Uncertainty in model-informed precision dosing

Predicting what you can and what you cannot predict



David Augustin

Corpus Christi College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2024

Supervisors

Prof. David Gavaghan, Department of Computer Science, University of Oxford

Dr. Ben Lambert, Department of Statistics, University of Oxford

Dr. Martin Robinson, Department of Computer Science, University of Oxford

Industrial supervisors

Dr. Ken Wang, Pharmaceutical Research and Early Development, Roche

Dr. Antje-Christine Walz, Pharmaceutical Research and Early Development, Roche

Examiners

Prof. Eamonn A. Gaffney

Prof. Piet H. van der Graaf

Studentship support

The author acknowledges the Clarendon Scholarship Fund and the Engineering and Physical Sciences Research Council for studentship support.

To Theodin, who made submitting so much easier

Abstract

Recent advances in Big Data, Machine Learning (ML), and Artificial Intelligence (AI) have affected almost all phases of the drug development life cycle, promising to reduce its extremely high attrition rates. However, drug development remains challenging, especially for those drugs displaying narrow therapeutic windows and large treatment response variability. For such drugs, dosing regimens often need to be individualised in order to balance the trade-off between their efficacy and their toxicity. ML & AI methods, among other modelling approaches, are increasingly used to support this dosing regimen individualisation – an approach known as model-informed precision dosing (MIPD).

In this thesis, we study aspects of the responsible use of MIPD in clinical practice, maximising its positive outcomes, while mitigating its risks. In this context, we particularly focus on the estimation of predictive uncertainty. We work in a Bayesian inference framework, where predictive uncertainty is accredited to the lack of information, resulting in structural and parametric uncertainty of models.

We begin our study by reviewing popular MIPD approaches, including pharmacokinetic & pharmacodynamic (PKPD) modelling as well as ML & AI approaches, such as regression and reinforcement learning. We then develop a simulation framework for clinical MIPD trials, enabling the development and evaluation of targeted MIPD approaches, while avoiding the astronomical costs of real clinical trials and guaranteeing the safety of patients throughout the development. Using this simulation framework, we find that MIPD approaches have different strengths and limitations, which demonstrates the importance of tailoring the MIPD approach to the use case. Overall, we find that for typical applications, where the availability of clinical data are limited, PKPD models provide the most responsible approach to MIPD – notably, PKPD modelling is not an ML or an AI approach.

We develop a new inference approach for population PKPD models, which we call 'filter inference'. Filter inference reduces the computational costs of nonlinear mixed effects (NLME) inference, making it applicable to datasets with measurements from thousands of patients, such as those obtained during post-market monitoring. The use of post-marketing data has the potential to improve estimates of inter-individual variability, which would improve uncertainty estimates during MIPD. We also study the strengths and limitations of information criteria for the estimation of structural uncertainty of PKPD models, identifying future directions for research in responsible MIPD.

As part of this thesis research, an open source Python package has been developed (https://chi.readthedocs.io), facilitating the responsible use of MIPD.

Contents

List of Abbreviations **ix**

1 Introduction **1**
1.1 Thesis outline . . . 4

2 Background **7**
2.1 MIPD in a nutshell . . . 8
2.2 The PKPD modelling approach to MIPD . . . 13

3 Simulating clinical trials for model-informed precision dosing **98**
3.1 Introduction . . . 99
3.2 Methods . . . 102
3.3 Results & discussion . . . 116
3.4 Conclusion . . . 135
3.5 Data and software availability . . . 137

4 Filter inference: A scalable nonlinear mixed effects inference approach **138**
4.1 Introduction . . . 139
4.2 Methods . . . 141
4.3 Results & discussion . . . 155
4.4 Conclusion . . . 175
4.5 Data and software availability . . . 176

5 On the locality of information criteria and its consequences for MIPD **177**
5.1 Introduction . . . 178
5.2 Methods . . . 180
5.3 Results & discussion . . . 193
5.4 Conclusion . . . 207
5.5 Data and software availability . . . 209

6 Discussion **210**
6.1 Main contributions . . . 211
6.2 Future research . . . 214

Appendices

A Software development: A Python package for treatment response modelling **218**
A.1 Introduction . . . 218
A.2 Methods . . . 219
A.3 Results & discussion . . . 222
A.4 Conclusion . . . 229
A.5 Data and software availability . . . 230

B Supporting information for Chapter 3 **231**
B.1 Warfarin clinical trial model . . . 232
B.2 The Regression model . . . 269
B.3 The Deep RL model . . . 272
B.4 The PKPD model . . . 280
B.5 Clinical trial simulation . . . 301
B.6 Supplementary tables . . . 305

C Supporting information for Chapter 4 **306**
C.1 Optimised implementation of filter inference . . . 306
C.2 Estimation of early cancer growth model parameters . . . 307
C.3 Estimation of variance estimation error . . . 307
C.4 IIV-noise distinguishability . . . 308
C.5 Estimating the average of a population distribution over the posterior distribution . . . 310
C.6 Estimation of EGF pathway model parameters . . . 311
C.7 Estimation of the evaluation time . . . 312
C.8 Hyperparameter tuning of MH algorithm . . . 313
C.9 Equivalence of filter inference with a Gaussian filter and ABC based on the mean and variance . . . 314
C.10 Equivalence of filter inference with identifiable summary statistics-based filters and ABC based on the same summary statistics . . . 315
C.11 Supplementary tables and figures . . . 316

D Supporting information – Chapter 5 **318**
D.1 Infliximab candidate models . . . 318
D.2 Data-generating population model for infliximab . . . 321
D.3 MAP estimation . . . 322
D.4 Dose optimisation using MAP estimates . . . 322
D.5 Estimation of posterior distributions . . . 322

D.6 Estimation of the posterior predictive distributions in Figure 5.8 323
D.7 Correctly individualised dose for simulated patient 324
D.8 Estimation of posterior predictive dose distribution 324
D.9 Estimation of probability assigned to dose interval 325
D.10 Estimation of exact cross-entropies 326
D.11 Proof of convergence to the local cross-entropy 327
D.12 Supplementary tables 329

References **331**

List of Publications

Publications relating to chapters of this thesis

1. **DA**. Chi: A Python package for treatment response modelling. *Journal of Open Source Software*. 2024.

2. **DA**, Lambert B, Robinson M, Wang K, Gavaghan D. Simulating clinical trials for model-informed precision dosing: Using warfarin treatment as a use case. *Front. Pharmacol. - Drug Metabolism and Transport*. 2023.

3. **DA**, Lambert B, Wang K, Walz AC, Robinson M, Gavaghan D. Filter inference: A scalable nonlinear mixed effects inference approach for snapshot time series data. *PLOS Computational Biology*. 2023.

4. **DA**, Wang K, Walz AC, Lambert B, Clerx M, Robinson M, Gavaghan D. Treatment response prediction: Is model selection unreliable?. *bioRxiv*. 2022.

Other publications

5. Creswell R[†], **DA**[†], Bouros I[†], Farm HJ[†], Miao S[†], Ahern A[†], et al. Heterogeneity in the onwards transmission risk between local and imported cases affects practical estimates of the time-dependent reproduction number. Philosophical Transactions of the Royal Society A. 2022.

6. Clerx M, **DA**, Dale-Evans AR, Mirams GR. Derivative-based Inference for Cell and Channel Electrophysiology Models. Computing in Cardiology (CinC). IEEE. 2022.

7. Foss-Solbrekk K, **DA**. The EU Orphan Drug Regulation 20 Years On: a Quantitative and Legal Analysis. European Intellectual Property Review. 2021.

[†] Equal contribution.

List of Abbreviations

ABC . . . . . Approximate Bayesian inference

ACMC . . . . . Adaptive covariance matrix Monte Carlo

ADME . . . . . Absorption, distribution, metabolism and excretion

AI . . . . . Artificial intelligence

AIC . . . . . Akaike information criterion

AUC . . . . . Area under the curve

BSL . . . . . Bayesian synthetic likelihood

CMAES . . . . Covariate matrix adaption evolution strategy

CT model . . . Clinical trial model

EGF . . . . . Epidermal growth factor

EGFR . . . . . Epidermal growth factor receptor

ESS . . . . . Effective sample size

FOSS . . . . . Free and open-source software

HMC . . . . . Hamiltonian Monte Carlo

iid . . . . . Independent and identically distributed

IIV . . . . . Inter-individual variability

IOV . . . . . Inter-occasion variability

INR . . . . . International normalised ratio

KDE . . . . . Kernel density estimation

KL divergence Kullback-Leibler divergence

LKO estimate . leave-k-out estimate

LOO estimate . leave-one-out estimate

MAP estimate Maximum a posteriori estimate

MCMC . . . . . Markov chain Monte Carlo

MH MCMC . . Metropolis-Hastings Markov chain Monte Carlo

MIPD . . . . . Model-informed precision dosing

MLE . . . . . Maximum likelihood estimate

MSE . . . . . Mean squared error

NLME model . Nonlinear mixed effects model

NSCLC . . . . . Non-small cell lung cancer

NUTS . . . . . No-U-Turn sampler

PBPK model . Physiology-based pharmacokinetics model

PD . . . . . Pharmacodynamics

PK . . . . . Pharmacokinetics

PKPD . . . . . Pharmacokinetics & pharmacodynamics

PSIS-LOO . . . Pareto-smoothed importance sampling leave-one-out

PSO . . . . . Particle swarm optimiser

QSP model . . Quantitative systems pharmacology model

RL . . . . . Reinforcement learning

SGD . . . . . Stochastic gradient descent

TR . . . . . Therapeutic range

TTR . . . . . Time in the therapeutic range

VKORC . . . . Vitamin K epoxide reductase complex

VKORC1 . . . VKORC's subunit 1

WAIC . . . . . Watanabe-Akaike information criterion

1 Introduction

Contents

1.1 Thesis outline 4

Treatment response variability across patients is a common phenomenon in clinical practice with complex and not always fully understood origins. Many drugs can be administered safely without accounting for this inter-individual variability. However, for some drugs, differences between treatment responses are severe and individualisation of dosages is needed to avoid harmful adverse events. Model-informed precision dosing (MIPD) is an emerging technique used to individualise dosing regimens of otherwise difficult-to-administer drugs (Keizer et al. 2018; Peck 2021; Polasek, Shakib, et al. 2018; Sheiner 1969).

Typical examples that would benefit from precision dosing are drugs with narrow therapeutic windows and large treatment response variability, including blood thinners, such as warfarin (Wadelius and Pirmohamed 2007), chemotherapies, such as docetaxel (Yuxiang Ma et al. 2021), and monoclonal antibody treatments, such as infliximab (Gill et al. 2016). Other examples include antibiotics, like vancomycin, where MIPD has been suggested and partially implemented to guide dosing strategies for critically ill patients, balancing the treatment of severe infections and the risks for harmful adverse events (Broeker et al. 2019; Matsumoto et al. 2022; Wicha et al. 2021). Medicines for

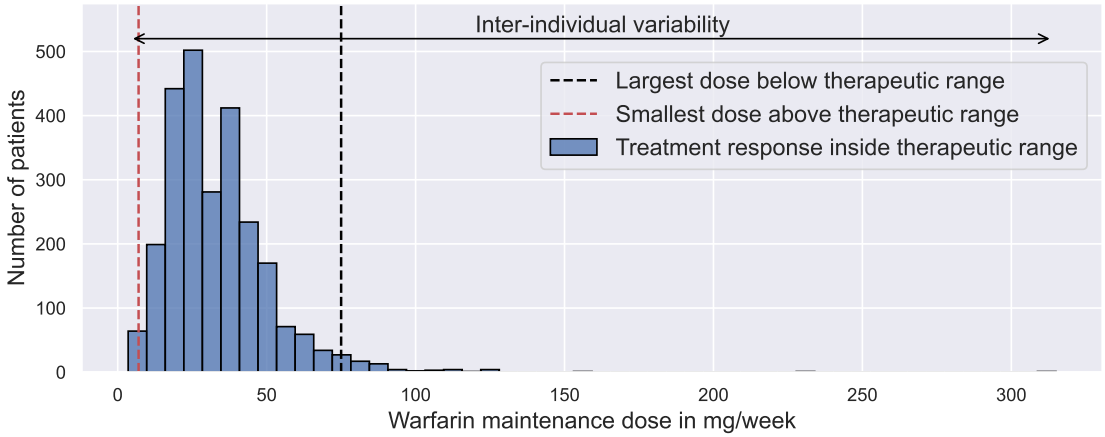


Figure 1.1: Warfarin treatment response variability. The figure shows clinically determined maintenance dosages of warfarin in a cohort of 3419 patients, targeting a therapeutic range of INRs between 2 and 3 (Wadelius and Pirmohamed 2007). The histogram illustrates the individualised maintenance dosages for patients with treatment responses inside the therapeutic window (2546 patients). The dashed lines illustrate the smallest dose with an INR above the therapeutic range (red) and the largest dose with an INR below the therapeutic range (black).

which dosing regimens need to be continuously adapted to changing conditions may also benefit from MIPD, for example in the context of diabetes where insulin treatment is used to stabilise patients' blood glucose levels (Y. Wang et al. 2019; T. Zhu et al. 2020).

Figure 1.1 illustrates the large degree of inter-individual variability possible in clinical practice. The figure shows the result of a clinical precision dosing study for warfarin, from a cohort of 3419 patients (Wadelius and Pirmohamed 2007). The objective of the study was to find individualised maintenance dosages for each patient, targeting a fixed therapeutic window. Treatment responses outside this therapeutic window were considered to be either too weak, failing to generate the desired therapeutic effect, or too strong, increasing the risk for harmful adverse events. An in-depth discussion of warfarin, its treatment effect, and its therapeutic window is deferred to Chapter 3.

The result of the study was that maintenance doses substantially varied across patients (see histogram in Figure 1.1). The typical (median) maintenance dose needed for patients to display treatment responses inside the therapeutic window was found to be 30 mg/week with the 90 % bulk of the dose distribution ranging from 12 mg/week to 63 mg/week across patients. The weakest responder of the cohort needed a weekly maintenance dose of almost 100 times the dose administered to the strongest responder

(dose for the weakest responder: 315 mg/week; dose for the strongest responder: 3.5 mg/week). Notably, only 2546 out of 3419 patients displayed treatment responses inside the therapeutic range. The smallest dose that led to an excessive treatment response was found to be 5 mg/week – a dose smaller than most doses that successfully targeted the therapeutic window for other patients in the cohort (see red dashed line in Figure 1.1). Similarly, the largest dose resulting in a too weak treatment response was found to be larger than the majority of the successful ones (largest dose below therapeutic window: 74 mg/week, see black dashed line in Figure 1.1).

The warfarin study demonstrates that treatment responses can display extreme inter-individual variability in clinical practice, show-casing the necessity of precision dosing for some drugs and the individualisation of their dosing regimens. But individualising dosages is not a straightforward task, as the dynamics of treatment responses are often nonlinear and delayed, making 'manual' dose adjustments from monitoring data without any additional guidance challenging. In addition, dose predictions are complicated by data quality constraints, featuring limitations of the monitoring frequency, the accessibility of quantities of interest, and the corruption of measurements through measurement noise. Inter-occasion variability as a result of time-varying pharmacokinetics, concomitant food-intake, comedication and disease progression also adds to the complexity of the picture (Keutzer et al. 2020). MIPD has emerged as a promising technique to overcome these challenges in clinical practice.

The most prominent MIPD methods are regression, reinforcement learning (RL) and pharmacokinetic and pharmacodynamic (PKPD) modelling (Johnson, Caudle, et al. 2017; Johnson, L. Gong, et al. 2011; Keizer et al. 2018; Ribba, Dudal, et al. 2020). These methods may be further categorised into different subvariants. For example, RL variants include model-free algorithms such as Q learning (Zadeh et al. 2023) and model-based algorithms such as Monte Carlo tree search (Maier, Hartung, Kloft, et al. 2021). MIPD variants of PKPD modelling include maximum a posteriori-guided dosing and Bayesian data assimilation-guided dosing (Maier, Hartung, de Wiljes, et al. 2020). Other variants include the modelling of virtual twins using physiology-based pharmacokinetic (PBPK)

and quantitative systems pharmacology (QSP) models (Polasek and Rostami-Hodjegan 2020). A detailed introduction to MIPD will follow in Chapter 2.

Although differences across MIPD methods exist, all approaches have in common that they produce individualised dosing regimen predictions – predictions that can provide guidance and can help to successfully target desired treatment responses despite substantial inter-individual variability. However, this upside of MIPD assumes that model predictions are generally, or at least mostly, correct, when, in fact, model predictions *can be erroneous* or plainly *wrong*. Such erroneous predictions can have severe consequences for the well-being of patients, especially in the context of MIPD, including undertreatments that leave patients unnecessarily exposed to disease progression or other risk factors, and overtreatments that potentially lead to overexposure, elevated toxicity, and dangerous adverse events. In this thesis, we will research how we can minimise the risk for erroneous or wrong predictions as much as possible by making the uncertainty estimation of predictions more transparent and complete.

1.1 Thesis outline

In the next chapter, Chapter 2, we will introduce the necessary background information for the rest of the thesis. This will include basic concepts of MIPD and an introduction to the most prominent MIPD approach: PKPD modelling. Basic introductions to regression, and RL are included in Appendix B. In the context of PKPD modelling, we will go beyond the basics and introduce common notions of predictive uncertainty, such as: 1. predictive uncertainty as a result of parametric uncertainty; and 2. predictive uncertainty as a result of structural uncertainty. Estimating parametric uncertainty will be the focus of Chapter 4, and estimating structural uncertainty will be the focus of Chapter 5.

In Chapter 3, we will introduce a simulation framework to test and develop MIPD methods, providing a platform to mature MIPD models before testing them in clinical practice (Augustin, Lambert, Robinson, et al. 2023). Central to the framework is a drug-specific clinical trial model that captures available pharmacological and physiological knowledge. It also emulates common challenges for precision dosing in clinical practice. We will demonstrate the use of the framework by implementing it for the use case

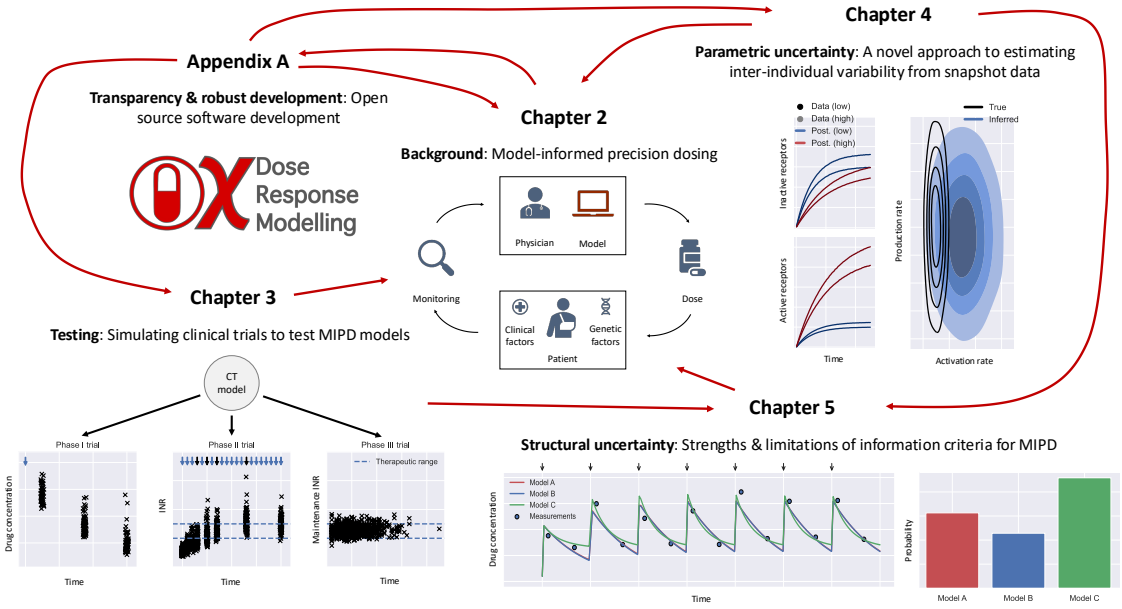


Figure 1.2: Graphical abstract of thesis.

of warfarin treatment. We will simulate three clinical MIPD trials: one trial for a regression model; one trial for a RL model; and one trial for a PKPD model. Our findings from this chapter identify PKPD modelling as the most promising approach to a wide class of MIPD problems.

In Chapter 4, we will take a step back and zoom in on PKPD modelling and the existing methodology for predictive uncertainty estimation. In particular, we will focus on nonlinear mixed effects (NLME) modelling – an approach that can be used to estimate parametric uncertainty of PKPD models as a result of inter-individual variability. NLME modelling estimates parametric uncertainty from population data prior to MIPD. As the number of measured individuals increases, the estimate of the inter-individual variability, and hence the estimate of the uncertainty, becomes more accurate. However, the computational costs also grow with the population size, making NLME modelling computationally intractable when data across too many individuals are collected. We introduce a novel inference approach, which we term 'Filter inference' (Augustin, Lambert, K. Wang, et al. 2023), making NLME modelling tractable for large population sizes. Filter inference enables the use of (sparse) clinical treatment response monitoring data across a large number of individuals for NLME modelling.

In the final research chapter of this thesis, Chapter 5, we will take another step back and zoom in on model assumptions and their role in producing erroneous dosing regimen predictions (Augustin, Walz, et al. 2022). We will review how structural uncertainty can be used as a framework to relax model assumptions when the pharmacological and physiological processes involved in a drug's PKPD are uncertain. The most common approach in PKPD modelling to handle such structural uncertainty is to estimate the consistency of multiple candidate models with the available measurements and to select the most consistent model for prediction. This consistency is typically quantified using information criteria, such as the Akaike information criterion (AIC). We will investigate the strengths and limitations of the information criteria for MIPD. We find that some information criteria are not able to provide robust estimates of the structural uncertainty in the context of MIPD. One of those information criteria is the AIC.

In Chapter 6, we will make concluding remarks on our findings and highlight some of the many research avenues that we were not able to pursue in this thesis but hopefully will be able to investigate in the future.

Alongside this thesis, we developed a free and open source Python package for PKPD modelling and MIPD (Augustin 2024). Chi was central to carrying out our research, and we hope it will help the wider community to further progress the field of MIPD. For details on Chi, we refer to Appendix A.

2 Background

Extracts of this chapter have been published in:

DA, Lambert B, Robinson M, Wang K, Gavaghan D. Simulating clinical trials for model-informed precision dosing: Using warfarin treatment as a use case. *Front. Pharmacol. - Drug Metabolism and Transport*. 2023.

DA. Chi: A Python package for treatment response modelling. *Journal of Open Source Software*. 2024.

DA. Documentation of 'Chi: A Python package for treatment response modelling'. https://chi.readthedocs.io. 2024

Contents

2.1 MIPD in a nutshell . . . 8
2.1.1 The dose response relationship . . . 8
2.1.2 The therapeutic window . . . 10
2.1.3 Inter-individual variability . . . 11
2.1.4 The problem statement . . . 13
2.2 The PKPD modelling approach to MIPD . . . 13
2.2.1 The mechanistic model . . . 13
2.2.2 The dosing regimen . . . 20
2.2.3 From mechanistic models to dosing regimens . . . 23
2.2.4 Estimating model parameters from data . . . 28
2.2.5 Parametric uncertainty & Bayesian inference . . . 35
2.2.6 Sampling from posterior distributions using MCMC . . . 42
2.2.7 Nonlinear mixed effects modelling . . . 54

2.2.8 Estimating parametric uncertainty from population data . 61
2.2.9 Structural uncertainty & Bayesian inference 70
2.2.10 Information criteria 84

In this chapter, we will introduce fundamentals of pharmacology, mathematics, and statistics relevant to this thesis. We will begin by reviewing the concept of a therapeutic window which we will then use to formalise the objective of MIPD. We will conclude the chapter by introducing different approaches to MIPD, including PKPD modelling, regression, and RL.

Note that this chapter provides by no means an exhaustive review of the MIPD literature, and the overview will be biased towards the topics relevant for this thesis.

2.1 MIPD in a nutshell

As alluded to in the Introduction, MIPD is a modelling technique that can be used to tailor dosing regimens to individuals such that desired treatment responses can be realised across patients despite inter-individual differences (see Chapter 1). In this section, we want to formalise what we mean by a 'desired treatment response' and cement the causal relationship between the dose of a drug and its treatment response.

2.1.1 The dose response relationship

In order to formalise the notion of a desired treatment response, we first need to appreciate that drugs are extremely versatile. No drug is like another. Drugs generally have different molecular compositions, active ingredients, modes of action, transport properties, and metabolic rates, just to name a few. But the most obvious distinction of drugs is, perhaps, their indication – some drugs may be used to treat cancer, others may prevent the formation of blood clots or the onset of neurodegenerative diseases, such as Alzheimer's and Parkinson's disease. Despite these differences, it is useful for the purpose of MIPD to introduce a drug-agnostic notion of a desired treatment response.

One way to approach this is the dose response curve of a drug which conceptualises the strength of a drug effect as a function of its dose (see left panel in Figure 2.1).

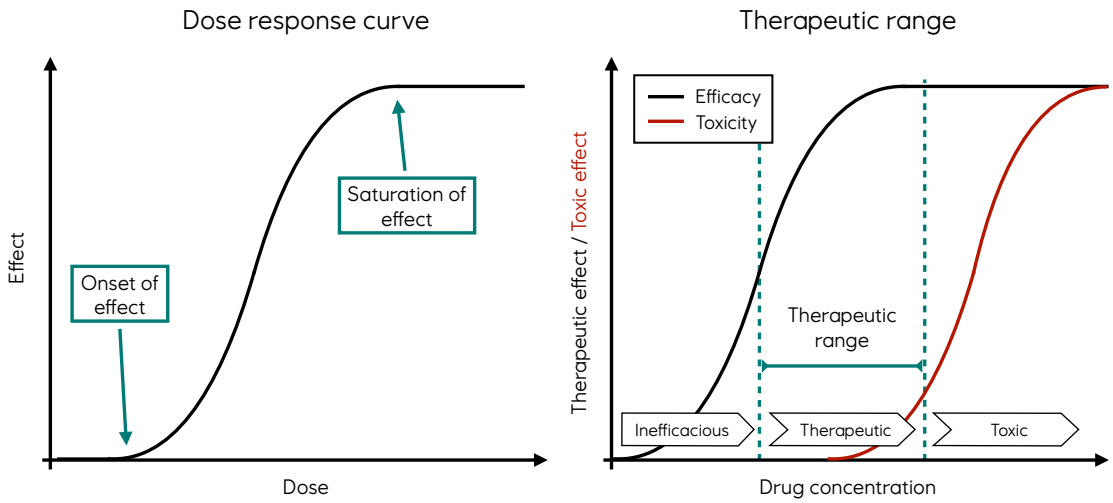


Figure 2.1: The dose response curve & the therapeutic range. The left panel of the figure shows a typical dose response curve of a drug, illustrating the effect of a drug as a function of its dose. The right panel of the figure shows the therapeutic effect (black curve) and the toxic effect (red curve) of a drug as a function of its concentration. The therapeutic range is indicated by teal dashed lines, indicating drug concentrations with an optimal trade-off between drug efficacy and drug toxicity.

The panel shows a typical dose response curve, featuring three of its key characteristics common to all drugs: 1. the effect of a drug does not depend linearly on the administered dose; 2. a minimum dose is needed for the onset of an effect; and 3. the effect saturates and stops increasing with the dose after a threshold dose is surpassed (Moini et al. 2023). This response behaviour is generic to all drugs. However, the exact dose for onset and saturation, as well as the steepness of the dose response curve generally varies between drugs.

To get a feeling for why drugs are bound to display dose response curves of the shape illustrated in Figure 2.1, it is worth taking a closer look at the mode of action of a particular drug: erlotinib. Erlotinib is a small molecule drug used to treat non-small cell lung cancer (NSCLC) (Bronte et al. 2014; Khozin et al. 2014). The therapeutic effect of erlotinib derives from its interference with tumour cell proliferation, differentiation, and development by binding to a central component involved in these processes – the epidermal growth factor receptor (EGFR) tyrosine kinase (Yano et al. 2003). EGFR is located at the cell surface of tumour cells, and each cell has a large number of EGFR copies. If too few EGFR tyrosine kinases are blocked by erlotinib, sufficiently many

copies function undisturbed, illustrating why a minimum dose of erlotinib is needed to see a therapeutic effect. Similarly, once a sufficient number of EGFR tyrosine kinases are disrupted, increasing the dose of erlotinib will not strengthen its therapeutic effect as the tumour cell proliferation will already have been maximally inhibited. This gives rise to a plateau of the therapeutic effect for large erlotinib doses similar to the one illustrated in the left panel in Figure 2.1.

While modes of action may vary across drugs – for example, some drugs promote biological processes as opposed to inhibiting them, such as monoclonal antibodies which label target cells for destruction by the adaptive immune system (S. Singh et al. 2018) – most modes of action will involve binding or interacting with a target protein or molecule to invoke their effect (Goodsell et al. 2020). This binding or interacting will be subject to similar onset and saturation dynamics as the erlotinib example.

2.1.2 The therapeutic window

The definition of a desired treatment response can be motivated with an extension of the dose response curve illustrated in the right panel of Figure 2.1. The panel shows the therapeutic effect, or efficacy, of a drug as a function of its concentration at the site of action (see black curve). In agreement with the dose response curve in the left panel, the efficacy has an onset and a plateau as a function of the drug concentration. The addition to the picture presented before is that the panel also shows the onset of a drug-induced toxicity for higher drug concentrations (see red curve). By toxicity we mean any adverse events caused by the drug as a result of disrupting or promoting untargeted biological processes (Gillette et al. 1974). The two primary toxicities associated with erlotinib are, for example, skin rash and diarrhoea (Rudin et al. 2008), but adverse events can, in general, also be more extreme, such as cardiac arrest or carcinogenesis (Laverty et al. 2011). Analogously to the therapeutic effect, the toxic effect of a drug also follows the general shape of a dose response curve, leading to a plateau of the toxicity for higher drug concentration levels.

The efficacy curve and the toxicity curve together partition the space of possible drug concentration levels into three domains: 1. the subtherapeutic domain, where the

drug exposure is insufficient to generate the desired therapeutic effect; 2. the therapeutic domain, where the drug exposure is sufficient to generate the desired therapeutic effect, but not too excessive to cause serious toxicity levels; and 3. the toxic domain, where the drug exposure leads to substantial toxicity and adverse events (see right panel in Figure 2.1). The size and the concentration ranges for the three domains are drug-specific, and for drugs that are not fit-for-purpose, the onset of toxicity may even precede the onset of efficacy (Moini et al. 2023).

The therapeutic domain, commonly referred to as the therapeutic range or therapeutic window, is the target of any dosing strategy and can be used to formalise a drug-agnostic notion of a desired treatment response: a desired treatment response is one resulting from a continual drug exposure inside the therapeutic range. Note that this definition uses the wording 'continual drug exposure', hinting at the dynamic nature of drug exposures during treatments. We will defer an in-depth discussion of the dynamics of treatment responses to section 2.2. For now, we just highlight that the transience of drug concentration levels as a result of continuous metabolism or excretion is a mandatory feature of safe drugs, guaranteeing that no permanent accumulation of drug residues in patients' bodies is possible. This dynamic behaviour of drug exposures needs to be accounted for during precision dosing.

2.1.3 Inter-individual variability

The high-level goal for drug development, once a drug candidate has been identified, is to find safe and efficacious dosing regimens – i.e. dosing regimens that target the therapeutic window (Kaitin 2010; Tamimi et al. 2009). The establishment of such dosing regimens unlocks the potential for regulatory approval and the more wide-spread use of the drug in clinical practice. A challenge with identifying safe and efficacious dosing regimens is that individuals generally respond differently to the same levels of drug exposures, resulting in inter-individual variability (IIV) of the therapeutic window (see left panel in Figure 2.2).

The panel shows a schematic illustration of a drug's efficacy and toxicity curves for four individuals. The therapeutic windows are illustrated using teal-shaded areas. The figure shows substantial differences in the treatment responses between individuals,

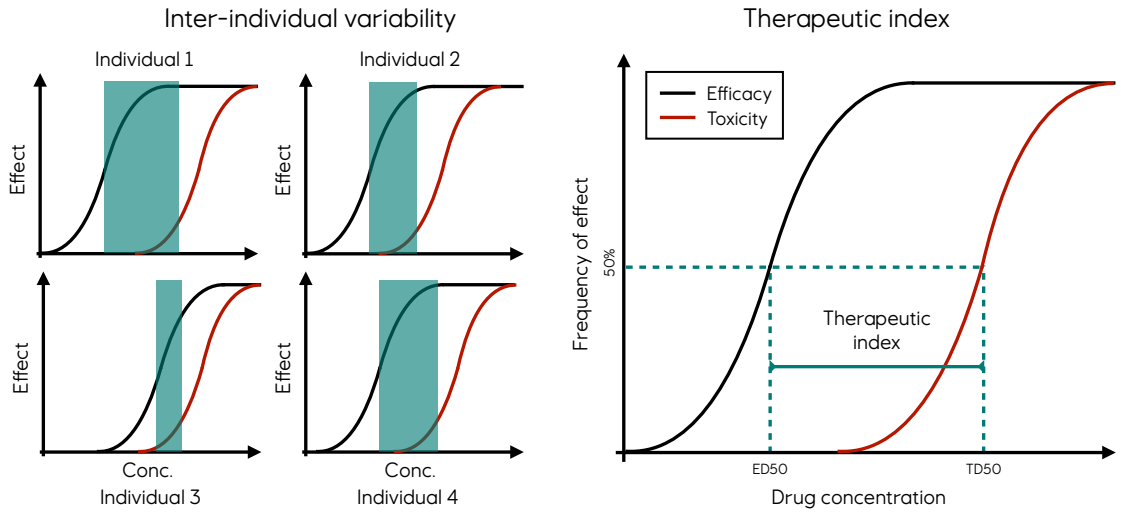


Figure 2.2: Inter-individual variability & the therapeutic index. The left panel of the figure shows the efficacy (black) and the toxicity of a drug for four individuals (top-left subpanel: individual 1; top-right subpanel: individual 2; bottom-left subpanel: individual 3; and bottom-right subpanel: individual 4). The therapeutic window for each individual is highlighted using a teal-shaded area. The right panel of the figure shows the cumulative frequency of therapeutic (black) and toxic (red) effects across a cohort of patients. The ED50 and the TD50 are highlighted using dashed lines.

illustrating that the therapeutic window does typically not generalise between patients (Y. S. Lin et al. 2021; Peck 2021; Thummel et al. 2014).

To overcome this lack of generalisability, therapeutic windows are traditionally established using a statistical approach, where the safety and efficacy of dosing regimens are studied in multiple phases of clinical trials with escalating numbers of patients (Tamimi et al. 2009). The result of these clinical trials is a distribution of therapeutic windows, which can be summarised using a therapeutic index. The therapeutic index[1] is defined as the ratio of the median efficacious dose, ED50, and the median toxic dose, TD50, across patients (Moini et al. 2023).

The right panel in Figure 2.2 illustrates this statistical approach. The panel shows the cumulative frequency of therapeutic effects (black) and toxic effects (red) in a patient cohort as a function of the drug concentration. The ED50 and the TD50 are illustrated using dashed lines. The range between the ED50 and the TD50 marks the

[1]Note that different definitions of the therapeutic index exist and we refer to Muller et al. (2012) for an extensive review.

concentration levels where the majority of the patients display the targeted therapeutic effect without surpassing threshold toxicity levels.

2.1.4 The problem statement

For drugs with therapeutic indices substantially below 1 – i.e. with typical toxic dosages much larger than typical efficacious dosages – it is often possible to find a single dosing regimen that can be safely and efficaciously administered to all patients, without accounting for IIV. However, for those drugs whose ED50 and TD50 are considerably closer together, or whose toxic effect is so extreme that even a rare occurrence cannot be tolerated, dosing regimens need to be tailored to the therapeutic window of each individual. The goal of MIPD is to determine these individualised dosing regimens.

2.2 The PKPD modelling approach to MIPD

In the previous section, we established the causal relationship between a drug's dose and its treatment effect. We also formalised the concept of a therapeutic window and reiterated the existence of treatment response variability across individuals. In this section, we introduce the PKPD modelling approach to MIPD, and discuss challenges of precision dosing relating to the dynamic nature of treatment responses, nonlinearities, and treatment response delays.

2.2.1 The mechanistic model

PKPD modelling is a widely used modelling framework in the pharmaceutical industry, dating back to 1937 when Torsten Teorell published two seminal papers on the 'Kinetics of distribution of substances administered to the body' (Teorell 1937a,b). In his articles, Teorell demonstrates the use of mathematical models in describing and simulating the time course of drug concentration levels in different parts of the body following dose administrations – a modelling approach that became later known as pharmacokinetic (PK) modelling (Paalzow 1995; Wagner 1981).

In PK modelling, the time course of administered substances is captured using approximate descriptions of the body in the form of simplified compartmental models,

focussing only on the circulation of the substance in the blood and in a few, select tissues (Teorell 1937a). In this approach, interactions between the substance and the body, such as the substance's absorption, distribution, metabolism and excretion (ADME), are not modelled in detail, and instead are captured using a semi-mechanistic description in terms of differential equations and transition rates between the compartments. Because of this mechanistic nature, we will refer to PK models (and later also to PKPD models) as mechanistic models.

To illustrate the idea of PK modelling, Figure 2.3 shows Teorell's original PK model for subcutaneously administered substances from 1937 (Teorell 1937a). The left panel shows a pictorial representation of the compartmental model and the right panel shows the simulated drug amounts. The mathematical representation of the model is a system of differential equations with three states

$$\begin{aligned} \frac{\mathrm{d}a_d}{\mathrm{d}t} &= -k_1\, a_d, & a_d(t=0,\psi) &= N_0, \\ \frac{\mathrm{d}a_b}{\mathrm{d}t} &= k_1\, a_d - k_2\, a_b + k_3\, a_t - k_4\, a_b, & a_b(t=0,\psi) &= 0, \\ \frac{\mathrm{d}a_t}{\mathrm{d}t} &= k_2\, a_b - k_3\, a_t - k_5\, a_t, & a_t(t=0,\psi) &= 0, \end{aligned} \tag{2.1}$$

where a_d denotes the drug amount in the depot compartment, a_b denotes the drug amount in the blood compartment, and a_t denotes the drug amount in a generic tissue compartment (see compartments from left to right in the left panel of Figure 2.3). The concentration of the drug in each compartment is derived from the drug amount and the effective volume of distribution in the respective compartment, e.g. $c_b = a_b/v_b$ for the blood compartment. The volumes of distributions are parameters of the model. k_1, k_2, k_3, k_4, and k_5 are also parameters of the model denoting 1. the absorption rate from the depot into the blood compartment, k_1; 2. the transition rate from the blood to the tissue, and from the tissue to the blood, k_2 and k_3; 3. the elimination rate of the drug through the liver, the lung, and the kidneys, k_4; and the fixation rate of the drug in the tissue through metabolic processes, k_5. We will denote the parameters of the model collectively as ψ. The initial conditions of the differential equations in Equation 2.1 represent the subcutaneous dose administration, where N_0 denotes the dose amount administered to the depot compartment.

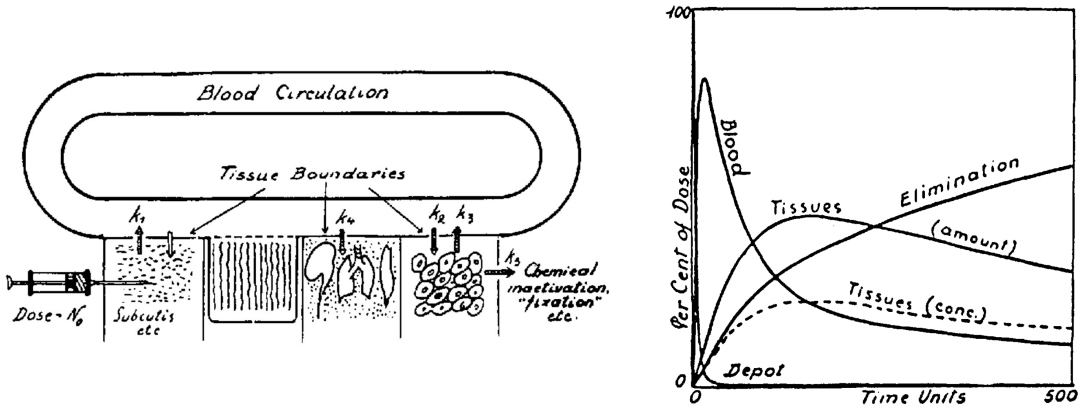


Figure 2.3: PK model of subcutaneous administration. The figure shows Teorell's compartmental PK model published in 1937 (Teorell 1937a). The left panel shows the compartments of the model: the 1st subpanel shows the depot compartment with subcutaneous drug administration; the 2nd subpanel shows the blood compartment connecting all compartments; the 3rd subpanel shows the elimination compartment, including the liver, the lung, and the kidneys; and the 4th subpanel shows a generic tissue compartment. The right panel shows the simulated time course of the drug in the different parts of the body. Drug amounts are illustrated using solid lines. Drug concentrations are illustrated using dashed lines.

Equation 2.1 is a simple linear system of differential equations which can be solved analytically in one step when expressed in its tensor representation

$$\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t} = \mathbf{A}\,\mathbf{a}, \quad \mathbf{a}(t=0,\psi) = \mathbf{a}_0 \quad \Leftrightarrow \quad \mathbf{a}(t,\psi) = \mathbf{a}_0 \mathrm{e}^{\mathbf{A}t}, \tag{2.2}$$

where $\mathbf{a} = (a_d, a_b, a_t)^T$ is the drug amount state vector, $\mathbf{a}_0 = (N_0, 0, 0)^T$ is the vector of initial drug amounts in the compartments, and

$$\mathbf{A} = \begin{pmatrix} -k_1 & 0 & 0 \\ k_1 & -k_2 - k_4 & k_3 \\ 0 & k_2 & -k_3 - k_5 \end{pmatrix} \tag{2.3}$$

is the matrix of change rates between the compartments.

Evaluating Equation 2.2 for given values of ψ traces out the drug amount profiles illustrated in the right panel of Figure 2.3. The panel shows that initially the full dose of the substance is concentrated in the depot compartment, where the drug is subcutaneously administered. Most of the substance then quickly transitions into the blood compartment from which it is slowly propagated to the tissue compartment.

Towards the end of the simulation, the majority of the substance has been removed from the system through the elimination compartment. Importantly, the relative speed of the transitions, and as a result the exact profile of the substance profiles, depends on the choice of the model parameters ψ. For example, substantially increasing the k_2 parameter of the model will elevate the substance levels in the tissue compartment, transforming the profile of the blood compartment into that of the depot compartment and reducing the overall eliminated substance amount. Interestingly, for this model the amount profiles shown in Figure 2.3 are independent of the actual dose amount, provided the amount is greater than zero, $N_0 > 0$, as the model is linear and the profiles are shown in units of the dose amount.

Since Teorell's seminal publications, the PK modelling framework has been extended in many directions, including its generalisation to nonlinear drug distribution and nonlinear drug elimination dynamics (An 2017; Bischoff et al. 1968; Csajka et al. 2006; Rowland et al. 1973). Arguably the most notable addition to the PK modelling framework is its extension to PKPD modelling (Salahudeen et al. 2017; Teorell et al. 1974). PKPD modelling complements the PK modelling approach with a pharmacodynamic (PD) element, focussing on the therapeutic or toxic effects incurred by a substance in the human body. Similar to PK modelling, PD models also make use of semi-mechanistic descriptions of the drug effects, avoiding any detailed accounts of the drug-mediated processes.

An example PKPD model is illustrated in Figure 2.4. The model provides a semi-mechanistic description of the response dynamics to warfarin treatment and consists of a PK component (left), and a PD component (right) with a total of 7 compartments (Hamberg, Wadelius, et al. 2010). Warfarin is an orally administered anticoagulant widely used for the prevention and treatment of venous thrombosis, pulmonary embolism and thromboembolic complications associated with atrial fibrillation and/or cardiac valve replacement (FDA 2010).

The mathematical representation of the model is a system of differential equations. Analogously to Teorell's model in Figure 2.3, the PK component consists of a depot

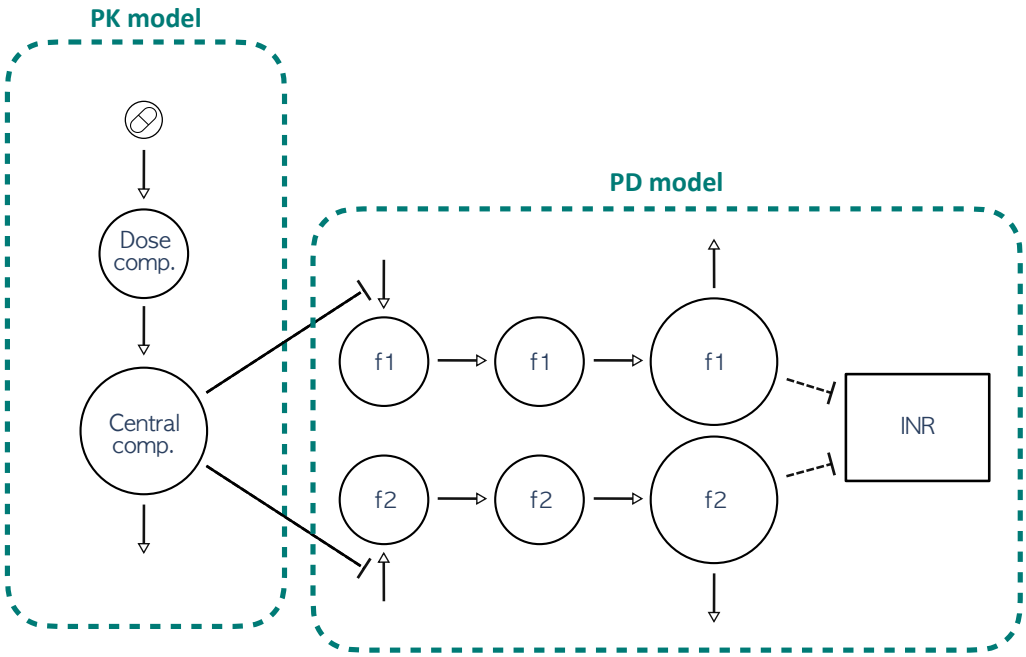


Figure 2.4: PKPD model of warfarin treatment response. The figure shows the PK model (left) and the PD model (right) of Hamberg et al's PKPD model of warfarin treatment (Hamberg, Wadelius, et al. 2010). The PK component consists of two compartments: the dose compartment; and the central compartment. The PD component consists of 7 compartments: two sets of three coagulation factor compartments, implementing a delay of the treatment response; and an INR compartment, capturing the quantity of interest of the warfarin treatment response. Warfarin interacts with the INR dynamics through the inhibition of the coagulation factor synthesis (see middle of the figure).

compartment and a blood compartment, in modern language referred to as dose compartment and central compartment, respectively

$$\begin{aligned} \frac{\mathrm{d}a_d}{\mathrm{d}t} &= -k_a a_d + r(t), & a_d(t=0,r,\psi) &= 0, \\ \frac{\mathrm{d}a_c}{\mathrm{d}t} &= k_a a_d - k_e a_c, & a_c(t=0,r,\psi) &= 0. \end{aligned} \tag{2.4}$$

Here, the dose compartment represents digestive processes following the ingestion of the drug, leading to a delayed increase of the drug amount in the blood, a_c. k_a denotes the corresponding absorption rate. k_e denotes the elimination rate of the drug from the system. $r(t)$ represents a time-dependent dose rate, generalising the administration of a single substance dose in Teorell's model (Equation 2.1) to arbitrary dosing regimens. We will come back to dosing regimens and their implementation in terms of the dose rate in the next section, subsection 2.2.2.

