be found in the appendix in Figure A.3 and A.4.

Overall, the fits were not as good as when all the parameters were unpooled, see Section 4.6.1. The resulting AIC score of the hierarchical model is 178.8 and $R^2 = 0.859$. But as expected, for all parameters there is less variance between the patients, see Table 4.5.
More in detail, for S_0 the hierarchical model estimated a mean of 0.98, with a standard deviation of $6.54 \cdot 10^{-3}$. It is worth mentioning that we restricted the possible values for S_0 to the interval $[0.85, 1]$, so the method did not explore extreme scenarios in which, the less sensitive population is more abundant. Nevertheless, it showed a clear preference towards the right end of the interval.

	σ		S_0		δ		C	
	mean	std	mean	std	mean	std	mean	std
Hierarchical	1.17	0.13	0.98	$6.54 \cdot 10^{-3}$	2.48	0.097	6.88	4.05
Unpooled	0.69	0.50	0.99	$1.27 \cdot 10^{-2}$	2.57	1.44	22.11	21.18

Table 4.5: The mean and standard deviation of each parameter over all patients for the hierarchical and the unpooled FS Exponential model.

Also for the δ_0 the hierarchical model estimated a mean of 2.48, with a standard deviation of $9.7 \cdot 10^{-2}$, this could mean that the sensitive part of this cancer decays more or less at the same rate, also across patients.
On the other hand, even the hierarchical model estimated C with quite a big variance 4.05, while the mean was 6.88. Compared to the unpooled optimisation, the variance obtained is much smaller, this is given by the fact that when optimising only on the single patients, for a few of them, the optimiser returns the boundary value of $C = 55$. If we assumed that the hierarchical model was somewhat more realistic, we could interpret this last result, by saying that the decay of the less sensitive population is more patient-specific, compared to the sensitive one.

Finally, for σ in the unpooled optimisation, all estimated σ values were below 1 except for two patients, with a maximum of 1.68. So it is not too surprising to see that the hierarchical estimated σ as 1.17, since the same value is used in all likelihoods, including the two patients that before already were above 1.

In Figure 4.15, the Fast-Slow Exponential with parameters estimated by the hierarchical model on two patients, is shown. It can be compared with the unpooled parameter estimation in Figure 4.11, in which the dynamic fitted the data clearly better, especially for patient C233. A difference can also be noticed for the patient C215. The hierarchical model converged to the threshold $\ln(\varepsilon)$ at $t = 80$, while the unpooled chose a middle ground between the last three points.

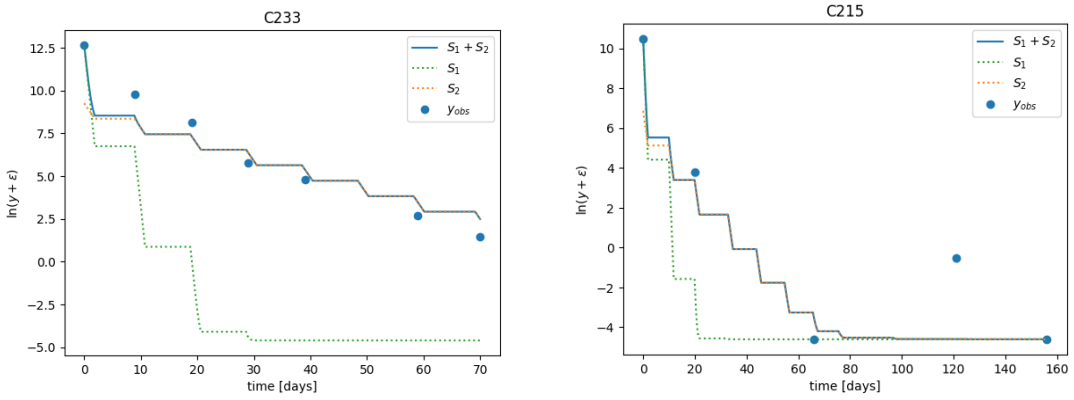


Figure 4.15: Example of hierarchical Fast-Slow Exponential model on two patients of the dataset.