Note that the PK model does not include a tissue compartment, reflecting the fact that, as a blood thinner, warfarin's site of action is in the blood (the central compartment). Warfarin also shows little to no signs of depositing its active ingredient in any secondary tissues, making the implementation of a tissue compartment redundant (Hamberg, Dahl, et al. 2007). In modern language, the warfarin PK model is therefore referred to as 1-compartment PK model – one main compartment, and one dose compartment whose form depends on the route of administration. For example, for intravenous infusion, the drug is administered directly into a patient's blood stream, making a separate dose compartment redundant. Analogously, Teorell's PK model (Equation 2.1) is now referred to as 2-compartment PK model.

The focus of the PD component is the international normalised ratio (INR) and its dynamics in response to warfarin exposure. The INR is a measure for the delay between the activation of blood clotting pathways and the formation of clots in the blood. The larger the INR, the longer the blood takes to clot. The formation of blood clots is an important immune response to injury, enabling the repair of tissues and the closing off of wounds. However, in some instances blood clotting can be dangerous, for example when clots block arteries and cause strokes (Bivard et al. 2013). Warfarin's therapeutic effect is to prolong the clotting time in order to avoid adverse effects of blood clot formation. We defer more details of warfarin's pharmacology to Chapter 3.

The model describes the INR treatment response in terms of the dynamics of two effective 'coagulation factors': a fast responding coagulation factor, f_1; and a slow responding coagulation factor, f_2,

$$\bar{y} = \bar{y}_0 + \bar{y}_{max}\left(1 - \frac{f_1 + f_2}{2}\right). \tag{2.5}$$

Here, $\bar{y}$ denotes the INR, $\bar{y}_0$ denotes the baseline INR, and $\bar{y}_{max}$ denotes the maximum INR increase. f_1 and f_2 represent the concentrations of the clotting factors relative to their baseline concentration prior to the treatment. The clotting factors in the model are an effective description of a large number of physiological coagulation factors that are not explicitly modelled (Wajima et al. 2009).

The dynamics of the effective coagulation factors are modelled using a system of differential equations

$$\begin{aligned}
\frac{\mathrm{d}f_{i,1}}{\mathrm{d}t} &= k_i\left(1 - \frac{\kappa c^\gamma}{c^\gamma + c_{50}^\gamma}\right) - k_i f_{i,1}, & f_{i,1}(t=0,\psi) &= 1,\\
\frac{\mathrm{d}f_{i,2}}{\mathrm{d}t} &= k_i f_{i,1} - k_i f_{i,2}, & f_{i,2}(t=0,\psi) &= 1,\\
\frac{\mathrm{d}f_i}{\mathrm{d}t} &= k_i f_{i,2} - k_i f_i, & f_i(t=0,\psi) &= 1,
\end{aligned} \tag{2.6}$$

where $i \in \{1, 2\}$. The states $f_{i,1}$ and $f_{i,2}$ denote the relative concentrations of clotting factor i in two sequential compartments, implementing a delay of the INR response to the warfarin exposure (see left side of the PD model in Figure 2.4). Only the clotting factor in the final compartment, f_i, affects the INR level (see right side of the PD model in Figure 2.4). The delay is parametrised by the transition rates, k_1 and k_2, between the compartments.

The first term in the expression for $f_{i,1}$ implements the interaction between the drug and the coagulation pathways (see Equation 2.6 and Figure 2.4). c denotes the concentration of warfarin in the central compartment, given by the ratio of its amount, a_c, and its effective volume of distribution, v: $c = a_c/v$. κ denotes warfarin's maximal inhibitory effect and c_{50} denotes its half maximal effect concentration (EC50). γ parametrises the steepness of warfarin's efficacy curve.

Note that this term also implements the typical nonlinear dose response behaviour discussed in subsection 2.1.1, introducing an onset of the inhibitory effect of warfarin for concentrations smaller than c_{50} and a plateau for concentrations much larger than c_{50}. Interestingly, this nonlinear dose response translates into a nonlinearity only for the dynamics of the PKPD model as a whole – the PK model component and the PD model component separately remain linear, and a closed-form solution for their systems of differential equations can still be derived analytically. However, in general, solutions to PKPD models can only be obtained using numerical integration methods (Seinfeld et al. 1970). In the remainder of this thesis, we will use the open source Python package `Myokit` to solve PKPD models numerically (Clerx, Collins, et al. 2016).

This concludes the mechanistic model section. To summarise: 1. PKPD models provide semi-mechanistic descriptions of treatments and include a PK model component and a PD model. The PK model is responsible for capturing the ADME processes of a drug in the body – sometimes referred to as 'what the body does to the drug', while the PD model is responsible for capturing the therapeutic and/or toxic effects of a drug – sometimes referred to as 'what the drug does to the body' (Glassman et al. 2019); and 2. PKPD models have parameters which can be pharmacologically or physiologically interpreted and which influence the treatment response simulations. The PKPD model for the warfarin treatment response has, for example, 8 parameters, $\psi = (k_a, k_e, v, k_1, k_2, \kappa, \gamma, c_{50})$, parametrising the absorption and elimination dynamics, the distribution, as well as the delay and the strength of the treatment response.

2.2.2 The dosing regimen

Having introduced PKPD modelling, we can now turn to their applications. An important application of PKPD modelling is the simulation of patients' treatment responses to different dosing regimens. For example, the warfarin model defined in Equations 2.4, 2.5, and 2.6 can be used to simulate the INR response to warfarin treatment. Central for this simulation is the dosing regimen implementation in terms of the time-dependent dose rate, $r(t)$.

The simplest ansatz to implement dosing regimens in PKPD models is to assume that dose rates are constant during dose administrations and otherwise zero. Then, the dose rate profile takes the form of a step function, switching between dose rates equal to zero outside dose administrations and dose rates equal to d/Δ during dose administrations, where d denotes the dose and Δ denotes the duration of the administration.

To illustrate this modelling approach, we show 4 dosing regimens in the left panel in Figure 2.5. The dosing regimens administer the same dose ($d = 5\,\text{mg}$) periodically every $24\,\text{h}$ at different different dose durations (blue: $\Delta = 0.001\,\text{h}$; grey: $\Delta = 6\,\text{h}$; red: $\Delta = 12\,\text{h}$; and black: $\Delta = 24\,\text{h}$). The right panel shows the corresponding drug concentration profiles in the central compartment simulated with the 1-compartment PK model defined in Equation 2.4 for $k_a = 2\,\text{h}^{-1}$, $k_e = 0.1\,\text{h}^{-1}$, and $v = 10\,\text{L}$.

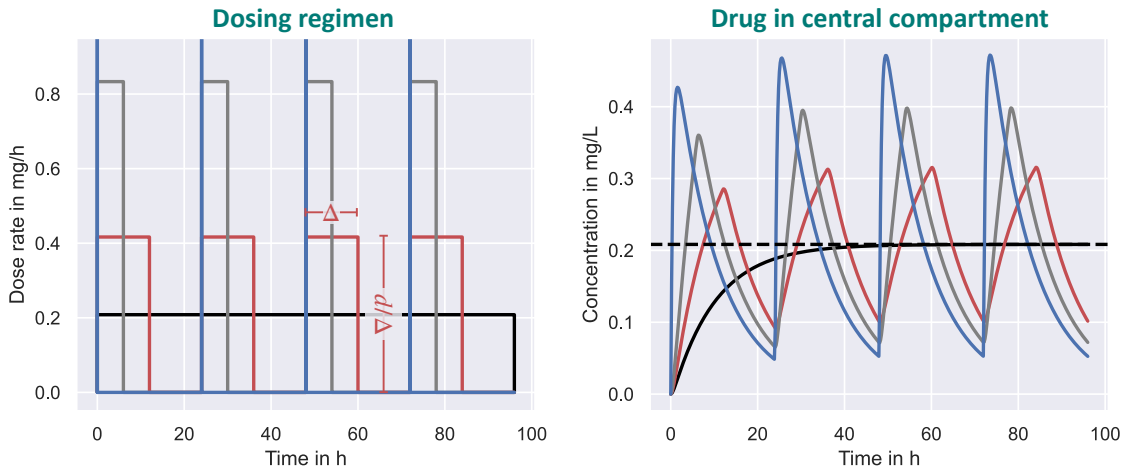


Figure 2.5: Dosing regimens. The left panel of the figure shows the dose rates of 4 different dosing regimens. All dosing regimens administer the same dose ($d = 5\,\text{mg}$) periodically every $24\,\text{h}$ at different different dose durations (blue: $\Delta = 0.001\,\text{h}$; grey: $\Delta = 6\,\text{h}$; red: $\Delta = 12\,\text{h}$; and black: $\Delta = 24\,\text{h}$). The corresponding drug concentration levels simulated using a 1-compartment PK model with indirect dose administration is show in the right panel. The dashed line illustrates the steady state drug concentration level at a constant dose rate of $r = 5\,\text{mg}/24\,\text{h}$.

Formally, the dosing regimen model can be expressed in terms of a sequence of step functions

$$r(t) = \sum_{k=1}^{K} \frac{d_k}{\Delta_k} \Theta(t - t_k)\,\Theta(t_k + \Delta_k - t), \tag{2.7}$$

where K denotes the total number of dose administrations making up the dosing regimen. d_k denotes the kth dose administered at time t_k. The duration of the administration is denoted by Δ_k. $\Theta(x)$ represents the Heaviside step function, which is equal to 0 for $x \leq 0$ and is equal to 1 for $x > 0$. As a result, the dose rate, $r(t)$, assumes the constant value d_k/Δ_k during the kth dose administration. For periods outside dose administrations, the dose rate is zero.

Equation 2.7 can represent a large class of dosing regimens, ranging from bolus administrations and infusions to single, periodic or irregular administrations. The route of administration is implemented by adding the dose rate to the rate of change of the drug amount in the corresponding compartment of the PK model. Typical routes of administration include intravascular adminisration (central compartment), as well as oral administration and subcutaneous administration (dose compartment), just to name a few

(Nebendahl 2000). For example, the warfarin model accounts for the oral administration of warfarin tablets by including the dose rate in the expression for the drug amount in the dose compartment (see Equation 2.4). A limitation of the dosing regimen model in Equation 2.7 is the unphysical, discrete jumping between dose rates, which in many instances may be more realistically modelled using smoother changes between dose rate levels. However, in this thesis we will use Equation 2.7 to implement dosing regimens, as this is the most widely used approach in the PKPD modelling community.

Besides illustrating the implementation of dosing regimens, Figure 2.5 also provides a first indication of the role of dosing regimens in the dynamic nature of treatment responses. In particular, it shows that the degrees of freedom of the dosing regimen – i.e. the sequence of dose amounts, the timings of dose administrations, and the durations of administrations – have real consequences for the treatment response. For example, the figure shows that the drug concentration oscillates for non-continuous dose administrations and that the amplitude of these oscillations depends on the duration of the administrations. The shorter the dose duration, the larger the amplitude. The bolus administrations with almost instantaneous doses ($\Delta = 0.001\,\text{h}$) result in a concentration time-course which shows the largest peaks and the lowest troughs (see blue line), while the constant infusion ($\Delta = 24\,\text{h}$) shows no oscillations (see black line). These oscillations are a result of the rapid absorption of the drug from the dose compartment into the central compartment and its continuous elimination. For a constant drug administration, the drug concentration reaches a steady state, which for the warfarin PK model (Equation 2.4) equals $c_{ss} = d/\Delta/v/k_e \approx 0.21\,\text{mg/L}$ (see dashed line in Figure 2.5). For non-continuous administrations, the drug concentration oscillates around this value.

Figure 2.5 also illustrates that drug concentration levels show a delayed response to dose administrations. This delay is the consequence of the drug-specific pharmacokinetics and the dosing regimen. For example, the constant infusion takes the longest time to reach a blood concentration value of 0.2 mg (see black line), while the drug concentration following a bolus dose surpasses this value within minutes of its administration (see blue line).

2.2.3 From mechanistic models to dosing regimens

In the previous two sections, we introduced PKPD models as (semi-)mechanistic or empirical descriptions of treatment response dynamics which formalise biomarkers or clinical readouts as functions of the time and the administered dosing regimen

$$\bar{y}(t, r, \psi), \tag{2.8}$$

where we use $\bar{y}$ to denote the quantity of interest, t to denote the time, r to denote the dosing regimen, and ψ to denote the parameters of the model. This formalised relationship can be used to simulate treatment responses. In this section, we will discuss how we can use PKPD models in the context of MIPD and derive dosing regimens that target desired treatment responses.

To begin the discussion, let us have another look at the simulations of the warfarin PKPD model defined in Equations 2.4, 2.5, and 2.6 for given values of ψ, this time focussing on its PD component (see Figure 2.6). The model describes the time course of both, the warfarin concentration in the blood and the clinical read-out of the treatment response, the prothrombin time. The prothrombin time quantifies the time it takes for blood to clot and is measured in terms of the international normalised ratio (INR).

The figure shows the model simulations for two different dosing regimens of warfarin: 1. daily (bolus) administrations of 5 mg (see blue lines); and 2. daily (bolus) administrations of 7.5 mg (see red lines). The time points of the dose events are illustrated in the left panel (see black arrows) together with the simulated warfarin concentrations. The right panel shows the corresponding INR level simulations. The boundaries of the therapeutic range are represented using dashed lines, marking INRs below 2 as inefficacious and INRs above 3 as dangerous. Overly low INRs do not prolong the clotting time sufficiently to prevent or treat adverse events from blood clotting, while INRs above 3 prolong the blood clot formation too much, risking extreme bleeding events from the smallest injuries (FDA 2011). The target of MIPD is therefore to find dosing regimens that result in INRs between 2 and 3.

The ability of PKPD models to simulate treatment responses for different dosing regimens makes it possible to systematically explore the space of dosing regimens and

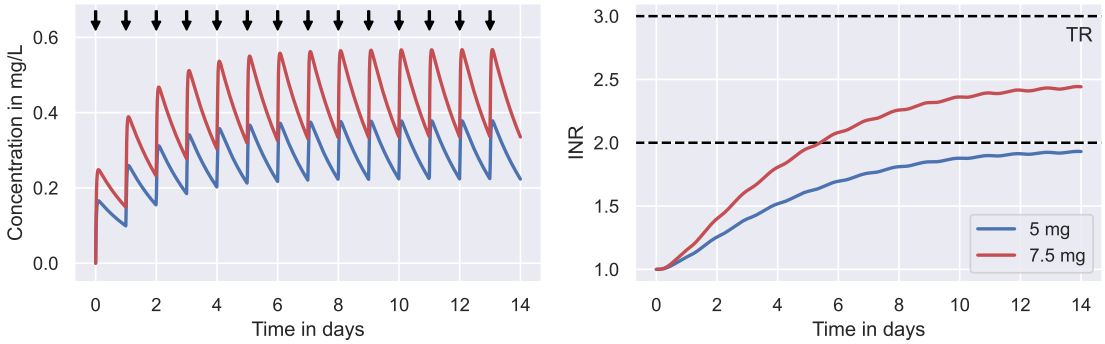


Figure 2.6: Treatment response simulation. The figure shows the simulated treatment response of the warfarin PKPD model for two dosing regimens: daily (bolus) administrations of 5 mg (blue curves); and daily (bolus) administrations of 7.5 mg (red curves). The simulated warfarin concentrations are illustrated in the left panel, and the associated INR values are shown in the right panel. The dose administrations are indicated by black arrows in the left panel. The therapeutic range (TR) is indicated by dashed black lines in the right panel.

to find the one dosing regimen that optimally targets the desired treatment response (Sheiner 1969). In the simplest case, this desired treatment response can be expressed in terms of a single target value for the quantity of interest

$$\bar{y}(t, r^*, \psi) = y^*. \tag{2.9}$$

We use y^* to denote the target value and r^* to denote the *optimal* dosing regimen. However, more complicated targets for MIPD are also possible. For example, weighted ranges or distributions of target values may be used to derive optimal dosing regimens for treatments with more complex requirements (Maier, Hartung, Kloft, et al. 2021). The appropriate approach for defining target treatment responses is specific to each application.

In practice, constraints on the dose amount, the route of administration, the frequency of administrations, and the delay of the treatment response dynamics make it difficult (or impossible) to realise the desired treatment response exactly. It is therefore customary to relax the optimisation target to treatment responses that only approximately equal the desired treatment response. Arguably, the simplest approach for this is based around the mean squared error (MSE) between the model simulation and the target response

$$\mathrm{MSE}(r) = \frac{1}{n} \sum_{j=1}^{n} \Big(y^* - \bar{y}(t_j, r, \psi) \Big)^2, \tag{2.10}$$

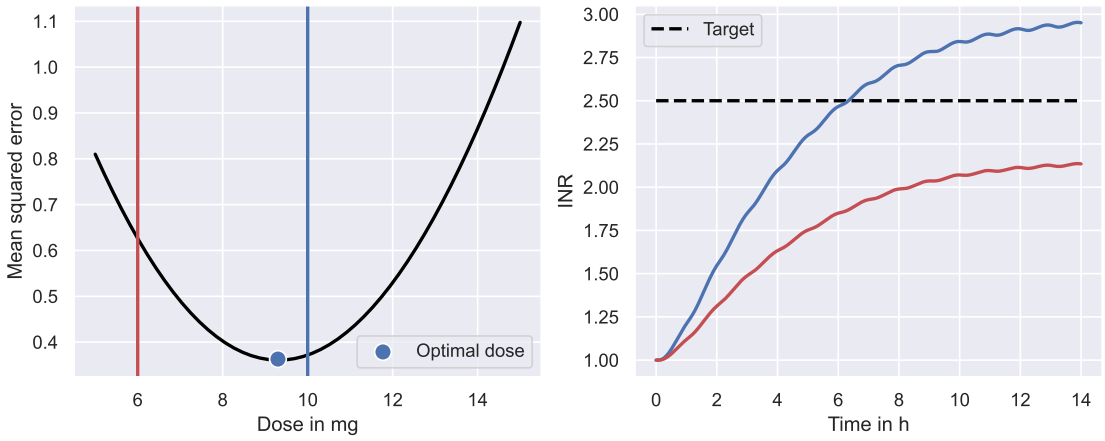


Figure 2.7: 1D dosing regimen optimisation. The figure illustrates the objective of a 1-dimensional dosing regimen optimisation where the dosing regimen is constrained to administering the same daily bolus for the duration of the treatment. The example optimises the treatment response simulated with the warfarin PKPD model. The left panel shows the MSE evaluated for different dose amounts (black), highlighting two suboptimal dose amounts (blue and red). The optimal dose is illustrated using a blue marker. The right panel shows the INR responses corresponding to the suboptimal dose amounts. The target INR level is highlighted using a dashed line.

where t_j denotes the jth time point at which the model simulation is compared to the target value. n represents the total number of evaluation time points. This defines the optimal dosing regimen as the dosing regimen that minimises the MSE between the simulation and the target value.

To illustrate this concept of an optimal dosing regimen, we show in Figure 2.7 a simplified optimisation problem, where the dosing regimen is constrained to administering the same bolus dose each day of the treatment period. We use $n = 5000$ equidistant time points from $t_0 = 0\,\text{d}$ to $t_{5000} = 14\,\text{d}$ to evaluate Equation 2.10 for each dose amount.

The left panel of the figure shows the MSE for dose amounts from 5 mg to 15 mg (black). The optimal dose is highlighted using a blue marker. Two suboptimal dose amounts and their associated treatment response simulations are illustrated in blue and red. The figure demonstrates that the MSE provides a simple objective for the dosing regimen optimisation: follow the gradient of the MSE to find its minimum; this minimum will represent the optimal dosing regimen.

For convex, 1-dimensional optimisation problems like the one illustrated in Figure 2.7,

it is possible to exhaustively explore the values of an objective function and to find its minimum manually. Here, by objective function we mean the function that defines the target of the optimisation, such as the MSE in Equation 2.10. However, for higher dimensional, non-convex optimisation problems, such as those arising from dosing regimens with varying dose amounts over time, it becomes exceedingly difficult to find the minimum of an objective function manually, and numerical methods are needed to support the optimisation.

Many optimisation algorithms for this purpose exist. Perhaps, one of the most widely known optimisers is the Gradient Descent algorithm, which iteratively steps through the parameter space with a fixed step size, following the gradient until a local minimum is reached (Boyd et al. 2004). Other gradient-based optimisers include Newtonian methods, such as LBFGS (C. Zhu et al. 1997), or batch optimisers, such as stochastic gradient descent (SGD) (Amari 1993) or ADAM (Kingma et al. 2014). Besides gradient-based optimisers, there are also gradient-free algorithms which avoid the computational costs of evaluating the gradient and, instead, sample multiple points to approximate gradient information. Examples of gradient-free optimisers include the CMAES algorithm (Hansen et al. 2003) and the particle swarm optimiser (PSO) (Kennedy et al. 1995).

Whether the gradient is evaluated or not, each optimiser comes with different strengths and limitations for different problems (Clerx, Augustin, et al. 2022). For example, constraining dose amounts to fixed multiples of commercially available tablets will discretise the parameter space, making gradient-free algorithms more suited for the dosing regimen optimisation than those that use gradients. However, gradient-free algorithms typically make use of stochastic sampling strategies to explore the parameter space which becomes extremely inefficient as the number of parameters for the optimisation grows (Fernandez-Martinez et al. 2020). It is therefore necessary to choose optimisation algorithms strategically, matching the strengths of an optimiser to the optimisation problem. In this thesis, we choose optimisers from an open-source Python library, `PINTS` (Clerx, Robinson, et al. 2019).

To conclude this section, note that besides illustrating the objective of optimisation algorithms, Figure 2.7 also demonstrates an important limitation of the 1-dimensional

optimisation scheme for the dosing regimen optimisation: the optimal dose does not actually lead to a treatment response close to the target INR (see e.g. the blue treatment response with an almost optimal MSE in Figure 2.7). The treatment response differs substantially from the target INR for the first and last third of the simulated period, and indicates a steady state INR value of around 3 – a value barely inside the therapeutic range. This unintended optimisation result is the consequence of a number of contributing factors, including the treatment response delay, the limited simulation period, the constraints of the dosing regimen, and the averaging of the objective function over all time points.

One approach to improve the optimisation result is to relax the constraints of the previous optimisation and allow varying dose amounts between days. The corresponding optimal dosing regimen is illustrated in Figure 2.8. We minimise the MSE numerically using PINTS' implementation of the CMAES algorithm (Hansen et al. 2003). During the optimisation, possible dosing regimens are constrained to clinically feasible dose amounts – i.e. those dose amounts that can be administered as combinations of commercially available warfarin tablets. This limits allowed dose amounts to warfarin doses of 0 mg, 1 mg and any dose $\geq$ 2 mg in 0.5 mg steps. We also constrain the maximum daily dose to be at most 30 mg.

The figure shows that the optimal dosing regimen begins the treatment with the maximally allowed daily dose of 30 mg, followed by the second largest dose of the regimen, and a drug holiday. In the remainder of the simulated period, the optimal dosing regimen schedules almost constant daily dose amounts. The simulated treatment response approaches the target INR substantially faster than the treatment response from the previous optimisation (see Figure 2.7), compensating for the treatment response delay with an initially elevated dose level which is reduced to zero before reaching the maintenance dose amount.

Beyond increasing the degrees of freedom of the dosing regimen, it is also possible to improve the optimisation result using more advanced objective functions. For example, the proximity of simulated treatment responses can be scored using distributions of target values (Maier, Hartung, Kloft, et al. 2021). In addition, discount factors can

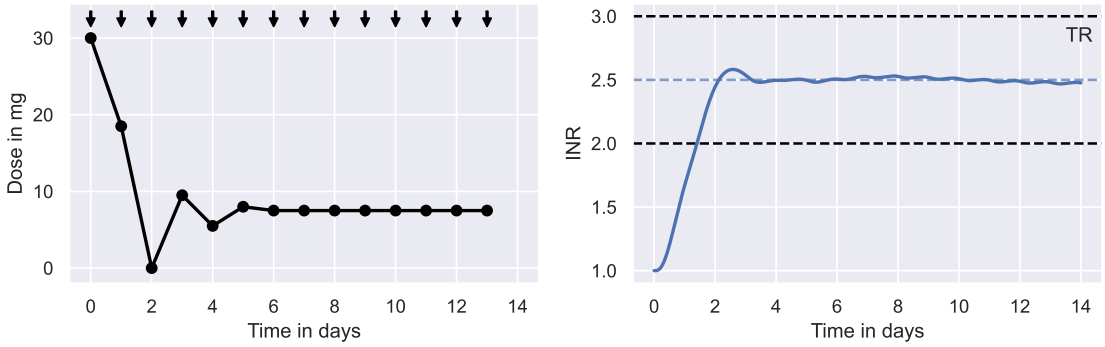


Figure 2.8: Higher dimensional dosing regimen optimisation. The figure shows the optimal dosing regimen for a desired INR level of 2.5 (left panel) and the associated INR response (right panel) simulated using the warfarin PKPD model. The dose administrations are indicated using black arrows in the left panel. In the right panel the therapeutic range (TR) (black) and the target value (blue) are indicated using dashed lines.

be used to put more emphasis on either the short-term or the long-term treatment responses of the simulation.

This concludes the basic introduction to the use of PKPD models for precision dosing. In the next section, we will discuss how model simulations can be related to real patients, starting with the simplest approach – maximum likelihood estimation – and closing with the current best practice – Bayesian inference.

2.2.4 Estimating model parameters from data

In the previous sections, we discussed the simulation and optimisation of treatment responses using PKPD modelling. In this section, we take a step back and address a more fundamental question for PKPD modelling in the context of MIPD: given that we optimise dosing regimens for *simulated* treatment responses, how do we know that those dosing regimens also translate into desired treatment outcomes for patients in clinical practice? I.e., how do we know that the simulations of a PKPD model actually describe the treatment response of a patient?

To approach this question, recall that in subsection 2.2.2 we already demonstrated that changing an input to the simulation – the dosing regimen – can substantially alter the simulated treatment response dynamics. Periodic bolus administrations led, for example, to oscillating drug concentration levels with more extreme peaks and troughs

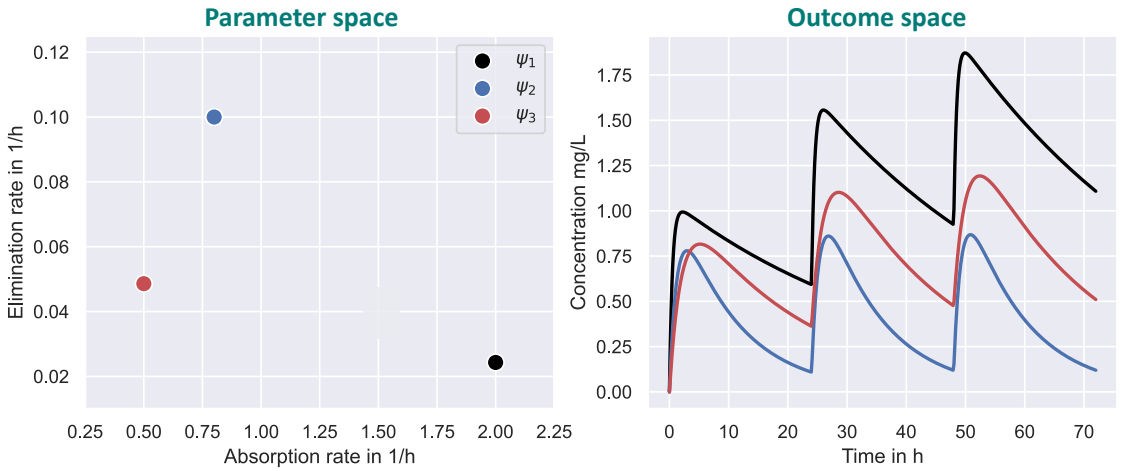


Figure 2.9: Influence of parameter values on simulated treatment responses. The left panel shows the absorption rate and the elimination rate of three sets of parameter values for the warfarin PK model (Equation 2.4). The volume of distribution is fixed for all parameter sets at $v = 14.3\,\mathrm{L}$. The corresponding warfarin concentration simulations in the central compartment following daily bolus administrations of $d = 15\,\mathrm{mg}$ corresponding are shown in the right panel.

than periodic infusions of the same drug dose. Other inputs to the simulation are the values of the model parameters, ψ (see Equation 5.8). To illustrate the importance of the parameter values for the outcome of the simulated treatment response dynamics, we show three treatment responses for different parameter values simulated with the warfarin PK model (Equation 2.4) in Figure 2.9. Recall that the model has three parameters: the absorption rate, k_a; the elimination rate, k_e; and the volume of distribution, v.

The left panel of the figure shows the values of the elimination rate and the absorption rate across the sets of model parameters. The volume of distribution is fixed at $v = 14.3\,\mathrm{L}$. The right panel shows the corresponding treatment response simulations for daily warfarin bolus doses of $d = 15\,\mathrm{mg}$. The figure illustrates the substantial influence of the parameter values on the simulated treatment response dynamics. For example, the simulation with the largest elimination rate results in the overall lowest drug concentration levels (see blue curve), manifesting the elevated clearance of the drug from the system.

With such substantial differences in simulated treatment responses for different choices of model parameters, it is apparent that the justification of model-informed

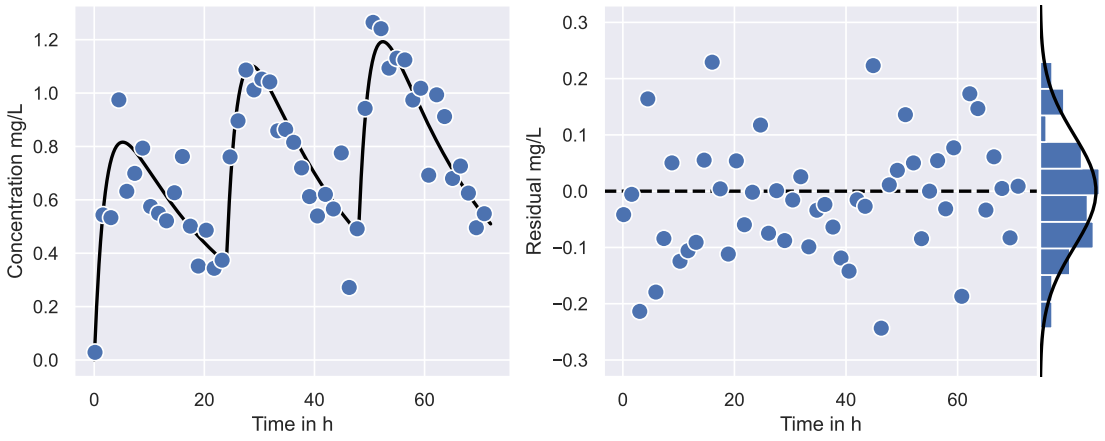


Figure 2.10: The measurement model. The left panel shows simulated measurements of the warfarin blood concentration (blue), together with the underlying, 'true' treatment response dynamics (black). The right panel shows the differences between the measurements and the underlying treatment response over time. The line of identity, where measurements equal the measured process, is indicated using a dashed line. The histogram over all residuals is shown in the right axis of the panel. The distribution from which the residuals are randomly sampled is illustrated in black.

dosing regimens crucially relies on the use of suitable parameter values, describing the treatment response dynamics of real patients. Parameter values with this property are typically unknown, provided they exist in the first place, and have to be estimated from measurements. The role of statistical inference is to perform this estimation (Cox 2006; Gelman, Carlin, et al. 1995).

For now, we will focus on estimating the parameters of models, assuming that there is no uncertainty about the best model structure to use for MIPD. We will come back to estimating uncertainty of model structures in subsection 2.2.9.

To motivate the framework of statistical inference for the estimation of model parameters from data, we again use the warfarin example and illustrate simulated warfarin blood concentration measurements in the left panel of Figure 2.10 (see scatter points). We indicate the underlying treatment response dynamics using a black line. The details of the measurement simulation are discussed below. In the figure, the measurements do not capture the treatment response exactly, but appear to idiosyncratically deviate from the true concentration levels. These seemingly unexplainable deviations from the

underlying process are assumed to be truly random for the purpose of statistical inference (see below).

In the context of PKPD modelling, this assumption of random measurement errors can be justified by the complexity of the measurement process, which, for example, for blood concentration measurements can include sequential sampling, anticoagulation, centrifugation, and filtration of the blood (Musteata 2011). Each step has the potential to corrupt the measurement result and introduce an error to the measured value. The measurement process of other biomarkers or clinical readouts can be similarly complex, see for example the details of the INR measurement process in Chapter 3.

Assuming that the individual processes that corrupt measurements are independent and do not bias the measured values systematically, the resulting measurement error – also referred to as measurement noise – can be modelled using an independent and identically distributed (iid) random variable, ϵ,

$$y(t, r, \psi) = \bar{y}(t, r, \psi) + \epsilon, \tag{2.11}$$

where y denotes possible measurements of the true quantity of interest. This measurement model represents treatment response measurements as a composite of two contributions: 1. the simulated treatment response; and 2. the realisation of the random measurement noise. For example, the measurements illustrated in the left panel of Figure 2.10 are composed of the warfarin PK model simulations (Equation 2.4) and the realisations of the random measurement error, here sampled from a normal distribution with mean $\mu = 0$ and standard deviation $\sigma = 0.1\,\text{mg/L}$. Other common noise models include heteroscedastic normal errors, where the variance scales with the treatment response, or lognormally distributed errors, which additionally guarantee non-negative measurement values (Lavielle 2014).

The right panel of Figure 2.10 highlights the composite nature of the warfarin measurement model, plotting the residuals between the measured values and the treatment response simulations, $y_j - \bar{y}(t_j, r, \psi)$, over time. These residuals are the realisations of the random variable, ϵ. Here, we use y_j to denote the measured value at time t_j. The distribution of the random variable, ϵ, from which the residuals are sampled

is illustrated on the right, vertical axis of the panel (see black curve). Aggregating the residuals across measurement time points recovers an approximate representation of this distribution (see blue histogram), which converges to the exact distribution as the number of measurements increases.

Note that the measurement model (Equation 2.11) defines a distribution of possible measurement values for each time point, t,

$$p(y|t, r, \psi), \tag{2.12}$$

making the simulation of treatment response measurements for any measurement times possible. Here, $p(y|x)$ denotes the probability density of a random variable, y, conditional on the value of x. For example, the measurements illustrated in Figure 2.10 are simulated by randomly sampling values from the distribution $p(y|t, r, \psi) = \mathcal{N}(y|\bar{y}(t, r, \psi), \sigma^2)$ – one measurement for each measurement time.[2] The dosing regimen was fixed to daily bolus doses of $d = 15\,\text{mg}$, and the model parameters were set to $k_a = 0.5\,\text{h}^{-1}$, $k_e = 0.0486\,\text{h}^{-1}$, $v = 14.3\,\text{L}$, and $\sigma = 0.1\,\text{mg/L}$.

Measurements can also be simulated for different dosing regimens, r, or different model parameters, ψ. Note that for ease of notation, we extend the definition of ψ to include the model parameters of the measurement model.

The measurement model (Equation 2.12) conceptualises the relationship between measurements and simulations of the treatment response. This relationship can be used, in the context of statistical inference, to infer model parameters, ψ, describing an observed treatment response. In particular, we can use it to estimate the likelihood for a given treatment response measurement, $\{(y_1, t_1), r\}$, under the measurement model by evaluating the model's probability density at the measurement, $p(y_1|t_1, r, \psi)$. The likelihood of a dataset with n measurements, $\mathcal{D} = \{(y_1, t_1), \ldots, (y_n, t_n), r\}$, can be similarly quantified by the product of the individual likelihoods of the measurements

$$p(\mathcal{D}|\psi) = \prod_{j=1}^{n} p(y_j|t_j, r, \psi). \tag{2.14}$$

[2] The probability density of a normal distribution is defined as

$$\mathcal{N}(y|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-(y-\mu)^2/2\sigma^2}, \tag{2.13}$$

where μ denotes its mean and σ denotes its standard deviation.

Note that this construction of the likelihood relies centrally on the assumption that the measurement noise is iid – i.e. the likelihood of observing a particular measurement is independent of all other measurements. For correlated measurement errors, the joint probability density does not factorise in the same way[3] and the evaluation of the likelihood requires more care. A natural mechanism leading to non-iid residuals is, for example, a misspecification of the treatment response model relative to the measured treatment response dynamics, which can cause a systematic under- or overestimation of the treatment response for periods of times. This introduces autocorrelations between sequential residuals, violating the iid assumption (Creswell et al. 2020). However, modelling non-iid residuals can substantially increase the computational costs of the likelihood evaluation. We will therefore adopt common community practice and assume iid measurement noise.

The likelihood in Equation 2.14 quantifies the likelihood of the observed treatment response, $\mathcal{D}$, under the measurement model with parameters, ψ. Choosing the parameters such that the likelihood is maximised provides an intuitive procedure to identify model parameters that best describe the observed treatment response. To illustrate this inference approach, we visualise the likelihood of a simulated dataset under the warfarin model for two sets of model parameters in Figure 2.11.

[3] To see that Equation 2.14 only holds for iid measurement errors, note that we use $p(\mathcal{D}|\psi)$ as a shorthand notation for the joint likelihood of the data under the measurement model

$$p(\mathcal{D}|\psi) = p(y_1, \dots, y_n | t_1, \dots, t_n, r, \psi). \tag{2.15}$$

Using the arithmetics of conditional probabilities – e.g. $p(y, x|z) = p(y|x, z)\, p(x|z)$ for three random variables, y, x and z, – we can decompose $p(\mathcal{D}|\psi)$ into a product of two conditional probabilities

$$p(\mathcal{D}|\psi) = p(y_1, \dots, y_{n-1} | y_n, t_1, \dots, t_n, r, \psi)\, p(y_n | t_1, \dots, t_n, r, \psi). \tag{2.16}$$

The conditioning on the time points $t_1, \dots, t_{n-1}$ in the second term has no effect as the measurement y_n dose not depend on times other than t_n under the model. However, the conditioning on y_n in the first term can only be dropped, if the measurements are independently distributed, in which case the expression reduces to

$$p(\mathcal{D}|\psi) = p(y_1, \dots, y_{n-1} | t_1, \dots, t_n, r, \psi)\, p(y_n | t_n, r, \psi). \tag{2.17}$$

This defines a recursive procedure that can be iteratively applied to the first term to factorise the joint likelihood into a product of n terms, one for each measurement. If the measurement errors are identically distributed, we obtain Equation 2.14, otherwise each measurement is captured by a different distribution.

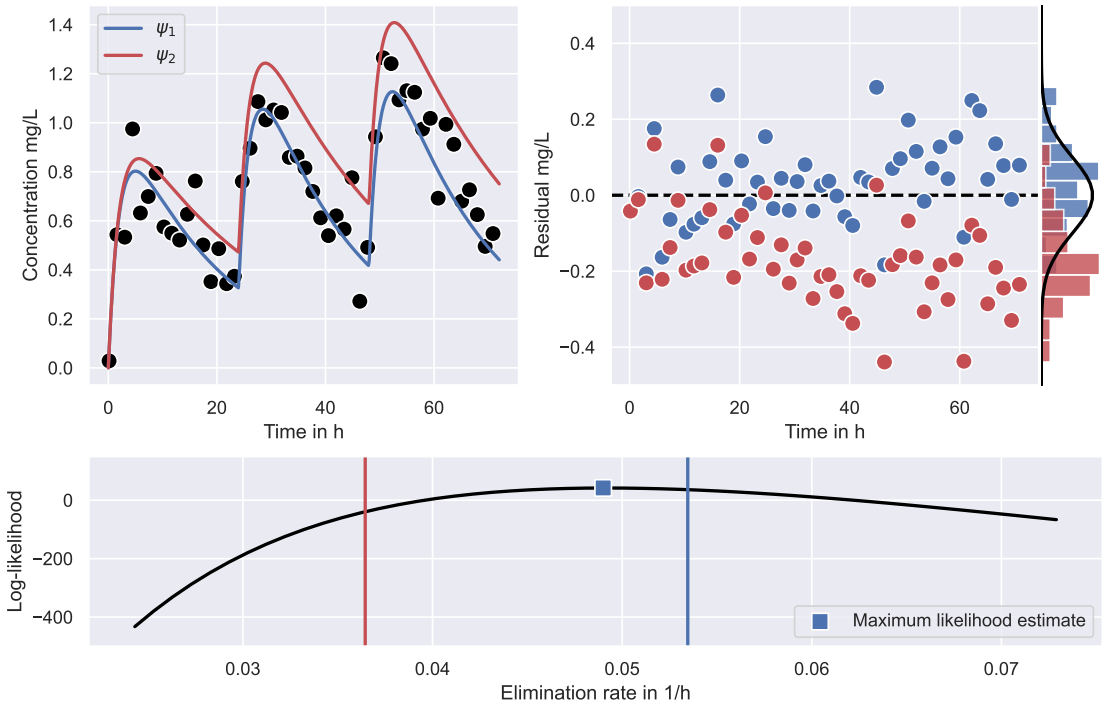


Figure 2.11: Maximum likelihood estimation. The top-left panel shows measurements of the warfarin blood concentration (black) following daily bolus doses of 15 mg. Simulations of the warfarin PK model (Equation 2.4) for two sets of parameters are illustrated in blue and red. The top-right panel shows the differences between the measurements and the model simulations (see scatter points). The histograms over the residuals (blue and red) as well as the measurement noise model (black) are shown on the right axis. The bottom panel shows the log-likelihood values of the data for different elimination rates. Other model parameters are fixed. The maximum likelihood estimate is highlighted using a blue marker. The log-likelihood values for the two parameter sets are illustrated using blue and red lines.

The top-left panel shows the treatment response measurements (black) overlaid with the simulations of the PKPD model for the parameter sets (blue and red). The residuals (blue and red) and the measurement noise model (black) are shown in the top-right panel. For simplicity, the noise parameter, σ, is fixed to the same value for both parameter sets, leading to an identical noise model. Also the absorption rate, k_a, and the volume of distribution, v, are fixed to the same values across the parameter sets. This leaves one parameter – the elimination rate, k_e, – to optimise (see bottom panel). Note that for numerical reasons, we show the logarithm of the likelihood – also referred to as the log-likelihood,

$$\log p(\mathcal{D}|\psi) = \sum_{j=1}^{n} \log p(y_j|t_j, r, \psi). \tag{2.18}$$

This avoids floating point imprecision resulting from the multiplication of n values between 0 and 1 when the product of the probability densities is evaluated. Crucially, the logarithm is a strictly monotonic function and therefore does not shift the maximum of the (log-)likelihood. The maximum likelihood estimate (MLE) of the elimination rate is illustrated using a blue marker (see bottom panel). The log-likelihood values for the two parameter sets are shown using blue and red lines.

The log-likelihood in Figure 2.11 shows that the treatment response measurements have different likelihoods under the measurement model for the two parameter sets (see blue and red lines in the bottom panel). Neither parameter set maximises the likelihood. We can find the maximum of the likelihood numerically, analogously to the dosing regimen optimisation discussed in subsection 2.2.3[4], recovering the data-generating parameter value (Data-generating: $k_e = 0.0486\,\mathrm{h}^{-1}$; MLE: $k_e = 0.0490\,\mathrm{h}^{-1}$).

This concludes the basic introduction to statistical inference. The key take-away from this section is that treatment response measurements are modelled in this thesis using a composite approach with contributions from a deterministic component – the PKPD model – and from a stochastic component – the measurement noise model. This enables the construction of a likelihood of the data under the measurement model, making the estimation of model parameters possible. A central assumption for this is that measurements display random errors relative to the measured treatment response.

2.2.5 Parametric uncertainty & Bayesian inference

Maximum likelihood estimation is a first step towards describing treatment responses of patients with simulations, enabling the prediction of clinically-relevant dosing regimens

[4]Note that, for a normal measurement model, maximum likelihood estimation has strong parallels to the dosing regimen optimisation discussed in subsection 2.2.3: Plugging in the normal probability density (Equation 2.50) in the expression for the log-likelihood (Equation 2.18) yields

$$\log p(\mathcal{D}|\psi) = -n \log \sqrt{2\pi\sigma^2} - \sum_{j=1}^{n} \Big(y_j - \bar{y}(t_j, r, \psi)\Big)^2 / 2\sigma^2, \tag{2.19}$$

which for a fixed noise parameter, σ, reduces to a MSE optimisation problem. MSE optimisation was also the approach we took for the dosing regimen optimisation. The main difference between the optimisation problems is that for parameter estimation the objective is to find simulations close to the measured treatment responses, $\{y_j\}$, while for the dosing regimen optimisation simulations are optimised to approximate the desired treatment response, y^*.

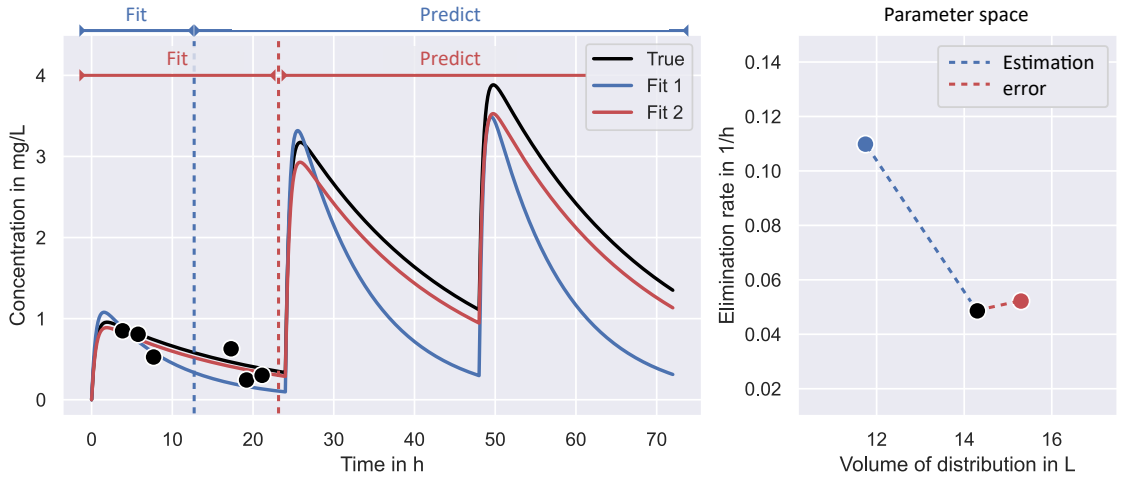


Figure 2.12: Measurement noise induces errors in parameter estimates. The left panel shows measurements of a treatment response (scatter points) simulated with the warfarin measurement model. The measured treatment response is illustrated using a black line. Two maximum likelihood fits of the warfarin model are shown in blue (fitted to the first three measurements) and red (fitted to all measurements). The fits are simulated beyond the measurement time points to illustrate the model predictions of the treatment response. The MLEs corresponding to the fits are shown in the right panel. The data-generating model parameters used for the simulation of the true treatment response are shown in black.

(see subsection 2.2.3). However, the clinical feasibility of maximum likelihood estimation hinges on the availability of sufficient data for the parameter inference. This is in conflict with typical clinical settings, where the ability to collect patient data is limited. In this section, we will show that having available only a limited number of measurements introduces errors to the values of parameter estimates – errors with consequences for predicted treatment responses. We will also show how these errors can be mitigated by estimating parametric uncertainty using Bayesian inference – an extended inference approach building on maximum likelihood estimation.

To illustrate the interplay between volume of data and accuracy of parameter estimation, we fit the warfarin model from subsection 2.2.4 repeatedly to simulated measurements (see Figure 2.12). The number and frequency of the measurements are reduced to match clinically feasible measurement burdens, measuring the drug blood concentration only on day 1 of the treatment. The measurements are illustrated using scatter points in the left panel of Figure 2.12. The measured, and in practice unknown, treatment response is illustrated using a black line.

We fit the warfarin model twice to the measurements using MLE. The first fit estimates the parameters using only the first three blood concentration measurements (4 h, 6 h, and 8 h after the first dose administration). The second fit estimates the parameters from all available measurements, including also the measured values obtained 17 h, 19 h, and 21 h after the first dose. The simulated treatment responses are illustrated in the left panel, and the estimated model parameters are illustrated in the right panel (see blue/red lines and scatter points). The data-generating model parameters corresponding to the measured treatment response are illustrated in black (see right panel). For illustrative purposes, only the volume of distribution, v, and the elimination rate, k_e, are estimated from the data. The absorption rate, k_a, and the noise parameter, σ, are fixed to the data-generating parameter values.

The figure shows that it is possible to estimate model parameters from a small number of measurements using maximum likelihood estimation. However, the estimates deviate from the data-generating parameter values (see right panel). These estimation errors translate into considerable predictive errors relative to the measured (true) treatment response, especially for the blue estimate derived from fewer measurements (see left panel).

The differences between the MLEs and the data-generating parameters in Figure 2.12 are the manifestation of an important, generic limitation of statistical inference: statistical inference cannot recover the data-generating parameters of a stochastic model from a finite number of measurements (Gelman, Carlin, et al. 1995). This means that even if we had a model that perfectly described the measured treatment response of a patient for some 'data-generating' parameters, MLEs estimated from a clinically feasible number of measurements would still differ from those parameter values. In fact, one can show that MLEs converge to data-generating parameters only in the limit where the number of measurements grows to infinity, $n \to \infty$ (Cox 2006). This convergence is caveated by an assumption relating to the identifiability of model parameters (Raue et al. 2009). It also assumes that the fitted model structure is the data-generating one. We will postpone the discussion of parameter non-identifiabilities to later chapters of this thesis when they arise. Structural uncertainty of models are discussed in subsection 2.2.9.

To see that a finite number of measurements cannot recover the data-generating model parameters of a stochastic model, let us consider the estimation of model parameters from noise-free measurements. Without measurement noise, the warfarin measurement model becomes deterministic,[5] with each measurement capturing the treatment response exactly. As a result, the model simulation has to equal the measurements at the measured time points to describe the observed treatment response. Setting the simulated values to the measured values removes one degree of freedom from the model for each measurement, fixing the two unknown model parameters, v and k_e, already after two measurements. In the presence of measurement noise, measurements are less informative about the treatment response and more measurements are needed to determine the model parameters. More specifically, each measurement constrains the model simulation only to a range of treatment responses that is consistent with the measured value under the measurement noise model (see e.g. Figure 2.11). The intersection of these treatment response ranges from all measurements translates into an area of parameter values consistent with the measurements. The MLEs are the point in this area of parameter space that is most consistent with the measurements. Notably, both the area of consistent parameter values and the MLEs are sensitive to the realisation of the measurements, especially when the number of measurements is small, explaining the estimation error in Figure 2.12 as a consequence of the randomness of measurement noise.

The inevitability of estimation errors (and consequently predictive errors) due to the limited availability of clinical data makes it apparent that maximum likelihood estimation is, in most cases, not suited for MIPD in clinical practice. An extended inference approach is needed to quantify the errors of the parameter estimation. One such inference approach is the profile likelihood estimation, which estimates confidence intervals for the MLEs (Raue et al. 2009). Another approach is Bayesian inference

[5] For $\sigma \to 0$, the normal measurement model (Equation 2.50) reduces to a delta distribution

$$p(y|t,r,\psi) = \lim_{\sigma\to 0} \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\left(y-\bar{y}(t,r,\psi)\right)^2/2\sigma^2} = \delta\Big(y-\bar{y}(t,r,\psi)\Big), \tag{2.20}$$

meaning that each measurement, y, captures the PKPD model simulation, $\bar{y}(t,r,\psi)$, exactly. Since the PKPD model is deterministic, also the measurement model becomes deterministic in this limit.

which was first used in the context of MIPD in 1979 (Sheiner, Beal, et al. 1979). Now, Bayesian inference is the dominant inference method for precision dosing, e.g. (Broeker et al. 2019; Faelens et al. 2021; Gastmans et al. 2022; J. H. Hughes et al. 2024). We will, therefore, focus on Bayesian inference in this thesis.

Bayesian inference takes a conceptually different approach to parameter estimation than maximum likelihood, positioning the area of data-consistent parameters at the center of the inference. As we will discuss below, this area is framed as the level of uncertainty with which the data-generating parameter values are known. This parametric uncertainty is reduced by the information gained from treatment response measurements (Sheiner and Beal 1982).

Formally, Bayesian inference captures and updates parametric uncertainty by translating the likelihood of the data under the model (Equation 2.14) into a distribution of parameter values consistent with the treatment response measurements using Bayes' rule[6]

$$p(\psi|\mathcal{D}) = \frac{p(\mathcal{D}|\psi)\,p(\psi)}{p(\mathcal{D})}. \tag{2.22}$$

This introduces three distributions: 1. the posterior distribution, $p(\psi|\mathcal{D})$, which quantifies the parametric uncertainty after the inference; 2. the prior distribution, $p(\psi)$, which quantifies the parametric uncertainty before the inference; and 3. the evidence, $p(\mathcal{D})$, which quantifies the likelihood of the data under the model, taking the level of uncertainty prior to the inference into account. The ability to quantify prior uncertainty – i.e., prior knowledge or expectations about feasible parameter values – in terms of the prior distribution is a novel feature of Bayesian inference. We will discuss algorithms used to sample from posterior distributions in the next section, subsection 2.2.6.

[6] Bayes' rule is an application of the conditional probability arithmetics that we have already applied in Equation 2.15 to obtain the factorised likelihood expression. Here, we make use of the symmetry of joint probability distributions to obtain Bayes' rule. For two random variables, x and y, we can rewrite their joint probability in two ways: 1. by conditioning on the value of x, $p(x,y) = p(y|x)\,p(x)$; and 2. by conditioning on the value of y, $p(x,y) = p(x|y)\,p(y)$. Since both expressions equal the joint probability, they have to equal each other

$$p(y|x)\,p(x) = p(x|y)\,p(y). \tag{2.21}$$

Dividing both sides by $p(x)$, assuming $p(x) > 0$, gives Bayes' rule: $p(y|x) = p(x|y)\,p(y)/p(x)$. In Equation 2.22, $y = \psi$ and $x = \mathcal{D}$.

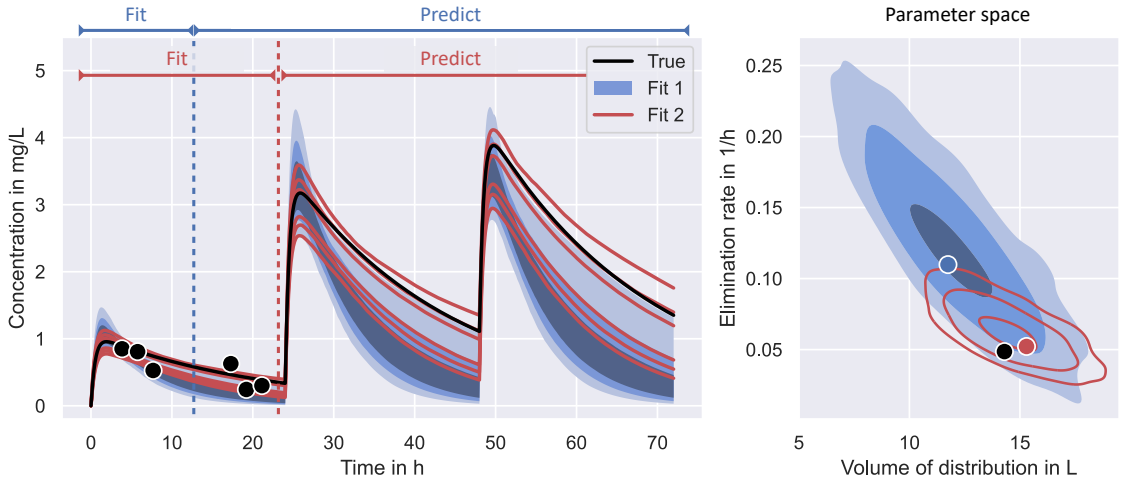


Figure 2.13: Estimation of parametric uncertainty using Bayesian inference. The left panel shows measurements of a treatment response (scatter points) simulated with the warfarin measurement model. The measured treatment response is illustrated using a black line. Two Bayesian fits of the warfarin model are shown in blue (fitted to the first three measurements) and red (fitted to all measurements). The fits are simulated beyond the measurement time points to illustrate the model predictions of the treatment response. The posterior distributions corresponding to the fits are shown in the right panel. The data-generating model parameters used for the simulation of the true treatment response are shown in black. For reference, we also illustrate the MLEs from Figure 2.11 (see blue and red scatter points).

The result of Bayesian inference is the posterior distribution (Equation 2.22) which assigns each region in parameter space with a probability of containing the data-generating parameters. For example, two posteriors inferred from the warfarin data are visualised in the right panel of Figure 2.13 using blue shaded areas and red contour lines. The shaded areas/contours illustrate the 67 %, 90 %, and 99 % bulk probability regions of the posterior distributions.

The right panel of the figure shows that both posterior distributions assign substantial probability to the neighbourhood of the data-generating parameter values. The blue posterior includes the data-generating parameters in its 99 % bulk probability region, while the red posterior includes them in its 90 % bulk probability region. This demonstrates the self-consistency of the inference approach – the data-generating parameters are consistent with the data. For the posterior inferred from more measurements, the total area assigned with substantial probability is more condensed than the area of the other posterior, reflecting the lower level of uncertainty about the data-generating parameters

as a result of the information gain from the additional measurements.

The ability to estimate the area of parameter space that is consistent with treatment response measurements is one of the primary reasons for the appeal of Bayesian inference for MIPD, making it possible to account for the limited information provided by noisy measurements. This helps to avoid predictive errors resulting from the use of erroneous parameter estimates. In particular, we can simulate treatment responses for different samples from the posterior distribution to predict a range of feasible treatment outcomes. Formally, this defines a distribution of treatment responses

$$p(y|t, r, \mathcal{D}) = \int \mathrm{d}\psi \, p(y|t, r, \psi) \, p(\psi|\mathcal{D}), \tag{2.23}$$

where $p(y|t, r, \mathcal{D})$ quantifies the probability of y being a measurement of the measured treatment response at time t. $p(y|t, r, \mathcal{D})$ is referred to as the posterior predictive distribution in Bayesian inference (Gelman, Carlin, et al. 1995). Focusing more on the treatment response than on the measurements, we can also simulate noise-free measurements to construct the posterior predictive distribution by choosing $p(y|t, r, \psi) = \delta\Big(y - \bar{y}(t, r, \psi)\Big)$, in which case $p(y|t, r, \mathcal{D})$ quantifies the probability density of the treatment response at time t. This noise-free limit is illustrated for both warfarin posteriors in the left panel of Figure 2.13. The measured treatment response (black) is contained within the 99 % bulk probability region at each time point for the early fit (see blue area). For the later fit, the posterior predictive distribution becomes narrower and contains the measured treatment response within its 90 % bulk probability region at each time point (see red area).

Another argument in favour of Bayesian inference is its mechanism for informing the parameter estimation by knowledge or expectations for feasible parameter values, quantified in terms of the prior distribution, $p(\psi)$. Such prior information may, for example, be based on experience with similar medicines or similar patient groups, or it may be obtained from earlier clinical trials (Sheiner and Beal 1982). Prior distributions complement the information gained from measurements and help to narrow down the parameter space, making the inference more data-efficient.

However, prior distributions also introduce additional complexity to the inference approach and can bias the inference results. For example, in Figure 2.13 the MLEs are not located at the peaks of the posterior distributions, demonstrating the influence of the prior distribution on the posterior distribution (see Equation 2.22). This influence of the prior can improve the inference results, if the prior information is consistent with the measured treatment response. But for prior expectations inconsistent with the data-generating process, the inference results can also be strongly biased, especially when prior distributions exclude important regions of the parameter space. In subsection 2.2.8, we will introduce an inference approach to derive suitable prior distributions for MIPD from clinical trial data using nonlinear mixed effects (NLME) modelling.

In summary, statistical inference cannot estimate parameter values without error from a finite number of noisy measurements. This makes point estimates of parameter values unsuitable for MIPD in clinical practice where the availability of clinical data is limited. Bayesian inference is one of several inference approaches that overcomes these shortcomings by estimating areas in parameter space that are consistent with available measurements, making the prediction of a range of feasible treatment responses possible. Bayesian inference can also improve the data-efficiency of the parameter estimation by providing a mechanism to capture prior information about feasible parameter values.

2.2.6 Sampling from posterior distributions using MCMC

The change from determining MLEs to estimating posterior distributions comes with a need for new algorithms to enable the inference. In this section, we introduce Markov chain Monte Carlo (MCMC) algorithms – one of the most popular (and in this thesis ubiquitously used) classes of algorithms for the estimation of posterior distributions (Gelman, Carlin, et al. 1995; Jones et al. 2022). Other common approaches, such as Variational Inference (Blei et al. 2017), are not going to be discussed and we refer to the literature as those approaches are, arguably, less generally applicable.

We begin the review with a brief primer on Markov chains, providing the theoretical foundation for MCMC. We then illustrate how the properties of Markov chains can be exploited to estimate posterior distributions in an iterative fashion. We conclude

the section with a discussion of numerical challenges and widely used criteria to assess the convergence of MCMC inference results.

Markov chains are stochastic models for processes that are difficult (or impossible) to model deterministically (Norris 1998). Such processes include the motion of a molecule in a solution, the fluctuations of stock prices on a stock exchange, or the dynamics of the weather (Agbinya 2022). Markov chains model such dynamics iteratively using transition probabilities which propagate one state of the system to the next. These transition probabilities are assumed to depend only on the current state of the system

$$p(s_{j+1}|s_j, s_{j-1}, \ldots, s_0) = p(s_{j+1}|s_j), \tag{2.24}$$

where s_j denotes the state of the system at time t_j, and s_{j+1} denotes the next state at time $t_{j+1} > t_j$. This independence from the states before s_j is the defining property of first order Markov chains, referred to as the 'memoryless' property.

While the study of stochastic processes with the use of Markov chain models is interesting in its own right, the focus in Bayesian inference is on the properties of the dynamics arising from their memorylessness (Brooks et al. 2011). For example, if a state in a discrete space is revisited at least once, with probability 1 and within a finite number of transitions, it is revisited infinitely many times under the Markov chain dynamics in the limit of infinitely many transitions. This is a direct consequence of the memorylessness of the dynamics. States that are being revisited infinitely times are referred to as *recurrent* states. If all states in a Markov chain are recurrent and the expected number of transitions for recurrence is finite, the Markov chain is called *positive recurrent*.

Positive recurrent Markov chains are the main focus of MCMC algorithms, as they are guaranteed to have stationary distributions, $p^*(s)$, which are invariant under the Markov chain dynamics (Norris 1998). This means states that are sampled from a stationary distribution and propagated one step using the transition probability will still be distributed according to the stationary distribution. Stationary distributions also exist for continuous state spaces, but the notion of recurrence is different as the probability of returning to the same state is zero for uncountable state spaces. A notion of recurrence that guarantees stationary distributions for continuous state spaces is

Harris-recurrence, which relaxes the recurrence condition to returning to an area of state space instead of returning to the same state (Roberts et al. 2004).

Formally, the probability of sampling a state, s', and transitioning to another state, s, is given by $p(s|s')\,p^*(s')$. The invariance of the stationary distribution implies that marginalising $p(s|s')\,p^*(s')$ over s' recovers the stationary distribution

$$p^*(s) = \int \mathrm{d}s'\, p(s|s')\, p^*(s'). \tag{2.25}$$

Stationary distributions of Markov chains are interesting for the estimation of posterior distributions because any distribution, $p(s)$, is guaranteed to converge to a stationary distribution, $p^*(s)$, under the dynamics of the Markov chain in the limit $n \to \infty$ (Norris 1998). If the Markov chain is *irreducible*, this stationary distribution is unique. Irreducibility for discrete state spaces means that any recurrent state revisits any other recurrent state infinitely many times in the limit $n \to \infty$. For continuous state spaces, irreducibility is again relaxed to areas of parameter space and is referred to as ϕ-irreducibility (Roberts et al. 2004).

The existence of unique stationary distributions for irreducible and positive recurrent Markov chains is exploited in MCMC algorithms to estimate posterior distributions. More specifically, the transition probability, $p(s|s')$, of an irreducible and positive recurrent Markov chain is chosen such that the posterior distribution is its stationary distribution, $p^*(s) = p(\psi|\mathcal{D})$. Any initial distribution can then be evolved iteratively using the Markov chain dynamics until it converges to the stationary distribution to obtain the posterior distribution. Here, the 'states' of the Markov chain are values for the model parameters, ψ. In Figure 2.14, we illustrate this concept using the warfarin model from subsection 2.2.5. For simplicity, we estimate the posterior distribution only for the elimination rate, k_e.

The figure shows the prior distribution for k_e in the left panel (see black distribution) and the estimated posterior distribution in the right panel (see red distribution). Samples from the prior distribution are illustrated in the middle panel using black scatter points. These samples are propagated 200 steps using the transition probability of a tailor-made Markov chain whose stationary distribution is constructed to equal the posterior distribution. The samples at each iteration are illustrated in grey. A randomly-chosen

trace of samples is highlighted in blue. Visually, the distribution of samples remains stationary after the first 100 iterations. The samples on the 200th iteration can therefore be interpreted as samples from the stationary distribution – here, the posterior distribution (see red scatter points).

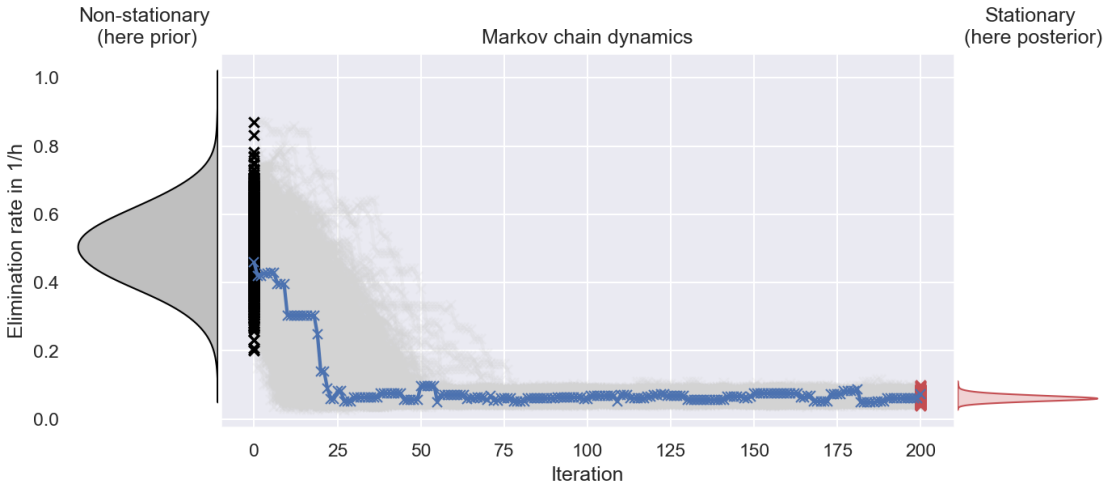


Figure 2.14: Stationary distributions of Markov chains. The figure illustrates the convergence of the Metropolis-Hastings Markov chain to its stationary distribution. The initial starting points of 1000 chains are drawn from an arbitrary (non-stationary) distribution (see black scatter points). The non-stationary distribution is chosen to equal the prior distribution of the elimination rate, k_e, (see black distribution in the left panel). The traces of the Markov chains over 200 iterations are illustrated in grey. An arbitrarily-chosen trace is highlighted in blue. The transition probability of the Markov chain is chosen such that its stationary distribution equals the posterior distribution of the elimination rate. The posterior distribution is illustrated in red (see right panel).

MCMC algorithms exploit the convergence of Markov chains for the purpose of estimating posterior distributions. However, in practice it is more common to use the trace of just a few samples to estimate the posterior distribution, instead of propagating many samples in parallel (see e.g. the blue trace in Figure 2.14). For this to work, MCMC algorithms make use of the irreducibility of the Markov chain, guaranteeing that all recurrent states can be visited from any other recurrent state. As a result, the trace of a single chain (of a sufficient length) can be histogrammed to reconstruct the posterior distribution.

The art of MCMC is to construct transition probability distributions such that the resulting Markov chain is irreducible, positive recurrent, and has a stationary distribution

that equals the targeted posterior distribution. The first who identified such a transition probability distribution were Metropolis and his collaborators, leading to the development of the first-ever MCMC algorithm (Metropolis et al. 1953). Since then, many other MCMC algorithms have been developed, e.g. (Haario, Saksman, et al. 2001; Hastings 1970; Hoffman et al. 2014; Neal 2010).

Metropolis et al realised that the construction of suitable Markov chains is simplified by focussing on a subclass of positive recurrent Markov chains called *reversible* Markov chains, which satisfy the time-reversal condition[7],

$$p(s'|s)\,p^*(s) = p(s|s')\,p^*(s'). \tag{2.26}$$

Here, time-reversibility means that sampling a state, s, from the stationary distribution and transitioning to another state, s', is as likely under the Markov chain dynamics as the reversed process of sampling a state s' and transitioning to s. Reversible Markov chains are positive recurrent by construction as can be shown by integrating both sides of Equation 2.26 over s', which recovers Equation 2.25. Metropolis et al used this time-reversal to prove the positive recurrence of a transition probability that is composed of a proposal distribution and an acceptance distribution,

$$p(s|s') = \int \mathrm{d}s''\, p(s|s'', s')\, \mathcal{N}\Big(s''\Big|s', \sigma^2\Big), \tag{2.27}$$

where $\mathcal{N}(s''|s', \sigma^2)$ is the proposal distribution and $p(s|s'', s')$ is the acceptance distribution. The standard deviation, σ, is a hyperparameter of the transition probability. Here, each transition can be broken down into two steps: first, a proposal state, s'', is drawn from the proposal distribution. This proposal is then either accepted or rejected. If a proposal is rejected, the Markov chain remains in the current state, s'. If a proposal is accepted, the Markov chain moves to the proposed state, s''. The acceptance probability is determined by the stationary distribution and equals $p(s = s''|s'', s') = \min\Big(p^*(s'')/p^*(s'), 1\Big)$.[8] The convergence to the stationary

[7] This condition is also referred to as *detailed balance*.

[8] Formally, the Metropolis acceptance distribution can be written as

$$p(s|s'', s') = \delta(s - s'')\,\min\Big(p^*(s'')/p^*(s'), 1\Big) + \delta(s - s')\,\Big(1 - \min\Big(p^*(s'')/p^*(s'), 1\Big)\Big), \tag{2.28}$$

where $\delta(x)$ denotes the Kronecker-delta distribution which is zero everywhere except for $x = 0$.

Algorithm 2.1: Metropolis-Hastings (MH) MCMC algorithm.

Input : 1. Likelihood: $p(\mathcal{D}|\psi)$;
2. Prior: $p(\psi)$;
3. Starting point: $\psi^{(0)}$;
4. Step size: σ;
5. Number of iterations: n

Output : Samples of MH Markov chain.

$samples = [\,]$ `// Initialise sampling`
$\psi' = \psi^{(0)}$
for $i \leftarrow 1$ **to** n **do**
  $\psi'' \sim \mathcal{N}(\,\cdot\,|\psi', \sigma^2)$ `// Sample proposal`
  $x \sim \mathcal{U}(\,\cdot\,|0, 1)$ `// Sample uniform random variable`
  $accepted = x < \Big(p(\mathcal{D}|\psi'')p(\psi'')/p(\mathcal{D}|\psi')p(\psi')\Big)$ `// Accept or reject`
  if $accepted$ **then**
    $\psi' = \psi''$ `// Continue from proposal`
  end if
  $samples$.append($[\psi']$) `// Store current sample`
end for
return *samples*

distribution can be shown using the time-reversal condition.[9] The uniqueness of this stationary distribution follows from the ϕ-irreducibility of the Markov chain.[10]

Setting the stationary distribution equal to the posterior distribution (Equation 2.22), we find that the acceptance probability of the Metropolis Markov chain is given by

$$p(\psi = \psi''|\psi'', \psi') = \min\left(\frac{p(\mathcal{D}|\psi'')\,p(\psi'')}{p(\mathcal{D}|\psi')\,p(\psi')}, 1\right), \tag{2.29}$$

[9] To prove the convergence to the stationary distribution, $p^*(s)$, Metropolis et al made use of the time-reversibility condition (Equation 2.26). More specifically, plugging in the Metroplis transition probability (Equation 2.27) into the time-reversibility condition yields

$$\int \mathrm{d}s''\, p(s'|s'', s)\,\mathcal{N}\Big(s''\Big|s, \sigma^2\Big)p^*(s) = \int \mathrm{d}s''\, p(s|s'', s')\,\mathcal{N}\Big(s''\Big|s', \sigma^2\Big)p^*(s').$$

Simplifying this expression using the properties of $\delta(x)$ and the invariance of the normal distribution under the exchange of s and s', $\mathcal{N}(s|s', \sigma^2) = \mathcal{N}(s'|s, \sigma^2)$ (see Equation 2.50), reduces the equality to

$$\min\Big(p^*(s')/p^*(s), 1\Big)\,p^*(s) = \min\Big(p^*(s)/p^*(s'), 1\Big)\,p^*(s').$$

To show that this equality holds, we distinguish three cases: 1. $p^*(s) = p^*(s')$; 2. $p^*(s) > p^*(s')$; 3. $p^*(s) < p^*(s')$. The equality holds for all three cases, proving that the Metropolis-Hastings Markov chain is reversible. This concludes the proof, as all reversible Markov chains are positive recurrent and have a stationary distribution, $p^*(s)$.

[10] The ϕ-irreducibility can be seen by noting that the Metropolis Markov chain permits transitions to any region in parameter space from any point in the parameter space in just one iteration with probability greater zero.

which only depends on the ratio of the likelihood-prior of the current set of parameters, ψ', to the proposed set of parameters, ψ''. The model evidence, $p(\mathcal{D})$, does not need to be evaluated.

As a result, posterior distributions can be estimated using an algorithm consisting of two recurring steps (see Algorithm 2.1): 1. a proposal step, where a proposal, ψ'', is drawn from a normal distribution which is centered at the previous parameters, ψ'; and 2. an acceptance step, where the proposal is accepted with probability $p(\psi = \psi''|\psi'', \psi')$ (see Equation 2.29). This is implemented by comparing the likelihood-prior ratio to the realisation of a uniform random variable (see line 6 in Algorithm 2.1). If the realisation of the random variable is smaller than the ratio, the proposal is accepted and the next iteration of the Markov chain continues from ψ''. Otherwise, it is rejected and the chain remains at ψ'. Algorithm 2.1 is commonly referred to as Metropolis-Hastings (MH) MCMC algorithm[11] in attribution of Hastings' substantial contributions to generalisations of the algorithm (Hastings 1970).

Note that at no point during the convergence proof of the MH MCMC algorithm did we specify the properties of the posterior distribution, meaning that MH MCMC can, in theory, be used to estimate *any* posterior distribution. Such a convergence guarantee is in stark contrast to numerical optimisation algorithms, e.g. those discussed in subsection 2.2.3, for which typically no guarantees for convergence to global optima exist. However, in practice, there are still challenges which can make MCMC algorithms impractical for the estimation of certain posterior distributions. In the remainder of this section, we discuss these challenges and highlight common metrics used to analyse the convergence of MCMC results.

The practical limitations of the MH MCMC algorithm (and many other MCMC algorithms) can be illustrated by attempting the estimation of a posterior distribution with different values for the hyperparameter, σ, measuring a typical step length in the proposal distribution. In Figure 2.15, we show three such attempts of estimating

[11] Note that for numerical stability, the acceptance criteria in line 6 is, in practice, evaluated using a log-transformation of the expression

$$\log x \quad < \quad \log p(\mathcal{D}|\psi'') + \log p(\psi'') - \log p(\mathcal{D}|\psi') - \log p(\psi').$$

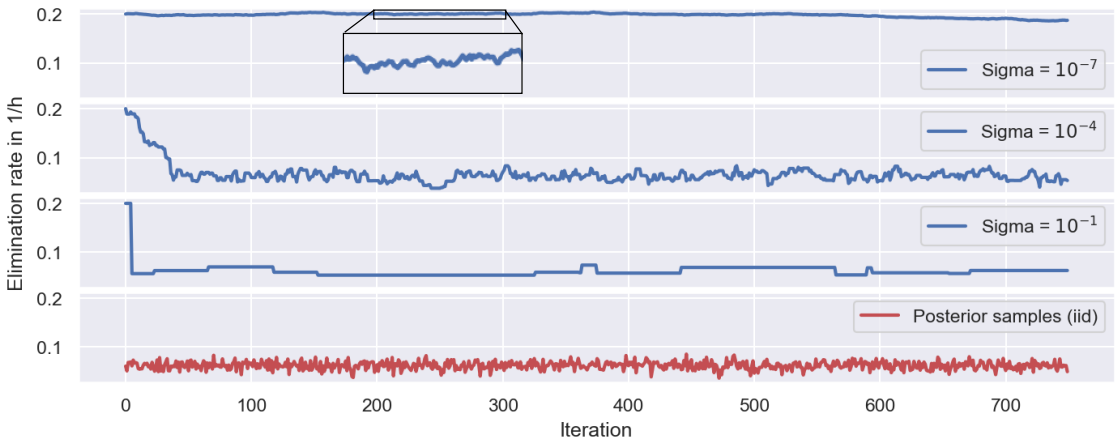


Figure 2.15: Autocorrelations of MCMC samples. The figure shows MCMC samples generated with the MH MCMC algorithm for different values of σ. The bottom panel shows iid values sampled from the targeted posterior distribution.

the warfarin posterior distribution. The figure shows three traces of the MH MCMC algorithm, each obtained with a different σ (panel 1: $\sigma = 10^{-7}$; panel 2: $\sigma = 10^{-4}$; and panel 3: $\sigma = 10^{-1}$). All three traces start at the same initial value, $k_e^{(0)} = 0.2\,1/\text{h}$. The trace in panel 4 is not generated using an MCMC algorithm but shows iid samples from the posterior distribution.[12]

The figure shows that the choice of the hyperparameter, σ, substantially influences the values sampled with the MCMC algorithm: for a very small value of σ, the Markov chain remains close to its starting point, seemingly not moving within 800 iterations (see panel 1 in Figure 2.15). Only at a smaller-scale-look is it visible that the Markov chain performs Brownian motion-like dynamics around the starting point (see inset in panel 1); for larger values of σ, the Markov chain quickly moves away from its initial value and converges to the area of parameter space supported by the posterior distribution (see panel 2). Once converged, the trace of the samples looks similar to the trace of the iid samples in panel 4; for very large values of σ, the Markov chain jumps to the relevant area of parameter space almost immediately but then seemingly does not move over many iterations (see panel 3); in total, the Markov chain for large σ seems to accept only around 10 proposals within the displayed 800 iterations of the MCMC algorithm.

[12] Note that, in practice, we can rarely sample from the posterior distribution directly to make this comparison.

The figure illustrates that the speed of convergence of the MH MCMC algorithm sensitively depends on the choice of the hyperparameter, σ. For very small values of σ, the convergence to the posterior distribution is visibly the slowest (see panel 1). This is because the average distance[13] from the current location to the proposed location is σ, here 10^{-7}, meaning that even if each step of the Markov chain was in the direction of the mean of the posterior distribution, the algorithm would need around 10^6 iterations to reach it (distance to posterior mean $\approx 0.1\,1/\mathrm{h}$). The convergence is slowed down even further by the mismatch between the standard deviation of the posterior distribution and the value of σ: the value of σ is 5 orders of magnitude smaller. This means that in each iteration the posterior density will not substantially change over the range of possible proposals and remain close to the value of the posterior density of the current location, resulting in an almost guaranteed acceptance of the proposal, regardless of whether the Markov chain steps towards or away from the mean of the posterior distribution. There is nevertheless a small bias towards areas with higher posterior probability density, explaining the marginal drift to smaller k_e values at the end of 800 MCMC iterations in the top panel of Figure 2.15.

As the value of σ approaches the standard deviation of the posterior distribution, changes of the posterior density with each proposal become more substantial and the Markov chain moves more efficiently towards the area of the bulk probability mass for the posterior distribution (see panel 2). Here, σ is approximately two orders of magnitude smaller than the standard deviation of the posterior distribution. The samples generated by the MCMC algorithm are more correlated than the iid values sampled directly from the posterior distribution (see panel 4). This has two reasons: 1. the two-orders-of-magnitude-smaller standard deviation of the proposal distribution naturally constrains new samples to be closer to each other (i.e. to be more correlated) than if the samples were sampled independently from the posterior distribution; and 2. the Markov chain has a nonzero probability at each iteration to remain at its current location, making subsequent samples occasionally identical to each other.

[13] Formally, average distance refers, here, to the root of the mean squared distance.

As the value of σ surpasses the standard deviation of the posterior distribution, the correlations between the MCMC samples become even more apparent (see panel 3). Here, σ is one order of magnitude larger than the standard deviation of the posterior distribution. This means that as soon as the Markov chain manages to reach a high posterior density region, proposals have a high probability of not being inside this high density region, resulting in almost guaranteed rejections and a stagnation of the MCMC algorithm.

To avoid the complication of needing to find a good step size for fast convergence of the MH MCMC algorithm, other MCMC algorithms have been developed. One of them is the Adaptive Covariance Matrix Monte Carlo (ACMC) algorithm, which adapts the covariance matrix of the proposal distribution during the initial iterations of the MCMC algorithm to reach a fixed acceptance rate of proposals (Haario, Saksman, et al. 2001). This makes it possible to automatically find proposal distributions that avoid the scenario in panel 3, and reduce the risk that the Markov chain does not leave high probability density regions of parameter space, once it has found them. This approach can work very well for low dimensional parameter spaces. However, the algorithm becomes less and less efficient as the dimension of the parameter space grows, transitioning from a sampling scenario similar to the one illustrated in panel 2 to a sampling scenario closer to that illustrated in panel 1. This is because the volume of parameter space with high probability mass increases in volume much slowly that the volume of the total space as the dimensionality of the space increases (Betancourt 2017), forcing the algorithm to choose smaller and smaller step sizes to remain in high probability regions, once it has found them, in order to maintain a fixed acceptance probability. This makes ACMC impractical for the inference of high dimensional posterior distributions.

Another MCMC approach, suitable for high dimensional posterior distributions, is the Hamiltonian Monte Carlo (HMC) algorithm, which uses geometric properties of the posterior distribution to efficiently sample from it (Neal 2010). More specifically, HMC translates the proposal process of the MH MCMC algorithm into a physics-motivated dynamical process which can be described using a Hamiltonian dynamical system. This dynamical system can be solved and evolved in time using numerical integration. The starting point of the dynamical system is determined by the current location of the

Markov chain, while the end point of the dynamical system, after evolving it over some time period, provides the proposal. The step size of the numerical integrator and the length of the evolution time period are hyperparameters of the algorithm. An interesting property of this mapping is that it also changes the formulation of the acceptance probability: the likelihood-prior ratio in Equation 2.29 is replaced by the 'energy' ratio of the system after and before evolving it. The insight for the HMC algorithm was that the energy of Hamiltonian dynamical systems is conserved, meaning that, in theory, proposals are accepted with probability 1, regardless of how long or how far the dynamical system is evolved. As a result, the sampling efficiency of the algorithm can approach that of iid sampling for appropriately chosen evolution times.

In practice, errors during the numerical integration can, however, lead to violations of energy conservation and to a reduction in the acceptance probability. These errors sensitively depend on the step size of the numerical integrator (Betancourt 2017). To avoid having to manually find optimal step sizes for the HMC algorithm, Hoffman et al developed an extension of the algorithm which, among other things, uses its initial iterations to automatically tune the step size and the evolution time of the algorithm. This algorithm is known as the No-U-Turn sampler (NUTS) (Hoffman et al. 2014).

A challenge with the HMC algorithm and the NUTS algorithm is that both tend to require 100s of evaluations of the likelihood, the prior and their gradients to evolve the Hamiltonian system in time, making the algorithms computationally more expensive per iteration than the ACMC algorithm. However, the rapid efficiency decline of the ACMC algorithm with the dimensionality of the posterior distribution makes HMC and NUTS the preferred choice for the inference of posterior distributions with many parameters. Many other MCMC algorithms also exist, see e.g. (Braak 2006; Haario, Laine, et al. 2006; Vrugt et al. 2009), but in this thesis, we will use the ACMC algorithm for low dimensional problems and the NUTS algorithm for high dimensional problems, because these algorithms were found to perform well.

Despite the plethora of MCMC algorithms and their different advantages and limitations, one challenge across all of them remains the assessment of whether the histogram of MCMC samples from a given Markov chain has converged to the posterior

distribution or not. To facilitate this assessment, two primary metrics have been developed: 1. the $\hat{R}$-metric; and 2. the effective sample size (ESS). The $\hat{R}$-metric requires multiple runs of an MCMC algorithm and compares the variance of MCMC samples within each chain to the variance across the chains (Vehtari, Gelman, Simpson, et al. 2021). This makes it possible to distinguish the scenario in panel 1 of Figure 2.15 from the scenario in panel 2, provided the starting points of the chains are well dispersed in the parameter space. If the within-chain-variance is much smaller than the across-chain-variance ($\hat{R} > 1$), the chains are not sampling from the same region of parameter space and, therefore, have not converged. If the within-chain-variance is equal to the across-chain-variance ($\hat{R} = 1$), the chains sample from the same region of parameter space, despite having started from very different initial starting points. This provides an indication that the Markov chains have converged.

Note that, for convergence, it can be helpful to discard the samples from the initial iterations of each Markov chain as those samples are typically artefacts of the initialisation of the algorithm and do not involve parameter values that are supported by the posterior distribution (Cowles et al. 1996). Without discarding these iterations, an enormous amount of MCMC samples may be needed to converge to the posterior distribution, as the parameter values obtained during the 'warm-up' phase of the algorithm need to be dominated by the values obtained post warm-up. For adaptive MCMC algorithms, like ACMC and NUTS, it is even *required* to discard the iterations during the tuning phase of the algorithms, as their convergence proofs typically assume fixed hyperparameters (Haario, Saksman, et al. 2001; Hoffman et al. 2014).

The ESS metric is a complementary metric to the $\hat{R}$-metric, which estimates the number of iid samples equivalent to the number of correlated samples generated by the MCMC algorithm (Cowles et al. 1996). This helps to assess the quality with which histograms of MCMC samples can approximate the posterior distribution. For example, if the equivalent number of iid samples is much less than 1000 (i.e. ESS $\ll$ 1000), we can expect histograms of each parameter to feature substantial stochastic deviations from the true marginal posterior distributions. The number of equivalent iid samples can be less, equal or greater than the number of MCMC samples. For example, the

Markov chain in panel 3 of Figure 2.15 has 800 MCMC samples, while its ESS is at most 10, as most of the MCMC samples are identical to each other. However, some MCMC algorithms achieve negative autocorrelations, sampling more efficiently from the posterior distribution than independent draws.

This concludes the section on sampling from posterior distributions using MCMC, closing the gap from subsection 2.2.5 in providing a procedure to estimate parametric uncertainty of PKPD models in practice. In the next section, we will further extend the PKPD modelling approach to MIPD, and discuss how we can use nonlinear mixed effects modelling (NLME) to estimate suitable prior distributions for MIPD.

2.2.7 Nonlinear mixed effects modelling

In subsection 2.2.5, we highlighted that Bayesian inference, in contrast to maximum likelihood estimation, is fundamentally an information-updating mechanism, building on prior information to arrive at inference results. In this section, we show that, in the context of MIPD, this updating mechanism is an opportunity to improve the data-efficiency of the parameter estimation (Sheiner and Beal 1982). More specifically, we will show that informative prior distributions, obtained from preceding clinical trials, make it possible to estimate better PKPD model parameter values from fewer monitoring measurements. The modelling framework that will enable this efficiency gain is nonlinear mixed effects (NLME) modelling (Lavielle 2014).

To motivate NLME modelling and its use for MIPD, we consider a typical dataset generated during a clinical trial. In this case, the clinical trial is simulated and focusses on the pharmacokinetics of warfarin (Augustin, Lambert, Robinson, et al. 2023). During the trial, the warfarin blood concentration levels of 1000 patients are monitored, each treated with the same dosing regimen.[14] Using Bayesian inference (subsection 2.2.5), we fit the warfarin PK model repeatedly to the data, obtaining one set of model parameters for each individual in the trial. Figure 2.16 illustrates the result of this approach. The left panel shows the fits of the model to the monitoring data, and the right panel shows the

[14] Note that the number of patients in a typical clinical trial depends on the phase and the objective of the trial. In this case, we choose 1000 patients for illustrative purpose. In practice, early phase clinical trials may have significantly fewer participants (Augustin, Lambert, Robinson, et al. 2023).

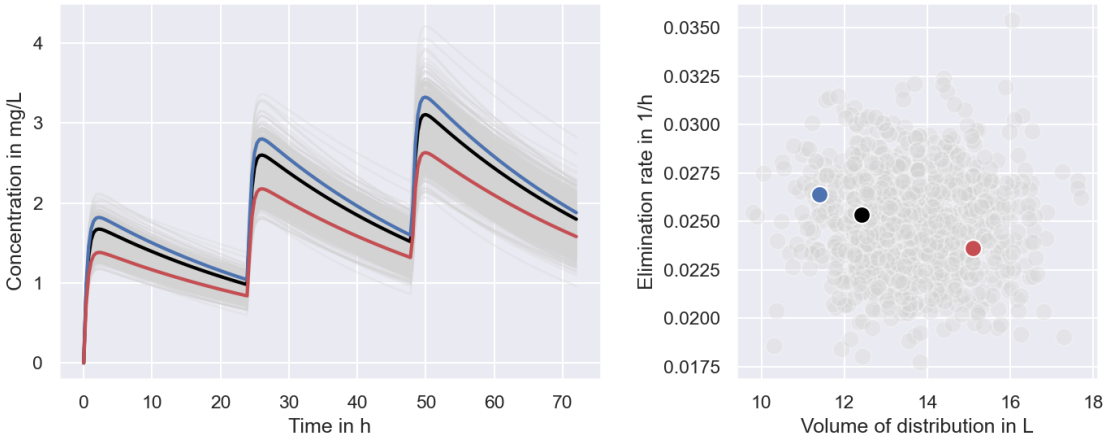


Figure 2.16: Inter-individual variability in PKPD modelling. The left panel of the figure shows PKPD model simulations fitted to (fictitious) clinical trial data. For visual purposes, the clinical trial data is not shown in the figure. The parameter estimates corresponding to the fits are shown in the right panel. In this figure, it is assumed that the parameters can be estimated without residual uncertainty. The parameters and fits of three arbitrarily-chosen individuals are highlighted in black, blue, and red.

corresponding model parameters, estimated from the data (see grey lines and markers). For illustrative purposes, we do not show the clinical trial data and assume (for now) that each individual is monitored sufficiently, such that all parameters can be estimated without any residual uncertainty (see markers in right panel). The fits and the model parameters of three arbitrarily chosen individuals are highlighted in black, blue, and red.

The figure shows that the fits (unsurprisingly) differ across individuals, reflecting the inter-individual variability (IIV) of warfarin treatment responses. The figure also shows that this treatment response variability is translated in PKPD modelling into varying model parameter values across individuals (see right panel in Figure 2.16). For example, the estimates of the volume of distribution range between 10 L and 18 L across individuals, while the elimination rate estimates assume values between 0.0175 1/h and 0.0350 1/h. Importantly, the parameter values occupy a relatively confined region in parameter space, aligning with the intuition that the pharmacological properties of a drug are broadly similar across individuals, unless specific conditions, such as genetic defects or co-medication, drastically alter important pathways for the drug-body or body-drug interactions (Mandema et al. 1992).

This variation of the model parameters in parameter space motivates the definition of a population model, $p(\psi|\theta)$, which describes the distribution of model parameters across individuals in a population (Lavielle 2014). Here, θ denotes the parameters of the population model. For clarity, we will henceforth refer to ψ as individual-level parameters and to θ as population-level parameters. In this modelling approach, a particular individual (e.g. the red individual in Figure 2.16) is interpreted as a random sample, ψ, from a population of individuals characterised by $p(\psi|\theta)$. In theory, the population model does not require any additional modelling choices, as the histogram over the parameter estimates from sufficiently many individuals converges to the distribution of the model parameters in the population. However, in practice, only a limited number of individuals are monitored in clinical trials, making modelling choices for the population distribution necessary to facilitate the estimation of the population model. Popular choices include multivariate normal distributions and lognormal distributions, but, in principle, any probability distribution can be used as a population model.

For example, a population model which may be used to describe the warfarin clinical trial data in Figure 2.16 is

$$p(\psi|\theta) = \mathcal{N}(k_e|\mu_{k_e}, \sigma^2_{k_e})\,\mathcal{N}(v|\mu_v, \sigma^2_v)\,\delta(k_a - \mu_{k_a}), \tag{2.30}$$

where $\psi = (k_a, k_e, v)$ are the individual-level parameters, defined by the warfarin PK model in Equation 2.4, and $\theta = (\mu_{k_e}, \sigma^2_{k_e}, \mu_v, \sigma^2_v, \mu_{k_a})$ are the population-level parameters. The IIV of the volume of distribution and the elimination rate are modelled using normal distributions, while the absorption rate is assumed to display no variation in the population.[15] In this population model, all individual-level parameters are modelled independently, allowing us to multiply their marginal distributions to construct $p(\psi|\theta)$. However, when individual-level parameters co-vary in a population, population models with correlations between the individual-level parameters are better suited to model the IIV (Lunn et al. 2002). Note that each individual-level parameter value is mapped to at least one population-level parameter, even when a parameter is assumed to be the same

[15] Recall, that in subsection 2.2.5, we fixed the absorption rate prior to the parameter estimation, making a population model which assumes varying k_a parameters across individuals, in this case, inconsistent with the parameter estimates.

across individuals. These population-level parameters can be estimated from clinical trial data, as we will discuss in the next section (subsection 2.2.8).

The population model, together with the PKPD model, can be used to simulate the treatment response variability in a population using:

$$p(y|t,r,\theta) = \int \mathrm{d}\psi \, p(y|t,r,\psi)\, p(\psi|\theta). \tag{2.31}$$

$p(y|t,r,\psi)$ refers to an individual's measurement distribution (Equation 2.12), which can be considered with or without measurement noise, depending on the focus of the simulation. $p(y|t,r,\theta)$ is commonly referred to as nonlinear mixed effects (NLME) model, where 'nonlinear' refers to the often nonlinear dynamics of PKPD models, and 'mixed effects' refers to the traditional decomposition of an individual's parameter value into a value that is fixed in the population, $\bar{\psi}$, and a random value that varies from individual to individual, η (Lavielle 2014). For example, for a population model which assumes normally distributed parameters, $p(\psi|\theta) = \mathcal{N}(\psi|\mu,\sigma^2)$, the traditional NLME picture interprets the population mean as a fixed effect, $\bar{\psi} = \mu$, making it possible to express an individual's parameter as a sum of the fixed effect and the individual-specific random effect, $\psi = \bar{\psi} + \eta$, where η is normally distributed with mean zero and variance σ^2. Note that the decomposition into fixed and random effects depends on the choice of the population model. For example, for a lognormal distribution, the decomposition is $\psi = \bar{\psi}\,\mathrm{e}^{\eta}$, where $\bar{\psi}$ is the median of the population distribution. In this thesis, we will, for most parts, focus on the distribution of individual-level parameters in a population, $p(\psi|\theta)$, and abstain from using the traditional picture of NLME modelling.

In the context of MIPD, it is worth noticing that the NLME model (Equation 2.31) is structurally identical to the posterior predictive distribution in Equation 2.23. In both instances, the PKPD model is simulated for different individual-level parameters, sampled from a distribution of feasible model parameters, to construct a range of treatment responses. For the NLME model, this distribution is the population model, $p(\psi|\theta)$, while for the posterior predictive distribution, it is the posterior distribution, $p(\psi|\mathcal{D})$. In subsection 2.2.5, we highlighted that the posterior distribution contains all parameter values consistent with the data and prior knowledge. The population

model can therefore be interpreted as containing all model parameters consistent with the IIV in the population. I.e., the parameter values of each patient are among those parameters supported by the population model.