4.7 Non-high-risk patients

In this final section, we take a brief look at the four patients that do not have high-risk neuroblastoma and thus have not gone through COJEC.
Not having access to the time data of when they received drugs, we simply tried to see if one of the classical models, see Section 2.2.1, or Fast-Slow Exponential, see Equation 4.4, would capture the overall dynamic.
The results are summarised in Figure 4.16, which shows that Fast-Slow Exponential is the best model for fitting the data, followed by Bertalanffy.

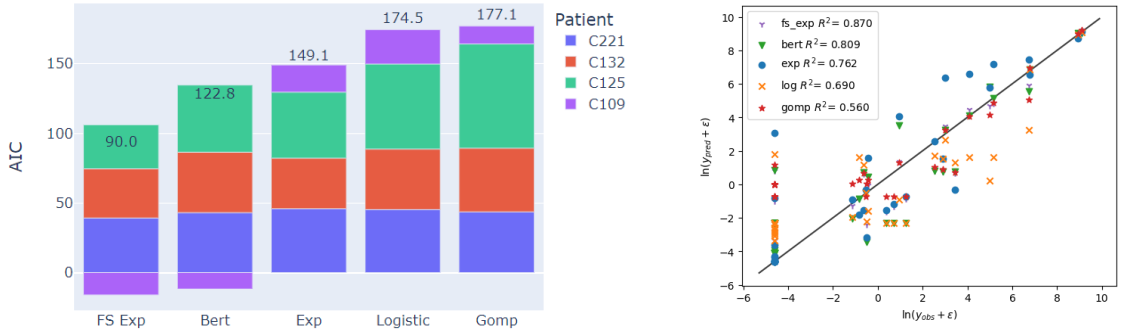


Figure 4.16: On the left are the AIC scores of the non-high-risk patients. The numbers on the bars are the sum over all patients.
On the right, the predictions are plotted against the observations in log-scale and the R^2 is computed.

Overall the MTM levels are smaller than in high-risk neuroblastoma patients, and many measurements are identically 0, which creates the vertical line in the bottom left corner of Figure 4.16.
In Figure 4.17 we can see an example of the FS Exponential for two patients. While both patients have a similar maximum likelihood, the model seems to get the data's trend quite good for patient C125, contrary to patient C132. For patient C132 no other model achieves a good fit due to oscillatory behaviour in the first four measurements. At the same time, the range of the MTM is very small, which results in a relatively high likelihood.

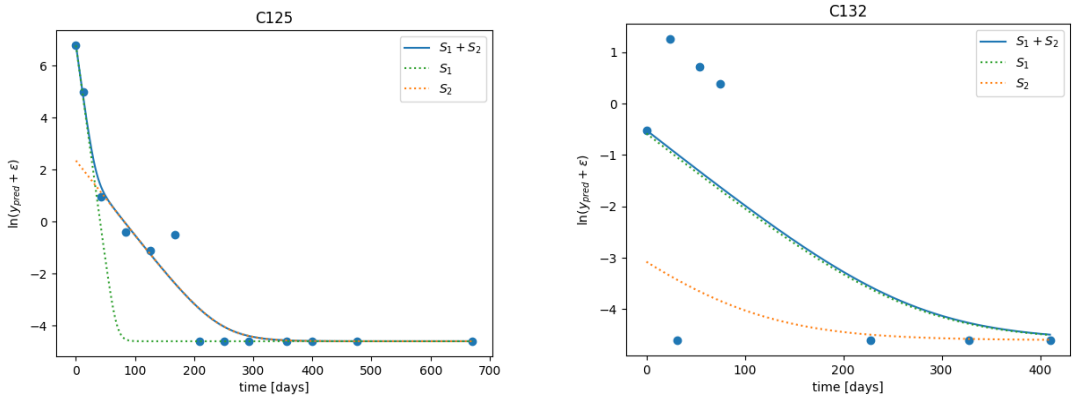


Figure 4.17: Fast-Slow Exponential for two non-high-risk patients.