As a result, the population model provides a meaningful starting point for the dosing regimen individualisation of patients, even when the exact parameter values of a patient are unknown. With patient-specific information, this starting point can be updated and improved using Bayesian inference, resulting in a posterior distribution that is consistent with both, preceding clinical trials and the patient-specific data

$$\log p(\psi|\mathcal{D}) = \log p(\mathcal{D}|\psi) + \log p(\psi|\theta) + \text{constant}. \tag{2.32}$$

Here, the posterior distribution, $p(\psi|\mathcal{D})$, is conditioned on the values of the population-level parameters, but we will leave this conditioning implicit for ease of notation. For clarity, Equation 2.32 is Bayes' rule (Equation 2.22) rewritten for the logarithm of the posterior distribution, where the population model takes the place of the prior distribution, and the model evidence is denoted as a ψ-independent, constant term.

Using the population model as a prior distribution during MIPD transfers learnings from preceding clinical trials and reduces the information-burden on individual-specific monitoring measurements. Consequently, fewer measurements are needed to estimate individual-specific model parameters, and MIPD becomes more data-efficient. In particular, parameters that remain constant across individuals can often be accurately estimated prior to MIPD, effectively removing degrees of freedom from the PKPD model for the parameter individualisation. But also knowing the range of parameter values consistent with the IIV in a population helps to improve the data-efficiency (Sheiner and Beal 1982).

For extreme variability of parameter values, it can still be necessary to collect a substantial amount of individual-specific monitoring measurements in order to reduce the parametric uncertainty sufficiently, making MIPD without extensive monitoring challenging. However, in some instances, variability of parameter values can be partially explained by other patient-specific information, shifting the information-burden away from monitoring measurements. Such variability-explaining information is referred to

as covariates (Lavielle 2014). Examples of covariates include genetic information and clinical factors, such as single nucleotide polymorphisms (SNPs), age, sex, and the weight of a patient. Figure 2.17 illustrates two covariates for the warfarin use case.

The figure shows the population distributions of the EC50 parameter (see left panel) and the elimination rate parameter (see right panel), which are part of a larger population model for the warfarin PKPD model defined in Equations 2.4, 2.5, and 2.6. The population model was developed by Hamberg et al using clinical data (Hamberg, Dahl, et al. 2007; Hamberg, Wadelius, et al. 2010), and is defined in Appendix B.4.2. The marginal distributions of the parameters are illustrated on the right side of each panel (see black distributions). Both distributions indicate extreme variability in the population, with individual parameters deviating by up to 100 % from the population median. However, the figure also shows that the distributions can be decomposed into lower variance distributions using additional patient-specific information – the covariates. For example, the population distribution of the EC50 parameter is decomposed into three subpopulations, each forming its own distribution, by grouping patients by their VKORC1 genotype (one of GG, GA, or AA). This grouping substantially reduces the variability of the EC50 in each subpopulation relative to the full population distribution (see left panel). Similarly, the distribution of the elimination rate parameter is decomposed into subpopulations defined by the age of the patients (see right panel). In contrast to the VKORC1 gentoype, the age of patients is continuous, resulting in a family of subpopulations – one for each possible age of a patient.

The EC50 parameter and the elimination rate parameter of the warfarin model are examples of parameter values with large IIV where variability can be partially accounted for by covariates. For the warfarin model, this correlation is rooted in warfarin's pharmacology. For example, warfarin's mode of action is the inhibition of the vitamin K epoxide reductase complex (VKORC), and mutations in VKORC's subunit 1 (VKORC1) affect warfarin's inhibitory activity (Hamberg, Wadelius, et al. 2010), explaining the change of the EC50 parameter with the VKORC1 genotype in Figure 2.17. This justifies the use of covariates as additional prior information,

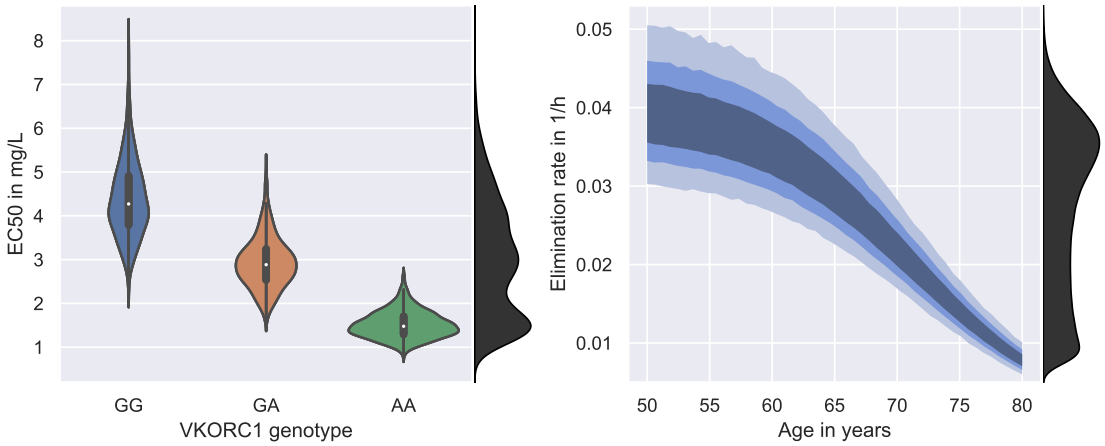


Figure 2.17: Covariate-based population models. The left panel shows Hamberg et al's population model for the EC50 of warfarin. The population distribution is illustrated in black. Subpopulations, obtained by grouping individuals in the population by their VKORC1 genotypes, are illustrated in blue, yellow, and green. The right panel shows Hamberg et al's population model for warfarin's elimination rate parameter. Here, subpopulations are defined by the age of patients (see blue distribution). The shades of blue indicate the bulk 67 %, 90 %, and 99 % probability regions of each subpopulation.

reducing the parametric uncertainty before estimating the posterior distribution from individual-specific monitoring measurements.

Formally, the decomposition of a population model into subpopulations is given by

$$p(\psi|\theta) = \int \mathrm{d}\chi \, p(\psi|\theta,\chi)\, p(\chi). \tag{2.33}$$

Here, χ denotes the covariates that define each supopulation, $p(\psi|\theta,\chi)$ denotes the subpopulation model, and $p(\chi)$ denotes the distribution of the covariates in the population.[16] During the parameter individualisation, the subpopulation model, $p(\psi|\theta,\chi)$, replaces the population model as prior distribution in Equation 2.32, reducing the initial parametric uncertainty to the subpopulation's IIV. For examples of subpopulation models, we refer to Appendix B.4.2.

[16] Note that for discrete covariates, the integral in Equation 2.33 is a sum over the values of the covariates. We are using an integral, irrespective of continuous or discrete covariates, just for ease of notation. For clarity, for discrete covariates, Equation 2.33 takes the form

$$p(\psi|\theta) = \sum_{\chi} p(\psi|\theta,\chi)\, p(\chi),$$

where the sum runs over the discrete values of the covariates. For example for the EC50 parameter in Figure 2.17, the sum runs over the VKORC1 genotypes (GG, GA, and AA).

This concludes the section on NLME modelling. In summary, NLME modelling, together with Bayesian inference, provides a procedure to improve the data-efficiency of MIPD. NLME models capture the variability of treatment responses across individuals and translate the IIV into varying individual-level parameters. The variation of these parameters is described using a population model which can serve as an informative prior distribution during MIPD, resulting in a reduced information-burden on individual-specific measurements. In the next section, we will discuss how we can use typical clinical trial data for the estimation of NLME model parameters, manifesting the so-far-asserted link between NLME modelling and the knowledge transfer from preceding clinical trials to MIPD.

2.2.8 Estimating parametric uncertainty from population data

In the previous section (subsection 2.2.7), we established that NLME modelling can be used to define informative prior distributions for MIPD, achieving low parametric uncertainty from fewer individual-specific monitoring measurements. The key to this efficiency gain is the population model which describes the variability of parameter values across individuals in a population. But, when informing inference results by model predictions, a similar question to the one in subsection 2.2.4 arises: how can we be sure that the model's *simulated* IIV truthfully represents the variability between real patients? I.e., how do we know that population models contain useful information for any to-be-treated patient in clinical practice?

Analogously to subsection 2.2.4, we can begin to address this question using statistical inference and estimate model parameter values from data. Before, we estimated the parameter values of PKPD models from measurements of treatment responses, ensuring that treatment response simulations were consistent with clinically observed treatment responses. Here, we can similarly ensure the consistency of the population model with clinical treatment response variability by estimating its parameters from 'measurements' of the individual-level parameters. More specifically, we can define a likelihood for values of the individual-level parameters under the population model

$$\log p(\Psi|\theta) = \sum_{i=1}^{N} \log p(\psi_i|\theta), \tag{2.34}$$

where ψ_i denotes the parameter values corresponding to an individual i. N denotes the number of measured individuals, and $\Psi = (\psi_1, \psi_2 \ldots, \psi_N)$ denotes the collection of parameter values across individuals. This construction is identical to the likelihood of treatment response measurements under the measurement model (Equation 2.14), with Ψ assuming the role of the measurements, and the population model assuming the role of the measurement model. The decomposition of $p(\Psi|\theta)$ into a product of individual parameter likelihoods assumes that the individual-level parameters are iid under the population model. For correlated individual-level parameters, the joint probability density does not factorise like this and the evaluation of the likelihood requires more care.[17]

The likelihood, $p(\Psi|\theta)$, can be translated into a posterior distribution using Bayes' rule, $\log p(\theta|\Psi) = \log p(\Psi|\theta) + \log p(\theta) +$ constant, analogously to subsection 2.2.5, where $p(\theta)$ denotes the prior distribution for the population-level parameters. This posterior distribution can be numerically estimated using MCMC sampling (see subsection 2.2.6).

An obvious challenge with this construction is that it assumes the availability of measurements of the individual-level parameters, ψ. The individual-level parameters are, however, a non-physical abstraction of an individual's treatment response, arising from the PKPD model description of the treatment response (see subsection 2.2.1), which makes it impossible to directly measure their values. We may, nevertheless,

[17] To see that Equation 2.34 only holds for iid individual-level parameters, note that we use $p(\Psi|\theta)$ as a shorthand notation for the joint likelihood of the individual-level parameters under the population model

$$p(\Psi|\theta) = p(\psi_1, \ldots, \psi_N|\theta). \tag{2.35}$$

Using the arithmetics of conditional probabilities, analogously to Equation 2.15, we can decompose this expression into two conditional probabilities

$$p(\Psi|\theta) = p(\psi_1, \ldots, \psi_{N-1}|\psi_N, \theta)\, p(\psi_N|\theta). \tag{2.36}$$

The conditioning on ψ_N in the first term can only be dropped, if the parameters are independently distributed, in which case the expression reduces to

$$p(\Psi|\theta) = p(\psi_1, \ldots, \psi_{N-1}|\theta)\, p(\psi_N|\theta). \tag{2.37}$$

This defines a recursive procedure that can be iteratively applied to the first term to factorise the joint likelihood into a product of N terms, one for each individual-level parameter. If the parameters are identically distributed, we obtain Equation 2.34, otherwise each set of parameters is captured by a different distribution.

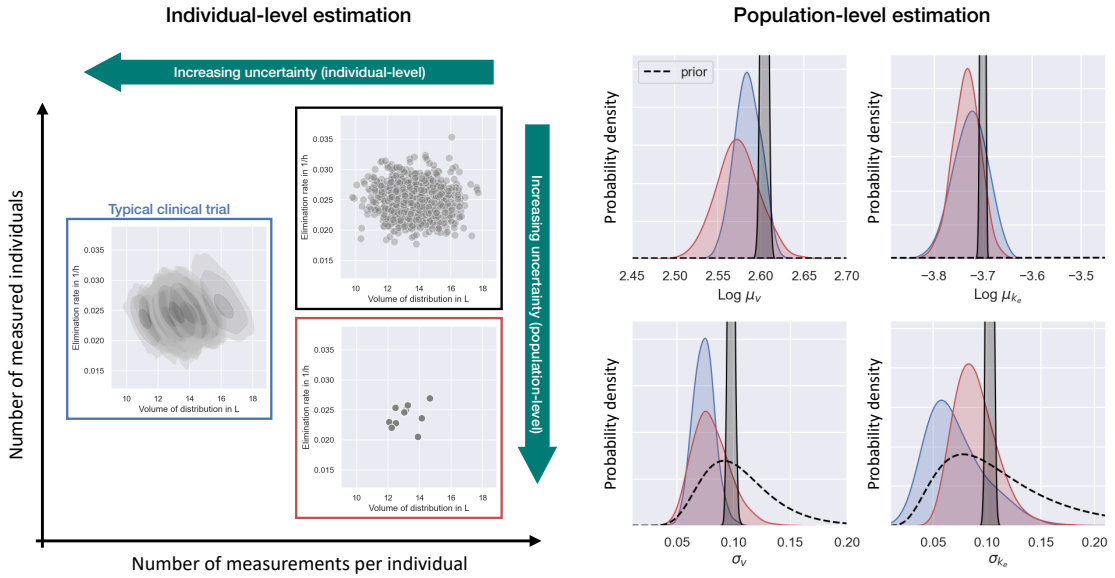


Figure 2.18: Estimation of population-level parameters. The figure illustrates the estimation of population-level parameters from three simulated datasets (black, blue, and red), containing treatment response measurements across varying numbers of individuals. The left panel shows the individual-level estimates, and the right panel shows the population-level estimates. For illustrative purposes, the figure does not show the treatment response measurements from which the individual-level parameters are inferred.

try to indirectly 'measure' the individual-level parameters by estimating their values from treatment response measurements. We can then use these inferred values of the individual-level parameters to estimate the parameters of the population model.

For closely monitored individuals, where individual-level estimates display negligible parametric uncertainty, this two-step inference approach poses no problems and the posterior distribution, $p(\theta|\Psi)$, can be used to estimate the population-level parameters. But, in instances where monitoring data are limited – as is typically the case in clinical trials – the two-step inference approach fails to capture the uncertainty of the population-level estimates appropriately and a more advanced inference approach needs to be used. One of those advanced inference approaches is NLME inference (Lavielle 2014).

To illustrate the challenges of estimating the parameters of population models from inferred individual-level parameters, Figure 2.18 shows inference results of the warfarin model from three simulated datasets, each dataset containing varying numbers of monitoring measurements from varying numbers of individuals. For illustrative purposes, the figure focuses on a subset of the model's parameters – the volume of distribution,

v, the elimination rate, k_e, and the corresponding population-level parameters: μ_v, σ_v, μ_{k_e}, and σ_{k_e}. For each dataset, the left panel of the figure shows the individual-level parameter estimates, and the right panel shows the population-level parameter estimates. The monitoring measurements, from which the individual-level parameters are inferred, are not shown in the figure. The population-level parameters used to simulated the data are: $\log \mu_v = 2.6$; $\sigma_v = 0.1$; $\log \mu_{k_e} = -3.7$; and $\sigma_{k_e} = 0.1$.

We will, for now, focus on the inference results obtained from the two datasets with many monitoring measurements per individual (see black box and red box in the left panel). For those datasets, the parametric uncertainty of the individual-level parameter estimates is negligible and the values of the population-level parameters can be inferred using $p(\theta|\Psi)$. The resulting posterior distributions are shown in the right panel – one marginal distribution for each population-level parameter (black-box dataset: black; and red-box dataset: red). The prior distribution, used to inform the inference results, is fixed for both estimations and is illustrated using dashed lines in the right panel.

The figure shows that both posterior distributions support the data-generating parameter values, illustrating the consistency of the inference approach with the data-generating process. The area in parameter space supported by the black posterior distribution is more confined than the area supported by the red posterior distribution. This shows that the black-box dataset is more informative about the data-generating parameter values than the red-box dataset – an observation in agreement with the results from Figure 2.13, where we found that a large number of monitoring measurements is more informative than a small number of monitoring measurements. Here, the number of measured individuals plays the role of the number of monitoring measurements.

Without individual-level parametric uncertainty, this would conclude this section on estimating the parameter values of population models. However, inference from typical clinical trial data often results in uncertain individual-level estimates, as monitoring capabilities in clinical practice are limited (see e.g. Figure 2.13). Uncertain parameter estimates make it difficult to use the above inference approach to estimate the population-level parameters, since $p(\theta|\Psi)$ requires conditioning on known parameter values. While it is technically possible to meet this requirement by conditioning on representative sets

of parameter values from each posterior distribution (e.g. the means of the posterior distributions), it would ignore the level of uncertainty of the individual-level estimates and result in an overestimation of the information contained in the monitoring measurements. NLME inference overcomes this limitation.

In NLME inference, the NLME model (Equation 2.31) is used to simultaneously estimate the individual-level and population-level parameters from the monitoring measurements, thereby propagating any uncertainty in individual-level estimates to the population-level estimates. This is done using the model's joined probability distribution for the value of a treatment response measurement and the values of the individual-level parameters, $p(y, \psi|t, r, \theta) = p(y|t, r, \psi)\, p(\psi|\theta)$.[18] More specifically, we can use $p(y, \psi|t, r, \theta)$ to quantify both, the likelihood of an individual's monitoring measurements under the measurement model, and the likelihood of the model's parameter values under the population model, $p(\mathcal{D}_i, \psi_i|\theta) = p(\mathcal{D}_i|\psi_i)\, p(\psi_i|\theta)$.[19] In this construction, the likelihood naturally factorises into a product with two contributions: 1. the measurement likelihood (Equation 2.14); and 2. the parameter likelihood, $p(\psi_i|\theta)$.

[18] Note that the joined probability distribution, $p(y, \psi|t, r, \theta)$, is implicitly defined in Equation 2.31. To show that $p(y, \psi|t, r, \theta)$ and Equation 2.31 are, in fact, equivalent, we can marginalise $p(y, \psi|t, r, \theta)$ over ψ

$$p(y|t, r, \theta) = \int \mathrm{d}\psi\, p(y, \psi|t, r, \theta) = \int \mathrm{d}\psi\, p(y|t, r, \psi)\, p(\psi|\theta). \tag{2.38}$$

This recovers Equation 2.31. The left equality holds by definition of marginal distributions, and the right equality holds by definition of the joined distribution in terms of the measurement model and the population model, $p(y, \psi|t, r, \theta) = p(y|t, r, \psi)\, p(\psi|\theta)$.

[19] Note that $p(\mathcal{D}_i, \psi_i|\theta)$ is a shorthand notation for the joined likelihood of an individual's measurements, $\mathcal{D}_i = \{(y_{i1}, t_{i1}), \ldots, (y_{in}, t_{in_i}), r_i\}$, and an individual's model parameters, ψ_i,

$$p(\mathcal{D}_i, \psi_i|\theta) = p(y_{i1}, \ldots, y_{in_i}, \psi_i|t_1, \ldots, t_{n_i}, r_i, \theta). \tag{2.39}$$

At this point, we make no assumptions about the model structure. Assuming that the NLME model describes the data-generating process, we can split this expression into a product of two likelihoods (one for the measurements; and one for the model parameters), as the measurements depend in the model only indirectly on the population-level parameters

$$p(\mathcal{D}_i, \psi_i|\theta) = p(y_{i1}, \ldots, y_{in_i}, |t_1, \ldots, t_{n_i}, r_i, \psi_i)\, p(\psi_i|\theta). \tag{2.40}$$

Comparing this expression to Equation 2.15, we find that its first term is the measurement likelihood, $p(\mathcal{D}_i|\psi_i)$,

$$p(\mathcal{D}_i, \psi_i|\theta) = p(\mathcal{D}_i|\psi_i)\, p(\psi_i|\theta), \tag{2.41}$$

recovering the in-text expression.

For treatment response measurements across N individuals, this likelihood can be straightforwardly extended to

$$\log p(\mathcal{D}, \Psi|\theta) = \sum_{i=1}^{N} \log p(\mathcal{D}_i|\psi_i) + \log p(\Psi|\theta), \tag{2.42}$$

by multiplying the individual measurement likelihoods and accounting for the parameter likelihood. We use $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$ to denote the measurements across the individuals. For iid individual-level parameters, $p(\Psi|\theta)$ factorises (see Equation 2.34) and $p(\mathcal{D}, \Psi|\theta)$ reduces to the product of the individual likelihoods, $p(\mathcal{D}_i, \psi_i|\theta)$.

To conclude the definition of the NLME inference approach, we translate the likelihood in Equation 2.42 into a joined posterior distribution for the individual-level and population-level parameters using Bayes' rule

$$\log p(\theta, \Psi|\mathcal{D}) = \log p(\mathcal{D}, \Psi|\theta) + \log p(\theta) + \text{constant}, \tag{2.43}$$

where the model evidence is, again, denoted as a parameter-independent constant term. The NLME posterior distribution can be numerically estimated using MCMC sampling by repeatedly evaluating Equation 2.43 up to its constant term (see subsection 2.2.6). Example inference results using NLME inference are illustrated in Figure 2.18 (see blue-box and blue distribution).

The left panel of the figure shows the individual-level posterior distributions, obtained by marginalising the NLME posterior distribution over the population-level parameters, $p(\Psi|\mathcal{D}) = \int \mathrm{d}\theta\, p(\theta, \Psi|\mathcal{D})$.[20] The right panel shows the population-level posterior distribution, obtained similarly by marginalising the NLME posterior distribution over the individual-level parameters, $p(\theta|\mathcal{D}) = \int \mathrm{d}\Psi\, p(\theta, \Psi|\mathcal{D})$. The figure shows that the posterior distribution (blue) is less confined than the black distribution, indicating that the blue-box dataset contains less information about the population-level parameters than the black-box dataset. This is in agreement with the earlier finding that measurements from more individuals are more informative than measurements from fewer individuals.

[20] Note that, after MCMC sampling, marginalisation requires no additional computation as MCMC sampling from the joined posterior distribution (Equation 2.43) results in pairs of samples: $\{(\theta^1, \Psi^1), \ldots, (\theta^S, \Psi^S)\}$, where S denotes the number of samples. Each subset of samples, $\{\theta^1, \ldots, \theta^S\}$ and $\{\Psi^1, \ldots, \Psi^S\}$, is distributed according to its respective marginal distribution. As a result, marginalisation is possible by simply considering only the subset of interest.

However, the comparison of the NLME posterior distribution with the red posterior distribution also shows that the number of measured individuals is not the only factor influencing the population-level information – the number of measurements per individual determine the available information as well. This is exemplified by the comparable level of parametric uncertainty of the blue distribution and the red distribution. Here, the red distribution is inferred from a dataset containing measurements from much fewer individuals (see red-box). However, for these individuals many more monitoring measurements are available, such that the uncertainty of the individual-level estimates is negligible. This illustrates that the parametric uncertainty of population-level parameter values is determined by two contributing factors: 1. the number of measured individuals; and 2. the number of measurements per individual (see green arrows in Figure 2.18).

Conceptually, the contributions to the population-level uncertainty can be understood by interpreting the posterior distribution, $p(\theta|\mathcal{D})$, as a superposition of many posterior distributions – one posterior distribution, $p(\theta|\Psi)$, for each set of parameter values supported by the individual-level posterior distribution, $p(\Psi|\mathcal{D})$.[21] This makes it intuitively clear that the population-level posterior distribution displays large uncertainty when also its constituent posterior distributions display large uncertainty, which will generally be the case when measurements from too few individuals are available. When measurements from many individuals are available, the constituent posterior distributions become more confined. However, the population-level posterior distribution may nevertheless remain uncertain about the data-generating parameter values in cases where the individual-level parameters (and thus the areas supported by the constituent posterior distributions) vary substantially. This may be the case when too few measurements per individual are available and the individual-level posterior distribution, $p(\Psi|\mathcal{D})$, remains uncertain about the data-generating parameter values for the measured individuals.

[21] For clarity: $p(\theta|\mathcal{D}) = \int \mathrm{d}\Psi\, p(\theta|\Psi)\, p(\Psi|\mathcal{D})$. To derive this equality, we first use the arithmetics of conditioning to express the joined posterior distribution in the definition of the posterior distribution, $p(\theta|\mathcal{D}) = \int \mathrm{d}\Psi\, p(\theta, \Psi|\mathcal{D})$, in terms of two conditional distributions: $p(\theta, \Psi|\mathcal{D}) = p(\theta|\Psi, \mathcal{D})\, p(\Psi|\mathcal{D})$. We then use the result from Equation 2.34 and its corresponding posterior distribution to show that the distribution of the population-level parameters is independent of the data, once conditioned on the individual-level parameters: $p(\theta|\Psi, \mathcal{D}) = p(\theta|\Psi)$.

To conclude this section, we highlight two other notable properties of NLME inference: 1. the NLME posterior distribution (Equation 2.43) can also be used to estimate the model parameters of covariate population models when covariate information of measured individuals is available. To this end, the covariate population model replaces the population model in the hierarchical likelihood

$$\log p(\mathcal{D}, \Psi|\theta) = \sum_{i=1}^{N} \log p(\mathcal{D}_i|\psi_i) + \sum_{i=1}^{N} \log p(\psi_i|\theta, \chi_i). \tag{2.44}$$

Here, we use $\mathcal{D}$ to denote both, the monitoring data and the covariate information of the measured individuals, $\mathcal{D} = \{(\mathcal{D}_1, \chi_1), \ldots, (\mathcal{D}_N, \chi_N)\}$. This likelihood can be translated into a posterior distribution and estimated using MCMC sampling analogously to Equation 2.43;

2. NLME inference results tend to underestimate treatment response variability – an important property to be aware of when using population models as prior distributions for MIPD. At its core, this underestimation is a consequence of entropic pressure[22] exerted by population models on individual-level estimates during hierarchical inference when individual-level parameter estimates are uncertain.

To see this, note that typical clinical data tend to result in individual-level estimates with considerable uncertainty (see e.g. Figure 2.13). This uncertainty provides freedom to consider different sets of parameter values (within the limits of the individual-level posterior distributions) for the estimation of population-level parameters during hierarchical inference (Equation 2.43). This, by itself, does not lead to the underestimation of IIV and simply propagates the uncertainty of the individual-level parameters to the population-level estimates. However, in a hierarchical inference approach, the population model serves as a prior distribution for the inference of individual-level parameters, providing a mechanism for population-level estimates to inform individual-level estimates (Equation 2.42). Changes in the population-level estimates can therefore result in changes in the individual-level posterior distributions, making biases of the individual-level estimates towards low IIV configurations principally possible, if the population model

[22] We use entropic pressure to refer to the tendency of population models to bias individual-level parameter estimates towards low entropy configurations during hierarchical inference. This use of the term 'entropic pressure' is different to its use in physics, where entropic pressure is used to describe the emergent phenomena of systems maximising their entropy over time (Klein et al. 2011).

were to favour such configurations. And in fact, all popular population models[23] do favour low IIV configurations of the individual-level estimates during hierarchical inference. This is because the likelihood of the individual-level parameters under the population model (2nd term in Equation 2.42) is maximised when the entropy of the population model is minimal, i.e. when the inter-individual variability under the population model is minimal.[24] As a result, the population model biases the individual-level estimates towards low entropy configurations which in turn biases population models to have lower entropy than the data-generating process – a mechanism we term entropic pressure.

The tendency of NLME inference to underestimate population variation is known as shrinkage in the literature (Yuan et al. 2019). An example of shrinkage is already present in Figure 2.18, where the posterior distribution of σ_v assigns the majority of its probability mass to values smaller than the data-generating value, $\sigma_v = 0.1$ (see blue posterior distribution). Here the degree of shrinkage is relatively minor, because the individual-level likelihoods and the population-level prior distribution counter the entropic pressure. However, in the extreme, when some individual-level parameter values are not informed by the available treatment response measurements and the population-level prior distribution permits zero-entropy population models, the entropic pressure can collapse the corresponding dimension of the population model, resulting in vanishing IIV of the inter-individual variability irrespective of the true variance of the data-generating process (Betancourt 2017).

[23] To be more precise: All population models whose entropy depends on its model parameters will exert entropic pressure on the individual-level parameters during hierarchical inference. In PKPD modelling, the most popular population models are a normal distribution and a lognormal distribution, whose entropy depends on their parameters.

[24] For clarity: The log-likelihood of N individual-level parameters $\Psi = \{\psi_1, \ldots, \psi_N\}$ under the population model is $\log p(\Psi|\theta) = \sum_{i=1}^{N} \log p(\psi_i|\theta)$ (see Equation 2.34), and the entropy of a random individual-level parameter ψ under the population model is $S = -\int \mathrm{d}\psi\, p(\psi|\theta) \log p(\psi|\theta)$. The entropy quantifies the lack of information about the value of a yet-to-be-drawn random sample ψ from the population model. The minimal entropy is zero, in which case we know the value of ψ before sampling it, i.e. there is no IIV, all individuals have the same model parameters and display the same treatment response. The larger the entropy, the more uncertain we are about the values of future random samples from the population model, implying greater levels of treatment responses variability across individuals. To see that maximal likelihood implies minimal entropy, note that the log-likelihood of the individual-level parameters under the population model approximates the population-level entropy up to a constant factor: $S \approx -\sum_{i=1}^{N} \log p(\psi_i|\theta)/N$. As a result, the entropy of the population model is minimised when the likelihood of the individual-level parameters under the population model is maximised.

In the context of MIPD, shrinkage of population-variance estimates can mean that parametric uncertainty may be underestimated during the parameter individualisation, and that parameter estimates may be biased away from the data-generating parameter values. It is therefore important to reduce shrinkage as much as possible prior to MIPD. For a fixed number of measured individuals, this can be achieved by monitoring individuals more closely, reducing the uncertainty of their individual-level estimates. However, the minimum level of shrinkage is determined by the number of measured individuals. Consequently, shrinkage can only be avoided by collecting measurements across sufficiently many individuals.[25] In fact, we find that for a large number of measured individuals, as little as one monitoring measurement per individual can be sufficient to avoid the underestimation of IIV, opening new opportunities for clinical applications of NLME modelling (see Chapter 4 for more details).

In summary, parameters of NLME models can be estimated from inter-individual treatment response measurements using NLME inference, ensuring the consistency of simulations with observations of clinical IIV. In the context of MIPD, this helps to utilise existing information, such as clinical data, to define informative prior distributions and to make MIPD more data-efficient in clinical practice. The uncertainty of the NLME model parameters is determined by two factors: 1. the number of measured individuals; and 2. the level of uncertainty of the individual-level parameter estimates. For typical clinical trial datasets, where the number of measured individuals and the number of measurements per individual are limited, NLME inference suffers from shrinkage, resulting in an underestimation of IIV. This needs to be considered during MIPD.

2.2.9 Structural uncertainty & Bayesian inference

This concludes the review of parametric uncertainty. In this section, we will discuss a so-far-neglected source of predictive uncertainty – structural uncertainty. Structural

[25] An important caveat to this statement is the varying sensitivity of model outputs to parameter values. If a model output is not sensitive to the values of some of its parameters, for example because the output is not linked to the parameters or the output's sensitivity to the parameters varies in time, measurements of an observable linked to this model output can fail to contain information about said model parameters. In those instances, shrinkage can lead to population-level estimates with vanishing population variance, irrespective of the number of measurements and the number of measured individuals, and regardless of whether the considered parameters display IIV in reality or not.

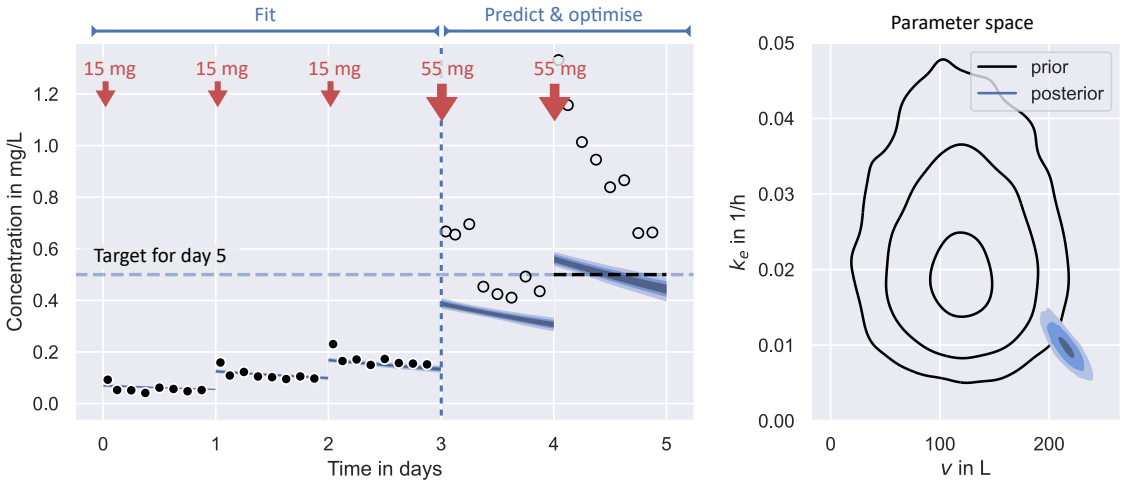


Figure 2.19: Predictive errors beyond parametric uncertainty. The figure shows the attempt to individualise the dosing regimen of a simulated patient treated with imirestat. The left panel shows measurements of the imirestat blood concentration using scatter points: filled scatter points are used for the inference of the model parameters; hollow scatter points are not used for the inference, but indicate the treatment response of the patient after the dose adjustment. The model fit and the model prediction are illustrated using shades of blue, indicating the bulk 67 %, 90 %, and 99 % of the posterior predictive distribution without measurement noise. Dose events are illustrated using red arrows. The right panel shows the posterior distribution associated to the parameter inference after the 3rd treatment day.

uncertainty and parametric uncertainty, together, provide a complete description of a model's predictive uncertainty (Draper 1995). We will begin the section by motivating structural uncertainty using an example from diabetes treatment. We will then outline how structural uncertainty can be quantified within a Bayesian inference framework and discuss practical challenges.

To motivate structural uncertainty, we consider the treatment of diabetes patients with imirestat. Imirestat is an aldose reductase inhibitor which can be used to prevent eye and nerve damage in diabetes patients with elevated glucose metabolism (Várkonyi et al. 2008). We choose imirestat as a use case for structural uncertainty because of its complicated pharmacokinetics which cannot be described using the models that we have discussed so far (Mager and Jusko 2001). In this illustration, we will nevertheless attempt to use a 1-compartment PK model to individualise a patient's imirestat dosing regimen.

To this end, we simulate measurements of a patient's treatment response using Mager et al's model of the imirestat pharmacokinetics (Mager and Jusko 2001) and use

those measurements to inform the dosing regimen individualisation. We fix the initial dosing regimen to three bolus administrations of 15 mg and measure the treatment response 8 times each day. After the initial three treatment days, we infer the parameters of the 1-compartment PK model from the measurements and use the resulting posterior predictive distribution to predict an individualised dosing regimen.[26] For illustrative purposes, we target an imirestat blood concentration level of 0.5 mg/L for the 5th treatment day. The results of the MIPD attempt are illustrated in Figure 2.19.

The left panel of the figure shows the blood concentration measurements using scatter points – filled scatter points indicate measurements used for the parameter inference; hollow scatter points are not used for the inference, but indicate the treatment response to the 'optimised' dosing regimen. The model fit and the model prediction are illustrated using shades of blue, indicating the bulk 67 %, 90 %, and 99 % of the noise-free posterior predictive distribution. The corresponding posterior distribution is illustrated in the right panel of Figure 2.19.

The figure shows that the initial treatment response measurements of the patient are well described by the 1-compartment PK model, with measurements leaving only a small level of residual parametric uncertainty relative to the prior distribution. This small parametric uncertainty translates into a low level of predictive uncertainty, suggesting that the model may be used for the prediction of optimised dosing regimens without substantial errors. However, when administering an 'optimised' dosing regimen to the simulated patient (daily doses of 55 mg), we find that only the model's predicted treatment response realises the targeted concentration level (see shades of blue in the left panel), while the patient's treatment response measurements overshoot the target value and display values of up to three times the targeted concentration level (see hollow scatter points).

For many use cases, such a three-fold overshoot can mean the difference between a treatment with no unintended side effects and a treatment with serious adverse events (see e.g. the clinical warfarin data in Chapter 1). The mismatch between the estimated predictive uncertainty and the actual treatment response of the simulated patient illustrates that parametric uncertainty alone does not provide a complete picture

[26] For simplicity, we use the maximum a posterior parameter values to predict the dosing regimen.

of a model's predictive uncertainty – the Bayesian inference framework introduced in subsection 2.2.5 fails to capture important sources of uncertainty.

The reason for this mismatch is intuitively clear from the outset of the use case: the 1-compartment PK model does not describe the pharmacokinetics of imirestat – i.e., the model is misspecified with respect to the data-generating process. So, why would we expect the model to predict meaningful dosing regimens? In the context of MIPD, the fact that a model may produce wrong predictions is not surprising – after all, models are only approximate descriptions of a patient's treatment response. It is more concerning that the predictions in Figure 2.19 do not portray an accurate picture of the extent of the residual predictive uncertainty – how can we trust a model to inform treatment decisions when the model's predictions are wrong and unaware of their uncertainty?

To understand why the current framework fails to capture the predictive uncertainty, and how we can begin to remedy this short-coming, let us review the so-far-developed Bayesian inference approach. We began the construction of the inference approach in subsection 2.2.4, where we defined the likelihood of treatment response measurements under a model of the data-generating process. This model was composed of a mechanistic model and a measurement noise model (see Equation 2.14). We then used Bayes' rule to translate this likelihood into a distribution of parameter values consistent with the measurements and prior knowledge – the posterior distribution (see Equation 2.22) – and used this posterior distribution to predict a range of treatment responses consistent with the available information – the posterior predictive distribution (see Equation 2.23). We constructed the inference approach in this way because we realised that noisy measurements do not uniquely define the data-generating parameters, and choosing any one set of parameter values permitted predictive errors.

But what does it mean for parameters to be 'data-generating' when the model structure is misspecified with respect to the data-generating process? To answer this question, we illustrate in Figure 2.20 the fit of the 1-compartment PK model to all available imirestat concentration measurements after the dosing regimen adjustment. The left panel shows that the model fit changes substantially from its fit after 3 treatment days, overall displaying a poor ability to describe the treatment response measurements.

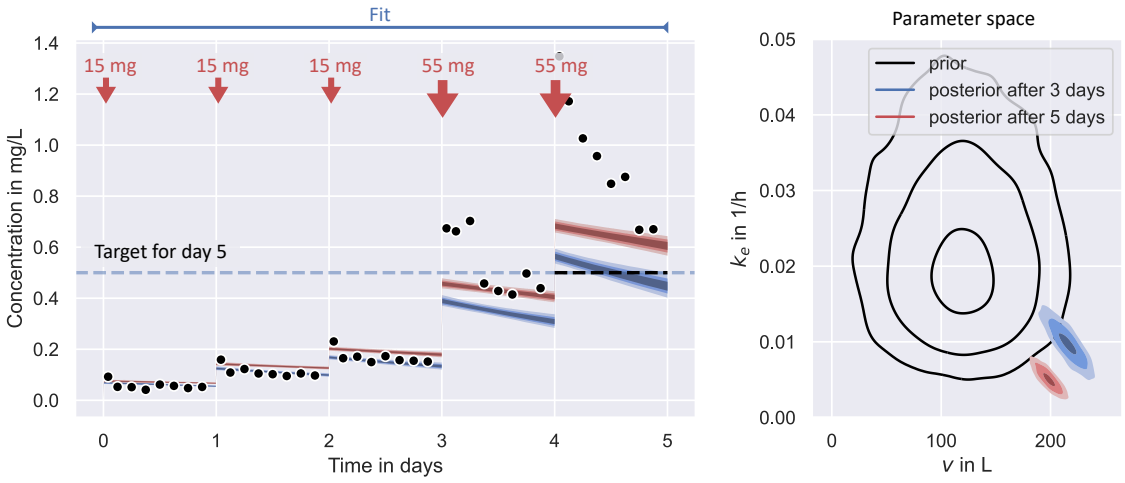


Figure 2.20: Breakdown of parametric uncertainty. The figure shows an extension of the MIPD experiment from Figure 2.19. This time, the model is also fit to the measurements after the dosing regimen adjustment (see red shaded areas). The fit from Figure 2.19 is shown for reference in blue.

Despite this lower quality fit to the data, the parametric uncertainty of the model appears to be lower after 5 treatment days than after 3 treatment days (see right panel). This demonstrates that the concept of parametric uncertainty, and with it the concept of data-generating parameter values, breaks down when the model structure is misspecified with respect to the data-generating process (at least for the purpose of estimating predictive uncertainty). This is because the posterior distribution does not assess how well a model captures the data-generating process – it assesses only which parameter values are consistent with the measurements and the prior knowledge, assuming that the chosen model structure is correct. As a result, the posterior distribution will converge for large numbers of measurements to the set of parameter values that best describes the available data, even if the overall quality of that description is poor.[27]

In many ways, the predictive errors in Figures 2.19 and 2.20 are not different from predictive errors induced by estimation errors, albeit not the result of erroneous parameter

[27] Note that in contrast to Figure 2.13, the update of the posterior distribution in Figure 2.20 is no longer consistent, in the sense that the posterior distribution inferred from less information (blue posterior) does not support the area in parameter space consistent with the additional information (see red posterior). However, for parametric uncertainty to be useful for predictive uncertainty estimation, the area in parameter space supported by posterior distributions should only shrink as more information becomes available (see e.g. Figure 2.13) and not jump to previously unsupported areas of parameter space. This behaviour on its own is an indication for model misspecfication.

estimation. In fact, all parameter values supported by the posterior distributions are, indeed, consistent with the treatment response measurements – they just do not translate into correct predictions because we have chosen the wrong model structure to make them. Choosing a model structure for prediction, despite inconclusive information about the correct model structure to use, is analogous to choosing a set of model parameters, despite inconclusive information about the correct values to use. Such a lack of conclusive structural information has gone unnoticed so far, because we have always presumed to know the model structure of the data-generating process. However, in clinical practice, we are often missing a complete picture of the pharmacological mechanisms of a drug. As a result, the uncertainty about the appropriate model structure to use for estimation and prediction can be substantial. This structural uncertainty needs to be captured, in addition to the parametric uncertainty, in order to provide a complete picture of predictive uncertainties.

Similar to parametric uncertainty, we can quantify and update structural uncertainty using Bayesian inference. In order to do this, we make the dependence of the measurement model (Equation 2.12) on the choice of the model structure explicit

$$p(y|t, r, \psi) \quad \longrightarrow \quad p(y|t, r, \psi, m). \tag{2.45}$$

Here, m denotes the model structure. We will clarify how exactly we use m to denote model structures below.

With this updated notation, we can re-derive the posterior distribution (Equation 2.22) using Bayes' rule, finding that its expression remains conceptually unchanged

$$p(\psi|\mathcal{D}, m) = \frac{p(\mathcal{D}|\psi, m)\, p(\psi|m)}{p(\mathcal{D}|m)}. \tag{2.46}$$

The only difference between this expression and Equation 2.22 is the explicit conditional dependence of all terms on the model structure. Intuitively, this conditional dependence makes sense, as statements about a model's parametric uncertainty can only be made after the model structure has been defined.[28]

[28] Note that the model parameters, ψ, depend on the model structure, including their number and whether they are discrete or continuous. For ease of notation, we will leave this dependence implicit.

However, perhaps, more interestingly, the updated notation allows us to use Bayes' rule also in a different way,[29] deriving a second distribution focused on the model structure

$$p(m|\mathcal{D}) = \frac{p(\mathcal{D}|m)\,p(m)}{p(\mathcal{D})}. \tag{2.48}$$

This distribution is structurally identical to the posterior distributions we have seen before (see e.g. Equation 2.22), permitting the interpretation of $p(m)$ as the prior distribution, and of $p(m|\mathcal{D})$ as the posterior distribution of the model structure (Draper 1995). As a result, $p(m)$ can be used to quantify prior knowledge of suitable model structures before learning from the data, while $p(m|\mathcal{D})$ can be used to quantify the consistency of those model structures with all of the available information. For clarity, we will refer to $p(\psi|\mathcal{D}, m)$ from now on as parametric posterior distribution and to $p(m|\mathcal{D})$ as structural posterior distribution, unless otherwise clear from the context.

Note that, similar to the parametric posterior distribution, the structural posterior distribution assigns non-zero probabilities only to those model structures supported by the prior distribution. Excluding model structures prematurely can therefore result in incorrect estimates of the predictive uncertainty.

Note also that the parametric posterior distribution and the structural posterior distribution are closely related, with the normalising factor of the parametric posterior distribution, $p(\mathcal{D}|m) = \int \mathrm{d}\psi\, p(\mathcal{D}|\psi, m)\, p(\psi|m)$, assuming the role of the measurement likelihood for the structural posterior distribution (see Equations 2.46 and 2.48). This link gives rise to technical and conceptual challenges which we will discuss in the next section, subsection 2.2.10.

[29] We can derive the expression for the structural posterior distribution by first using Bayes' rule to translate the measurement likelihood into a joint posterior distribution

$$p(\psi, m|\mathcal{D}) = \frac{p(\mathcal{D}|\psi, m)\,p(\psi, m)}{p(\mathcal{D})}. \tag{2.47}$$

Note that this expression follows from Bayes' rule when considering ψ and m as one random variable, e.g. $X = (\psi, m)$. From this expression we can derive the structural posterior distribution in two steps: 1. we use the arithmetics of conditional probabilities to split both, the joint posterior distribution and the joint prior distribution, into two separate distributions for the parameters and the model structure, $p(\psi, m|\mathcal{D}) = p(\psi|\mathcal{D}, m)\,p(m|\mathcal{D})$ and $p(\psi, m) = p(\psi|m)\,p(m)$; and 2. we replace the parametric posterior distribution by its expression in terms of the likelihood and the prior (Equation 2.46). Plugging all expression into the joint posterior distribution yields the expression for the structural posterior distribution.

With the expressions for parametric uncertainty and structural uncertainty in place, we can define an updated version of the posterior predictive distribution (Equation 2.23)

$$p(y|\mathcal{D}) = \sum_{m=1}^{M} \int \mathrm{d}\psi \, p(y|\psi, m) \, p(\psi, m|\mathcal{D}), \tag{2.49}$$

– a version which provides a complete picture of a model's predictive uncertainty. Here, $p(\psi, m|\mathcal{D})$ denotes the joint posterior distribution, composed of the parametric posterior distribution and the structural posterior distribution, $p(\psi, m|\mathcal{D}) = p(\psi|\mathcal{D}, m) \, p(m|\mathcal{D})$. M denotes the number of candidate models supported by the structural prior distribution. We choose to enumerate different model structures using integer values, $m \in \{1, \ldots, M\}$, – one integer for each candidate model – because model structures do not vary continuously in the context of PKPD modelling. This makes any unique labelling of the structures equally valid. The order of the enumeration is not important.

To develop an intuition for this extended framework for predictive uncertainty estimation, we illustrate an example pool of candidate models for the imirestat use case in Figure 2.21. The first column shows the first candidate model ($m = 1$) – the 1-compartment PK model with lognormal measurement noise that we have used for the dosing regimen individualisation in Figure 2.19.[30] The second column shows the second candidate model ($m = 2$) – a variant of the first model where the measurement noise model is replaced by a normal noise model.[31] While it is less common to consider different measurement noise models in practice, we include model $m = 2$ as an example for structural differences due to differences in non-mechanistic model components. The third candidate model ($m = 3$) is a 2-compartment PK model with normal measurement noise – a variant of Teorell's PK model (Equation 2.1) without metabolic processes in the peripheral compartment. The final candidate model ($m = 4$) has a mechanistic

[30] For clarity, $p(y \mid t, r, \psi, m{=}1) = \mathrm{LN}(y \mid \bar{y}(t, r, \psi, m{=}1), \sigma)$, where

$$\mathrm{LN}(y|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \frac{1}{\mu} \mathrm{e}^{-(\log y - \log \mu)^2 / 2\sigma^2}, \tag{2.50}$$

denotes the probability density of the lognormal measurement noise model, and $\bar{y}(t, r, \psi, m{=}1)$ denotes the output of the 1-compartment PK model, $c(t, r, \psi)$, defined by $\mathrm{d}c/\mathrm{d}t = -k_e c + r(t)/v$ with $c(t = 0) = 0$.

[31] For clarity, $p(y \mid t, r, \psi, m{=}2) = \mathcal{N}(y \mid \bar{y}(t, r, \psi, m{=}2), \sigma^2)$, where, in this case, the mechanistic model is the same as for the first candidate model, $\bar{y}(t, r, \psi, m{=}2) = \bar{y}(t, r, \psi, m{=}1)$.

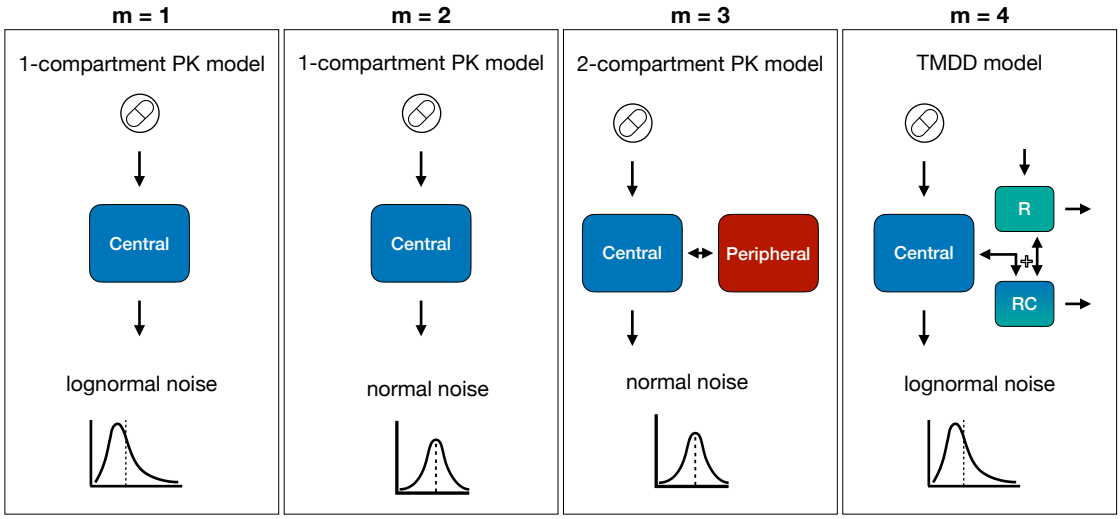


Figure 2.21: Pool of candidate models. The figure shows candidate models for the imirestat use case. Each candidate model consists of a mechanistic model and a measurement noise model.

model that we have not yet considered in this thesis.[32] This model is a target-mediated drug disposition (TMDD) model, where the pharmacokinetics of the drug are influenced by its target (Mager and Jusko 2001). In particular, the TMDD model captures temporary complex formations between a drug and its target, resulting in changes in its clearance behaviour, especially when the complex elimination rate is different from the elimination rate of the free drug.

Formally, the TMDD model is defined by a set of differential equations for the drug concentration, c, the target receptor concentration, c_r, and the drug-target complex concentration, c_{rc},

$$\begin{aligned}
\frac{\mathrm{d}c}{\mathrm{d}t} &= -k_e c - k_f c\, c_r + k_{ds} c_{rc} + r(t)/v, & c(t=0,\psi) &= 0,\\
\frac{\mathrm{d}c_r}{\mathrm{d}t} &= k_s - k_{dg,r} c_r - k_f c\, c_r + k_{ds} c_{rc}, & c_r(t=0,\psi) &= \frac{k_s}{k_{dg,r}},\\
\frac{\mathrm{d}c_{rc}}{\mathrm{d}t} &= -k_{dg,rc} c_{rc} + k_f c\, c_r - k_{ds} c_{rc}, & c_{rc}(t=0,\psi) &= 0.
\end{aligned} \tag{2.51}$$

Here, k_e and v denote the elimination rate and the volume of distribution of the drug. k_s and $k_{dg,r}$ denote the synthesis rate and the degradation rate of the target receptor. $k_{dg,rc}$ denotes the degradation rate of the complex, and k_f and k_{ds} denote the formation

[32] For clarity, $p(y \mid t, r, \psi, m{=}4) = \mathrm{LN}(y \mid \bar{y}(t, r, \psi, m{=}4), \sigma)$, where $\bar{y}(t, r, \psi, m{=}4) = c(t, r, \psi)$ defined in Equation 2.51.

rate and the disassociation rate of the complex. In total, the TMDD model has 7 parameters, $\psi = (k_e, v, k_s, k_{dg,r}, k_{dg,rc}, k_f, k_{ds})$.

Starting with this pool of candidate models, we can re-run the imirestat MIPD experiment from Figure 2.19, this time using the extended framework for predictive uncertainty estimation (Equation 2.49). For illustrative purposes, we fix all but two parameters across the models – the drug elimination rate, k_e, and the drug volume of distribution, v, remain free.[33] We then infer the parametric and structural posterior distributions using two different datasets: 1. using only the measurements from the first treatment day; and 2. using only the measurements from the first three treatment days. The remaining measurements serve as a visual check for the predictive uncertainty estimation.

Figure 2.22 shows the result of the experiment. The treatment response measurements and the model fits are shown in the left panel. Each fit is illustrated by the 99 % bulk probability region of its noise-free posterior predictive distribution, $p(y|\mathcal{D}, m)$ – one posterior predictive distribution for each candidate model ($m = 1$: purple; $m = 2$: grey; $m = 3$: red; $m = 4$: blue). The average of these distributions over the structural posterior distribution recovers the posterior predictive distribution in Equation 2.49. We do not show this average in the figure. The structural prior distribution and the estimates of the structural posterior distribution are shown in the right panel.[34] The parametric uncertainty is estimated but not illustrated in the figure for visual purposes.

[33] We reduce the dimension of the posterior distribution for computational reasons which we will further discuss later in this section.

[34] The estimation of the structural posterior distribution requires the evaluation of both, the model evidence, $p(\mathcal{D}|m)$, and the structural prior distribution. The denominator of Equation 2.48 does not need to be explicitly calculated, as the posterior distribution is normalised

$$p(m|\mathcal{D}) = \frac{p(\mathcal{D}|m)\, p(m)}{\sum_m p(\mathcal{D}|m)\, p(m)}. \tag{2.52}$$

The evaluation of the model evidence requires solving an integral, $p(\mathcal{D}|m) = \int \mathrm{d}\psi\, p(\mathcal{D}|\psi, m)\, p(\psi|m)$. Solving this integral in high dimensions is challenging, for reasons we will outline in subsection 2.2.10. However, in this use case, the candidate models have only two free model parameters which makes it possible to estimate the model evidences using a simple Monte Carlo integration approach

$$p(\mathcal{D}|m) \approx \frac{1}{S} \sum_s p(\mathcal{D}|\psi_s, m). \tag{2.53}$$

Here, ψ_s denote samples from the prior distribution.

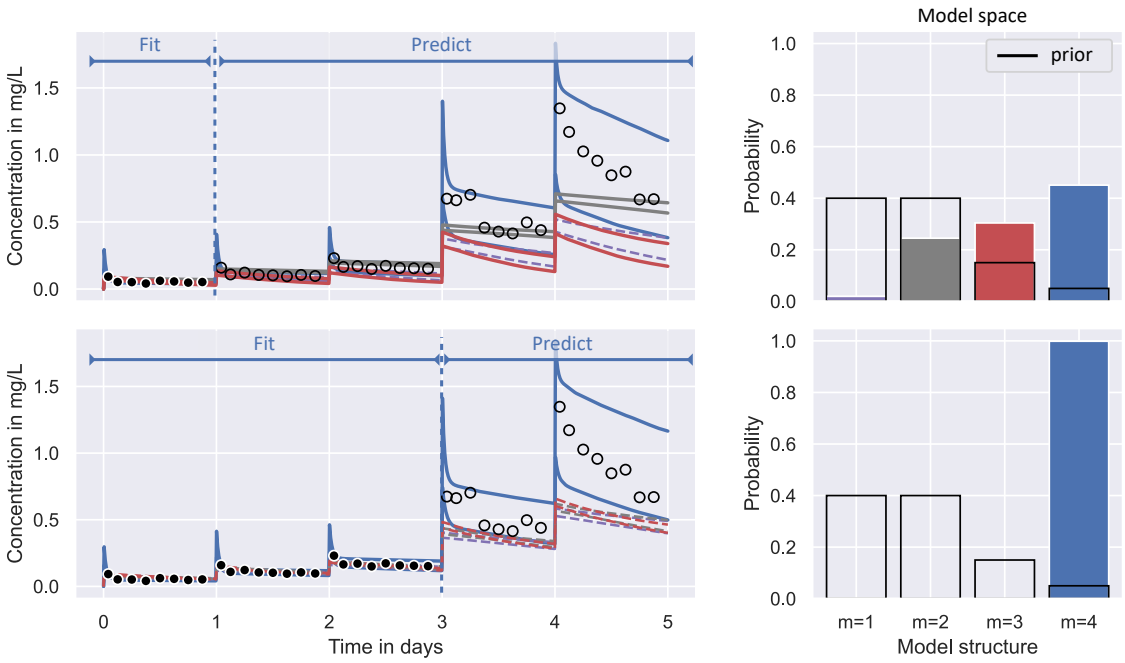


Figure 2.22: Estimation of structural uncertainty using Bayesian inference. The figure shows the treatment response prediction for a simulated patient treated with imirestat. The left panel shows measurements of the imirestat blood concentration using scatter points: filled scatter points are used for the inference of the model parameters; hollow scatter points are only used to indicate the treatment response of the patient at later times. The model fits and the model predictions are illustrated using the bulk 99 % of the posterior predictive distributions without measurement noise ($m = 1$: purple; $m = 2$: grey; $m = 3$: red; $m = 4$: blue). Dose are administered on day 0, 1, 2, 3, and 4. The right panel shows the structural posterior distributions associated to inference after the first treatment day (row 1) and after the third treatment day (row 2).

The figure shows that the extended framework for predictive uncertainty estimation is able to provide a faithful representation of residual uncertainties. Both after the first treatment day (see row 1) and after the third treatment day (see row 2), all candidate models are able to fit the patient's treatment response measurements, with visually negligible differences. Despite this agreement, the models' predictions for the following treatment days differ substantially, illustrating that structural uncertainty can lead to a large degree of predictive uncertainty. However, the available information – as indicated by the structural posterior distributions – does not support all model structures equally (see right panel). For example, after the first treatment day, the structural posterior distribution assigns the first candidate model with negligible probability mass (see row 1), indicating that its predictions are unlikely to capture the patient's future

treatment response – an insight that we failed to have in Figure 2.19 even after three treatment days. After the third treatment day, the information contained in the additional measurements eliminates any residual structural uncertainty, singling out the TMDD model as the most consistent model with the available information (see row 2). This model turns out to provide consistent predictions with the patient's treatment response, illustrating that the structural posterior distribution can, indeed, provide meaningful updates of the structural uncertainty.

However, despite this successful demonstration of the Bayesian framework for structural uncertainty estimation, there is a computational challenge hindering its widespread use in practice (Fong et al. 2020; Gelman, Carlin, et al. 1995; Gelman, Vehtari, et al. 2020). Calculating the model evidence needed to estimate the structural posterior distribution (Equation 2.48) requires solving an integral, $p(\mathcal{D}|m) = \int \mathrm{d}\psi\, p(\mathcal{D}|\psi, m)\, p(\psi|m)$ – an integral that turns out to be extremely difficult to calculate.

To illustrate this difficulty, we demonstrate a naive attempt to estimate the integral using sampling in Figure 2.23. Here, we average the model evidence over samples from the prior distribution to approximate the value of the integral

$$p(\mathcal{D}|m) \approx \frac{1}{S}\sum_{s=1}^{S} p(\mathcal{D}|\psi_s, m). \tag{2.54}$$

ψ_s denotes iid samples from the prior distribution and S denotes the sample size. For large enough sample sizes, the average converges to the value of the integral.[35] However, as shown in the top panel of the figure, this convergence is slow and requires impractically

[35] Formally, the convergence of Equation 2.54 can be shown by showing that its expectation over the prior distribution is $p(\mathcal{D}|m)$ and that its variance vanishes for $S \to \infty$. The value of the expectation value is

$$\mathbb{E}\left[\frac{1}{S}\sum_{s=1}^{S} p(\mathcal{D}|\psi_s, m)\right] = \frac{1}{S}\sum_{s=1}^{S} \mathbb{E}\Big[p(\mathcal{D}|\psi, m)\Big] = p(\mathcal{D}|m)\frac{1}{S}\sum_{s=1}^{S} = p(\mathcal{D}|m),$$

equalling the model evidence. For the first equality, we used that the number of samples, S, is independent of the values sampled from the prior distribution, allowing us to pull the expectation into the sum. For the second equality, we use that the expectation of the measurement likelihood over the prior distribution is the model evidence. The model evidence does not depend on any specific sample from the prior distribution and can therefore be pulled in front of the sum.

The convergence of the variance of Equation 2.54 can be similarly shown

$$\mathrm{Var}\left[\frac{1}{S}\sum_{s=1}^{S} p(\mathcal{D}|\psi_s, m)\right] = \frac{1}{S^2}\sum_{s=1}^{S} \mathrm{Var}\Big[p(\mathcal{D}|\psi, m)\Big] = \frac{1}{S}\mathrm{Var}\Big[p(\mathcal{D}|\psi, m)\Big],$$

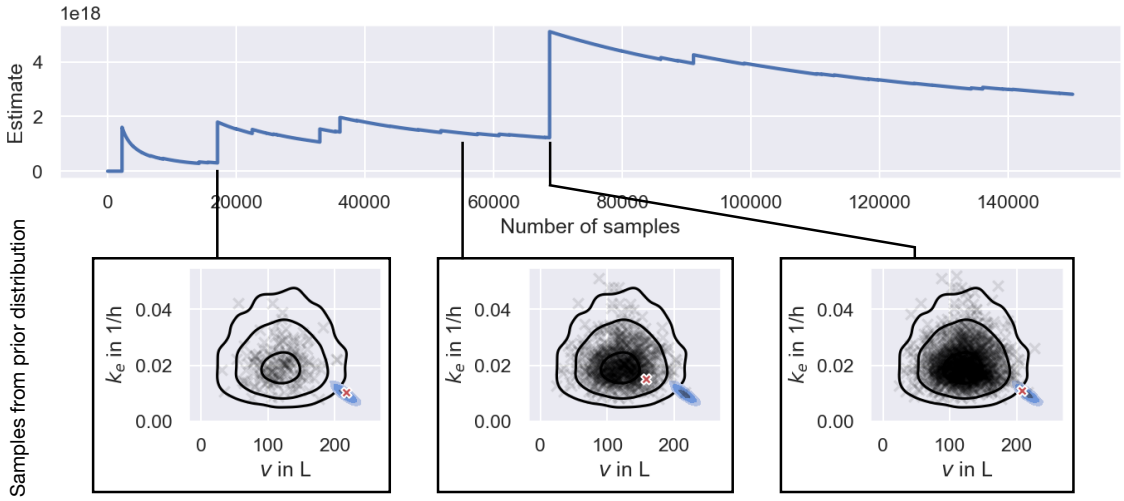


Figure 2.23: Estimation of model evidence. The top panel shows the estimate of the model evidence using Equation 2.54 for increasing sample sizes. The bottom panel illustrates the corresponding samples from the prior distribution, highlighting the most recent sample in red. The prior distribution is illustrated using black contour lines. The posterior distribution is illustrated using shades of blue.

many samples. Here, we estimate the model evidence of candidate model $m = 1$. The estimate displays substantial fluctuations even for up to 150 000 samples from the prior distribution, indicating that many more samples are needed for convergence.

The reason for the slow convergence of Equation 2.54 is the extreme variance of the measurement likelihood for different samples from the prior distribution – samples from regions in parameter space supported by the posterior distribution tend to result in much larger measurement likelihoods than samples from other regions. The mean of the measurement likelihood is therefore dominated by samples that randomly coincide with the area in parameter space supported by the posterior distribution (see bottom panel). However, the probability of randomly drawing prior samples from areas supported by the posterior distribution is small, especially when the information contained in the measurements is large and the dimensionality of the parameter distribution is high – both factors contribute to an increasingly negligible volume of parameter space with support from the posterior distribution relative to the prior distribution. This makes a very large number of iid samples from the prior distribution necessary to

which converges to zero, $\lim_{S\to\infty} \mathrm{Var}\Big[p(\mathcal{D}|\psi, m)\Big]/S = 0$, provided the variance of the measurement likelihood is finite.

adequately represent posterior samples in the average, making the use of Equation 2.54 for most use cases impractical.

While other estimators of the model evidence have been developed, such as the Harmonic mean estimator (Friel et al. 2012; Robert et al. 2009), the challenge of robustly estimating the model evidence remains an unsolved research challenge. This has led to the widespread adoption of an alternative, information theoretic framework for the estimation of structural uncertainty, dating back to Akaike in 1974 (Akaike 1974, 1978). In his approach, Akaike proposes the use of a model's cross-entropy with the data-generating process to estimate its consistency with the measurements – estimates later referred to as information criteria, e.g. (Cavanaugh et al. 2019). Information criteria are computationally more robust than the model evidence making them an attractive alternative for estimating structural uncertainty. Since then research efforts over many years have further developed Akaike's framework, e.g. (Bürkner et al. 2021; Burnham et al. 2002; Gelman, Hwang, et al. 2014; Stone 1977; Vehtari, Gelman, and Gabry 2017), culminating in a recent unification of Akaike's framework with the Bayesian approach to structural uncertainty estimation (Fong et al. 2020), showing that the model evidence can be expressed as an average over information criteria. This makes the approximate calculation of model evidences possible. In the next section, we will briefly outline this approach to conclude the review of the PKPD modelling approach to MIPD. Strengths and limitations of information criteria for PKPD modelling are further explored in Chapter 5.

To summarise, Bayesian inference can be used to estimate a model's predictive uncertainty in its full breadth. To this end, two types of uncertainties are estimated: 1. structural uncertainty; 2. and parametric uncertainty. Structural uncertainty is quantified using a structural posterior distribution which indicates the consistency of candidate model structures with the available information. The parametric uncertainty of each candidate model is quantified using parametric posterior distributions. Together, structural and parametric uncertainties lead to a predictive distribution which captures all predictive uncertainties of a model. This contributes to the responsible use of MIPD in clinical practice. However, it is important to note that a model's baseline

predictive uncertainty is determined by the available prior information. The quality of the uncertainty estimation is therefore sensitive to the quantification of prior knowledge. Overconfident prior distributions can result in substantially underestimated predictive uncertainty, e.g. in cases where important model structures are prematurely excluded from consideration.

2.2.10 Information criteria

In the preceding sections, we outlined the Bayesian approach to PKPD modelling and its use for predictive uncertainty estimation. We noted that one of the Bayesian uncertainty components – structural uncertainty – faces computational challenges in practice. In this section, we will conclude the review of PKPD modelling and discuss an alternative approach to structural uncertainty estimation that does not display such computational challenges. This approach was first introduced by Akaike in 1974 (Akaike 1974), and has recently emerged as an approximate approach to Bayesian structural uncertainty estimation (Fong et al. 2020).

In 1974, Hirotugo Akaike published a seminal paper on 'a new look at the statistical model identification', where he promoted the simple idea that models should be selected for prediction based on their ability to capture the data-generating process (Akaike 1972, 1974). To quantify this ability, he used a well established measure which assesses the similarity of distributions – the Kullback-Leibler (KL) divergence

$$\mathrm{KL}(q||m) = \int \mathrm{d}y\, q(y) \log q(y) - \int \mathrm{d}y\, q(y) \log p(y|\psi, m). \tag{2.55}$$

Here, $q(y)$ denotes the measurement distribution of the data-generating process, and $p(y|\psi, m)$ denotes the candidate model of the measurement distribution (see Equation 2.12). For pedagogical reasons, we begin the discussion with a non-dynamical modelling problem, i.e. a modelling problem with no time dependence. We also consider a fixed set of parameters for the candidate model. The Bayesian treatment of parametric uncertainty will be re-introduced shortly in this section. The intricacies of the cross-entropy for dynamic models are revisited in Chapter 5.

The KL divergence measures the similarity of the model with the data-generating process, in the sense that its value decreases the more the model distribution resembles the data-generating distribution, reaching a value of zero only when the two distributions coincide. This makes it possible to find the best model description of the data-generating process by minimising the KL divergence.

To illustrate this approach, we show the KL divergence for different candidate models in Figure 2.24. We use the imirestat example from the previous section as a use case and presume to know its data-generating measurement distribution at an arbitrarily chosen treatment day, here the 5th treatment day.[36] We can then calculate the KL divergence with the data-generating measurement distribution for different candidate models. For illustrative purposes, the figure does not consider different model structures, but simply varies the mean of the measurement distribution across the candidate models. However, the concept of using the KL divergence to identify the best model remains the same.

The top panel shows the probability densities of the data-generating process (black) and two arbitrarily chosen candidate models (blue and red). The bottom panel shows the KL divergences between all candidate models and the data-generating process. The model parameters corresponding to the distributions highlighted in the top panel are illustrated using vertical lines.

The figure shows that the KL divergence varies across the candidate models (see bottom panel). For example, the highlighted model in red displays a larger KL divergence than the other highlighted model (blue), indicating that the blue model provides a better description of the data-generating process. This is supported by the probability densities in the top panel, showing that the overlap with the data-generating measurement distribution is, indeed, much greater for the blue candidate model than for the red one. The figure also shows that the KL divergence is zero only for one of the models – the model with a measurement mean identical the data-generating process (see scatter point in bottom panel).

This suggests a simple solution to the structural uncertainty challenge: select the model with a vanishing KL divergence for prediction, as this model has the same

[36] We choose a time point for this illustration to convert the dynamic modelling problem into a static one.

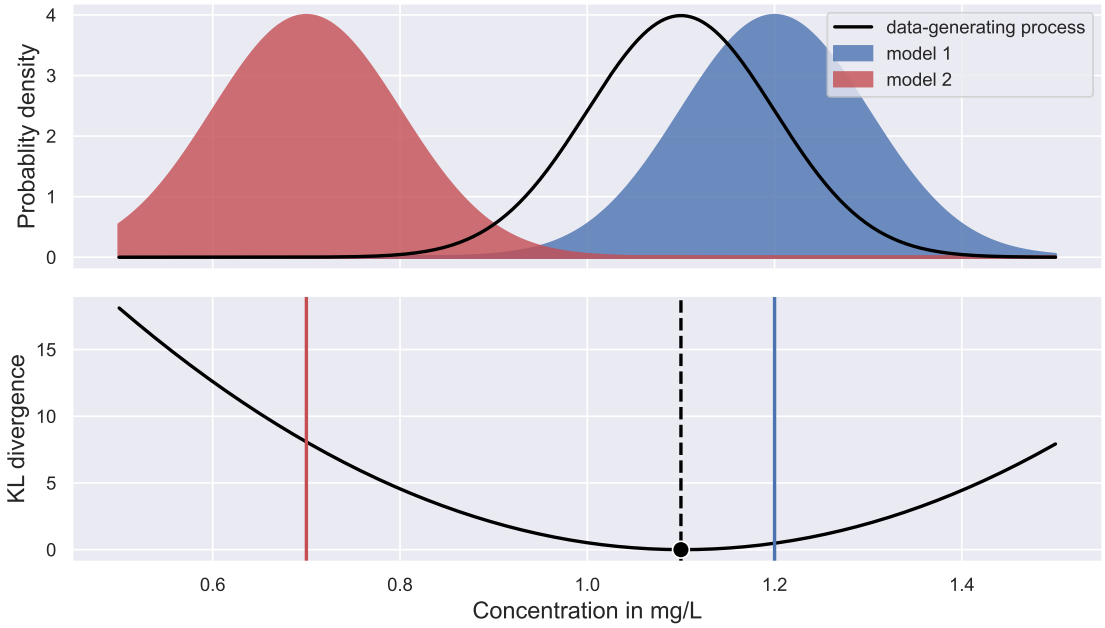


Figure 2.24: Model selection based on the KL divergence. The figure illustrates the values of the KL divergence between the data-generating distribution and different candidate models. The candidate models only differ in their measurement distribution means – the shape and variance is kept constant across models. The top panel shows the data-generating distribution in black, and two arbitrarily chosen candidate models in blue and in red. The KL divergences for all candidate models are shown in the bottom panel (see black curve). The values for the model equalling the data-generating distribution is highlighted using a dashed line, while the values for the highlighted candidate models are illustrated using blue and red lines.

measurement distribution as the data-generating process, and, therefore, captures its processes the best. The obvious challenge with this approach is that the data-generating distribution, $q(y)$, is typically unknown, making Equation 2.55 impossible to compute. However, Akaike realised that we can estimate parts of the KL divergence using only measurements of the data-generating process, overcoming this practical limitation (Akaike 1974).

More specifically, he noted that the second term of the KL divergence – the cross-entropy of the candidate model with the data-generating process, $S(q||m) = -\int \mathrm{d}y\, q(y) \log p(y|\psi, m)$, – can be estimated using the model's measurement likelihood

$$S(q||m) \approx -\frac{1}{n}\sum_{j=1}^{n} \log p(y_j|\psi, m) = -\log p(\mathcal{D}|\psi, m)/n, \tag{2.56}$$

where $\mathcal{D} = \{y_1, \ldots, y_n\}$ denotes the measurements sampled from the data-generating process, and $p(\mathcal{D}|\psi, m)$ denotes the measurement likelihood (see Equation 2.18). The first term of the KL divergence – the entropy of the data-generating process, $S(q) = -\int \mathrm{d}y\, q(y) \log q(y)$ – does not depend on the choice of the candidate model and can, therefore, be omitted for the purpose of model selection. This translates the model selection task from finding the candidate model with minimal KL divergence to finding the model with minimal cross-entropy.

To illustrate this updated approach to model selection, we repeat the experiment from Figure 2.24 and estimate the cross-entropy of the candidate models using two sets of measurements sampled from the data-generating process: 1. a dataset with $n = 30$ measurements (blue); and 2. a dataset with $n = 10$ measurements (red). The exact cross-entropies, calculated using the probability density of the data-generating process, are illustrated in the bottom panel for reference (see black line). The models that minimise the different cross-entropy estimates are illustrated using dashed lines.

The figure shows that the estimates are close to the exact values of the cross-entropy. However, the estimated values do depend on the realisation of the measurements, resulting in different model recommendations for different sets of measurements – the red dataset suggests, for example, that the model with mean 1.07 mg/L provides the best description of the data-generating process, while the blue dataset identifies the model with mean 1.09 mg/L as the best description. In reality, the data-generating distribution has a mean of 1.10 mg/L.

The existence of such estimation errors due to measurement noise is not new to us, and was the very motivation in preceding sections for pursuing an inference approach that can quantify residual uncertainties (see subsection 2.2.5). Also Akaike realised that model selection based on erroneous estimates causes problems for the robustness of model predictions. He therefore proposed the construction of cross-entropy-based model weights that quantify the residual uncertainty about the best model to use for prediction (Akaike 1978)

$$w(m) = \frac{\mathrm{e}^{-n\, S(q||m)}}{\sum_{m=1}^{M} \mathrm{e}^{-n\, S(q||m)}}. \tag{2.57}$$

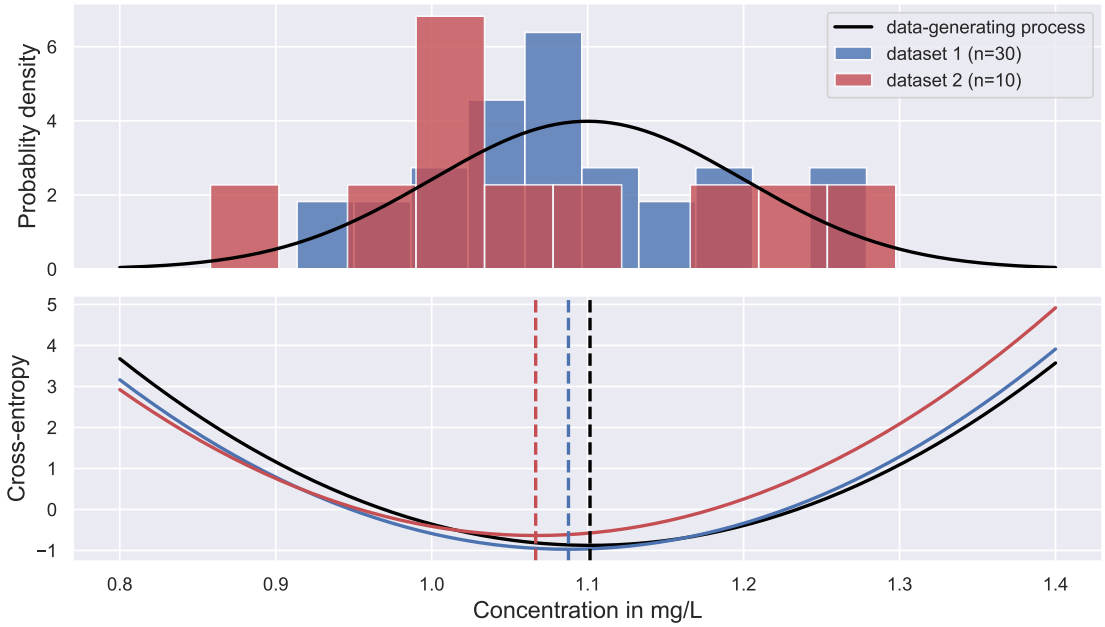


Figure 2.25: Model selection based on cross-entropy estimates. The figure illustrates the values of the cross-entropy between the data-generating distribution and different candidate models. The candidate models only differ in their measurement distribution means – the shape and variance is kept constant across models (see Figure 2.24). The top panel shows the data-generating distribution in black, and two datasets sampled from the data-generating distribution in blue and in red. The exact values of the cross entropy for all candidate models are shown in the bottom panel (see black curve). The values of the cross entropy estimated from the datasets are shown in blue and red. The models minimising the cross-entropies are highlighted using dashed lines.

Remarkably, Akaike's construction of the model weights was not rooted in strong theoretical arguments, but rather built on his intuition: 1. the model weights should zero-in on the model with the smallest KL divergence the more measurements are available;[37] and 2. for few measurements, all models with the potentially smallest KL divergence should receive substantial weights. Since then, several other arguments for

[37] To show that the model weights in Equation 2.57 singles out the candidate model with smallest KL divergence for a large number of measurements, note that the model weights can be expressed in terms of the KL divergence

$$w(m) = \frac{\mathrm{e}^{-n\,S(q||m)}}{\sum_{m=1}^{M}\mathrm{e}^{-n\,S(q||m)}} = \frac{\mathrm{e}^{n\,S(q)}\mathrm{e}^{-n\,S(q||m)}}{\mathrm{e}^{n\,S(q)}\sum_{m=1}^{M}\mathrm{e}^{-n\,S(q||m)}} = \frac{\mathrm{e}^{-n\,\mathrm{KL}(q||m)}}{\sum_{m=1}^{M}\mathrm{e}^{-n\,\mathrm{KL}(q||m)}}, \tag{2.58}$$

where, in the third equality, we use the definition of the KL divergence, $\mathrm{KL}(q||m) = S(q||m) - S(q)$ (see Equation 2.55). This expression converges to a delta-distribution for sufficiently many measurements, $\lim_{n\to\infty} w(m) = \delta_{m,\,m^*}$, assigning only the model with the minimal KL divergence, $m^* = \mathrm{argmin}_m \mathrm{KL}(q||m)$, with non-zero weight.

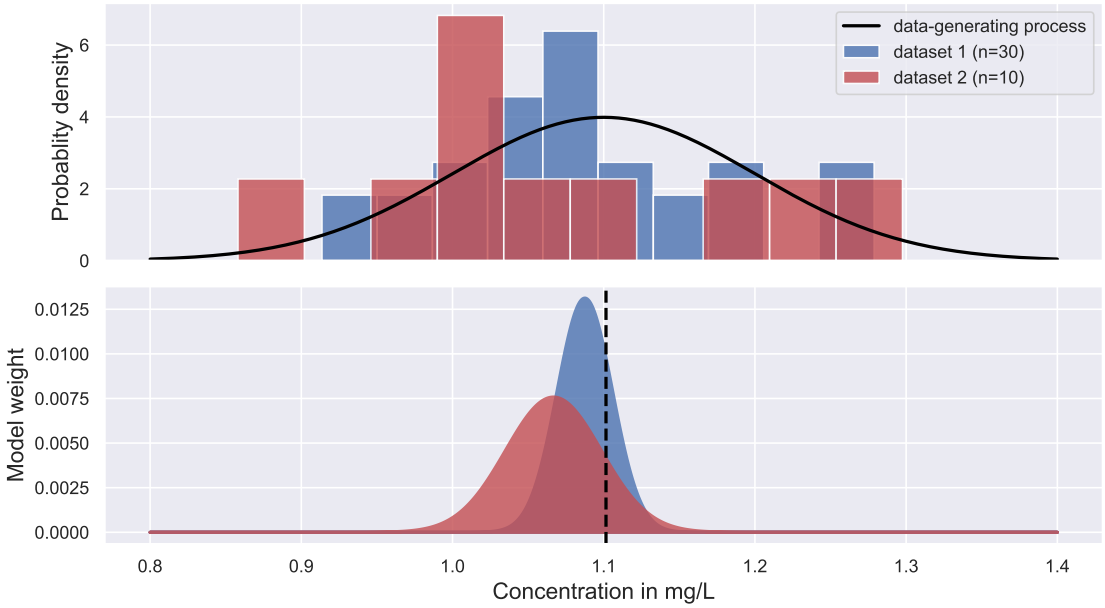


Figure 2.26: Estimating uncertainty using cross-entropy estimates. The figure illustrates the model weights of different candidate models for two datasets. The candidate models only differ in their measurement distribution means – the shape and variance is kept constant across models (see Figure 2.24). The top panel is identical to the top panel in Figure 2.25 and shows the data-generating distribution in black, as well as two datasets sampled from the data-generating distribution in blue and in red. The model weights estimated from the datasets are shown in blue and red.

Akaike's model weights have been brought forward, see e.g. (Cavanaugh et al. 2019).

Figure 2.26 illustrates Akaike's approach to uncertainty estimation using the imirestat use case from Figure 2.25. The top panel of the figure shows the data-generating distribution and two datasets sampled from the distribution. The bottom panel shows the estimates of the model weight for the different candidate models. Recall that, in this example, the models' measurement distributions only differ in their means – the shapes and variances are identical. The figure shows that Akaike's model weights assign substantial probability density to the model with the smallest KL divergence for both datasets (see dashed line). The weight distribution is wider for the red dataset than for the blue dataset. This is consistent with our so-far-developed intuition that more measurements contain more information about the data-generating process, demonstrating that Akaike's model weights can provide a meaningful quantification of the model uncertainty.

However, Akaike's approach, as we have introduced it so far, has one practical limitation – it assumes that the available measurements are only used to estimate the cross-entropy of the model (see Equation 2.56). But, in practice, it is more common to first fit models to the data in order to estimate their parameters, and then use the same data for the cross-entropy estimation. This biases estimates of the cross-entropy towards smaller values, resulting in a systematic overestimation of models' ability to describe the data-generating process.

In his seminal paper from 1974, Akaike found that this bias is asymptotically proportional to the number of model parameters fitted to the measurements when the model parameter values are estimated using maximum likelihood estimation (Akaike 1974). He used this insight to construct an estimator for the cross-entropy that is asymptotically unbiased in the limit where the number of measurements grows large

$$S(q||m)_{aic} = -\log p(\mathcal{D}|\psi^*, m)/n + k/n. \tag{2.59}$$

Here, ψ^* denotes the MLEs estimated from the measurements, $\mathcal{D}$, and k denotes the number of fitted parameters. This estimator is commonly referred to as Akaike information criterion (AIC).[38]

Since then, many other information criteria have been developed, see e.g. (Cavanaugh et al. 2019). For example, one particularly widely-used estimator is the leave-one-out (LOO) estimator (Stone 1977), where the model is fitted repeatedly to the available data, reserving one measurement for the evaluation of the log-likelihood at a time. The average of the log-likelihoods can then be used to estimate the model's cross-entropy

$$S(q||m)_{loo} = -\frac{1}{n}\sum_{j=1}^{n} \log p(y_j|\psi^*_{-j}, m). \tag{2.60}$$

Here, we use ψ^*_{-j} to denote the MLEs estimated from the dataset without the jth measurement. Note that this estimator is unbiased by construction, as each left-out measurement is not used for the parameter estimation. It does, however,

[38] In the literature, the more widely used notation for the AIC is

$$\text{AIC} = 2\log p(\mathcal{D}|\psi^*, m) - 2k,$$

which is equivalent to Equation 2.59 up to a constant normalisation factor.

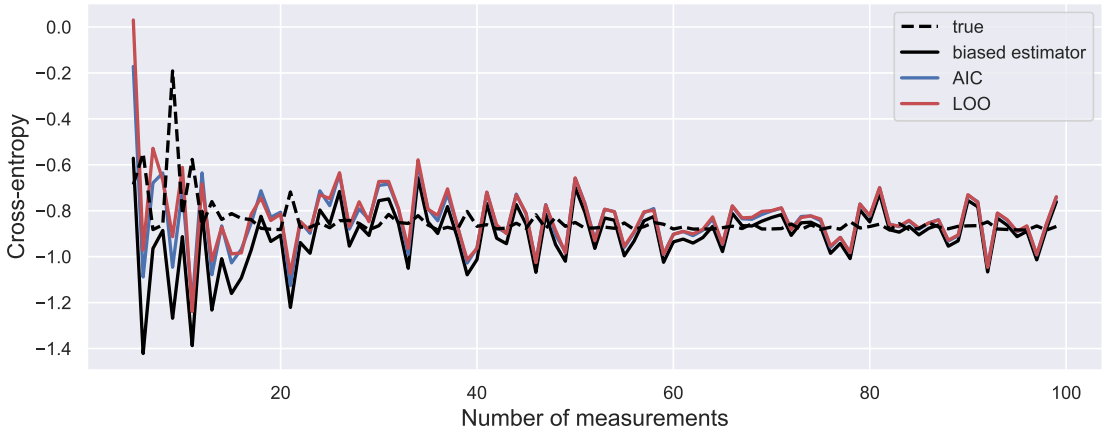


Figure 2.27: Information criteria in the context of maximum likelihood estimation. The figure shows estimates of the cross-entropy for the imirestat example from Figures 2.25 and 2.26. Here, the candidate model is fit repeatedly to data sampled from the data-generating process. The cross-entropy is estimated using different information criteria (unbiased estimator: black; AIC: blue; LOO: red). The true cross-entropy is calculated for reference using the probability density of the data-generating process (see dashed line).

require the repeated fitting of the candidate models to the data, which can become computationally expensive.

Figure 2.27 illustrates the use of information criteria for the estimation of the cross-entropy using the imirestat example from Figures 2.25 and 2.26 as a use case. In this illustration, we sample increasing numbers of measurements from the data-generating distribution. We then fit the model repeatedly to the different datasets and estimate the cross-entropies using the information criteria (biased estimator: black; AIC: blue; LOO: red). The candidate model is again a normal distribution but this time we fit both its mean parameter and its variance parameter to the data using maximum likelihood estimation. The true cross-entropy of the model is visualised using a dashed line.

The figure shows that, for small numbers of measurements, the true cross-entropy of the model varies substantially (see dashed line). For more measurements (approx. $n > 30$), the cross-entropy begins to converge to a constant value of around -0.9, showing that after a certain number of measurements the fit of the model no longer changes substantially. This indicates that around 30 measurements are sufficient to reduce the parametric uncertainty of the normal distribution to a negligible level. However, despite this convergence of the model fit, the information criteria-based estimates of the

cross-entropy do not converge within 100 measurements and fluctuate around the true values for all dataset sizes illustrated in the figure. These fluctuations are estimation errors due to measurement noise. Notably, the errors are, at least visually, unbiased for sufficiently many measurements, indicating that the bias correction of the AIC and LOO estimators works as expected. For smaller dataset sizes, the cross-entropy is systematically underestimated, especially by the biased estimator (see estimates for less than 30 measurements). The AIC and LOO estimators do better but also tend to estimate too small values for this few measurements.

One reason for the bias of the estimates for small datasets, despite the bias 'correction' of the information criteria, is the tendency of the MLEs to overfit the available data – this overfitting is only asymptotically accounted for, in the limit where a large number of measurements is available. This poses a problem for the robust estimation of the cross-entropy in PKPD modelling, as the data size requirement for unbiased cross-entropy estimation becomes more demanding the more model parameters are estimated, and PKPD models tend to have more than two parameters. In addition, the number of measurements is typically limited in clinical practice.

Bayesian inference provides a framework that is less prone to overfitting, estimating posterior predictive distributions based on all model parameters consistent with the available measurements and prior knowledge (see subsection 2.2.5). Cross-entropy estimates, obtained using Bayesian inference, therefore, quantify a model's ability to capture the data-generating process more faithfully when measurements are limited. We can estimate the cross-entropy in a Bayesian inference framework analogously to Equation 2.56, this time using the model's posterior predictive distribution

$$S(q||m) \approx -\frac{1}{n}\sum_{j=1}^{n} \log p(y_j|\mathcal{D}, m). \tag{2.61}$$

Here, $p(y|\mathcal{D}, m)$ denotes the posterior predictive distribution (Equation 2.23). Analogously to Equation 2.56, this estimator is unbiased only if the measurements used for the parameter inference are different from the measurements used for cross-entropy estimation. If the data for both estimations is the same, the cross-entropy estimate

will be biased – a bias that we need to accounted for, e.g. by using information criteria for the cross-entropy estimation.

One widely used information criterion for Bayesian models is the Watanabe-Akaike information criterion (WAIC) (Watanabe et al. 2010)

$$S(q||m)_{waic} = -\frac{1}{n}\sum_{j=1}^{n}\log p(y_j|\mathcal{D},m) + \frac{1}{n}\sum_{j=1}^{n}\mathrm{Var}\Big[\log p(y_j|\psi,m)\Big], \tag{2.62}$$

where the variance of the measurement likelihood in the second term is with respect to the posterior distribution, $p(\psi|\mathcal{D},m)$. Similar to the AIC (Equation 2.59), the WAIC attempts to correct for the bias of the cross-entropy estimation by accounting for the degrees of freedom of the model. However, for Bayesian models, it is not straightforward to count the degrees of freedom, as the uncertainty of model parameters can vary and is updated – a parameter constrained by less prior information may be considered more 'free' than a parameter that is almost entirely determined by prior knowledge. The WAIC, therefore, uses the variance of the measurement likelihood under the posterior distribution as a proxy for the degrees of freedom of a model.

Many more information criteria for Bayesian models have been developed (Gelman, Hwang, et al. 2014). One information criterion that remained applicable to both maximum likelihood models and Bayesian models is the LOO estimator

$$S(q||m)_{loo} = -\frac{1}{n}\sum_{j=1}^{n}\log p(y_j|\mathcal{D}_{-j},m). \tag{2.63}$$

Here, we use $\mathcal{D}_{-j}$ to denote the dataset when the jth measurement is left-out. The main difference between the use of the LOO estimator in a Bayesian inference framework and in a maximum likelihood framework is a practical one: in a Bayesian framework, the model does not need to be fitted repeatedly to the data to compute the LOO estimate. This is because the posterior distribution allows for an importance sampling strategy which can be used to approximate the terms of the LOO estimate, $\log p(y_1|\mathcal{D}_{-1},m),\ldots,\log p(y_n|\mathcal{D}_{-n},m)$ (Vehtari, Gelman, and Gabry 2017). This importance sampling strategy results in an approximate LOO estimate known as Pareto-smoothed importance sampling-LOO (PSIS-LOO) estimate, avoiding the computational costs of inferring the posterior distribution n times.



Figure 2.28: Information criteria in the context of Bayesian inference. The figure shows estimates of the cross-entropy for the imirestat example from Figures 2.25 and 2.26. Here, the candidate model is fit repeatedly to data sampled from the data-generating process. The cross-entropy is estimated using different information criteria (unbiased estimator: black; WAIC: blue; PSIS-LOO: red). The true cross-entropy is calculated for reference using the probability density of the data-generating process (see dashed line).

Figure 2.28 illustrates the Bayesian information criteria for the imirestat use case, analogously to Figure 2.27. The figure shows that the Bayesian model and the information criteria behave similarly to their maximum likelihood analogues for large dataset sizes (approx. $n > 30$). The true cross-entropy converges to a value of approximately -0.9 and the cross-entropy estimates fluctuate, visually unbiased, around this value. The main difference between the Bayesian and the maximum likelihood version is their behaviour for small dataset sizes: in the Baysian case, the true cross-entropy of the model steadily improves with the number of measurements, not displaying any extreme fluctuations that would indicate overfitting. The information criteria provide visually unbiased estimates of the cross-entropy, representing the ability of the model to capture the data-generating distribution more faithfully than the information criteria for maximum likelihood models (compare to Figure 2.27).

Using information criteria for model selection has now become a widely used approach in PKPD modelling, see e.g. (Aranzana-Climent et al. 2024; Kobuchi et al. 2024; L.-H. Lin et al. 2023), and also their use for structural uncertainty estimation has gained traction in recent years (Buatois et al. 2018; Goncalves et al. 2020; Schorning et al. 2016; Sébastien et al. 2016; Uster et al. 2021). This, despite the, arguably, poor theoretical foundations

of Akaike's model weights (Equation 2.57). However, in 2020, Fong and Holmes provided a new perspective on the use of information criteria, placing Akaike's approach into the context of Bayesian inference and giving it a stronger theoretical footing (Fong et al. 2020). This new perspective also solves, in some sense, the computational challenges faced during Bayesian inference when estimating structural uncertainty.

Fong and Holmes realised that the computationally challenging term of the structural posterior distribution (Equation 2.48) – the model evidence – can be expressed in terms of a sum over cross-entropy estimators – i.e. in terms of a sum over information criteria. More specifically, they showed that the model evidence for a dataset, $\mathcal{D}$, is equivalent to the cumulative leave-k-out (LKO) estimate of the cross-entropy

$$\log p(\mathcal{D}|m) = -\sum_{k=1}^{n} S(q||m)_{lko}, \tag{2.64}$$

where the LKO estimator, $S(q||m)_{lko}$, is an extension of the LOO estimator, reserving k measurements of the dataset for the cross-entropy estimation at a time.[39]

This equality can be interpreted to mean that the model evidence, naturally born out of the Bayesian inference framework, attempts to estimate a model's cross-entropy with the data-generating process. The accuracy of this estimate is limited by the availability of the measurements, inviting the replacement of the model evidence by the (scaled) true cross-entropy of the model, $\log p(\mathcal{D}|m) \to -n\, S(q||m)$. This results in a new expression for the structural posterior distribution (Equation 2.48)

$$p(m|\mathcal{D}) \to \frac{\mathrm{e}^{-n\, S(q||m)}\, p(m)}{\sum_{m=1}^{M} \mathrm{e}^{-n\, S(q||m)}\, p(m)}. \tag{2.65}$$

[39] Fong et al define the LKO estimator of the cross-entropy as the average over the measurement likelihoods obtained for all possible n-choose-k ways of leaving out k measurements (Fong et al. 2020)

$$S(q||m)_{lko} = -\frac{1}{\binom{n}{k}} \sum_{t=1}^{\binom{n}{k}} \frac{1}{k} \sum_{j=1}^{k} \log p\left(y_j^{(t)} \middle| \mathcal{D}_{1:n-k}^{(t)}, m\right).$$

Here, $\mathcal{D}_{1:n-k}^{(t)}$ denotes the tth dataset, out of n choose k possible datasets, resulting from leaving out k measurements at a time. $y_j^{(t)}$ denotes the jth measurement of those left-out measurements. Note that the inner sum in this expression is just the average log-likelihood for the left-out measurements – a cross-entropy estimate. The outer sum averages over all possible combinations of leaving out k measurements.

It should be noted that the model evidence does *not* converge to the scaled cross-entropy, even for infinitely many measurements, leading to conceptual controversy. We will come back this below.

Equation 2.65 shows that Akaike's model weights (Equation 2.57) can be interpreted as a special case of the Bayesian structural posterior distribution, if one accepts the replacement of the model evidence by the cross-entropy. In this special case, no prior structural information is available and all model structures are equally likely prior to the inference. This provides a generalisation of Akaike's approach to structural uncertainty estimation, where prior information about suitable candidate models can be quantified using structural prior distributions. It also suggests a much stronger theoretical footing for the use of information criteria-based model weights than previously appreciated (Fong et al. 2020).