5
Discussion

In conclusion, the model that we propose to describe the dynamic of MTM levels during therapy, is the **Fast-Slow Exponential**, see Equation 4.4.
Several reasons led us to this result, the main ones being: the simplicity of the model (compared to S-LS/S-LS*, for example), the good results obtained in terms of AIC and R^2 scores, the model's identifiability (see Section 4.6.2), and finally the fact that the dynamic aligns with the doctors' expectations.
It is worth mentioning that it is also the best-performing model on the non-high risk patients.
The scarcity of data often does not allow us to draw strong conclusions about the real dynamic. But what can already be seen from this project, is that multiple-populations models seem to be performing significantly better than single-population ones. The best example is indeed Fast-Slow Exponential, which outperforms all 1D-models, including Exponential Stairs or Bertalanffy Stairs. This is also backed up by medical research, which states that in neuroblastoma often multiple sub-populations with different chemosensitivity are present [7]. Furthermore, the starting ratio of less sensitive cells in the total population is always estimated to be below 5%, both in the unpooled and hierarchical optimisation. This aligns with our expectation that these cells have some disadvantages when there is no treatment.

Parallel to the selection of a model, we also investigated a number of parameter estimation methods.
If all parameters are assumed to be patient-specific, MLE is a relatively fast and easy way to estimate the parameters. Whenever the number of parameters is sufficiently small, then the best method to optimise the likelihood function is L-BFGS-B on a grid, see Section 4.2.3, which is fully deterministic. The downside of this method is that it scales exponentially with the number of parameters. If the number of parameters is big, PSO+L-BFGS-B (Algorithm 4) is a good compromise between execution time and results.
Another approach to the parameter estimation was hierarchical modelling, see Section 4.6.4. This method is based on the assumption that the parameters of the patients are not independent from each other, but rather come from the same hyper-distribution. Compared to the previous methods, this method requires higher computational time and resources, and the final results, in terms of goodness of fit on a single patient, are naturally worse, as the choice of parameters is more restricted.
However, when the assumptions behind this method are well founded, then despite its time-intensive nature, it still could be useful. For example, if a patient has very few data points, by combining data from other patients, it might still be possible to

infer realistic parameters. Furthermore, this approach limits the risk of overfitting individual patients.

Multiple future directions could be considered as a follow-up to this project. We suggest testing the models on a bigger dataset, both in number of patients, and also number of measurements for each patient. Especially measurements between the treatments of the COJEC schedule would be of interest. Those could be used to confirm, or reject, the flat dynamic assumed in the Fast-Slow Exponential. In general, more data would help with the model selection by making the AIC more accurate, and give us a better idea of the noise levels.
To rule out certain noise sources, the data should be collected by a standardised procedure, e.g. always at the same hour.
Another interesting aspect to investigate would be the separate effect of drugs within the COJEC regime. If more data was available, as an extension of the FS Exponential or S-LS models, one could model different effects for each of the five drugs. This is something of high medical interest, as based on the model then decisions about further treatment could be made.

In conclusion, despite the small dataset, we still managed to successfully draw meaningful biological conclusions, i.e. that multiple sub-populations are essential to explain the data. The heterogeneity could also explain why high-risk neuroblastoma has a bad prognosis.
Overall, this highlights the usefulness of mathematical modelling for medical applications.

Bibliography

[1] Oriol Abril-Pla et al. "PyMC: a modern, and comprehensive probabilistic programming framework in Python". In: *PeerJ Computer Science* 9 (2023). ISSN: 23765992. DOI: 10.7717/peerj-cs.1516.

[2] Daniel Andersson et al. "Circulating cell-free tumor DNA analysis in pediatric cancers". In: *Molecular Aspects of Medicine* 72 (2020). Liquid biopsy analysis in cancer diagnostics, p. 100819. ISSN: 0098-2997. DOI: https://doi.org/10.1016/j.mam.2019.09.003.