In addition, Equation 2.65 solves, in some sense, the computational challenges for the Bayesian approach to structural uncertainty estimation, as robust cross-entropy estimators exist for Bayesian models (Fong et al. 2020). However, the replacement of the model evidence by the scaled cross-entropy does not come without controversy, stemming from the convergence behaviour of the model evidence. The model evidence does, in fact, not converge to the cross-entropy, even for infinitely many measurements. One way to see this is by realising that one of the LKO terms, specifically the leave-n-out estimate, $S(q||m)_{lno} = -\sum_j \log p(y_j|\mathcal{D}_{1:0}, m)/n = -\sum_j \log\Big(\int \mathrm{d}\psi\, p(y_j|\psi, m)\, p(\psi|m)\Big)/n$, is the cross-entropy estimate of the prior predictive distribution – a term that is independent of the posterior predictive distribution. This term does, therefore, not estimate the ability of the model to capture the data-generating process after the parameter inference, but, instead, its ability to capture the data-generating process *before* the parameter inference.

This importance of the prior distribution for the model evidence provides sheds light on its computational challenges, providing an argument in favour of using the amended structural posterior distribution (Equation 2.65) which focuses exclusively on the cross-entropy of the posterior predictive distribution (Fong et al. 2020). However, the importance of the prior distribution for the model evidence can also be used as an argument against the validity of its amended expression, as the amendment

changes the nature of the model evidence, making it, strictly speaking, inconsistent with Bayesian inference.

This concludes the review of predictive uncertainty estimation in PKPD modelling. In Chapter 5, we will dive deeper into, so far, underappreciated limitations of information-criteria-based estimates of the structural uncertainty in PKPD modelling. These limitations are related to the dynamic nature of treatment responses.

In summary, the cross-entropy of a model can be used to quantify its ability to capture the data-generating process. The cross-entropy can be estimated from measurements of the data-generating process, but, when those measurements are also used to estimate the model's parameters, the estimates are biased. In those cases, information criteria provide asymptotically unbiased estimates of the cross-entropy. For a limited number of measurements, the bias correction of information criteria fails for maximum likelihood models. In a Bayesian inference framework, the bias correction of information criteria tends to be better. However, irrespective of the bias correction, information criteria remain sensitive to measurement noise for all typical dataset sizes available in clinical practice, leading to estimation errors. This motivates the construction of cross-entropy-based model weights, estimating the structural uncertainty due to measurement noise. These model weights can be tied to the Bayesian approach to structural uncertainty estimation, providing a solution to the computational challenges of Bayesian structural uncertainty estimation.

3

Simulating clinical trials for model-informed precision dosing

This chapter has been published in:

DA, Lambert B, Robinson M, Wang K, Gavaghan D. Simulating clinical trials for model-informed precision dosing: Using warfarin treatment as a use case. *Front. Pharmacol. - Drug Metabolism and Transport*. 2023.

Contents

3.1 Introduction . . . 99
3.2 Methods . . . 102
3.2.1 General framework for MIPD trial simulation . . . 102
3.2.2 Workflow of MIPD trial simulation . . . 103
3.2.3 Components of the CT model . . . 104
3.2.4 Implementation of warfarin trial simulation . . . 110
3.2.5 MIPD methods . . . 112
3.3 Results & discussion . . . 116
3.3.1 Simulating trial phases prior to MIPD . . . 116
3.3.2 Fitting the MIPD models . . . 120
3.3.3 Simulating the MIPD trials . . . 120
3.3.4 Results of the MIPD trials . . . 121
3.3.5 Degrees of dosing regimen individualisation . . . 123
3.4 Conclusion . . . 135
3.5 Data and software availability . . . 137

3.1 Introduction

Model-informed precision dosing (MIPD) is an emerging approach used to guide the individualisation of dosing regimens for drugs that are otherwise difficult to administer (Keizer et al. 2018; Sheiner 1969). Several MIPD approaches have been suggested (see Chapter 2), but identifying the best (or even an appropriate) MIPD approach for a given application remains a challenge as ethical and financial constraints limit the ability to run clinical MIPD trials repeatedly, thereby leaving no margin for error during the development process. An alternative to real clinical trials are *simulated* clinical trials (Holford, Kimko, et al. 2000; Holford, S. C. Ma, et al. 2010). In this chapter, we develop a holistic simulation framework which attempts to emulate all complexities of clinical practice for MIPD. Such simulations provide a cost and time efficient way to exhaustively test different MIPD approaches, and help to bring developers, researchers, and regulators closer together, as simulation results can be reproducibly shared in accordance with data protection laws.

Typical examples that would benefit from MIPD are drugs with narrow therapeutic windows and large treatment response variability, such as warfarin, docetaxel and infliximab (Gill et al. 2016; Yuxiang Ma et al. 2021; Wadelius and Pirmohamed 2007). Other examples include antibiotics, like vancomycin, where MIPD has been suggested and partially implemented to guide dosing strategies for critically ill patients, balancing the treatment of severe infections and the risks for harmful adverse events (Broeker et al. 2019; Matsumoto et al. 2022; Wicha et al. 2021). MIPD may also be used to efficiently adapt dosing regimens to continuously changing conditions, for example to stabilise the blood glucose levels of diabetes patients with insulin treatment (Y. Wang et al. 2019; T. Zhu et al. 2020).

The most prominent MIPD methods are regression, reinforcement learning (RL) and pharmacokinetic and pharmacodynamic (PKPD) modelling (Johnson, Caudle, et al. 2017; Johnson, L. Gong, et al. 2011; Keizer et al. 2018; Ribba, Dudal, et al. 2020). These methods may be further categorised into different subvariants. RL variants include, for example, model-free algorithms such as Q learning (Zadeh et al. 2023) and model-based algorithms such as Monte Carlo tree search (Maier, Hartung, Kloft, et al.

2021). MIPD variants of PKPD modelling include maximum a posteriori-guided dosing and Bayesian data assimilation-guided dosing (Maier, Hartung, de Wiljes, et al. 2020). Other variants include the modelling of virtual twins using physiology-based PK (PBPK) and quantitative systems pharmacology (QSP) models (Polasek and Rostami-Hodjegan 2020). Although differences across MIPD methods exist, all have in common that they need to process data in two steps in order to make individualised predictions. First models are fitted to population data. This step establishes a relationship between patient characteristics and dosing strategies. In a second step, data specific to the to-be-treated patient is used to predict individualised dosing regimens (see Chapter 2 for details). For example, regression models have been fitted using records of genetic information and dosages across patients (Gage et al. 2008; I. Y. Gong et al. 2011; IWPC 2009), enabling the prediction of individualised dosages based on genetic information.

The data used for model fitting and dosing regimen individualisation differ substantially across MIPD methods and include clinical factors, genetic factors and monitoring data. The type and volume of data are key to both the accuracy of predictions and the ease of implementation in clinical practice (Darwich et al. 2017; Ribba, Bräm, et al. 2022). The more data are collected, the better dosing regimens can be individualised. However, practical constraints limit how much and what type of data may be available for MIPD. As a result, the trade-off between accuracy and practicality needs to be considered when applying MIPD approaches to medicines. The systematic study of this trade-off for a specific application is, however, complicated by the astronomical costs of clinical trials, rendering repeated clinical trials for different MIPD methods infeasible. In this study, we propose a framework for the simulation of clinical MIPD trials, facilitating a resource-efficient way to test and develop MIPD approaches.

Using simulations to understand MIPD approaches is not a new concept and several MIPD simulation studies in the literature exist. Moore et al and Ribba et al use simulated treatment responses to investigate RL approaches as a strategy to individualise dosages of anaesthetics in intensive care units (Moore et al. 2004; Ribba, Bräm, et al. 2022). For the simulations, they used a semi-mechanistic PKPD model to emulate the time course of treatment responses and a nonlinear mixed effects (NLME) model structure

to capture inter-individual variability (IIV). A similar approach is adopted by Maier et al to study the individualisation of paclitaxel-based chemotherapy using different MIPD approaches, including PKPD modelling, RL and a hybrid PKPD-RL approach (Maier, Hartung, de Wiljes, et al. 2020; Maier, Hartung, Kloft, et al. 2021). An NLME model simulation approach is also used by Zadeh et al to study deep RL as a candidate for MIPD of warfarin (Zadeh et al. 2023) . Abrantes et al extend this NLME simulation approach, adding inter-occasion variability (IOV) of treatment responses as an extra dimension to the MIPD simulation (Abrantes et al. 2019). They model IOV following Karlsson et al and randomly vary the PKPD model parameters of virtual patients over time (Karlsson et al. 1993). An analogous approach is used by Keutzer et al to understand MIPD-based treatment individualisation of rifampicin (Keutzer et al. 2020).

We propose an extended framework for the simulation of clinical MIPD trials. Our framework complements the emulation of IIV and IOV by other previously established elements of clinical trial simulation (Holford, Kimko, et al. 2000; Holford, S. C. Ma, et al. 2010). In particular, our framework includes treatment response emergence from complex interactions of pharmacological and physiological processes as a central feature of the trial simulation. In addition, deviations of dose administrations from nominal dosing regimens, as well as delayed monitoring measurements are incorporated in the simulation to more faithfully represent practical limitations of monitoring-based MIPD approaches.

The chapter is divided into three sections: Methods; Results & discussion; and Conclusion. In the Methods, we first introduce the general framework for MIPD trial simulation and subsequently demonstrate its implementation in terms of a clinical trial model for the warfarin use case. We then introduce the three MIPD models, which we will investigate with the help of the clinical trial model. The considered models are: 1. a neural network regression model; 2. a deep reinforcement learning model; and 3. a PKPD model. In the Results & discussion, we use the clinical trial model to simulate MIPD trials for each of the three models and analyse the strengths and limitations of the MIPD models. In our analysis, we pay careful attention to attributing generic strengths and limitations to the MIPD methodology and specific strengths and

limitations to our implementations of the models. In the Conclusion, we summarise our findings and propose future directions for MIPD research.

3.2 Methods

We first introduce the proposed framework for clinical MIPD trial simulation and discuss the specific implementation for warfarin. We then outline the investigated MIPD methods. The data, models, and scripts used in this chapter are hosted on GitHub (https://github.com/DavAug/mipd-warfarin). A user-friendly API for MIPD approaches using PKPD modelling has been implemented in the open source Python package `chi` (Augustin 2024). MIPD approaches using neural networks were implemented in `PyTorch` (Paszke et al. 2019).

3.2.1 General framework for MIPD trial simulation

The objective of MIPD is to achieve desired treatment outcomes across individuals despite large treatment response variability by optimising individual dosing regimens. Challenges for this individualisation are nonlinear and delayed treatment responses (Dirks et al. 2010; Mager 2006; Véronneau-Veilleux et al. 2020), making it difficult to adjust and extrapolate dosages based on feedback from monitoring. Variability of the treatment response as a result of time-varying pharmacokinetics, concomitant food-intake, comedication and disease progression further complicate the dosing regimen adjustment (Keutzer et al. 2020). A realistic assessment of MIPD approaches in a simulated trial therefore needs to account not only for IIV, but also for treatment response delays, nonlinearities and IOV. However, PKPD-related aspects are not the only factors influencing the success of MIPD methods. There are also practical limitations for MIPD, including imperfect measurements and deviations from nominal dosing and monitoring schedules (Holford, Kimko, et al. 2000; Holford, S. C. Ma, et al. 2010). While measurement noise is commonly included in simulations, the variability in the execution of the trial often remains neglected. Deviations from the nominal schedule can impact the success of MIPD methods, especially when they are not fed back into the model.

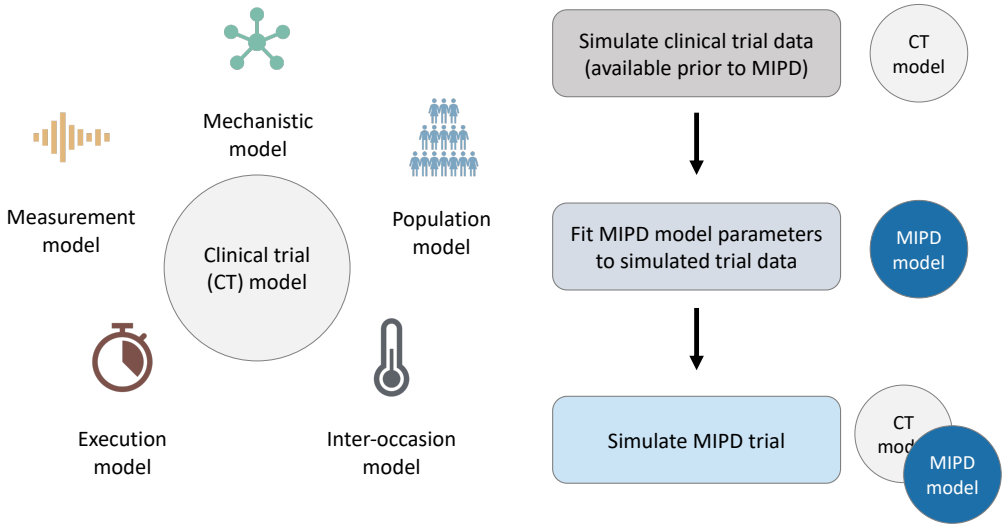


Figure 3.1: Framework for MIPD trial simulation. The left panel shows the components of the clinical trial (CT) model: 1. a mechanistic model; 2. a population model; 3. an inter-occasion model; 4. an execution model; and 5. a measurement model. The right panel shows the workflow leading up to the MIPD trial simulation. First, the CT model is used to simulate typical clinical trial data, available prior to the MIPD trial. Second, the simulated trial data is used to fit the parameters of the MIPD model. This emulates the starting point of MIPD trials in practice. Finally, the CT model and the fitted MIPD model are used to simulate the MIPD trial.

To faithfully emulate the performance of MIPD in clinical practice, our simulation framework accounts for these PKPD-related and practical challenges using a clinical trial (CT) model composed of five components: 1. a mechanistic model; 2. a population model; 3. an inter-occasion model; 4. an execution model and 5. a measurement model (see left panel in Figure 3.1). We describe each of these components in subsection 3.2.3. The right panel of the figure illustrates the workflow of using the CT model for simulating MIPD trials.

3.2.2 Workflow of MIPD trial simulation

Our method of MIPD trial simulation involves three steps (see right panel in Figure 3.1): 1. simulation of pre-MIPD clinical trial data; 2. fitting of the MIPD model to this simulated data; 3. simulation of the MIPD trial. The first two steps emulate the MIPD model development that takes place in practice prior to MIPD trials. In our method, we first simulate typical clinical data from e.g. phase I trials, phase II trials and/or

phase III trials using the CT model. We then fit the MIPD model parameters to the simulated data. The details of the fitting process are specific to the MIPD model and are presented in subsection 3.2.5. It is important that the simulated data are used for the fitting, even if real clinical data are available, in order to facilitate a clear attribution of limitations observed in a simulated MIPD trial to the MIPD model. If instead, the MIPD model was fitted to real clinical data, the approximation error of the CT model with respect to the data-generating process of the real clinical trial may also contribute to limitations observed in the simulated trials, making it harder to draw conclusions about the MIPD model. The real data should, instead, be used to calibrate the CT model prior to the data simulation in order to minimise the approximation error as much as possible. The final step of the workflow is the simulation of the MIPD trial using the fitted MIPD model and the CT model.

3.2.3 Components of the CT model

We now describe the components of the CT model. By design, the components of the simulation are non-overlapping and are as modular as possible. This simplifies the model structure and enables an iterative development of CT models, making it possible to replace or further develop individual components without requiring changes in other model components.

1. The mechanistic model: This component models the dynamics of treatment responses as a function of time, t, and the dosing regimen, r,

$$\bar{y}(t, r, \psi). \tag{3.1}$$

$\bar{y}$ denotes quantities of interest that may be monitored in clinical practice, and ψ denotes the parameters of the model. The main purpose of the mechanistic model is to faithfully reflect nonlinearities and delays of the treatment response, emergent from complex cascades of pharmacological and physiological processes. Emulating this complexity provides a tool to test the ability of different MIPD approaches to approximate the treatment response and predict individualised dosing regimens. Popular choices to simulate treatment response dynamics include PKPD models and quantitative

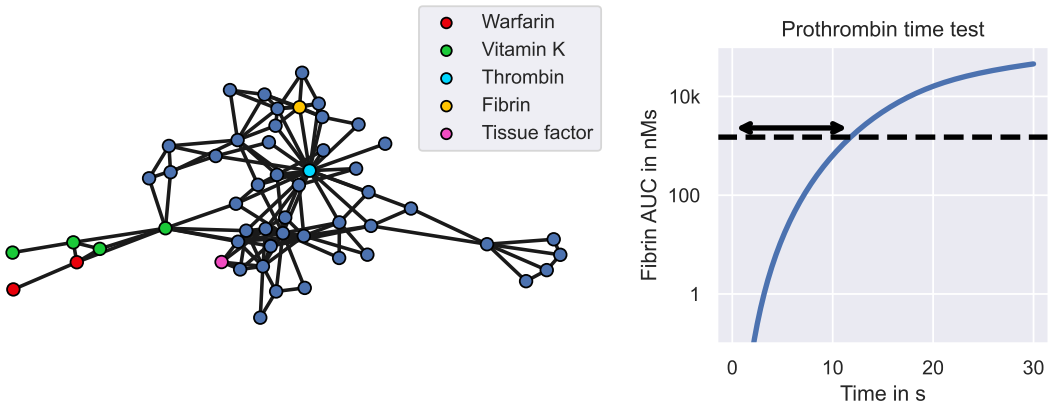


Figure 3.2: Warfarin clinical trial simulation - the mechanistic model. The figure shows Wajima et al's QSP model of the warfarin treatment response mechanism. The left panel shows the model. Nodes represent states of the model, including warfarin (red), vitamin K (green) and different coagulation factors (blue). Central coagulation factors include thrombin (light blue) and fibrin (yellow). Multiple nodes for warfarin refer to the drug amount in different compartments, while multiple nodes for vitamin K also refer to different forms of vitamin K, such as vitamin K epoxide and vitamin K hydroquinone. Interactions between states, such as transitions, inhibitions or activations, are represented by edges. The right panel shows the model simulation of the prothrombin time test. The arrow indicates the prothrombin time which marks the time elapsed between exposure to tissue factor and reaching a fibrin AUC of 1500 nMs.

systems pharmacology (QSP) models (Azer et al. 2021; Maier, Hartung, Kloft, et al. 2021; Ribba, Dudal, et al. 2020).

An example mechanistic model of warfarin treatment developed by Wajima et al is illustrated in Figure 3.2 (Wajima et al. 2009). As discussed in Chapter 1 and Chapter 2, warfarin is an oral anticoagulant widely used for the prevention and treatment of venous thrombosis, pulmonary embolism and thromboembolic complications associated with atrial fibrillation and/or cardiac valve replacement (FDA 2010). Wajima et al's model is conceptually similar to the PKPD model from subsection 2.2.1, but captures the treatment response dynamics in substantially more detail. The left panel shows the 53 blood components described by the model, including warfarin, vitamin K and different coagulation factors, such as thrombin and fibrin. Edges between the components represent interactions, such as transitions, inhibitions or activations. The monitored quantity of the treatment response, $\bar{y}$, is the prothrombin time. The prothrombin time measures the time it takes for plasma to clot after exposure to a thromboplastin reagent

and is routinely measured in clinical practice. In Wajima et al's model, this prothrombin time test is simulated by measuring the time elapsed between adding the reagent (300 nM tissue factor) and reaching a fibrin area-under-the-curve (AUC) of 1500 nMs (see right panel in Figure 3.2). The prothrombin time is commonly reported in terms of the international normalised ratio (INR), which measures the prothrombin time of a patient's blood sample in units of the prothrombin time of a reference sample. We will use this model in our warfarin clinical trial simulation (see subsection 3.2.4 for details).

2. The population model: This component models the variability in the treatment response across individuals using a mixed effects model extension of the mechanistic model. A mixed effects model defines a population distribution of model parameters

$$p(\psi|\theta), \tag{3.2}$$

capturing the differences between individuals, i.e. the IIV (Augustin, Lambert, K. Wang, et al. 2023; Lavielle 2014). θ denotes the parameters of the population distribution. Each sample, ψ, from the population distribution represents an individual with treatment response $\bar{y}(t, r, \psi)$. Thus, differences between individuals arise in this model structure from differences in the mechanistic model parameters. For some applications, these differences can be partially explained by covariates, χ, which may divide the population into subpopulations, $p(\psi|\theta, \chi)$. The full population distribution across covariates is then given by the average of the subpopulations weighted by the relative frequency of the covariates, $p(\psi|\theta) = \mathbb{E}_{\chi}[p(\psi|\theta, \chi)]$. For more details on NLME modelling, we refer back to subsection 2.2.7.

Covariates of the variability can range from clinical factors to genetic factors. For example for warfarin treatment, the VKORC1 genotype explains 27 % of the observed response variability (Wadelius, L. Y. Chen, et al. 2009). Other covariates of warfarin treatment include mutations in the CYP2C9 gene and the age of the patient. The link between covariates and pharmacological or physiological processes makes it possible to define mixed effects models that reflect the mechanistic relationship between covariates and the treatment response variability (Hamberg, Dahl, et al. 2007; Hamberg, Wadelius, et al. 2010; Hartmann, Biliouris, Lesko, et al. 2016, 2020). For example, warfarin's

mode of action is the inhibition of the vitamin K epoxide reductase complex (VKORC), and mutations in VKORC's subunit 1 (VKORC1) affect the inhibitory activity, which can be implemented with a reduced EC50 parameter in Wajima et al's mechanistic model.

The left panel of Figure 3.3 illustrates the emergence of inter-individual treatment response variability in our warfarin clinical trial model. The figure shows simulated INR treatment responses of 6 individuals to daily administrations of warfarin. 3 individuals have the GG genotype (VKORC1), the *1/*1 genotype (CYP2C9) and are 71 years old (see blue lines). The remaining 3 individuals have the GA genotype (VKORC1), the *1/*2 genotype (CYP2C9) and are 46 years old (see red lines). The dashed line indicates the treatment response of an average 71 year old individual with the GG and *1/*1 genotypes. We can see that individuals with $\chi = (\text{GG}, \text{*1/*1}, 71)$ tend to respond less strongly to warfarin treatment than individuals with $\chi = (\text{GA}, \text{*1/*2}, 46)$. However, there remains substantial IIV that is not explained by covariates.

3. The inter-occasion model: This component models the variability of the treatment response over time using time-varying modifications of the model parameters

$$\psi \rightarrow \psi'(t) = \psi\, \eta(t). \tag{3.3}$$

η denotes the alterations of the model parameters, and ψ' denotes the new model parameters. The treatment response of an individual with parameters ψ is now given by the mechanistic model simulation using the time-varying parameters, $\bar{y}(t, r, \psi'(t))$ (Karlsson et al. 1993). The role of the inter-occasion model is to implement changes of the treatment response that are not accounted for by the mechanistic model. Such changes can be externally driven, e.g. by concomitant food intake or comedication, or of completely unknown origin (Keutzer et al. 2020). The dynamics of η, $p(\eta|t)$, are a modelling choice.

A source of inter-occasion variability for warfarin is, for example, the time-varying consumption of vitamin K (Xue et al. 2016), changing the amount of vitamin K available in the blood. Since warfarin's mode of action is to inhibit VKORC – a complex converting one form of vitamin K into another, clotting factor-activating form of vitamin K – an increased availability of vitamin K can reverse the treatment effects of warfarin. In the

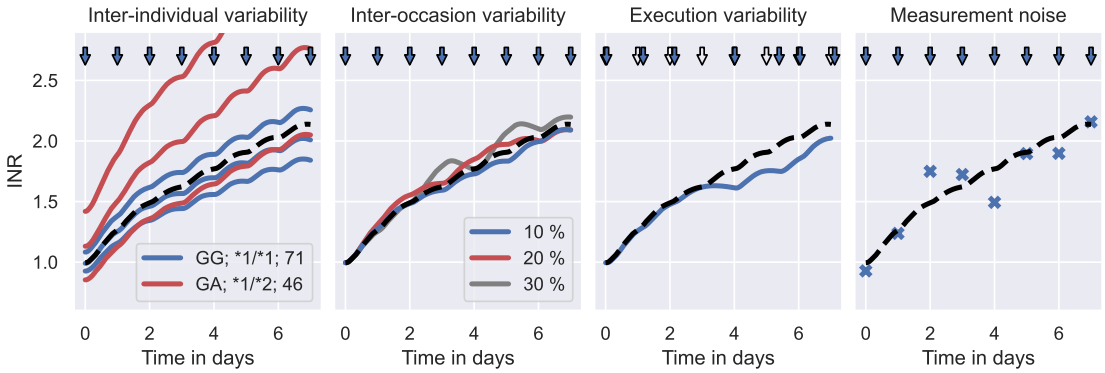


Figure 3.3: Warfarin clinical trial simulation - sources of treatment response variability. The figure shows each contribution to the treatment response variability in isolation. Panel 1: Illustrates the effect of the population model on the clinical trial simulation. The solid lines indicate the treatment responses of 6 simulated individuals to daily warfarin administrations: 3 of which have the covariates $\chi = (\text{GG}, \text{*1/*1}, 71)$ (see blue lines); and the remaining 3 have the covariates $\chi = (\text{GA}, \text{*1/*2}, 46)$ (see red lines). The typical treatment response across individuals is indicated by a dashed line. The administration times are indicated by blue arrows. Panel 2: Illustrates the effect of the inter-occasion model on the clinical trial simulation. The dashed line shows the treatment response of an individual with a constant vitamin K input rate, i.e. no IOV. The solid lines show the treatment response of the same individual with vitamin K input rates that randomly vary by 10 % (blue), 20 % (red) and 30 % (grey) between days. Panel 3: Illustrates the effect of the execution model on the clinical trial simulation. The dashed line indicates the treatment response of an individual associated with the nominal dosing regimen (hollow arrows). The blue line indicates the treatment response of the same individual associated with the delayed dose administrations (blue arrows). Panel 4: Illustrates the effect of the measurement model on the clinical trial simulation. The dashed line shows the simulated treatment response of an individual and the scatter points show the associated monitoring measurement.

2^{nd} panel of Figure 3.3 we illustrate the effects of varying vitamin K consumption on the warfarin treatment response in the clinical trial simulation. The dashed line shows the treatment response simulation with a constant vitamin K input rate parameter, i.e. no variability in the vitamin K consumption. The solid lines show the treatment response simulations with vitamin K input rates that randomly vary by 10 % (blue), 20 % (red) and 30 % (grey) from day to day.

4. The execution model: This component models unintended deviations from nominal dosing regimens and monitoring schedules during the trial. Nominal dosing regimens are defined by a sequence of doses and administration times, $r = \{(d_j, t_j)\}$, where d_j denotes the j^{th} dose and t_j denotes the associated administration time.

Nominal monitoring schedules are similarly defined by a sequence of measurement times. The actual doses and times are modelled in the execution model using random deviations from the nominal schedules

$$d_j \to d'_j = d_j + \Delta d, \qquad t_j \to t'_j = t_j + \Delta t, \tag{3.4}$$

where Δd and Δt denote the deviations. The treatment response corresponding to the actual dosing regimen, $r' = \{(d'_j, t'_j)\}$, is simulated using $\bar{y}(t, r', \psi'(t))$. The role of the execution model is to test the robustness of MIPD approaches in clinical practice by emulating the limited control over dose administrations and monitoring. By choosing to report nominal schedules as opposed to actual schedules, the execution model can also be used to mirror common inaccuracies of clinical data. The distributions of dose and time deviations, $p(\Delta d)$ and $p(\Delta t)$, are modelling choices and may differ between dosing and measurement schedules. While not considered in this article, the execution model may be extended to include missed administrations or missed measurements during the trial. Assuming no persistence, this can be implemented using draws of Bernoulli random variables associated with each time point, indicating whether or not a dose was administered or a measurement was taken (Holford, Kimko, et al. 2000; Holford, S. C. Ma, et al. 2010). For infusions, deviations in the duration of the administration may also be modelled.

In the 3^{rd} panel of Figure 3.3 we illustrate the effects of delayed dose administrations on the treatment response in the warfarin trial simulation. Doses are administered daily. The nominal dosing regimen is illustrated by hollow arrows. The actual dosing regimen with delayed administrations is illustrated by blue arrows. The associated treatment response simulations are indicated by the dashed line (nominal) and by the solid line (actual).

5. The measurement model: This component models the limited accuracy of treatment response measurements

$$y = \bar{y}(t, r', \psi'(t)) + \varepsilon, \tag{3.5}$$

where y denotes the measurement and ε denotes the measurement error. This defines a distribution of measurements around the mechanistic model output, $\bar{y}$, at each

time point t, $p(y|t, r', \psi')$, where measurements may be expected. The wider the measurement distribution, the larger the measurement noise. During the trial simulation, monitoring measurements are sampled from the measurement distribution. This model can be extended to include noise also in the measurement process of covariates, for example to reflect genotyping errors of the VKORC1 or CYP2C9 genes. This is however not considered in this article.

In the 4^{th} panel of Figure 3.3, we illustrate INR measurements sampled from the measurement distribution during warfarin treatment (blue scatter points). The dashed line shows the treatment response simulation of the mechanistic model without measurement noise.

3.2.4 Implementation of warfarin trial simulation

We develop the warfarin clinical trial model following the framework for MIPD trial simulation introduced in subsection 3.2.1. The mechanistic model is implemented using Wajima et al's model of the humoral coagulation network (see Figure 3.2). The model provides a mechanistic description of warfarin's PKPD using a system of nonlinear differential equations

$$\frac{\mathrm{d}a_d}{\mathrm{d}t} = -k_a a_d + r(t), \quad \frac{\mathrm{d}a_c}{\mathrm{d}t} = k_a a_d - k_e a_c, \quad \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}, a_c, \psi). \tag{3.6}$$

a_d and a_c describe the pharmacokinetics of warfarin and denote the amount of the drug in the dose compartment and the central compartment, respectively. k_a denotes the absorption rate and k_e denotes the elimination rate. $r(t)$ denotes the dose rate and implements the dosing regimen. The pharmacodynamics of warfarin are captured by $\mathbf{x}$, denoting the 51 remaining states of the model. The prothrombin time is simulated as a function of the states, $\bar{y}(t, r, \psi) = g(\mathbf{x}(t, r, \psi), \psi)$, involving the computation of the fibrin AUC after exposure to $300\,\text{nM}$ tissue factor. For full details of the mechanistic model, we refer to (Wajima et al. 2009) and Appendix B.1.

The population model is implemented using a hybrid of two mixed effects models developed by Hartmann et al and Hamberg et al (Hamberg, Wadelius, et al. 2010; Hartmann, Biliouris, Lesko, et al. 2016, 2020). Hartmann et al (2016) provide a mixed

effects model extension of Wajima et al's model, capturing the treatment response variability emergent from varying production rates of selected coagulation factors, including prothrombin, protein S, protein C and coagulation factors V, VII, IX, X, XI, XII and XIII. In a subsequent publication, they extend their model to include variability explained by covariates, such as the genotypes of the VKORC1 gene and the CYP2C9 gene (Hartmann, Biliouris, Lesko, et al. 2020). VKORC1 is used to model the variability in warfarin's EC50, while CYP2C9 is used to model the variability in warfarin's elimination rate. In our population model, we modify Hartmann et al's model further using elements from Hamberg et al's model to incorporate age and heterozygosity in the genotypes as covariates of the IIV (Hamberg, Wadelius, et al. 2010). This results in a population distribution whose subpopulations, $p(\psi|\theta, \chi)$, are defined by the age of a patient, one of three VKORC1 genotypes (GG; GA; AA) and one of six CYP2C9 genotypes (*1*1; *1*2; *1*3; *2*2; *2*3; *3*3). For full details of the population model, we refer to Appendix B.1. The population model parameters, θ, used to simulate the clinical trial are provided on GitHub (https://github.com/DavAug/mipd-warfarin).

The inter-occasion model is implemented using time-varying vitamin K input rates (see 2^{nd} panel in Figure 3.3). The input rate alterations are assumed to be normally distributed

$$p(\eta|t) = \mathcal{N}(\eta|\mu_\eta, \sigma_\eta^2), \tag{3.7}$$

with constant mean and standard deviation: $\mu_\eta = 1$ and $\sigma_\eta = 0.1$. To allow a change of vitamin K consumption over time, we sample a new η for each simulation day, such that the altered input rate may be interpreted as the daily average of the vitamin K consumption.

The execution model is implemented using exponentially distributed delays of the administration and monitoring times

$$p(\Delta t|\tau) = \frac{1}{\tau}\mathrm{e}^{-\Delta t/\tau}, \tag{3.8}$$

where $\tau = 30\,\text{min}$ denotes the average delay. For each nominal administration time and each nominal monitoring time, we independently sample delays from $p(\Delta t|\tau)$ and

compute the actual times according to Equation 3.4. If dose administrations and monitoring measurements are scheduled at the same nominal times, we only draw one delay random variable for both events.

The measurement model is implemented using lognormally distributed measurements around the mechanistic model output

$$p(y|t, r', \psi') = \mathrm{LN}(y|\mu, \sigma), \tag{3.9}$$

where $\mu = \bar{y}(t, r', \psi')$ denotes the median, and $\sigma = 0.1$ denotes the scale of the distribution. This implements a measurement error that scales proportionally with the measured quantity, producing measurement errors of approximately $10\,\%$ of the mechanistic model output.

3.2.5 MIPD methods

We investigate three MIPD models for warfarin treatment individualisation: 1. a neural network regression model ('Regression model'); 2. a deep reinforcement learning model ('Deep RL model'); and 3. a pharmacokinetic and pharmacodynamic model ('PKPD model'). All three models use data specific to the to-be-treated patient to predict individualised dosing regimens. While the details differ substantially between the models, Figure 3.4 illustrates their general workflow. The bottom of the figure shows the to-be-treated patient characterised by covariates, such as clinical and genetic factors. The top of the figure shows the treating physician and the MIPD model. The physician uses the model and the patient characteristics to predict an individualised dose. For some MIPD approaches, this dose can be iteratively refined using measurements of the patient's treatment response. Some models can also predict the full future dosing regimen at each iteration rather than just the next dose. Below, we discuss the three methods investigated in this study in more detail.

1. Regression model – This MIPD approach follows Anderson et al and Verhoef et al and uses a static model of the daily maintenance dose to individualise treatments (Anderson et al. 2012; Verhoef et al. 2013). The maintenance dose refers to the constant

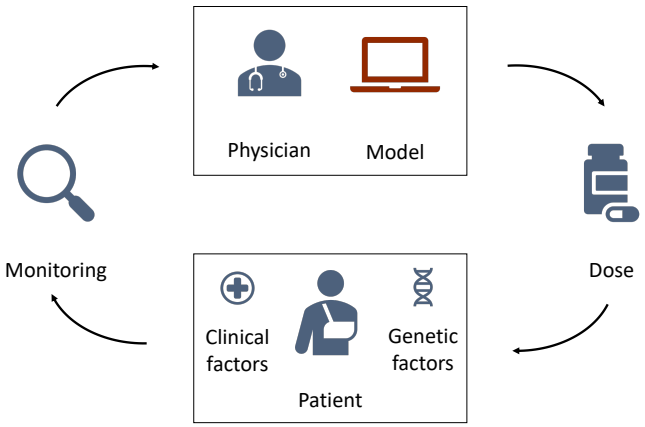


Figure 3.4: Schematic illustration of model-informed precision dosing. The top of the figure shows the treating physician and the MIPD model. The bottom of the figure shows the to-be-treated patient characterised by clinical and genetic factors. The physician uses the MIPD model and the patient characteristics to predict an individualised dose. If the treatment response is monitored over time, the physician can use the treatment response measurements and the MIPD model to iteratively refine the dose predictions.

warfarin dose administered daily to maintain a desired INR level. The maintenance dose is modelled as a function of the desired response, y^*, and the patient's covariates, χ,

$$d^* = d^*(\chi, y^*). \tag{3.10}$$

d^* denotes the maintenance dose. This makes it possible to predict individualised maintenance doses based on the covariates of an individual. In contrast to the general workflow in Figure 3.4, the model does not iteratively update its predictions using monitoring measurements. For more details on regression approaches to MIPD, we refer to Appendix B.2.

The model can be implemented using a variety of regression approaches, including linear regression, spline regression and tree regression (IWPC 2009). In this article, we choose a neural network approach. Neural networks are universal function approximators and can therefore learn to approximate the maintenance dose function in Equation 3.10 from data, even when the relationship between the dose and (χ, y^*) is nonlinear.

For those who are more familiar with neural network regression, in summary we compose the network of three sequential fully connected layers implemented in `PyTorch` (Paszke et al. 2019); the two inner layers of the network are of width 1024 and use

ReLU activations. The output layer uses a sigmoid activation. The network is trained on simulated trial data (see subsection 3.3.2) to minimise the mean squared error objective function using Adam (Kingma et al. 2014). For full details on the implementation and the training of the model, we refer to Appendix B.2.

2. Deep RL model – This MIPD approach is a deep reinforcement learning approach similar to Zadeh et al (Zadeh et al. 2023) where treatments are individualised using a model that iteratively predicts the next-to-administer dose. The dose is modelled as a function of the covariates and the current monitoring data

$$d_j = d_j(\chi, y_j). \tag{3.11}$$

d_j denotes the dose at time t_j and y_j denotes the INR measurement at time t_j. This makes it possible to iteratively predict individualised dosages based on monitoring data and the covariates of an individual, as illustrated in Figure 3.4. The target treatment response, y^*, is specified before the training of the model (Zadeh et al. 2023).

Conceptually, RL learns dosing strategies from trial and error: the model sequentially administers dosages and evaluates the 'goodness' of the dose decisions based on the feedback from the treatment response (see Chapter 2). The learned dosing strategy can be shown to optimally target the desired treatment response under certain technical assumptions and takes the form of a function for the next-to-administer dose (see Equation 3.11). We discuss the limitations of these assumptions in subsection 3.3.5. While trial and error in a clinical setting raises ethical concerns, RL models can also be trained on treatment response emulators (Ribba, Bräm, et al. 2022). Popular treatment response emulators are PKPD models (Zadeh et al. 2023). To reduce the number of trial and error iterations needed for convergence of the dosing strategy, RL can be performed in conjunction with function approximators (Baird 1995). In this article, we choose a deep neural network as the function approximator – an approach commonly referred to as deep RL. For more details on reinforcement learning approaches to MIPD, we refer to Appendix B.3.

For those who are more familiar with deep RL: we use a DQN model to implement the Deep RL model (Mnih et al. 2013). We compose the network of four sequential

fully connected layers implemented in `PyTorch` (Paszke et al. 2019): three hidden layers of widths (256, 128, 64) with ReLU activations, and an output layer of width 58. The outputs of the network are the predicted Q-values for 58 dose amounts that are considered by the model (Mnih et al. 2013). The dose with the maximum Q-value is suggested for administration. Following Zadeh et al, the network is trained online using a PKPD model as a treatment response emulator (Zadeh et al. 2023). Prior to the training, the PKPD model is fitted to simulated trial data (see subsection 3.3.2). We train the model to minimise the temporal difference error using Double Q-learning (Van Hasselt et al. 2016) and the Adam optimiser (Kingma et al. 2014). For full details on the implementation and training of the model, we refer to Appendix B.3.

3. PKPD model – This MIPD approach is a PKPD modelling approach similar to Hamberg et al (Hamberg, Hellman, et al. 2015) where treatments are individualised using a model that predicts patient-specific dosing regimens. The dosing regimen is modelled as a function of the covariates, the desired treatment response and the monitoring data

$$r = r(\chi, y^*, \mathcal{D}_j). \tag{3.12}$$

$\mathcal{D}_j = \{(y_1, t_1), \ldots, (y_j, t_j), r_j\}$ denotes the monitoring data available up to and including time t_j, where we use r_j to denote the administered dosing regimen for $t < t_j$. The function makes it possible to predict individualised dosing regimens based on monitoring data and the covariates of an individual (see Figure 3.4). The predictions can be iteratively refined as more monitoring data become available.

PKPD modelling is a semi-mechanistic modelling approach which makes use of approximate descriptions of the physiological and pharmacological processes to predict the treatment response dynamics (see Chapter 2). Conceptually, PKPD modelling is similar to the mechanistic model component of the CT model (see e.g. Wajima et al's model in Figure 3.2), but generally models the biological processes in lower detail. An example PKPD model is illustrated in Figure 2.4. By comparing the predicted and desired treatment responses, this approach is able to determine the optimal dosing regimens for each individual. Analogously to the CT model in subsection 3.2.3, inter-individual variability is described by differences in the model parameters across individuals. Patient-specific parameters are derived from the patient's covariates and monitoring data.

For those who are more familiar with PKPD modelling, in summary we use Hamberg et al's model (Hamberg, Wadelius, et al. 2010) implemented in `chi` (Augustin 2024) to model the warfarin treatment response dynamics. We fit the model to simulated trial data (see subsection 3.3.2) using hierarchical Bayesian inference and the No-U-Turn sampler (NUTS) (Hoffman et al. 2014) implemented in `pints` (Clerx, Robinson, et al. 2019). Individualised dosing regimens are predicted in two steps: 1. the model is fit to an individual's monitoring data using the population model and the individual's covariates as prior knowledge (Maier, Hartung, de Wiljes, et al. 2020); and 2. the dosing regimen is optimised to minimise the mean squared error between the model predictions and the desired treatment response. We use Bayesian inference and `pints`' implementation of the adaptive covariance matrix Markov chain Monte Carlo (ACMC) sampler to fit the model to an individual's data. For the dosing regimen optimisation, we use `pints`' implementation of the covariate matrix adaption evolution strategy (CMAES) optimiser (Hansen et al. 2003). For full details on the implementation and the dosing regimen prediction, we refer to Appendix B.4.

3.3 Results & discussion

We simulate three MIPD trials – one trial for each MIPD model – and analyse their relative strengths and limitations. To this end, we follow the workflow illustrated in Figure 3.1 and, first, fit the MIPD models to simulated clinical trial data to emulate a typical starting point for MIPD trials. The data imitate typical data collected during each of the three phases of clinical trials, and are simulated using the CT model. After the model fitting, we simulate the MIPD trials.

3.3.1 Simulating trial phases prior to MIPD

At present, most clinical trials are conducted not having MIPD in mind. As a result, the data available for the fitting of MIPD models will often not be tailored to the needs of the method. To reflect this practical limitation in our study, we simulate data from three typical phases of clinical trials for the warfarin use case, not taking the data-requirements for MIPD into account. All MIPD models in this article are fitted using only the data

from these trials. The simulated trial data as well as the code to reproduce the trials are hosted on GitHub (https://github.com/DavAug/mipd-warfarin).

Trial phase I: Phase I trials are primarily used to establish the safety of drugs and often monitor a drug's absorption, distribution, metabolism and elimination (ADME) in a relatively small cohort of patients. We emulate such a trial by mirroring a clinical trial reported by Hamberg et al (Hamberg, Dahl, et al. 2007). In this trial, the pharmacokinetics of $N = 60$ individuals is monitored. Each individual is administered with a single 10 mg dose of warfarin. Following the nominal administration time, the warfarin concentration in the blood is measured at 10 h, 35 h and 60 h after the administration. The data collected during the trial are the warfarin concentration measurements, the dosing regimens and the covariates for each of the 60 individuals. The simulated warfarin concentrations are illustrated in the left panel of Figure 3.5. The demographics of the cohort are reported in Appendix B.6. Pseudo-code outlining the implementation of the trial is presented in Algorithm B.1.

The panel shows the simulated warfarin concentration measurements. Measurements are illustrated using scatter points. Measurements taken from the same individual are connected using a solid line. The nominal administration time of the warfarin dose is indicated using a blue arrow.

Trial phase II: Phase II trials are primarily used to establish the efficacy of drugs and monitor a drug's pharmacodynamics. We simulate a phase II trial using a design similar to the trials reported by Hamberg et al (Hamberg, Wadelius, et al. 2010). We monitor the INR response of 100 simulated individuals for 3 weeks. Each individual is treated with daily warfarin doses. During the first 3 days, all individuals receive the same treatment: $d_1 = 10$ mg; $d_2 = 7.5$ mg; and $d_3 = 5$ mg. The 4th dose is adjusted for each individual based on the INR treatment response by a medical professional, here emulated using a simple linear heuristic: $d_j = d_{j-1}\, y^*/y_j$, where y_j denotes the INR measurement taken just before the j^{th} dose administration. This heuristic computes a personalised warfarin dose targeting the desired treatment response, y^*, assuming a linear relationship between the INR and the dose amount. The dose amounts are adjusted three more times for each individual on day 5, 7, and 13 of the trial using the

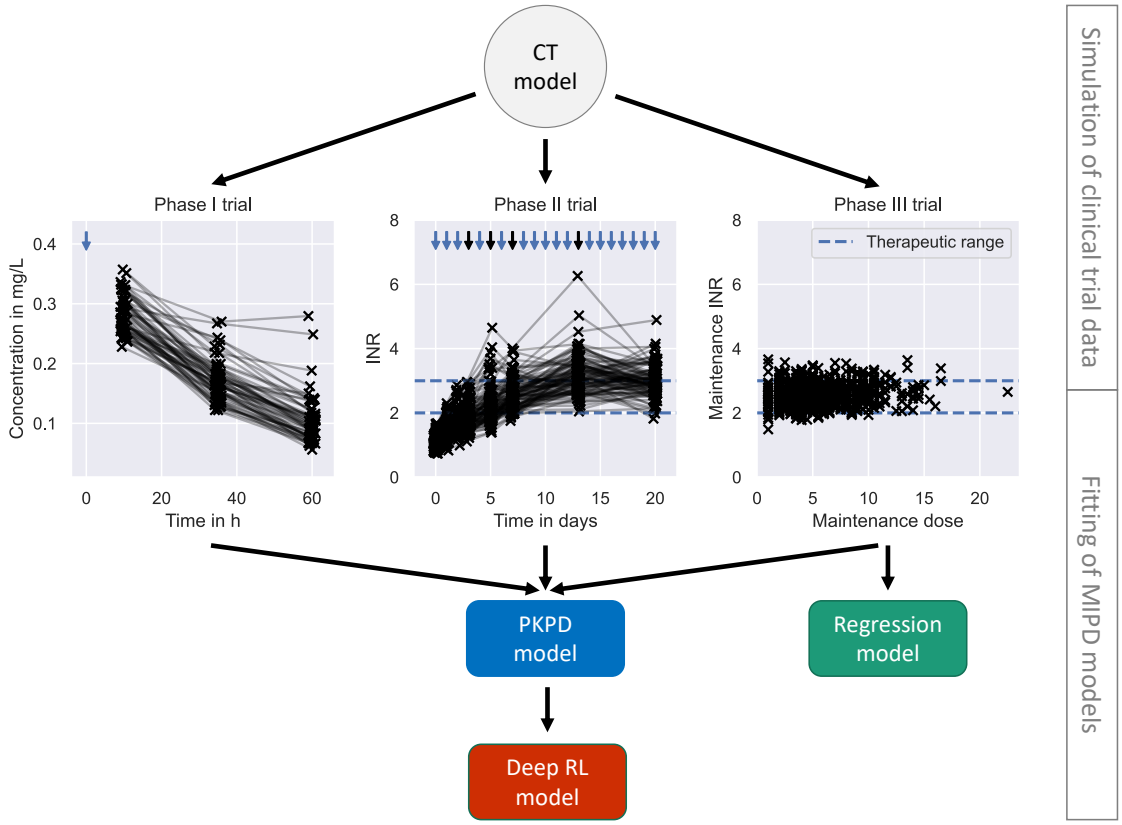


Figure 3.5: Pre-MIPD trial workflow. The figure shows the two steps performed prior to the MIPD trial simulation: 1. Simulation of clinical trial data (top); and 2. fitting of MIPD models (bottom). The CT model is used to simulate the trial data. The simulated data is illustrated in the middle of the figure: phase I (left); phase II (middle); phase III (right). Measurements of the blood warfarin concentration or the INR are illustrated using scatter points. Measurements connected by solid lines are taken from the same individual at different time points. Nominal administration times are illustrated by blue arrows. Dose administrations with individualised, on-the-fly adjustments of the dose amount are highlighted in black. The therapeutic range is illustrated in the middle and right panel using blue dashed lines. The bottom of the figure shows the fitting workflow of the MIPD models: The PKPD model is fitted to the data from all trials; the Regression model is fitted only to the data from clinical trial phase III; and the Deep RL model is fitted to treatment responses simulated with the PKPD model.

same heuristic. The INR of the individuals is closely monitored during the induction phase of the trial (day 0, day 1, day 2, day 3) and less frequently measured as the trial progresses (day 5, day 7, day 13 and day 20). To emulate safety constraints of real clinical trials, the trial is discontinued when an individual displays three consecutive INR measurements above 5. The data collected during the trial are the INR measurements, the nominal dosing regimens and the covariates for each of the 100 individuals.