[3] Joseph E. Cavanaugh and Andrew A. Neath. "Akaike's Information Criterion: Background, Derivation, Properties, and Refinements". In: *International Encyclopedia of Statistical Science*. Ed. by Miodrag Lovric. Springer Berlin Heidelberg, 2011, pp. 26–29. ISBN: 978-3-642-04898-2. DOI: 10.1007/978-3-642-04898-2_111.

[4] Emily Crowley et al. "Liquid biopsy: monitoring cancer-genetics in the blood". In: *Nature Reviews Clinical Oncology* 10 (2013), pp. 472–484. ISSN: 17594774. DOI: 10.1038/nrclinonc.2013.110.

[5] Martin Haugh. *MCMC and Bayesian Modeling*. 2017. URL: https://www.columbia.edu/~mh2078/MonteCarlo/MCMC_Bayes.pdf (visited on 05/30/2024).

[6] Matthew D. Hoffman and Andrew Gelman. "The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo". In: *Journal of Machine Learning Research* 15 (2014). ISSN: 15337928.

[7] Danielle M. Hsu et al. "G-CSF receptor positive neuroblastoma subpopulations are enriched in chemotherapy-resistant or relapsed tumors and are highly tumorigenic". In: *Cancer Research* 73.13 (2013). ISSN: 00085472. DOI: 10.1158/0008-5472.CAN-12-4056.

[8] Lester James V. Miranda. "PySwarms: a research toolkit for Particle Swarm Optimization in Python". In: *The Journal of Open Source Software* 3.21 (2018). DOI: 10.21105/joss.00433.

[9] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. 2006.

[10] Ekaterina Kalashnikova et al. "Correlation between variant allele frequency and mean tumor molecules with tumor burden in patients with solid tumors". In: *Molecular Oncology* (2023). ISSN: 18780261. DOI: 10.1002/1878-0261.13557.

[11] Eunjung Kim et al. “Adaptive therapy for metastatic melanoma: Predictions from patient calibrated mathematical models”. In: *Cancers* 13.4 (2021). ISSN: 20726694. DOI: 10.3390/cancers13040823.

[12] Dieter Kraft. *A Software Package for Sequential Quadratic Programming*. Wiss. Berichtswesen d. DFVLR, 1988.

[13] Ruth Ladenstein, Dominique Valteau-Couanet, and Claudia Pasqualini. *High-risk neuroblastoma: standard clinical practice recommendations*. 2020. URL: https://siope.eu/media/documents/escp-high-risk-neuroblastoma-standard-clinical-practice-recommendations.pdf (visited on 05/20/2024).

[14] Narmin Ghaffari Laleh et al. “Classical mathematical models for prediction of response to chemotherapy and immunotherapy”. In: *PLoS Computational Biology* 18.2 (Feb. 2022). ISSN: 15537358. DOI: 10.1371/journal.pcbi.1009822.

[15] Erika A. Newman et al. “Update on neuroblastoma”. In: *Journal of Pediatric Surgery* 54.3 (Mar. 2019), pp. 383–389. ISSN: 0022-3468. DOI: 10.1016/J.JPEDSURG.2018.09.004.

[16] Mina Nikanjam, Shumei Kato, and Razelle Kurzrock. “Liquid biopsy: current technology and clinical applications”. In: *Journal of Hematology and Oncology* 15.1 (2022). ISSN: 17568722. DOI: 10.1186/s13045-022-01351-y.

[17] Larry Norton. “A Gompertzian Model of Human Breast Cancer Growth”. In: *Cancer Research* 48 (1988). ISSN: 15387445.

[18] Kiona Ogle and Jarrett J. Barber. “Ensuring identifiability in hierarchical mixed effects Bayesian models”. In: *Ecological Applications* 30.7 (2020). ISSN: 19395582. DOI: 10.1002/eap.2159.

[19] Frank Peinemann et al. “Rapid COJEC versus standard induction therapies for high-risk neuroblastoma”. In: *Cochrane Database of Systematic Reviews* 5 (2015). ISSN: 1465-1858. DOI: 10.1002/14651858.CD010774.pub2.

[20] Yannik Schälte et al. “pyABC: Efficient and robust easy-to-use approximate Bayesian computation”. In: *Journal of Open Source Software* 7.74 (2022). DOI: 10.21105/joss.04304.