The trial was not terminated early for any of the simulated individuals. The simulated INR measurements are illustrated in the middle panel of Figure 3.5. The demographics of the cohort are reported in Appendix B.6. Pseudo-code outlining the implementation of the trial is presented in Algorithm B.2. The panel shows the simulated INR measurements. Measurements are illustrated using scatter points. Measurements taken from the same individual are connected using a solid line. The nominal administration times of the warfarin doses are indicated using arrows. Black arrows indicate personalised adjustments of the dose amount. The therapuetic range is indicated using blue dashed lines.

Trial phase III: Phase III trials can vary in scope, but tend to involve larger cohorts and have a stronger focus on treatment endpoints. We emulate a phase III trial by mirroring a clinical trial reported in a study by the International Warfarin Pharmacogenetics Consortium (IWPC 2009). The focus of the trial is to understand the variability of the maintenance warfarin dose. We simulate the trial analogously to trial phase II, but with a larger cohort, $N = 1000$, and for a longer duration (8 weeks). Following the initial, identical phase of the trial, the daily doses are adjusted two more times on day 27 and 34. During the final three weeks of the trial, the dose amounts remain unchanged to guarantee that individuals equilibrate to the maintenance treatment response by the end of the trial. To emulate safety constraints of real clinical trials, the trial is discontinued when an individual displays three consecutive INR measurements above 5. The data collected during the trial are the INR measurements at the end of the trial (day 55), the maintenance warfarin doses, and the covariates for each of the 1000 individuals.

The trial was not terminated early for any of the simulated individuals. The simulated maintenance doses and INR measurements are illustrated in the right panel of Figure 3.5. The demographics of the cohort are reported in Appendix B.6. Pseudo-code outlining the implementation of the trial is presented in Algorithm B.3. The panel shows the maintenance dose on the x-axis and the measurement of the INR on the y-axis. Measurements are illustrated using scatter points. The therapeutic range is indicated using blue dashed lines.

3.3.2 Fitting the MIPD models

We fit the MIPD models to the simulated clinical trial data, as schematically illustrated in Figure 3.5. The fitting is the second step of the MIPD trial simulation workflow (see Figure 3.1). Only one of the models – the PKPD model – can be fitted to all of the available clinical trial data. The Regression model is fitted using the data from the phase III trial. The Deep RL model is trained indirectly on the trial data through simulations from the fitted PKPD model. For details on the fitting, we refer to subsection 3.2.5 and Appendices B.2, B.3, and B.4.

3.3.3 Simulating the MIPD trials

We simulate one MIPD trial for each of the three methods in subsection 3.2.5. All trials are conducted using the same cohort. The cohort includes $N = 1000$ individuals. The demographics of the cohort are modelled after a trial reported by Hamberg et al (Hamberg, Wadelius, et al. 2010) and are visualised in Figure 3.6. The figure shows the CYP2C9 genotype distribution, the VKORC1 genotype distribution and the age distribution in the cohort.

In each trial, individuals are treated with daily warfarin doses for 19 days. The dose amounts are individualised using the respective MIPD model, as described in subsection 3.2.5, and target a treatment response of $y^* = 2.5$. The data available for the individualisation are the INR measurements taken daily before each dose administration and the covariates of the individual. The Regression model individualises the treatments by predicting the maintenance dose based on the covariates of the patients. This maintenance dose is administered every day throughout the trial. The Deep RL model and the PKPD model predict the warfarin doses iteratively based on a patient's covariates and INR measurements. In contrast to the simulated trials in the previous section, we do not emulate safety constraints for the MIPD trials in order to expose possible weaknesses of the models more clearly. Pseudo-code outlining the implementation of the trial is presented in Algorithm B.4.

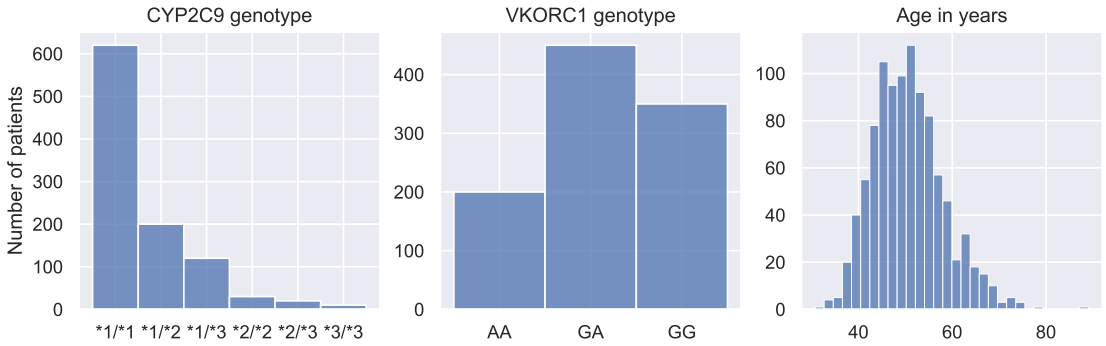


Figure 3.6: Demographics of MIPD trial cohort. The figure shows the CYP2C9 genotype distribution (left panel), the VKORC1 genotype distribution (middle panel) and the age distribution (right panel) of the MIPD trial cohort. The cohort includes 1000 simulated individuals.

3.3.4 Results of the MIPD trials

We use three metrics to quantify the success, the safety, and the efficiency of the dosing regimen individualisation: the maintenance INR; the peak INR; and the time in the therapeutic range (TTR). The success of the individualisation is quantified using the maintenance INR measured on the last day of the trial. INR measurements inside the therapeutic range indicate success, while measurements outside the therapeutic range indicate poor dosing regimen individualisation. The safety of the individualisation is quantified using the largest INR measurement recorded during the trial. This peak of the INR response indicates the risk for major bleeding events while transitioning into maintenance treatment. The efficiency of the individualisation is quantified using the TTR, i.e. the number of INR measurements inside the therapeutic range. For successful individualisations, the TTR indicates how quickly the desired treatment response has been achieved.

The results of the trials are visualised in Figure 3.7. Row 1 shows the results for the Regression model, row 2 shows the results for the Deep RL model, and row 3 shows the results for the PKPD model. The left panel shows the maintenance INR distribution in the cohort. INRs inside the therapeutic range (see blue dashed lines) are highlighted in blue and INRs outside the therapeutic range are highlighted in red. The target INR is illustrated using a black dashed line. The middle panel shows the

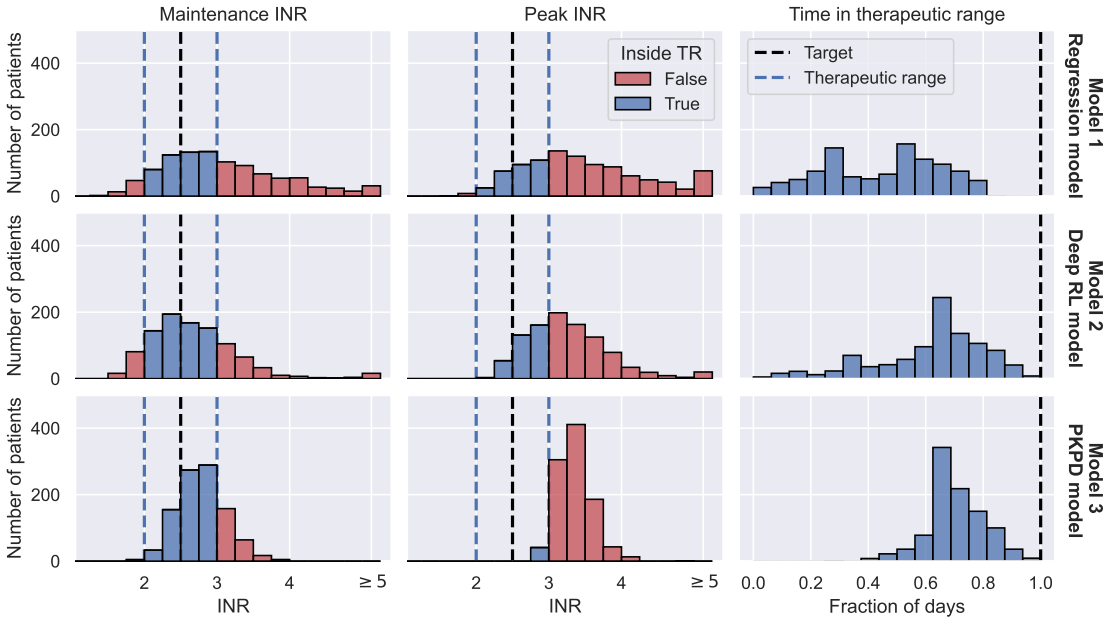


Figure 3.7: MIPD trial results. The figure shows the outcome of three MIPD trials conducted with an identical cohort of size $N = 1000$ with different MIPD models. The top row shows the results for the Regression model; the middle row shows the results for the Deep RL model; and the bottom row shows the results for the PKPD model. The outcome of the trials is illustrated using three metrics: the maintenance INR measured on the last day of the trial (left panel); the largest INR value measured during the trial (middle panel); and the time in the therapeutic range (right panel). The panels show the distributions of these metrics across individuals. The therapeutic range is illustrated using blue dashed lines. INR measurements inside the therapeutic range are highlighted in blue, and INR measurements outside the therapeutic range are highlighted in red. Target values of the dosing regimen individualisation are visualised using black dashed lines.

peak INR distribution in the cohort. The right panel shows the TTR distribution across individuals. The target TTR is illustrated using a dashed line.

The maintenance INR distributions (left panel) show that in a time span of 19 days all three MIPD methods are able to successfully target the therapeutic window for a large number of individuals. The Regression model successfully individualises the dosing regimen for 47.0 % of the individuals, while the Deep RL model and the PKPD model have success rates of 65.8 % and 75.1 %, respectively (see blue histograms). For the remaining individuals, the severity of the failed dosing regimen individualisation varies substantially. For the Regression model, 149 individuals display maintenance INRs above 4 with the largest value being 7.51. For the Deep RL model, only 32 individuals exceed

maintenance INRs of 4. However, the largest maintenance INR is 29.03 – a value almost 4 times larger than for the Regression model, raising serious safety concerns. For the PKPD model the largest maintenance INR is 3.90. 97.8 % of the individuals display maintenance INR measurements less than 3.5, showing that the PKPD model is able to most consistently achieve maintenance INRs close to the therapeutic window.

The peak INR distributions (middle panel) show that, for the majority of the cohort, the methods are not able to individualise the dosing regimens without overshooting the therapeutic window. In the Regression model trial, 67.1 % of the individuals display peak INR measurements above the therapeutic range. The Deep RL model controls the treatment response marginally better, overshooting the therapeutic range for 65.1 % of the individuals. The PKPD model misses the therapeutic window for almost all individuals (95.9 %) before reaching maintenance treatment. However, the panel also shows that the largest INR value measured in the PKPD model trial is substantially smaller than the largest INR values in the other two trials (Regression model: 7.99; Deep RL model: 29.03; PKPD model: 4.91), indicating that the PKPD model is the safest MIPD approach among the tested models.

The TTR distributions (right panel) show that the time spent inside the therapeutic range varies between individuals and MIPD approaches. For the Regression model, the median TTR is 45 % of the trial duration, with TTRs ranging between 0 % and 85 %. The other two methods achieve substantially larger TTRs across individuals. For the Deep RL model, the median TTR is 68 % of the trial duration, with individual values ranging between 5 % and 95 %. The PKPD model achieves a median TTR of 74 %, with a minimum TTR of 26 % and a maximum TTR of 100 %. This shows that across individuals the PKPD model takes the least time to successfully reach the therapeutic window.

3.3.5 Degrees of dosing regimen individualisation

The simulated trials in subsection 3.3.4 show that different MIPD approaches have different strengths and limitations. In this section, we study the dosing strategies of the models in more detail to gain a better understanding about their practical and methodological differences. We pay particular attention to attributing generic

strengths and limitations to the methodology and specific strengths and limitations to our implementation.

We investigate the dosing strategies by studying the dose decisions suggested by each of the models for three representative individuals from the simulated trials in subsection 3.3.4. The first individual, with ID 716, is characterised by the covariates $\chi_A = (*1*1, \text{GG}, 50)$. The other two individuals (ID 269; ID 305) are both characterised by the covariates, $\chi_B = (*1*2, \text{GA}, 50)$. The different dosing strategies and treatment responses are visualised in Figure 3.8. The figure shows the doses administered during the trials in the top panel and the corresponding treatment response measurements in the bottom panel. Doses or measurements belonging to the same individual are connected using solid lines. The therapeutic range is illustrated using dashed lines. The left panel shows the trial results for the Regression model, the middle panel shows the trial results for the Deep RL model, and the right panel shows the trial results for the PKPD model.

The Regression model

The left panel of the figure shows that the Regression model predicts a maintenance dose of 9 mg for the individual with ID 716 (black scatter points), and a maintenance dose of 4.5 mg for the other two individuals (see top left panel in Figure 3.8). These maintenance doses are administered daily throughout the trial. For the individual with ID 305, the maintenance INR measurement at the end of the trial is inside the therapeutic range, while the maintenance INR measurements for the other two individuals overshoot the therapeutic range (see bottom left panel in Figure 3.8). Notably, the individual with the successful individualisation (ID 305) has the same covariates as one of the individuals with the failed individualisation (ID 269).

This illustrates both a strength and a limitation of the Regression model: a strength of the model is that it is able to predict individualised dosages using information only about the covariates of individuals, making the model easier to implement in clinical practice than monitoring-based approaches. However, the figure also shows that dosing regimens exclusively derived from covariates will, at best, successfully target the desired

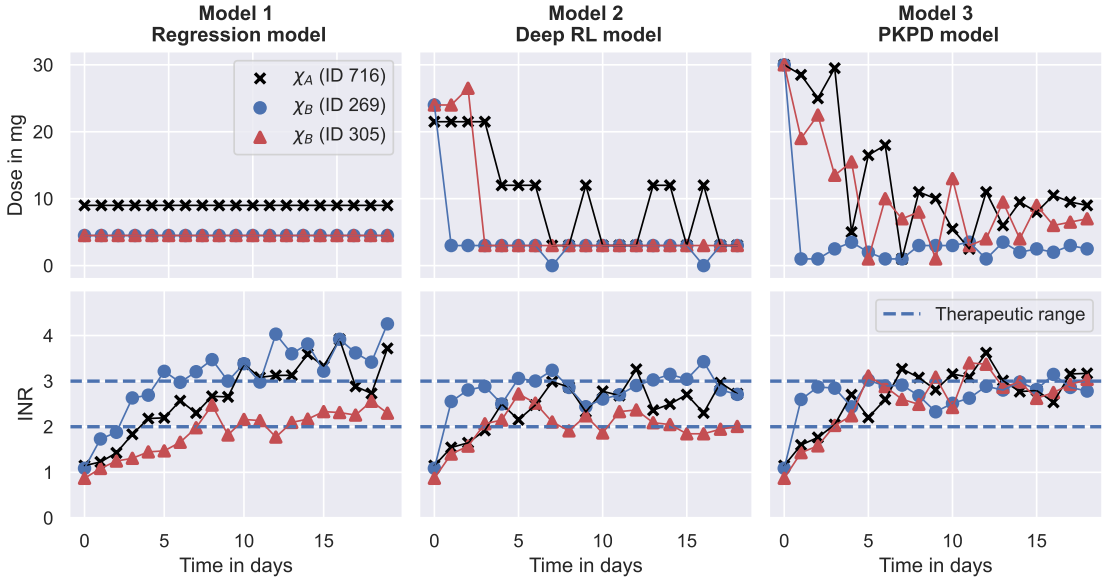


Figure 3.8: Degrees of dosing regimen individualisation. The figure shows the dosing regimen individualisations achieved by the MIPD models in the simulated MIPD trials for three representative individuals. The left panel shows the results for the Regression model, the middle panel shows the results for the Deep RL model, and the right panel shows the results for the PKPD model. The top row illustrates the administered dose amounts and the bottom panel illustrates the INR monitoring data. Dose amounts or measurements belonging to the same individual are connected using solid lines. The therapeutic range is illustrated using dashed lines. One of the individuals (ID 716) is characterised by the covariates $\chi_A = (*1*1, \text{GG}, 50)$. The other two individuals (ID 269; ID 305) are both characterised by the covariates $\chi_B = (*1*2, \text{GA}, 50)$.

treatment response for an average individual characterised by the covariates. Inter-individual differences that are not explained by covariates are not accounted for. In this case, the individual with ID 305 happens to respond similarly to an average individual[1] with the covariates, χ_B, resulting in a successful dosing regimen individualisation. The individual with ID 269, on the other hand, responds more strongly to warfarin than an average individual[2], yielding a maintenance INR above the therapeutic range. Not being represented by an average individual also explains the failed individualisation for the individual with ID 716. The inability to account for unexplained IIV is a generic limitation of approaches exclusively based on covariates.

[1] In this case, the close-to-average response can be explained by the individual's model parameters, ψ, being close to the means of the population distribution (Equation 3.2).

[2] Similarly to the above footnote, the stronger-than-average response can be explained by key model parameters being further away from the means of the population distribution.

In addition to this limited ability to account for IIV, the figure also shows that the Regression model is incapable of accounting for inter-occasional variation and differences in the execution of the treatment. This limitation is, again, a direct consequence of exclusively using covariates for the dosing regimen individualisation. Without quantifying the individual-specific IOV and EV, predicted dosing regimens can, at best, be successful when the IOV and the EV of the treated individual happen to be close to the average IOV and EV observed in the dataset used for the model fitting.

Another limitation, illustrated in Figure 3.8, is that the Regression model cannot account for treatment response delays. The model only predicts maintenance dosages, and therefore provides no guidance for individualisation of the induction phase of the treatment. In our implementation, we choose to overcome this limitation of the model by administering the predicted maintenance dosages from the beginning of the trial, not attempting to individualise induction dosing regimens without model guidance. This has the consequence that treatment responses take time to reach their maintenance level, limiting the efficiency attainable by the model. From Figure 3.8, we can, for example, see that the INR measurements of the individual with ID 305 reach the therapeutic range for the first time on day 8 of the treatment. After that, two more measurements are outside the therapeutic range, giving rise to a TTR of 10 days during the 19 day trial. Together with the top right panel in Figure 3.7, this indicates that even when the Regression model individualises maintenance dosages successfully, it achieves, at best, an average TTR of around $55\,\%$-$65\,\%$. This limit to the efficiency is specific to the treatment response delay of warfarin and our implementation of the Regression model.

A summary of the strengths and limitations specific to the Regression model are presented in the left column of Table 3.1.

The Deep RL model

In comparison, the Deep RL model is able to predict more individualised dosing regimens than the Regression model (see middle panel in Figure 3.8). The panel shows that for all three individuals, the model begins the treatment with elevated warfarin dosages of more than 20 mg. During the maintenance phase, it reduces the dosages. For the

	Regression model	Deep RL model	PKPD model
Requires covariates	✓	✓	(✓)
Requires monitoring	✗	✓	(✓)
Requires indefinite monitoring	✗	✓	✗
Accounts for explained IIV	✓	✓	✓
Accounts for unexplained IIV	✗	✓	✓
Accounts for IOV	✗	✓	✓
Accounts for EV	✗	✓	✓
Accounts for response delays	✗	✗	✓
Robust to unseen responses	✓	✗	✓
Robust to misspecification	✓	✗	✗
Fully individualises regimens	✗	✗	✓

Table 3.1: Summary of the strengths and limitations of the MIPD models. The properties of the models are grouped into data-related properties (rows 1-3), variability-related properties (rows 4-8), and strategy-related properties (rows 9-13). The parentheses around the property of the PKPD model in the top right corner indicate that the model, as defined in Section 3.2.5, can be used without covariates, but, in the simulated MIPD trial in Section 3.3.4, we did not explore this possibility. Similarly, parentheses around the property below indicate that the PKPD model can be used without the use of monitoring data.

individual with covariates χ_A (ID 716), the model alternates between administering dosages of 12 mg and 3 mg. For the other two individuals with covariates χ_B, the model administers a constant dose of 3 mg, from which it only occasionally deviates for one of the individuals (ID 269). Overall, the figure shows that the Deep RL model is more successful in individualising the dosing regimens of the individuals, as indicated by their maintenance INR measurements at the end of the trial. The individualisation of the induction phase reduces the time needed to reach the therapeutic range, leading to all three individuals displaying INR measurements inside the therapeutic range within the first four treatment days.

The main reason for the improved performance of the Deep RL model is the use of feedback control from the monitoring data in addition to the covariate information. The model derives predictions from both monitoring data and covariates using the dose function, defined in Equation 3.11, which predicts the next-to-administer dose based on the most recently measured INR value and the covariates of the to-be-treated individual. We visualise the predictions of the dose function for the three individuals

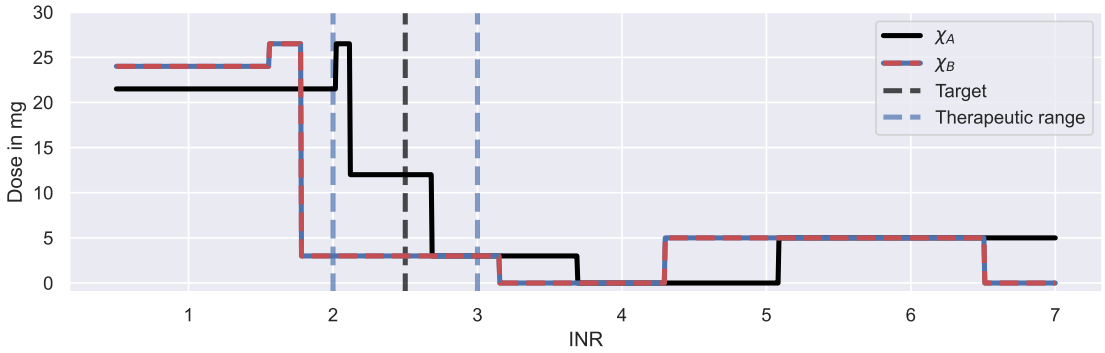


Figure 3.9: Dosing strategy of the Deep RL model. The figure illustrates the dose function of the Deep RL model, defined in Equation 3.11, for a range of possible INR monitoring measurements and two sets of covariates: $\chi_A = (*1*1, \text{GG}, 50)$ (black line); and $\chi_B = (*1*2, \text{GA}, 50)$ (red-blue line). The function determines the next-to-administer dose based on the most recently measured INR value (bottom axis) and the covariates of an individual. The target treatment response is illustrated using a black dashed line. The therapeutic range is indicated using blue dashed lines.

in Figure 3.9. For illustrative purposes, the figure focuses on INR values between 0.5 and 7. The predicted dose values are illustrated using a black line for individuals with the covariates χ_A and a red-blue line for individuals with the covariates χ_B. The target treatment response is illustrated using a black dashed line, and the therapeutic range is indicated using blue dashed lines.

The figure shows that the Deep RL model has a clear strategy for INR monitoring measurements below 4, and a less clear strategy for INR measurements above 4. We will first focus on the dose decisions for INRs below 4: for INR measurements below the therapeutic range, the model administers large warfarin doses; for INRs inside the therapeutic range, the model administers intermediate warfarin doses; and for INRs above the therapeutic range, the model administers low warfarin doses. The exact change points and dose amounts are specific to the covariates. For example, for individuals with the χ_A covariates, the model starts the treatment with a dose of 21.5 mg and switches to an intermediate dose of 12 mg for INR measurements between 2.1 and 2.7 (see black line in Fig 3.9). This dosing strategy is consistent with the dose decisions observed in Figure 3.8.

The dose function illustrates both a strength and a limitation of the Deep RL

model (see Figure 3.9). A strength of the model is that it bases its dose decisions on a simple and interpretable feedback control mechanism: when INR values are too low, it increases the warfarin dose; and when INR values are too high, it decreases the warfarin dose. These dose decisions are tailored to an individual based on the individual's covariates. This strategy leads to a substantially higher success rate and efficiency of the dosing regimen individualisation relative to the Regression model (see Figure 3.7), as the feedback control enables the model to maintain treatment responses within a desired range, even when unexplained IIV, IOV and EV are present. This strength is generic to reinforcement learning approaches.

However, while the ability to utilise monitoring data for control is a strength, the model's inability to learn from the measurements is a limitation of the Deep RL model, making the approach indefinitely reliant on monitoring data. The reliance on monitoring data is specific to the Deep RL model and is a consequence of its learning approach: the model establishes the dose function (Equation 3.11) prior to the treatment of individuals (see subsection 3.3.2), never individualising the dose function based on the individual-specific monitoring data. Instead, the model treats individuals with dosing strategies exclusively based on covariates, where the monitoring data is only used as a control mechanism to steer treatment responses back to the desired treatment response, if needed (see e.g. the dosing strategy in Figure 3.8 for ID 269). In principle, reinforcement learning approaches can update the dose function using individual-specific monitoring data (Maier, Hartung, Kloft, et al. 2021). However, for the Deep RL model, meaningful updates of the dose function are challenging, as the large number of model parameters of its neural network complicates the balance between fine-tuning and overfitting to the limited number of measurements.

The absence of fully individualised dosing strategies explains the observed tendency of the Deep RL model to overshoot the therapeutic range (see middle panel in Figure 3.7): to successfully target the therapeutic range across individuals with the same set of covariates, the initial warfarin dose predicted by the dose function needs to be high enough, so that even the weakest responders in a subpopulation reach the therapeutic range. This dose leads to INR responses above the therapeutic range for strong

responders in the same subpopulation due to the substantial level of warfarin IIV not explained by covariates. The over-treatment of the strong responders is compensated for by reducing the dose for INRs above the therapeutic range so much that even the strongest responders are guaranteed to regress back to the therapeutic range (see large dose steps in Figure 3.9). This explains the substantial fraction of individuals with peak INRs above the therapeutic range in Figure 3.7, and points to a general lack of precision of the Deep RL model. This 'control over precision' strategy is a generic limitation of reinforcement learning approaches that do not individualise their dose functions based on the individual-specific monitoring data.

The dose function in Figure 3.9 also demonstrates that the Deep RL model can fail to learn meaningful dosing strategies for the full range of possible INR measurements. When INR measurements become large (≥ 5.05 for χ_A; and ≥ 4.3 for χ_B), the model suggests increasing the warfarin dose to 5 mg, despite the fact that higher warfarin dosages inevitably lead to even higher INR values. The root for these dose decisions lies in the training of the Deep RL model (see subsection 3.3.2): large INR monitoring measurements remain under-explored during the training, leading to poorly tested dose decisions for large INR measurements. Those dose decisions can remain without consequence, when the model is able to control treatment responses well enough to never reach large INR values (see middle panel in Figure 3.8). But when an individual responds more strongly to warfarin than expected, for example due to unexplained IIV, IOV or EV, poorly tested dose decisions for large INR values can cause a failure of the model's feedback control mechanism. This failure explains the severe over-treatments of a few individuals observed in the simulated MIPD trial in Figure 3.7.

The lack of exploration during the training, despite the use of a standard training procedure (see ϵ-greedy policy in Appendix B.3), is the consequence of a mismatch between the technical assumptions of the Deep RL model and the reality of warfarin treatment responses: the Deep RL model assumes that there is no delay between dose administrations and the feedback from INR measurements, when, in fact, warfarin treatment responses have delays of up to 10 days (see e.g. Figure 3.3). This reduces the ability of the standard training procedure to contribute to the exploration of large

INR measurements. The mismatch between the model assumptions and the treatment response delay is generic to reinforcement learning models and difficult to overcome, as the convergence and optimality of reinforcement learning centrally relies on the assumption that transition dynamics can be modelled as a Markov decision process with i.i.d. actions and i.i.d. states (Sutton et al. 2018). For the Deep RL model, this implies that the model has to assume that INR measurements on one day depend only on the INR measurement and the administered dose on the previous day, i.e. any dose administrations and INR measurements on earlier days cannot be taken into account. Despite those technical constraints, Zadeh et al show that their deep reinforcement learning model is able to improve its performance, when the i.i.d. assumption of the states is explicitly violated and dose decisions are conditioned on more than one recent INR measurement (Zadeh et al. 2023). As a result, the technical assumptions of reinforcement learning, needed for convergence and optimality, may limit the ability of reinforcement learning models to account for treatment response delays in theory, but in practice, it may be possible to overcome those limitations (Gaon et al. 2020).

A summary of the strengths and limitations specific to the Deep RL model are presented in the middle column of Table 3.1.

The PKPD model

The right panel in Figure 3.8 shows that the PKPD model achieves the highest degree of dosing regimen individualisation among the tested models. The model begins the treatment for all three individuals with a high dose of 30 mg, followed by a gradual reduction of the dose over the following days. For the individual with ID 269, this dose reduction is more rapid than for the other two individuals. Towards the end of the trial, the dosages converge to constant, individual-specific maintenance dosages. The maintenance INRs are located at the upper threshold of the therapeutic range in close proximity to each other, indicating success of the dosing regimen individualisation. All three individuals display INR measurements inside the therapeutic range within the first three treatment days.

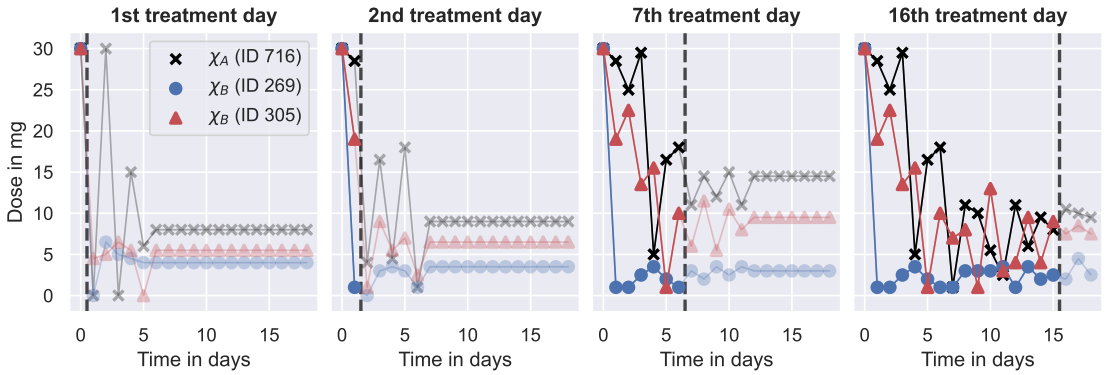


Figure 3.10: Dosing strategy of the PKPD model. The figure illustrates the dosing regimen function of the PKPD model, defined in Equation 3.12, for the three individuals from Figure 3.8 on four days of the trial: on the 1st treatment day (panel 1), on the 2nd treatment day (panel 2); on the 7th treatment day (panel 3) and on the 16th treatment day (panel 4). Dosages are illustrated using scatter points in two opacity levels: opaque scatter points for already administered dosages; faded scatter points for future dose administrations. Dosages belonging to the same individual are connected using solid lines. The day of the treatment is illustrated using black dashed lines.

The main reason for the good performance of the PKPD model is its use of both covariate information and monitoring data similar to the Deep RL model. The model uses the covariates to predict initial dosing regimens that target the desired treatment response for average individuals. These initial dosing regimens are further individualised based on the individual-specific monitoring data, achieving an increasing degree of individualisation as more monitoring data becomes available. This increasing degree of individualisation provides a distinct advantage over the other two MIPD models. The PKPD model derives predictions from both monitoring data and covariates using the dosing regimen function, defined in Equation 3.12, which predicts individualised dosing regimens based on all available INR measurements of the to-be-treated individual, the covariates, and the already-administered dosages. We visualise the values of the dosing regimen function for the three individuals in Figure 3.10. For illustrative purposes, the figure focuses on four days of the simulated trial: day 1; day 2; day 7; and day 16. The predicted dose values are illustrated using scatter points in two opacity levels: opaque scatter points indicate already administered dosages; and faded scatter points indicate future dose administrations. The treatment day is illustrated using dashed lines.

The figure shows that the dosing regimen predictions are iteratively updated as the treatment progresses. On day 1, the model provides rough estimates of the dosing regimens, scheduling maintenance dosages of 8 mg (ID 716), 4 mg (ID 269), and 5.5 mg (ID 305) after an initial induction phase of the treatment. With the next INR measurement on day 2, both the induction dosing regimen as well as the maintenance dosages are updated (see second panel in Figure 3.10). On day 16, the model predicts fully individualised dosing regimens which are almost identical to the dosing regimens administered in the simulated trial (see right panel in Figure 3.8).

The dosing regimen function illustrates both a strength and a limitation of the PKPD model (see Figure 3.10). A strength of the model is that it predicts full dosing regimens from any number of monitoring measurements, making the dosing schedule transparent, foreseeable, and less reliant on frequent or regular monitoring. The prediction of full dosing regimens is enabled by the model's explicit description of the pharmacological processes in terms of a semi-mechanistic model. This permits the prediction of treatment responses, and thus optimal dosing regimens, for any future time points. It also helps the model to account for treatment response delays and nonlinearities of the dose-response relationship (see subsection 3.2.5). However, the explicit model of the treatment response bears a risk for model misspecification (Merlé et al. 2004). In particular, neglecting or oversimplifying important treatment response mechanisms can lead to inaccurate treatment response predictions, which, in turn, can impact the quality of the dosing regimen individualisation. For example, the results from the simulated MIPD trial show that the PKPD model tends to administer too large warfarin dosages during the trial, resulting in a systematic bias towards INR measurements larger than the target INR (see bottom panel of Figure 3.7). This indicates that the PKPD model oversimplifies crucial elements of the treatment response mechanisms, resulting in a tendency to underestimate the treatment response of individuals. The risk for model misspecification is a generic limitation of PKPD modelling, which needs to be mitigated prior to clinical applications, for example by quantifying the structural uncertainty of PKPD models using model selection criteria or probabilistic model averaging (Augustin, Walz, et al. 2022; Uster et al. 2021).

The dosing regimen function in Figure 3.10 also illustrates that the updates of the dosing regimens themselves can result in a bias of the treatment strategy. The comparison between the predicted dosing regimens shows that although more individual-specific monitoring measurements are available on day 7 of the treatment, the maintenance dosages predicted on day 2 are closer to the actual maintenance dosages administered towards the end of the trial. This suggests that the quality of the dosing regimen individualisation can temporarily decrease with the number of monitoring measurements – a limitation specific to our implementation of the dosing regimen individualisation. The potential for worse dosing regimens despite more monitoring data is related to the estimation of the individual-specific model parameters from the monitoring data: we estimate only one set of model parameters instead of considering all model parameters consistent with the monitoring data (see subsection 3.2.5). Using Bayesian inference, we estimate a distribution of parameter values consistent with the data, also known as posterior distribution. In our implementation, we estimate the individual-specific model parameters using the modes of this distribution, also known as maximum a posteriori (MAP) estimates. MAP estimates are a popular choice to reduce posterior distributions to just one set of model parameters (Sheiner, Beal, et al. 1979). However, by disregarding the other model parameters that are also consistent with the data, it is possible to introduce biases in the treatment response predictions with consequences for the dosing regimen individualisation. In particular for nonlinear treatment responses, the treatment response predicted with the MAP estimates is, generally, not the treatment response that maximises the probability of the posterior predictive distribution (Maier, Hartung, de Wiljes, et al. 2020). This increases the risk for inaccurate treatment response predictions.

A bias of the MAP-based treatment response predictions explains the lack of safety observed during the simulated MIPD trial (see bottom panel in Figure 3.7). The MAP-based predictions tend to underestimate the warfarin treatment response for small INR values which leads to individualised dosing regimens with elevated dose amounts early in the trial. As the treatment progresses, the uncertainty about the model parameters becomes smaller, reducing the error of the MAP estimation, and with it, the bias of the treatment response predictions (see maintenance INR distribution in Figure 3.7).

This bias is also supported by the goodness-of-fit plot from the model fit to the pre-MIPD trial data in subsection 3.3.2, where also the predictions with the maximum probability parameters of the population distribution show a tendency to provide biased treatment response predictions (see middle and bottom right panel in Figure B.11). This limitation of our PKPD model can be mitigated by predicting the treatment response with each parameter set that is consistent with the data, i.e. the parameters in the posterior distribution, producing a distribution of treatment responses which reflects the uncertainty in the treatment response predictions (Maier, Hartung, de Wiljes, et al. 2020). The distribution of treatment responses can be optimised to obtain a dosing regimen with less risk for bias (Maier, Hartung, Kloft, et al. 2021).

A summary of the strengths and limitations specific to the PKPD model are presented in the right column of Table 3.1.

3.4 Conclusion

Simulated clinical trials provide a resource-efficient way to test and develop fit-for-purpose models for precision dosing. We show that we can emulate clinical trials using a clinical trial model with five independent model components: 1. a mechanistic model; 2. a population model; 3. an inter-occasion model; 4. an execution model; and 5. a measurement model. Each model component captures a different complexity of clinical practice that challenges the successful individualisation of treatments, ranging from PKPD-related challenges, such as nonlinear and delayed treatment responses, to practical challenges, such as unintentional deviations from nominal dosing schedules (see subsection 3.2.1). The modularity of the model components simplifies the development process of the clinical trial model, allowing for independent updates of its components throughout the drug development pipeline. This makes it possible to iteratively improve the trial simulations and develop a promising companion MIPD tool as more understanding and information about the drug under trial becomes available (Polasek, Rayner, et al. 2019).

Simulating trials for precision dosing of warfarin, we find that different MIPD models have different strengths and limitations. These strengths and limitations can be generic

to the methodology or specific to the model implementation. Modelling approaches that predict dosing regimens exclusively based on covariates of the treatment response variability are generically limited when unexplained IIV, IOV and EV are present (see section 3.3.5). However, when the majority of the treatment response variability can be explained by covariates, and those covariates are available in clinical practice, such approaches provide an excellent solution to the individualisation of dosing regimens. Otherwise, MIPD models based on monitoring data are better at accounting for treatment response variability, and achieve a higher degree of dosing regimen individualisation in the simulated warfarin trials (see subsections 3.3.4 and 3.3.5).

But there are also challenges with monitoring-based MIPD approaches. We find that the Deep RL model adopts a 'control over precision' treatment strategy, where doses are adjusted based on the feedback response from the monitoring data with limited foresight (see section 3.3.5). This lack of precision makes the model reliant on indefinite monitoring and limits its ability to account for treatment response delays. Deep reinforcement learning may, nevertheless, provide a good solution to the individualisation of dosing regimens for applications where monitoring is not a challenge, such as for treatments in the intensive care unit (Moore et al. 2004) or for dosing devices that are physically attached to patients, like insulin pumps (T. Zhu et al. 2020).

The PKPD model achieves the highest degree of dosing regimen individualisation among the tested models. The model predicts fully individualised dosing regimens whose level of individualisation increases with the amount of available monitoring data. This indicates that across applications PKPD models are the most promising approach for precision dosing. However, PKPD models are more susceptible to model misspecifications than the other two approaches (see section 3.3.5), necessitating a careful evaluation of the predictive uncertainty prior to MIPD, for example by means of model selection criteria or probabilistic model averaging (Augustin, Walz, et al. 2022).

Overall we find that MIPD approaches vary in their ability to account for the different sources of treatment response variability, meaning an ideal approach depends on the context. Distinguishing four sources of treatment response variability – 1. unexplained IIV; 2. explained IIV; 3. IOV; and 4. EV – our results suggest that

PKPD modelling is more successful than the other two approaches in cases where the treatment response variability is dominated by unexplained IIV. PKPD models account for unexplained IIV by individualising treatment response predictions from just a small number of monitoring measurements. We expect PKPD models to also be the favourable choice when the treatment response variability is dominated by EV, as the other tested models cannot adapt their predictions to irregular administration and monitoring schedules. However, when IOV dominates the treatment response variability, deep reinforcement learning may perform best, as its 'control over precision' dosing strategy is agnostic to random fluctuations in the treatment response. Lastly, when the treatment response variability is dominated by explained IIV, all three models likely provide similar maintenance predictions, making regression the preferred choice as long as treatment response delays are not a concern.

While the sources of the variability can indicate which MIPD approach may perform best, the volume and type of the available data determine which MIPD approach is possible. PKPD modelling works with both large and limited amounts of data, provided the treatment response mechanisms are well known and the available data can be linked to the processes involved. In contrast, regression and deep reinforcement learning need more data to be feasible but can process more complex data types and do not rely on a mechanistic understanding of the treatment response.

3.5 Data and software availability

The data, models, and scripts used in this chapter are fully documented and hosted on GitHub (see repository: https://github.com/DavAug/mipd-warfarin), making the presented results easily reproducible. A user-friendly API for MIPD approaches using PKPD modelling has been implemented in the open source Python package `chi` (Augustin 2024).

4

Filter inference: A scalable nonlinear mixed effects inference approach

This chapter is based on our publication

DA, Lambert B, Wang K, Walz AC, Robinson M, Gavaghan D. Filter inference: A scalable nonlinear mixed effects inference approach for snapshot time series data. *PLOS Computational Biology*. 2023.

Contents

4.1 Introduction . . . 139
4.2 Methods . . . 141
4.2.1 Nonlinear mixed effects inference . . . 143
4.2.2 Filter inference . . . 146
4.2.3 Filters . . . 152
4.3 Results & discussion . . . 155
4.3.1 Early cancer growth model inference . . . 155
4.3.2 EGF pathway model inference . . . 158
4.3.3 Scaling & computational costs . . . 160
4.3.4 Sampling efficiency . . . 164
4.3.5 Relationship to ABC and BSL . . . 166
4.3.6 Information loss and bias . . . 169
4.4 Conclusion . . . 175
4.5 Data and software availability . . . 176

4.1 Introduction

In the previous chapter (Chapter 3), we have seen that the PKPD modelling approach to MIPD is the most promising for a successful dosing regimen individualisation across many applications. In the remainder of this thesis, we will therefore exclusively focus on PKPD modelling. More specifically, in this chapter, we will develop an inference method, which we call 'filter inference', that reduces the computational costs of estimating NLME model parameters from measurements originating from a large number of individuals. This new method makes it computationally feasible to estimate suitable prior distributions for MIPD from sparsely collected data across a large number of patients, making the improved estimation of inter-individual variability from measurements collected as a byproduct of post-market safety monitoring, at least in principle, possible.

However, the applications of filter inference extend beyond MIPD as variability is an intrinsic property of biological systems and is often the reason for their complex behaviour (Hallgrimsson et al. 2005). Examples are plentiful. One is the evolution of organisms, whereby variability in the genetic material across individuals is one of the key drivers for adaptation of populations (Mueller 2019). Another is the human adaptive immune system, wherein variability in the antigen binding sites across antibodies is crucial for the defence against a large variety of pathogens (Abbas et al. 2021). However, variability in the function and regulation of cells is also the cause of many diseases, such as cancer and Alzheimer's disease (Hanahan et al. 2000, 2011; A. Kumar et al. 2015). Quantifying variability is therefore central to understanding many biological systems.

As introduced in subsection 2.2.7, nonlinear mixed effects (NLME) modelling is a popular approach to model variability in populations (Lavielle 2014; Ribba, Holford, et al. 2014). NLME models introduce a set of model parameters which typically need to be estimated from measurements. However, the inference of NLME models from measurements quickly becomes prohibitively expensive when the number of measured individuals increases (Hasenauer et al. 2011). This shortcoming is particularly limiting when individual entities can only be measured once, since such 'snapshot' measurements do not capture individual trajectories and are therefore relatively uninformative about the dynamics across individuals, requiring large numbers of snapshot measurements

for good inference results. In this chapter, we introduce a novel inference approach: Filter inference. We demonstrate that filter inference provides a scalable inference approach for snapshot time series measurements.

Snapshot measurements are particularly common in cell biology, where experimental techniques, such as single-cell RNA sequencing and flow cytometry, provide high-throughput measurements without the possibility to repeatedly measure individual cells (Adan et al. 2017; A. J. Hughes et al. 2014; Luecken et al. 2019). The availability of snapshot measurements paired with the limitations of the NLME inference has led to the development of a variety of inference methods. Hasenauer et al. (2011) simulate measurements to construct an approximate likelihood for the model parameters using kernel density estimation (KDE). To this end, they make explicit assumptions about the population parameter distribution. P. D. Dixit et al. (2020) use the simulated measurements to fit the histogram of the observed measurements exactly, making no explicit assumptions about the population parameter distribution. Instead, they require that the entropy of the model parameter distribution is maximised. Lambert, Gavaghan, et al. (2021) use exhaustive simulations from a prior distribution of the model parameters to construct a contour volume distribution which enables an efficient inference of the model parameters. Similar to Dixit et al's method, Lambert et al's approach also does not make any explicit assumptions about the population parameter distribution. However, it does require the simplifying assumption that measurement noise is negligible for the inference. Browning, Ansari, et al. (2022) use an ABC inference approach to infer NLME models from snapshot measurements, where summary statistics of simulated and observed measurements are compared. A comprehensive treatment of estimating NLME model parameters with likelihood-free methods, such as ABC and the Bayesian synthetic likelihood (BSL) approach, is provided by Drovandi et al. (2022).

With filter inference, we extend Hasenauer et al's inference approach. We show that the simulation-based KDE approximation of the likelihood can be generalised to approximating the likelihood with any distribution based on simulated measurements. The choice of the distribution acts as an information filter for the comparison between simulated and real measurements and influences the quality of the inference results.

We show that choosing distributions, taking into account the nature of the problem, can improve the inference results. The distribution or *filter* choice in filter inference has similarities with the choice of summary statistics in ABC or BSL. We use this similarity to systematically study the properties of filter inference and the consequences of different filter choices on the parameter estimates. We also introduce a differentiable form of the approximate likelihood, making filter inference applicable for state-of-the-art gradient-based sampling algorithms, such as Hamiltonian Monte Carlo (HMC) and the No-U-Turn sampler (NUTS) (Betancourt 2017; Hoffman et al. 2014; Neal 2010). This improves the inference efficiency, especially for NLME models with many parameters.

The body of this chapter is divided into two sections: a methods and a results section. In the methods, we review the NLME modelling framework and introduce filter inference. In the results, we demonstrate the performance of filter inference for two NLME inference problems, which both suppose access to snapshot measurements: 1. for an early cancer growth model; and 2. for an epidermal growth factor (EGF) pathway model. We also use these modelling problems to demonstrate the reduction of the computational costs when using filter inference, and draw comparisons between filter inference, ABC and BSL. We conclude the article by addressing potential sources for information loss and bias.

4.2 Methods

NLME models account for the dynamics of heterogeneous populations using a hierarchical modelling structure (Lavielle 2014; Ribba, Holford, et al. 2014). First, a time series model, $\bar{y}(\psi, t)$, is used to model the dynamics of an individual. Here, $\bar{y}$ denotes a quantity of interest, t denotes the time and ψ denotes the parameters of the model. An example time series model for early cancer growth is illustrated in red in Figure 4.1, where the quantity of interest,

$$\bar{y}(\psi, t) = y_0 \mathrm{e}^{\lambda t}, \tag{4.1}$$

captures a patient's tumour volume over time. The parameters of the model, $\psi = (y_0, \lambda)$, are the initial tumour volume and the growth rate.