[21] Tina Toni et al. “Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems”. In: *Journal of the Royal Society Interface* 6.31 (2009). ISSN: 17425662. DOI: 10.1098/rsif.2008.0172.

[22] Pauli Virtanen et al. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature Methods* 17.3 (2020). ISSN: 15487105. DOI: 10.1038/s41592-019-0686-2.

[23] Mattias Wahde. “Biologically inspired optimization methods: an introduction”. In: *Choice Reviews Online* 46.07 (2009). ISSN: 0009-4978. DOI: 10.5860/choice.46-3899.

[24] Wikipedia. *Particle swarm optimization*. URL: https://en.wikipedia.org/wiki/Particle_swarm_optimization (visited on 05/30/2024).

[25] Ignacio I. Wistuba et al. “Methodological and practical challenges for personalized cancer therapies”. In: *Nature Reviews Clinical Oncology* 8.3 (2011). ISSN: 17594774. DOI: 10.1038/nrclinonc.2011.2.

[26] Anyue Yin et al. “A Review of Mathematical Models for Tumor Dynamics and Treatment Resistance Evolution of Solid Tumors”. In: *CPT: Pharmacometrics & Systems Pharmacology* 8 (2019), pp. 720–737. ISSN: 21638306. DOI: 10.1002/psp4.12450.

[27] Ciyou Zhu et al. “Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-Scale Bound-Constrained Optimization”. In: *ACM Transactions on Mathematical Software* 23.4 (1997). ISSN: 00983500. DOI: 10.1145/279232.279236.

A

Appendix

All code can be found at `https://github.com/CharlotteP1/Master-Thesis`.

A.1 Supplementary Tables

Model	Parameter	Lower bound	Upper bound
	σ	0.01	10
	$y_{\text{pred}}(0)$	$0.8 \cdot y_{\text{obs}}(0)$	$1.2 \cdot y_{\text{obs}}(0)$
Exponential	r	0	2
Logistic	r	0	2
	K	0.001	5
Bertalanffy	r	0	5
	δ	0	5
Gompertz	r	0	1.2
	δ	0	15
Exponential Stairs	r	0	5
	δ	0	2
S-LS	r	0.001	1
	δ	1	10
	C	1	30
	S_0	0.9	1
	K	1	1.3
FS Exp	S_0	0.92	1
	δ	0	5
	C	1	55

Table A.1: Bounds for parameters for each model.

A.2 Supplementary Figures

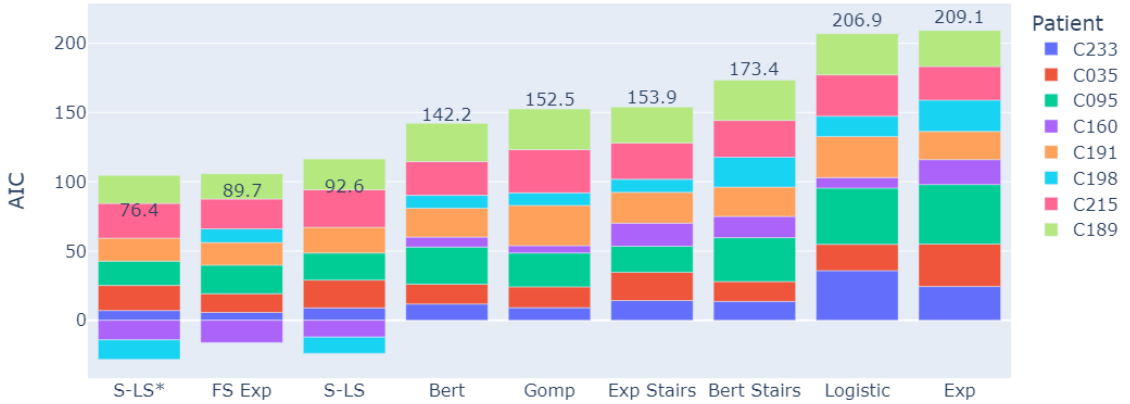


Figure A.1: AIC scores for each model, note that some scores are negative. The number at the bars is the sum over all patients.

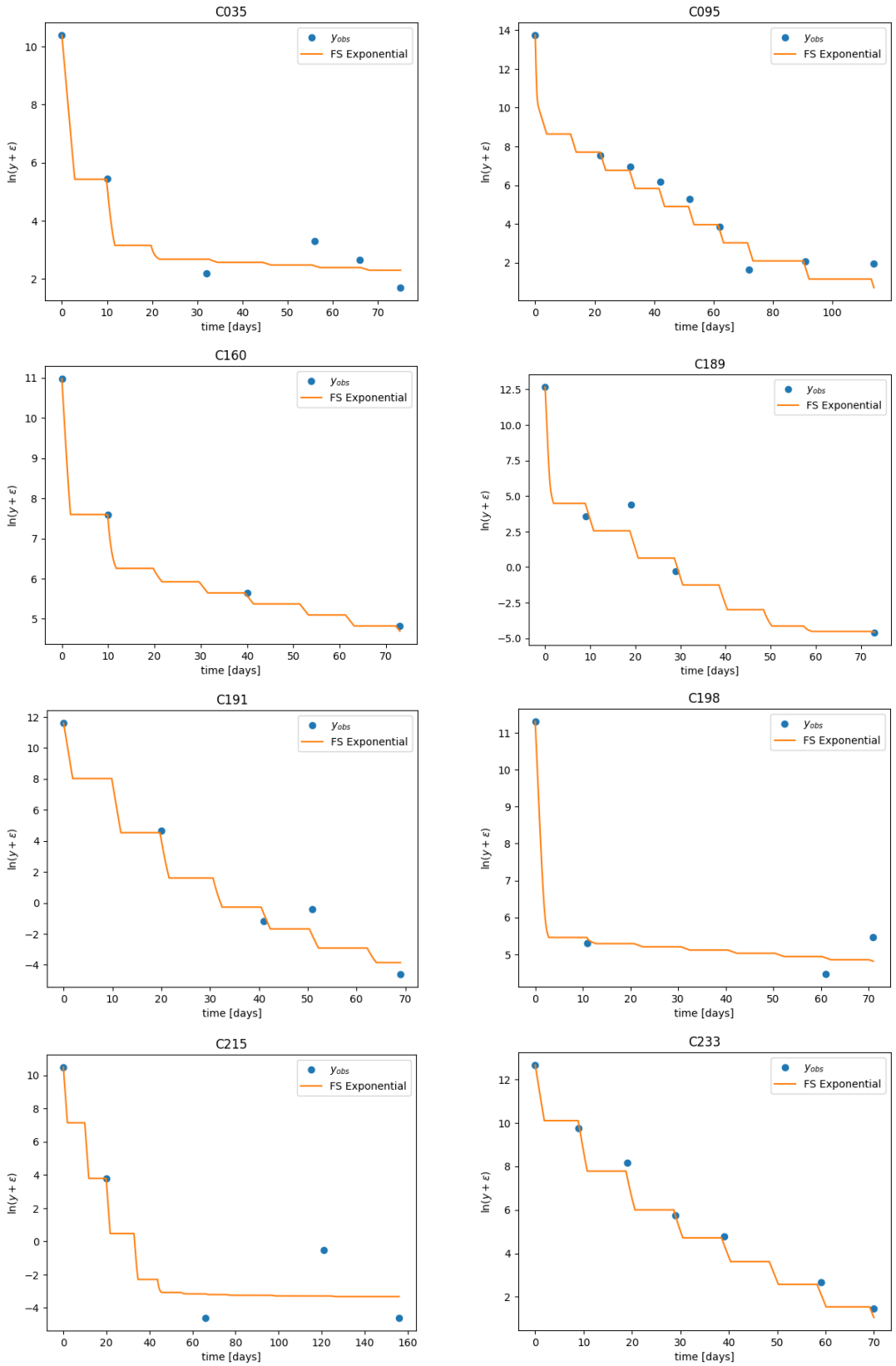


Figure A.2: All high-risk patients with the fitted FS Exponential model.

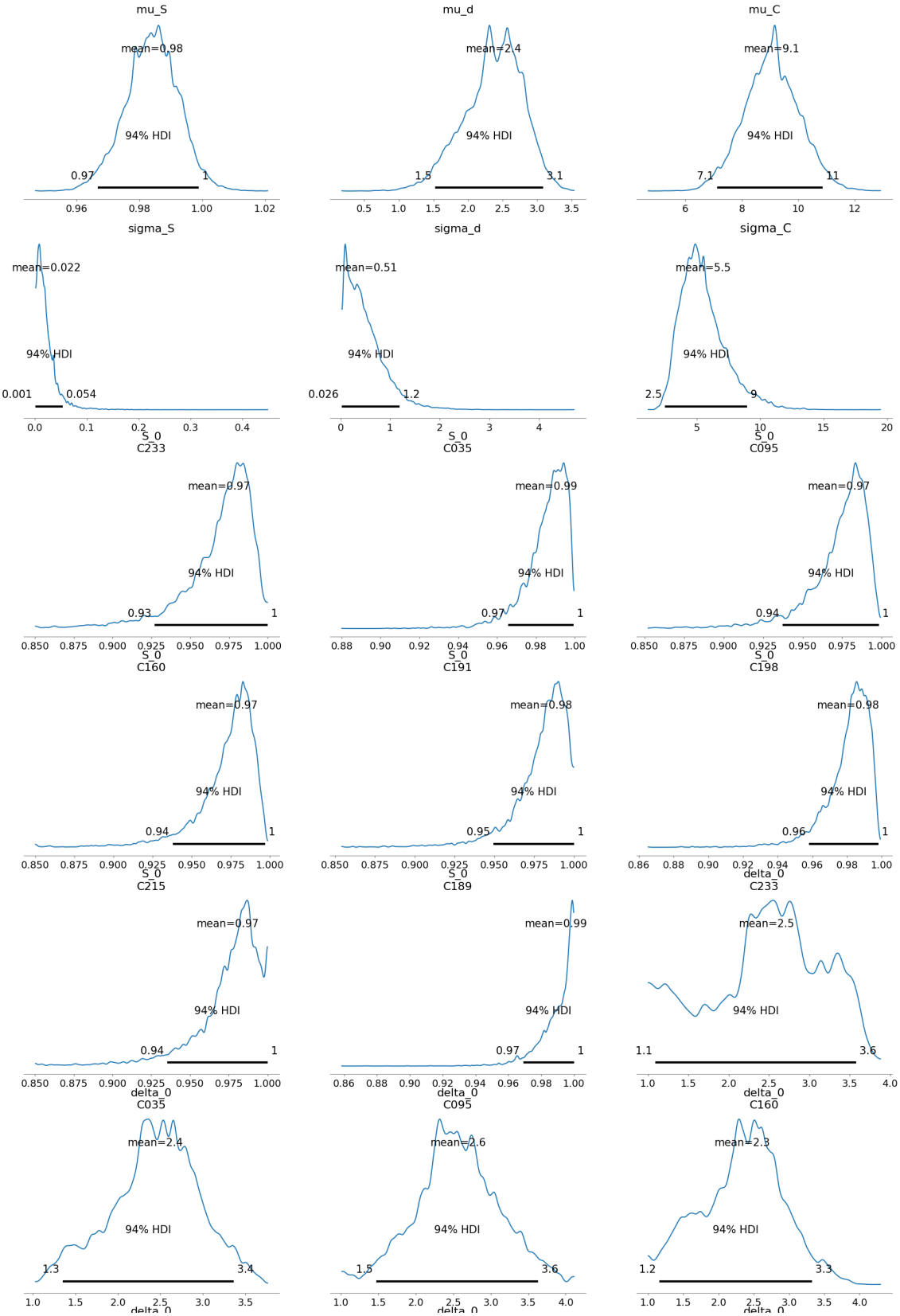


Figure A.3: Posteriors of the hierarchical model.

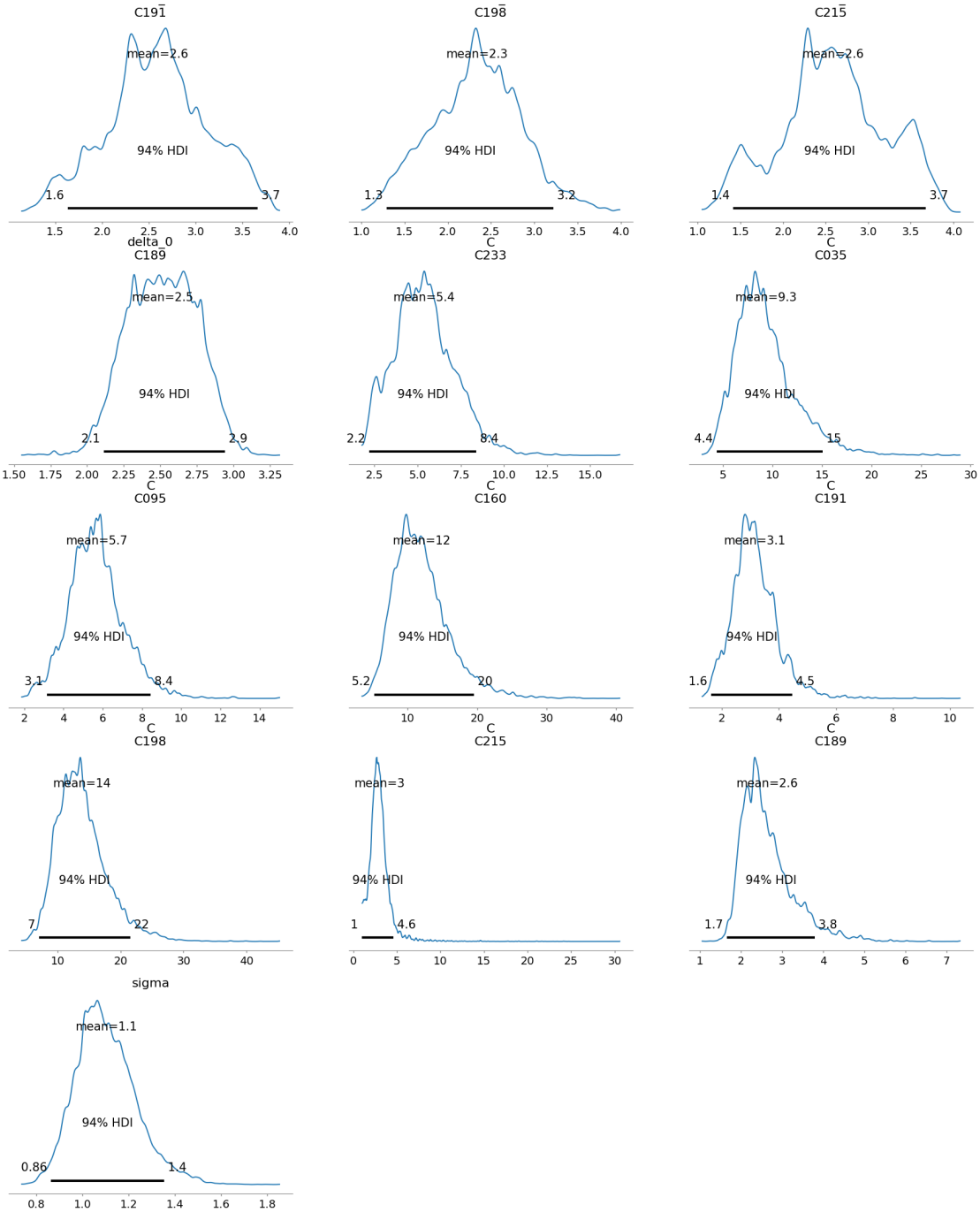


Figure A.4: Posteriors of the hierarchical model, continued.

UNIVERSITY OF
GOTHENBURG