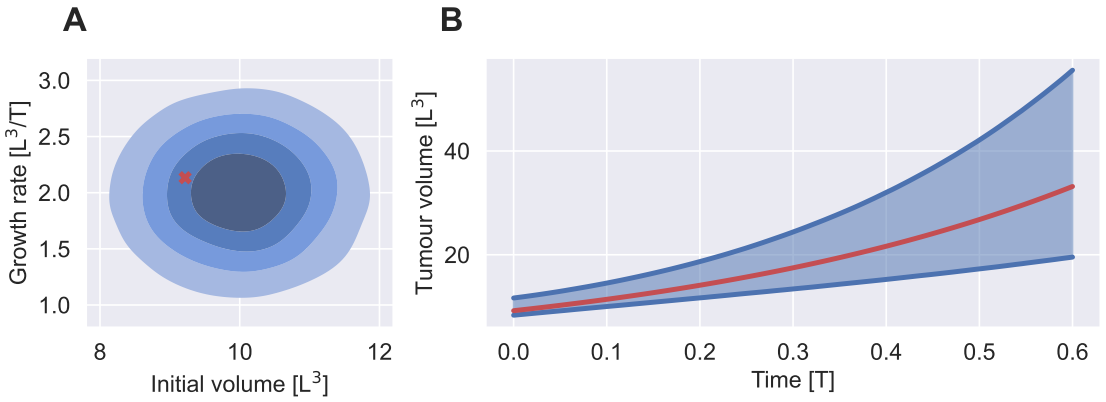


Figure 4.1: NLME model of early cancer growth. A: Shows the population model, $p(\psi|\theta)$, for $\theta = (10, 1, 2, 0.5)$ in shades of blue and the parameters of a randomly chosen individual in red. The shades of blue indicate the bulk $20\,\%$, $40\,\%$, $60\,\%$ and $80\,\%$ probability of the distribution. B: Shows the distribution of tumour volumes across individuals in the population, $p(y|\theta, t)$, in blue and the tumour volume of a randomly chosen individual over time in red. The blue lines indicate the 5th and 95th percentile of the tumour volume distribution at each time point. The quantities are shown in arbitrary units. L denotes length dimensions and T denotes time dimensions.

Second, a population model, $p(\psi|\theta)$, with population parameters θ is used to capture the inter-individual variability (IIV) by modelling the distribution of ψ in the population. For example in Figure 4.1A, the initial tumour volume and the growth rate are normally distributed across patients

$$p(\psi|\theta) = \mathcal{N}(y_0|\mu_{y_0}, \sigma^2_{y_0})\,\mathcal{N}(\lambda|\mu_\lambda, \sigma^2_\lambda), \tag{4.2}$$

where $\theta = (\mu_{y_0}, \sigma_{y_0}, \mu_\lambda, \sigma_\lambda)$ denotes the population means and standard deviations of ψ. For clarity, we will refer to ψ as individual-level parameters and to θ as population-level parameters. Although equivalent, note that our notation deviates from the standard NLME literature, where the individual-level parameters are decomposed into fixed and random effects, $\psi = \bar{\psi} + \eta$ (Lavielle 2014). In this notation, we recover the population model in Equation 4.2, by letting the fixed effects, $\bar{\psi}$, be equal to the population means, and the random effects, η, be normally distributed with zero means and variances equal to the population variances.

With this hierarchical structure, the variability in the dynamics can be simulated by repeatedly evaluating the time series model for different samples of the individual-level parameters. Each sample of ψ represents an individual in the population. As the number

of samples increases, the histogram over the simulations converges to the population distribution of the quantity of interest, which is illustrated in blue in Figure 4.1B. Formally, this population distribution is defined as

$$p(y|\theta,t) = \int \mathrm{d}\psi\, \delta(y - \bar{y}(\psi,t))\, p(\psi|\theta), \tag{4.3}$$

where $\delta(x)$ denotes the Dirac delta distribution. In the cancer growth example, $p(y|\theta,t)$ quantifies the probability with which a randomly chosen individual in the population has a tumour of volume y at time t. In general, the spread of $p(y|\theta,t)$ quantifies the IIV of the quantity of interest. NLME models assume that each set of individual-level parameters, ψ, fully characterises the dynamics of an individual. As a result, the heterogeneity of the dynamics in the population arises exclusively from the population model, $p(\psi|\theta)$.

4.2.1 Nonlinear mixed effects inference

The NLME model, as defined in Equation 4.3, is fully characterised by the population parameters θ. For most biological modelling problems these parameters are unknown and need to be estimated from data. Many algorithms and software packages for the inference of NLME models have been developed and excellent reviews exist (Augustin 2024; Bauer 2019; Rackauckas et al. 2020; Stegmann et al. 2018). A key feature of these inference approaches, henceforth referred to as 'traditional NLME inference', is a hierarchical representation of the data-generating process, where one time series model is calibrated to each measured individual. Here, we will review the Bayesian variant of traditional NLME inference, partly to exposit existing approaches but also to introduce concepts necessary to understand Filter inference.

In order to infer parameters from measurements, it is customary to include an error model, $y = \bar{y} + \epsilon$, in the NLME model definition (Lavielle 2014). The error model accounts for both measurement noise and discrepancies between the model and the true process – these two processes are collectively accounted for by ϵ. This extends the deterministic time series model output to a distribution of measurements, $p(y|\psi,t) = p(y|\bar{y}(\psi,t),\psi)$. For ease of notation, we extend the definition of ψ to also include the parameters of the error model. A common model choice is assuming normally distributed

residual errors, $p(y|\psi,t) = \mathcal{N}(y|\bar{y}(\psi,t),\sigma^2)$. The distribution of measurements across individuals in the population can then be defined analogously to Equation 4.3

$$p(y|\theta,t) = \int \mathrm{d}\psi\, p(y|\psi,t)\, p(\psi|\theta), \tag{4.4}$$

where $p(y|\psi,t)$ replaces the Dirac delta distribution. Thus, Equation 4.3 can be interpreted as a special case of Equation 4.4, where measurements are assumed to capture the value of the time series model output, $\bar{y}$, without any error. Note that Equation 4.4 implicitly defines a joint probability distribution for a measurement and individual-level parameters, $p(y,\psi|\theta,t) = p(y|\psi,t)\, p(\psi|\theta)$, which is marginalised over ψ on the right hand side of Equation 4.4: $p(y|\theta,t) = \int \mathrm{d}\psi\, p(y,\psi|\theta,t)$.

Given measurements across individuals, the joint probability distribution, $p(y,\psi|\theta,t)$, can be used to define a hierarchical log-likelihood for the model parameters

$$\log p(\mathcal{D},\Psi|\theta) = \sum_{ij} \log p(y_{ij}|\psi_i,t_j) + \sum_i \log p(\psi_i|\theta), \tag{4.5}$$

quantifying the likelihood of parameter values, (θ,Ψ), to capture the observed dynamics, $\mathcal{D} = (Y,T)$. We use $\Psi = (\psi_1,\psi_2,\ldots,\psi_N)$ to denote the individual-level parameters across N measured individuals, and Y to denote the associated matrix of measurements across individuals and times. In particular, the ijth element of Y, y_{ij}, denotes the measurement of individual i at time t_j, where the vector of all unique measurement times is denoted by T. As a result, Y has N rows, and $K = \dim(T)$ columns. Missing measurement values do not contribute to the likelihood. Equation 4.5 shows that the hierarchical likelihood comprises a term accounting for the likelihoods of individual-level parameters to describe the measurements, and a term accounting for the likelihood of the population parameters to describe the distribution of the individual-level parameters.

An example dataset suitable for the inference of the early cancer growth model is outlined in Table 4.1. For simplicity, we neglect challenges of the measurement process and assume that it is possible to measure the tumour volume across patients *in vivo*. In practice, it may be more feasible to use the *in vitro* proliferation of cancerous cells

	ID	Time $[T]$	Tumour volume $[L^3]$
1	1	1.5	11.00
2	2	1.5	8.30
3	1	2.1	11.52
4	3	2.1	9.80
5	1	4	12.03
6	2	4	8.50
⋮	⋮	⋮	⋮

Table 4.1: Outline of an example tumour volume dataset. The dataset contains (fictitious) time series measurements of tumour volumes across patients. Patients are labelled with unique IDs. The time and tumour volume are presented in arbitrary units. T indicates the time dimension and L the length dimension.

from tissue samples as a proxy for the tumour growth. For inference, the dataset can be expressed in matrix form

$$Y = \begin{pmatrix} 11.00 & 11.52 & 12.03 & \dots & y_{1K} \\ 1.5 & \text{NA} & 8.50 & \dots & y_{2K} \\ \text{NA} & 9.80 & \text{NA} & \dots & y_{3K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{N1} & y_{N2} & y_{N3} & \dots & y_{NK} \end{pmatrix} \tag{4.6}$$

with $T = (1.5, 2.1, 4, \dots, t_K)$, where the first row contains the measurements of the patient with ID 1, the second row contains the measurements of the patient with ID 2, and so on. NA denotes missing values. An important feature of the dataset is that individuals are not necessarily measured with the same frequency, or at the same time points. In the extreme, the dataset may contain only one measurement per individual, i.e. snapshot measurements.

In a Bayesian inference approach, Bayes' rule is used to translate the hierarchical likelihood into a distribution of parameter values consistent with the observations and prior knowledge, also known as the posterior distribution

$$\log p(\theta, \Psi|\mathcal{D}) = \log p(\mathcal{D}, \Psi|\theta) + \log p(\theta) + \text{constant}, \tag{4.7}$$

where $p(\theta)$ is the prior distribution of the population-level parameters (**Lunn2002**). $p(\theta)$ is used in Bayesian inference to quantify knowledge about parameter values and is a modelling choice. The model parameters can now be inferred from $p(\theta, \Psi|\mathcal{D})$

Algorithm 4.1: Traditional Bayesian NLME inference using the MH MCMC algorithm. The details of the proposal and acceptance step are omitted for clarity, but are described in subsection 2.2.6.

Input : 1. Hierarchical log-likelihood: $\log p(\mathcal{D}, \Psi|\theta)$;
2. Log-prior: $\log p(\theta)$;
3. Starting point: $(\theta^{(0)}, \Psi^{(0)})$;
4. Metropolis-Hastings sampler: `Sampler`;
5. Number of iterations: n

Output : Correlated samples from the posterior $p(\theta, \Psi|\mathcal{D})$.

$samples = [\,]$ `// Initialise sampling`
$\theta = \theta^{(0)}$, $\Psi = \Psi^{(0)}$
$score = \log p(\mathcal{D}, \Psi|\theta) + \log p(\theta)$ `// Eval.` $\log p(\theta, \Psi|\mathcal{D})$ `up to const.`
for $i \leftarrow 1$ **to** n **do**
  $\theta', \Psi' =$ `Sampler.propose`(θ, Ψ) `// Propose next sample`
  $score' = \log p(\mathcal{D}, \Psi'|\theta') + \log p(\theta')$ `// Eval.` $\log p(\theta', \Psi'|\mathcal{D})$ `up to const.`
  $accepted =$ `Sampler.check`$(score', score)$ `// Accept/reject proposal`
  if $accepted$ **then**
    $\theta = \theta'$, $\Psi = \Psi'$, $score = score'$ `// Continue from proposal`
  end if
  $samples$.append$([\theta, \Psi])$ `// Store current sample`
end for
return *samples*

using sampling algorithms, such as Markov chain Monte Carlo (MCMC) algorithms (Betancourt 2017), see Algorithm 4.1. This concludes the review of traditional Bayesian NLME inference.

For simplicity, Algorithm 4.1 illustrates the inference using the Metropolis-Hastings (MH) algorithm (Chib et al. 1995). However, the large dimensionality of the posterior, $p(\theta, \Psi|\mathcal{D})$, will often limit the sampling efficiency of the MH algorithm in practice, necessitating the use of more advanced MCMC algorithms, such as Hamiltonian Monte Carlo (HMC) or the No-U-Turn sampler (NUTS) (Hoffman et al. 2014; Neal 2010).

4.2.2 Filter inference

The intractability of traditional NLME inference for snapshot data stems from the increasing cost of evaluating the log-likelihood as the number of measured individuals grows. This is because the evaluation of the log-likelihood defined in Equation 4.5

requires one evaluation of the time series model for each observed individual, resulting in computational costs that increase at least linearly with the number of measured individuals. This expense renders traditional NLME inference intractable when thousands of individuals are measured, especially when time series models are defined by systems of differential equations that need to be solved numerically.

In theory, this intractability can be avoided by fitting to the measurements on a population-level, removing the need to evaluate the time series model for each individual separately. More specifically, using the population distribution of measurements defined in Equation 4.4, a log-likelihood for the population-level parameters can be defined directly, $\log p(\mathcal{D}|\theta) = \sum_{ij} \log p(y_{ij}|\theta, t_j)$. From this population-level log-likelihood, we can derive a log-posterior, $\log p(\theta|\mathcal{D}) = \log p(\mathcal{D}|\theta) + p(\theta) + \text{constant}$, which can be inferred using MCMC sampling. However, in practice, the integral in Equation 4.4 is too expensive to compute to make $p(y|\theta, t)$ tractable for inference.

To address the computational costs, Browning et al (2022) propose a moment matching algorithm, where $p(y|\theta, t)$ is approximated by its first moments (Browning, Drovandi, et al. 2022). Using a truncated Taylor expansion, they derive approximate expressions for these moments, making their estimation from just one evaluation of the time series model and its higher order derivatives possible, thereby resolving the computational bottleneck.

Two alternative inference approaches, widely applicable to problems with intractable likelihoods, are ABC and BSL (Beaumont 2019; Wood 2010). Here, summary statistics of simulated measurements are compared to summary statistics of the observed measurements in order to construct an approximate log-likelihood. In ABC, the similarity of the summary statistics is quantified using kernel functions whose acceptance scale is defined by manually chosen error margins. In contrast, BSL uses repeated simulation of summary statistics and parametric distributions to construct synthetic likelihoods for the summary statistics (Frazier et al. 2022; Price et al. 2018; Wood 2010). For a detailed introduction to NLME inference using ABC or BSL, we refer to Drovandi et al (Drovandi et al. 2022).

Filter inference is a conceptually related NLME inference approach, with elements from moment matching, ABC, and BSL. Similar to moment matching, Filter inference approximates the population measurement distribution to estimate the log-likelihood of model parameters. However, instead of using a Taylor expansion, it uses parametrised distributions constructed from simulated measurements to do so

$$\log p(\mathcal{D}|\theta) \approx \sum_{ij} \log p(y_{ij}|\tilde{Y}_j(\theta)). \tag{4.8}$$

These parametrised distributions are a modelling choice and we refer to them as *filters*. $p(y|\tilde{Y})$ denotes the filter. $\tilde{Y}_j = (\tilde{y}_{1j}, \ldots, \tilde{y}_{Sj})^t$ denotes S simulated measurements sampled from $p(y|\theta, t_j)$ at time t_j. Filters can be constructed from any summary statistics of the simulated measurements, including their moments. As a result, Filter inference may be seen as a sampling-based generalisation of Browning et al's moment matching algorithm.

The filter choice in Filter inference has similarities with the choice of summary statistics in ABC or BSL and can be used to capture different information about the measurement distribution. For example, a Gaussian filter, introduced below in subsection 4.2.3, compares the mean and variance of the measurements, while a Lognormal filter compares the median and the scale of the measurements. In contrast to ABC, Filter inference does not require manually chosen error margins, while also avoiding BSL's repeated simulation of summary statistics per log-likelihood evaluation (see subsection 4.3.5). The algorithmic details of Filter inference are presented in Algorithm 4.2.

Algorithm 4.2 uses MH MCMC sampling, similar to Algorithm 4.1, to infer the posterior distribution. The main difference between the algorithms is the replacement of the hierarchical log-likelihood evaluation by the estimate of the population-level log-likelihood, defined in Equation 4.8. In particular, we estimate the log-likelihood by simulating measurements from the model, $p(y|\theta, t)$, by first sampling simulated individuals, $\tilde{\Psi}$, from the population model, $p(\psi|\theta)$, see Algorithm 4.2 lines 17-21. We then simulate measurements for each simulated individual by sampling from $p(y|\psi_s, t)$ in lines 24-29. Here, we use s to label simulated individuals, instead of i which we

Algorithm 4.2: Filter inference using the MH MCMC algorithm. For details of the proposal and acceptance step see subsection 2.2.6.

Input : 1. Measurements: $\mathcal{D} = (Y, T)$;
2. Filter: $p(y|\tilde{Y})$;
3. Number of simulated individuals: S;
4. NLME model: $p(y|\psi, t)$, $p(\psi|\theta)$;
5. Log-prior: $\log p(\theta)$;
6. Starting point: $\theta^{(0)}$;
7. Metropolis-Hastings sampler: `Sampler`;
8. Number of iterations: n

Output : Correlated samples from the posterior $p(\theta|\mathcal{D})$.

$samples = [\,]$ `// Initialise sampling`
$\theta = \theta^{(0)}$
$estimate =$ `estimateLogLikelihood(`θ`)`
$score = estimate + \log p(\theta)$ `// Approx.` $\log p(\theta|\mathcal{D})$ `up to const.`
for $i \leftarrow 1$ **to** n **do**
  $\theta' =$ `Sampler.propose(`θ`)` `// Propose next sample`
  $estimate =$ `estimateLogLikelihood(`θ'`)`
  $score' = estimate + \log p(\theta')$ `// Approx.` $\log p(\theta'|\mathcal{D})$ `up to const.`
  $accepted =$ `Sampler.check(`$score'$, $score$`)` `// Accept/reject proposal`
  if *accepted* **then**
    $\theta = \theta'$, $score = score'$ `// Continue from proposal`
  end if
  $samples$`.append(`$[\theta]$`)` `// Store current sample`
end for
return *samples*
define `estimateLogLikelihood(`θ`)`
  $\tilde{\Psi} = [\,]$ `// Simulate individuals`
  for $s \leftarrow 1$ **to** S **do**
    $\psi_s \sim p(\psi|\theta)$ `// Sample an individual`
    $\tilde{\Psi}$`.append(`ψ_s`)`
  end for
  $estimate = 0$ `// Initialise estimate`
  $n_times = \text{length}(T)$
  for $j \leftarrow 1$ **to** n_times **do**
    $\tilde{Y} = [\,]$ `// Simulate measurements at` t_j
    for $\psi_s \in \tilde{\Psi}$ **do**
      $\bar{y} = \bar{y}(\psi_s, t_j)$ `// Evaluate time series model`
      $\tilde{y}_{sj} \sim p(y|\bar{y}, \psi_s)$ `// Sample a measurement`
      $\tilde{Y}$`.append(`$\tilde{y}_{sj}$`)`
    end for
    $f = \sum_i \log p(y_{ij}|\tilde{Y})$ `// Compute filter log-likelihood at` t_j
    $estimate$ `+=` f `// Add filter log-likelihood to estimate`
  end for
  return $estimate$

reserve for real individuals. Using the simulated measurements, we construct a filter that summarises population-level information of the measurements. The details of this construction are filter-specific and are discussed below. The filter defines a population-

level distribution of measurements, which we use to estimate the likelihood of the model parameters, see lines 21-29. From this estimate we can derive an estimate of the posterior which is computationally tractable.

Filter inference makes the number of time series model evaluations independent of the number of observed individuals. In this way, Filter inference remains tractable even when millions of snapshot measurements are used for parameter estimation. In particular, in Algorithm 4.2 the number of time series model evaluations is determined by the number of simulated individuals, S, and the number of measured time points, see line 27 and its surrounding for-loops. In an optimised implementation, this number can be reduced to a total of S time series model evaluations per log-likelihood estimation, see Appendix C.1. The dominant computational costs of Filter inference therefore do not scale with the number of measured individuals, but instead are set by the number of simulated individuals.

This form of approximate inference was first introduced by Hasenauer et al for a specific filter choice: the lognormal KDE filter introduced below (Hasenauer et al. 2011). Algorithm 4.2 generalises this approach to a framework, where filters can be chosen specific to the needs of the inference problem. However, Hasenauer et al's algorithm reportedly becomes inefficient for models with more than a few parameters (P. D. Dixit et al. 2020). This is because the approach samples from the posterior using the Metropolis-Hastings (MH) MCMC algorithm, whose sampling efficiency is known to scale poorly with the dimension of the posterior distribution (Chib et al. 1995).

As highlighted for traditional NLME inference, efficient sampling algorithms for high dimensional models exist, such as the HMC algorithm and its variants (Betancourt 2017; Neal 2010). HMC uses gradient-information to produce better proposals, resulting in a higher sampling efficiency per step. However, the estimate of the log-likelihood from Equation 4.8 changes non-deterministically with the population-level parameters θ, making its derivatives less useful for the HMC algorithm. In particular, the estimation of the log-likelihood involves random sampling from the population model and from the individual-level measurement distributions, see lines 19 and 28 in Algorithm 4.2. As a result, estimates of the log-likelihood will vary due to the stochasticity inherent

in both the population-level distribution and the measurement noise distribution, even for fixed population-level parameters.

To make gradient-based methods useful for Filter inference, we can recast the log-likelihood estimate from Equation 4.8 into a hierarchical form that changes deterministically with its input parameters. In particular, we can define a joint distribution of measurements, simulated measurements and simulated individual-level parameters, $p(y, \tilde{Y}, \tilde{\Psi}|\theta, t) = p(y|\tilde{Y}) \prod_s p(\tilde{y}_s|\psi_s, t)\, p(\psi_s|\theta)$, making the dependence on the realisation of the simulated measurements and parameters explicit. From this joint distribution, we can define a hierarchical log-likelihood comprising the filter estimate of the population-level log-likelihood, and the log-likelihood of model parameters $(\theta, \tilde{\Psi})$ to describe the simulated measurements, $\tilde{Y}$,

$$\log p(\mathcal{D}, \tilde{Y}, \tilde{\Psi}|\theta) = \sum_{ij} \log p(y_{ij}|\tilde{Y}_j) + \sum_{sj} \log p(\tilde{y}_{sj}|\psi_s, t_j) + \sum_s \log p(\psi_s|\theta). \quad (4.9)$$

We can use this log-likelihood and Bayes' rule to derive a log-posterior analogously to the hierarchical log-likelihood and the NLME log-posterior in Equations 4.5 and 4.7

$$\log p(\theta, \tilde{Y}, \tilde{\Psi}|\mathcal{D}) = \log p(\mathcal{D}, \tilde{Y}, \tilde{\Psi}|\theta) + \log p(\theta) + \text{constant}. \quad (4.10)$$

This log-posterior depends deterministically on its parameters, $(\theta, \tilde{Y}, \tilde{\Psi})$. As a result, we can use HMC to efficiently sample from $p(\theta, \tilde{Y}, \tilde{\Psi}|\mathcal{D})$, even for high-dimensional NLME models. Once $(\theta, \tilde{Y}, \tilde{\Psi})$ are inferred, the approximate posterior for the population-level parameters can be obtained by considering only the θ estimates (i.e. marginalisation). The algorithmic details of the approach are presented in Algorithm 4.3. The algorithm is illustrated using a MH sampler for easier comparison with Algorithms 4.1 and 4.2. Note that line 11 in Algorithm 4.3 already implements the marginalisation over $\tilde{Y}$ and $\tilde{\Psi}$.

Importantly, the posteriors for θ inferred using the stochastic likelihood (Equation 4.8) and using the deterministic likelihood (Equation 4.9) are identical. The main difference is that the implementation of the stochastic form uses ancestral sampling to simulate individuals and measurements to estimate the log-likelihood, which makes its estimates change non-deterministically with its input parameters. This stochastic dependence is eliminated in Equation 4.9 by explicitly formulating a log-likelihood term for each random

Algorithm 4.3: Filter inference (deterministic form) using the MH MCMC algorithm. For details of the proposal and acceptance step see subsection 2.2.6.

Input : 1. Filter log-likelihood: $\log p(\mathcal{D}, \tilde{Y}, \tilde{\Psi}|\theta)$;
2. Log-prior: $\log p(\theta)$;
3. Starting point: $(\theta^{(0)}, \tilde{Y}^{(0)}, \tilde{\Psi}^{(0)})$;
7. Metropolis-Hastings sampler: `Sampler`;
8. Number of iterations: n

Output : Samples from the posterior $p(\theta|\mathcal{D})$.

$samples = [\,]$ `// Initialise sampling`
$\theta = \theta^{(0)}$, $\tilde{Y} = \tilde{Y}^{(0)}$, $\tilde{\Psi} = \tilde{\Psi}^{(0)}$,
$score = \log p(\mathcal{D}, \tilde{Y}, \tilde{\Psi}|\theta) + \log p(\theta)$ `// Eval.` $\log p(\theta, \tilde{Y}, \tilde{\Psi}|\mathcal{D})$ `up to const.`
for $i \leftarrow 1$ **to** n **do**
  $\theta', \tilde{Y}', \tilde{\Psi}' =$ `Sampler.propose`$(\theta, \tilde{Y}, \tilde{\Psi})$ `// Propose next sample`
  $score' = \log p(\mathcal{D}, \tilde{Y}', \tilde{\Psi}'|\theta') + \log p(\theta')$ `// Eval.` $\log p(\theta', \tilde{Y}', \tilde{\Psi}'|\mathcal{D})$ `up to const.`
  $accepted =$ `Sampler.check`$(score', score)$ `// Accept or reject proposal`
  if *accepted* **then**
    $\theta = \theta'$, $\tilde{Y} = \tilde{Y}'$, $\tilde{\Psi} = \tilde{\Psi}'$, $score = score'$ `// Continue from proposal`
  end if
  $samples$`.append`$([\theta])$ `// Store current sample`
end for
return *samples*

variable contributing to the overall likelihood of the population-level parameters. In this way, the likelihood becomes a deterministic function of the random variables $(\theta, \tilde{Y}, \tilde{\Psi})$.

4.2.3 Filters

Filters are the central element of Filter inference, making inference of NLME models from measurements of thousands of individuals possible. Below we introduce five filters that we have found useful in our experiments.

1. Gaussian filter: A Gaussian filter summarises the population measurement distribution using a Gaussian distribution

$$p(y|\tilde{Y}_j) = \mathcal{N}(y|\tilde{\mu}_j, \tilde{\sigma}_j^2), \tag{4.11}$$

where $\tilde{\mu}_j$ and $\tilde{\sigma}_j^2$ are given by the empirical mean and variance of the simulated

measurements, $\tilde{\mu}_j = \sum_s \tilde{y}_{sj}/S$ and $\tilde{\sigma}_j^2 = \sum_s (\tilde{y}_{sj} - \tilde{\mu}_j)^2/(S-1)$, where S denotes the number of simulated individuals.

A Gaussian filter is illustrated, together with other filters, in Figure 4.2, where we repeatedly simulate $S = 100$ measurements from the early cancer growth model at $t = 1$, using the population model introduced in subsection 4.3.1. We use the simulated measurements to construct the filters, e.g. for the Gaussian filter we compute the mean and variance of the simulations. A random realisation of each filter for one set of $S = 100$ sampled measurements is illustrated in red. Sampling filters in this way 1000 times, we estimate the 5th and 95th percentile of the filter density distribution, illustrated in blue. As a reference for the filter approximations, the figure shows the true population measurement distribution, $p(y|\theta, t)$, in black.

2. Lognormal filter: A lognormal filter summarises the population measurement distribution using a lognormal distribution

$$p(y|\tilde{Y}_j) = \mathrm{LN}(y|\tilde{\mu}_j, \tilde{\sigma}_j), \tag{4.12}$$

where the location and scale of the lognormal distribution, $(\tilde{\mu}_j, \tilde{\sigma}_j)$, are given by the empirical mean and standard deviation of the log-transformed simulated measurements.

3. Gaussian mixture filter: A Gaussian mixture filter summarises the population measurement distribution using a Gaussian mixture distribution

$$p(y|\tilde{Y}_j) = \frac{1}{M} \sum_{m=1}^{M} \mathcal{N}(y|\tilde{\mu}_{j,m}, \tilde{\sigma}_{j,m}^2), \tag{4.13}$$

where M is a hyperparameter and determines the number of Gaussian kernels. For computational efficiency, the mean and the variance of the Gaussians are estimated from the simulated measurements by assigning S/M simulated individuals to each subpopulation. The parameters of the mth kernel are then estimated using the empirical mean and variance of the measurements of the mth subpopulation.

A more general implementation of the Gaussian mixture filter, not explored in this publication, estimates the maximum likelihood estimates of the filter parameters from the simulated measurements using the expectation-maximisation (EM) algorithm (Moon 1996). While computationally less efficient, this implementation provides more

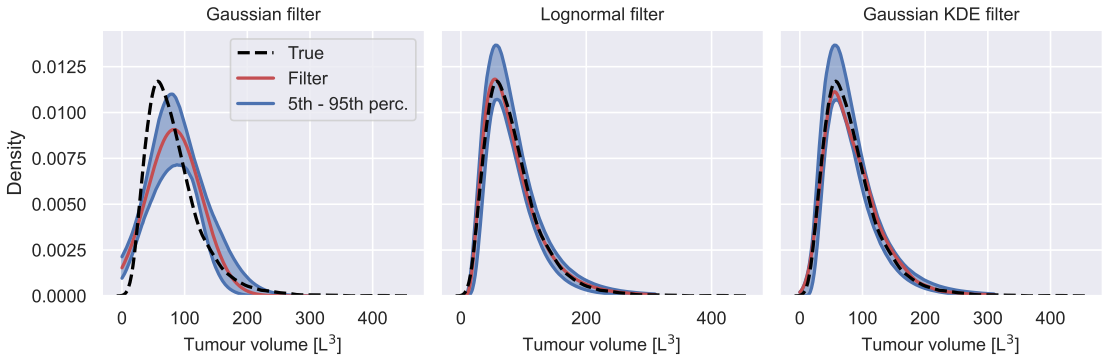


Figure 4.2: Filters in Filter inference. The figure shows a Gaussian filter, a lognormal filter and a Gaussian KDE filter of the early cancer growth model for $S = 100$ simulated individuals at time $t = 1$. Each filter is illustrated by a randomly chosen realisation, illustrated in red, and the 5th to 95th percentile of the filter distribution for different sets of simulated individuals. As a reference, the exact population measurement distribution is illustrated in black.

flexibility, making it possible to approximate the population measurement distribution more faithfully.

4. Gaussian KDE filter: A Gaussian KDE filter summarises the population measurement distribution using a Gaussian kernel density estimation

$$p(y|\tilde{Y}_j) = \frac{1}{S}\sum_{s=1}^{S}\mathcal{N}(y|\tilde{y}_{sj}, b_j^2), \tag{4.14}$$

where S is the number of simulated individuals. A Gaussian KDE population filter is a Gaussian mixture population filter, where each individual is assigned to its own subpopulation, i.e. $M = S$. The bandwidth of the kernels, $\tilde{b}_j^2$, is a hyperparameter of the population filter. In this article, we use the widely used rule-of-thumb for the bandwidth selection, $b_j^2 = (4/3/S)^{2/5}\tilde{\sigma}_j^2$, following Hasenauer et al (Hasenauer et al. 2011). Here, $\tilde{\sigma}_j^2$ is the empirical variance of the simulated measurements at time t_j.

5. Lognormal KDE filter: A lognormal KDE filter summarises the population measurement distribution using a lognormal kernel density estimation

$$p(y|\tilde{Y}_j) = \frac{1}{S}\sum_{s=1}^{S}\mathrm{LN}(y|\tilde{y}_{sj}, b_j), \tag{4.15}$$

The bandwidth is computed using the rule-of-thumb, $b_j^2 = (4/3/S)^{2/5}\tilde{\sigma}_j^2$, where $\tilde{\sigma}_j^2$ is the empirical variance of the log-transformed simulated measurements.

4.3 Results & discussion

To demonstrate the properties of Filter inference, we first infer posterior distributions from snapshot measurements for two modelling problems: 1. early cancer growth; and 2. EGF pathway signalling. We then compare the computational costs of traditional NLME inference and Filter inference and quantify the impact of using NUTS on the sampling efficiency. We conclude the section by highlighting the similarities between the filter choice in Filter inference and choosing summary statistics in ABC or BSL. We also illustrate how inappropriate choices of filters may result in information loss or bias.

Models, datasets, and Python scripts to reproduce the results are hosted on https://github.com/DavAug/filter-inference. All models are implemented in the open-source Python package `chi` (Augustin 2024), which we have extended to provide a user-friendly API for Filter inference. For the inference, we use `pints`' implementations of NUTS and the MH algorithm (Clerx, Robinson, et al. 2019). The gradients of the log-posterior, needed for NUTS, are automatically computed by `chi`, using the open-source Python package `myokit` (Clerx, Collins, et al. 2016).

4.3.1 Early cancer growth model inference

To establish that Filter inference is a sound approach for the inference of NLME models, we compare the results of Filter inference and NLME inference on a common dataset. We synthesise snapshot measurements from the early cancer growth model (Equation 4.1) with a Gaussian error model, $p(y|\psi, t) = \mathcal{N}\left(y|y_0 \mathrm{e}^{\lambda t}, \sigma^2\right)$, by first sampling individual-level parameters from the population model, $p(\psi|\theta) = \mathcal{N}(y_0|\mu_{y_0}, \sigma^2_{y_0})\,\mathcal{N}(\lambda|\mu_\lambda, \sigma^2_\lambda)\,\delta(\mu_\sigma - \sigma)$. We then measure each individual by sampling from $p(y|\psi, t)$. The population parameters are set to $\theta = (\mu_{y_0}, \sigma_{y_0}, \mu_\lambda, \sigma_\lambda, \mu_\sigma) = (10, 1, 2, 0.5, 0.8)$ for the data-generation. 15 snapshot measurements are synthesised for 6 time points between 0 and 0.6 time units. The resulting dataset, with a total of 90 measured individuals, is illustrated using scatter points in Figure 4.3A. This dataset is still tractable for NLME inference. The details of the inference procedure and the convergence assessment are reported in Appendix C.2.

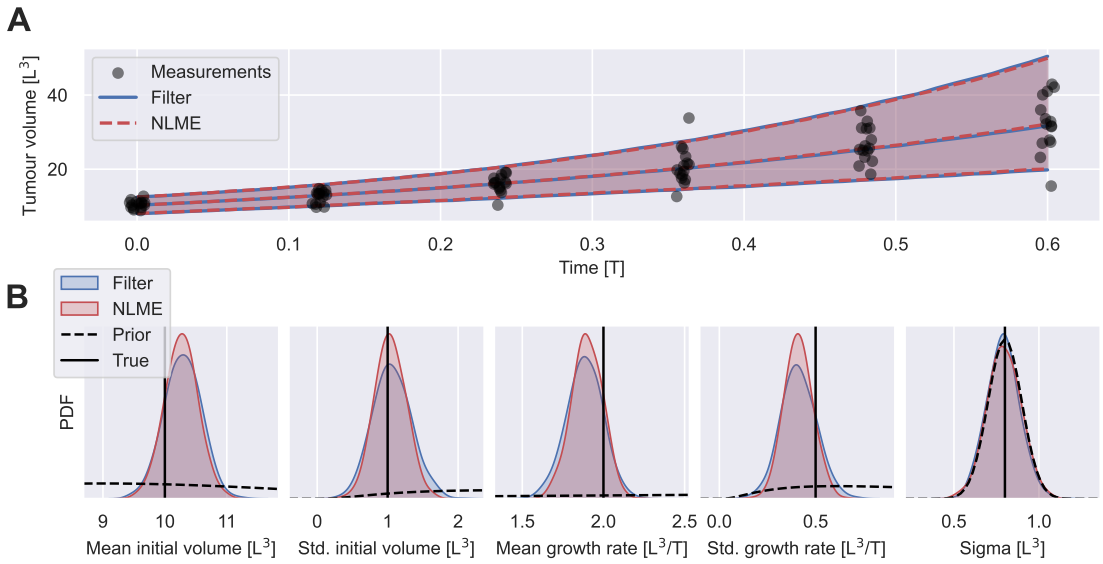


Figure 4.3: Filter inference versus traditional NLME inference. A: Shows 90 snapshot measurements in arbitrary units, generated from the early cancer growth model, and the fitted NLME models obtained using Filter inference with a Gaussian filter (blue) and traditional Bayesian NLME inference (red). The measurements are illustrated with jitter on the time axis. The fitted models are illustrated by the medians and the 5th to 95th percentile range of the inferred measurement distributions, $\mathbb{E}_{\theta|\mathcal{D}}[p(y|\theta,t)]$. The filter is constructed using $S = 100$ simulated individuals. B: Shows the inferred posterior distributions obtained using Filter inference (blue) and NLME inference (red). The data-generating parameters (solid lines) as well as the prior distributions used for the inference (dashed lines) are also shown.

The inference results show that Filter inference, with a Gaussian filter and $S = 100$ simulated individuals, and NLME inference produce almost identical fits to the measurements (see Figure 4.3A) and similar posterior distributions (see Figure 4.3B). Notably, the posterior distributions of both approaches encompass the data-generating parameter values within their main bulk probability mass. The measurements appear highly informative about the population means of the parameters but less so about the variability in the population. Importantly, the measurements are not informative about the noise parameter, μ_σ, since the posterior distributions do not differ substantially from the prior distribution. This can be explained by the difficulty to attribute the observed variability to either IIV or to measurement noise when individuals are not measured repeatedly. While the increase of the tumour volume variability over time indicates that IIV is present at least in one of (y_0, λ), since a Gaussian error model cannot capture heteroscedasticity, the measurements in Figure 4.3A leave room for attributing

all observed variability at $t = 0$ to just noise, or just IIV in y_0. Thus, as long as the combined variability of the model is of the same magnitude as the observed variability, each contribution may assume any magnitude between zero and the observed variability at $t = 0$. The variance of the measurements at $t = 0$ is 1.0 ± 0.4 (see Appendix C.3 for the details of the estimation), while the prior of μ_σ focusses on values between 0.5 and 1, as shown in the right-most figure in Figure 4.3, constraining the noise variance to be at most of order $\mathcal{O}(1)$. As a result, all values from the prior of μ_σ are compatible with the observations, and the posterior is not informed by the measurements.

This lack of IIV-noise identifiability can be overcome when variability contributions from IIV and noise lead to distinct shapes of the measurement distribution, $p(y|\theta, t)$, and sufficiently many measurements are available to resolve such distributional differences. However, NLME inference from large snapshot datasets is computationally intractable and Filter inference requires a large number of simulated individuals in order to distinguish IIV and noise, diminishing its computational advantages (see Appendix C.4 for a detailed discussion). The efficient inference from snapshot measurements using Filter inference with only a small number of simulated individuals is therefore reliant on informative noise priors. In many applications, such priors may be informed by the specifications of measurement devices. Where possible, repeated measurements of a few individuals may also be used to estimate noise parameters.

The inferred population distributions, represented by the averages over the posterior distributions, $\mathbb{E}_{\theta|\mathcal{D}}[p(\psi|\theta)]$, are illustrated together with the data-generating distribution in the left panel of Figure 4.4A (see Appendix C.5 for details on the computation of $\mathbb{E}_{\theta|\mathcal{D}}[p(\psi|\theta)]$). The comparison shows that the inference from 90 snapshot measurements with either inference method provides only a crude estimate of the IIV inherent in the data-generating process. This inaccuracy is a direct consequence of sampling bias – 90 individuals do not faithfully depict the variability in the whole population. To test whether this sampling bias can be mitigated by increasing the number of measured individuals, we exponentially increase the number of snapshot measurements per time point from 15 to $3 \times 15 = 45$, $3^2 \times 15 = 135$ and $3^3 \times 15 = 405$ snapshot measurements, resulting in datasets totalling $N = 270$, $N = 810$ and $N = 2430$ measurements. The

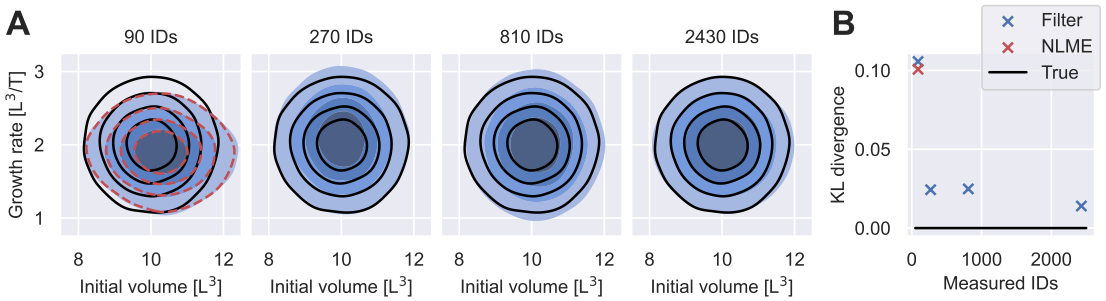


Figure 4.4: Quality of IIV estimates for varying dataset sizes. A: Shows inferred population distributions from measurements of varying numbers of individuals (IDs) using Filter inference with a Gaussian filter and $S = 100$ simulated individuals in blue. The different shades of blue indicate the bulk $20\,\%$, $40\,\%$, $60\,\%$ and $80\,\%$ probability regions. The distribution from Figure 4.3, inferred from measurements of 90 individuals using traditional NLME inference, is illustrated using red dashed lines. The data-generating distribution is illustrated in black. B: Shows the KL divergences between the data-generating population distribution and the inferred distributions from A.

inferred population distributions using Filter inference are illustrated in panels 2, 3 and 4 of Figure 4.4A. The figure demonstrates that the estimation of the IIV improves with the number of measured individuals, as also quantified by the Kullback-Leibler (KL) divergence between the data-generating population distribution and the inferred distributions shown in Figure 4.4B. Overall, Figure 4.4 illustrates that many more than 90 snapshot measurements are needed to obtain accurate estimates of the IIV for the early cancer growth model. For more complex models, including nonlinear dynamics and more model parameters, measurements from even more individuals are likely required, causing issues for NLME inference. In contrast, the computational costs of Filter inference are not dominated by the number of measured individuals, and thus inference remains tractable.

4.3.2 EGF pathway model inference

The EGF pathway plays an important role in regulating the behaviour of epithelial cells and tumours of epithelial origin (Herbst 2004). Understanding the cell-to-cell variability in the biochemical signalling is therefore of great interest (P. D. Dixit et al. 2020). Here, we demonstrate the ability of Filter inference to estimate the parameters of a published EGF signalling pathway model from snapshot measurements.

In particular, we consider a model of inactive and active EGF receptor (EGFR) concentrations (P. D. Dixit et al. 2019)

$$\frac{\mathrm{d}c_r}{\mathrm{d}t} = p - k_{\mathrm{on}} c_l c_r + k_{\mathrm{off}} c_a - k_{\mathrm{deg},r} c_r, \quad \frac{\mathrm{d}c_a}{\mathrm{d}t} = k_{\mathrm{on}} c_l c_r - k_{\mathrm{off}} c_a - k_{\mathrm{deg},a} c_a. \quad (4.16)$$

c_r denotes the concentration of inactive EGFR, and c_a denotes the concentration of active EGFR. The model assumes that inactive EGFR is produced at a constant rate, p. Upon binding of EGF, inactive EGFR is activated at a rate proportional to the surrounding EGF concentration, c_l. Once activated, receptors deactivate at a rate k_{off}. Both, active and inactive receptors are assumed to degrade over time at rates $k_{\mathrm{deg},a}$ and $k_{\mathrm{deg},r}$, respectively. In our study, the cell-to-cell variability is modelled by varying production and activation rates. The remaining model parameters are fixed across cells. We synthesise two distinct datasets, each comprising 1200 cells and use the collective data from both datasets to perform Filter inference. The first dataset contains snapshot measurements of cells exposed to a constant EGF concentration of $c_l = 2\,\mathrm{ng/mL}$, henceforth denoted as 'data (low)'. The second dataset, 'data (high)', contains snapshot measurements of the same experiment with a higher constant EGF concentration of $c_l = 10\,\mathrm{ng/mL}$. The data are simulated using a lognormal error model with scale parameter μ_σ, centred at the model outputs. The measurements are generated over a period of $25\,\mathrm{min}$ with population parameters $\theta = (\mu_p, \sigma_p, \mu_{k_{\mathrm{on}}}, \sigma_{k_{\mathrm{on}}}, \mu_{k_{\mathrm{off}}}, \mu_{k_{\mathrm{deg},r}}, \mu_{k_{\mathrm{deg},a}}, \mu_\sigma) = (1.7, 0.05, 1.7, 0.05, 8, 0.25, 0.015, 0.05)$, initialising both receptor concentrations at $0\,\mathrm{ng/mL}$. The generated datasets are illustrated by black scatter points (data (low)) and grey scatter points (data (high)) in Figure 4.5A. We infer the model parameters from the synthetic datasets using Gaussian filters with $S = 100$ simulated cells. The noise parameter, μ_σ, is fixed to the data-generating value during the inference. Details on the inference procedure are reported in Appendix C.6.

Figure 4.5A shows that Filter inference is able to infer measurement distributions that capture the dynamics of the observed EGF signalling pathway. Figure 4.5B shows that the inferred posteriors assign substantial weights to the data-generating parameters. The inferred cell-to-cell variability of the model parameters is of a reasonable magnitude, as the comparison of the data-generating population distribution and the inferred

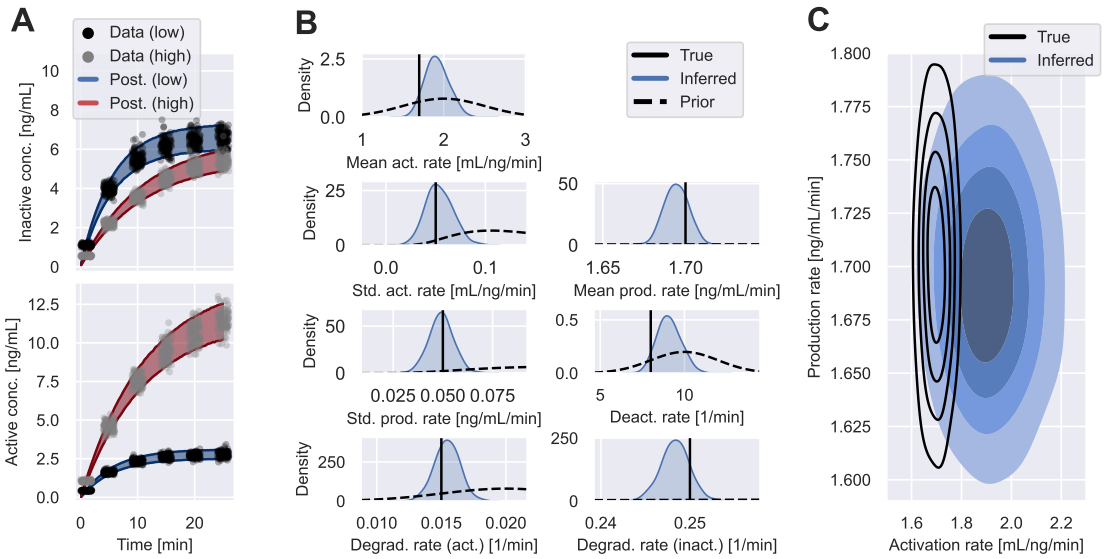


Figure 4.5: Inference results of EGF pathway model I. A: Shows snapshot measurements of active and inactive EGFR concentrations across cells. The cells are exposed to one of two EGF concentrations: data (low) with $c_l = 2\,\text{ng/mL}$ (black scatter points); and data (high) with $c_l = 10\,\text{ng/mL}$ (grey scatter points). The shaded areas illustrate the 5th to 95th percentile of the inferred measurement distributions using Filter inference with Gaussian filters and $S = 100$ simulated cells. B: Shows the inferred posterior distributions of the model parameters illustrated by blue density plots, together with the data-generating parameter values illustrated by black solid lines. The density of the prior distribution is illustrated by dashed lines for each parameter. C: Shows the inferred population distribution of the production rate and the activation rate in blue. The different shades of blue indicate the bulk 20%, 40%, 60% and 80% probability regions. The probability regions of the data-generating distribution are illustrated by black contours.

distribution in Figure 4.5C shows. However, the data appear to be more informative about the production rate variability than the activation rate variability. To investigate this further, in Figure 4.6A, we plot the posterior distribution of the population-level mean activation rate versus the deactivation rate. The strong correlation of the joint posterior distribution indicates that it is not possible to identify both the activation rate and deactivation rates from the synthesised datasets. In Figure 4.6B, we show inference results when we fix the deactivation rate to its known fixed value – in this case, the inferred population distribution is closer to the true one.

4.3.3 Scaling & computational costs

Filter inference is able to infer the parameters of NLME models from thousands of snapshot measurements. This is because the dominant computational costs of the

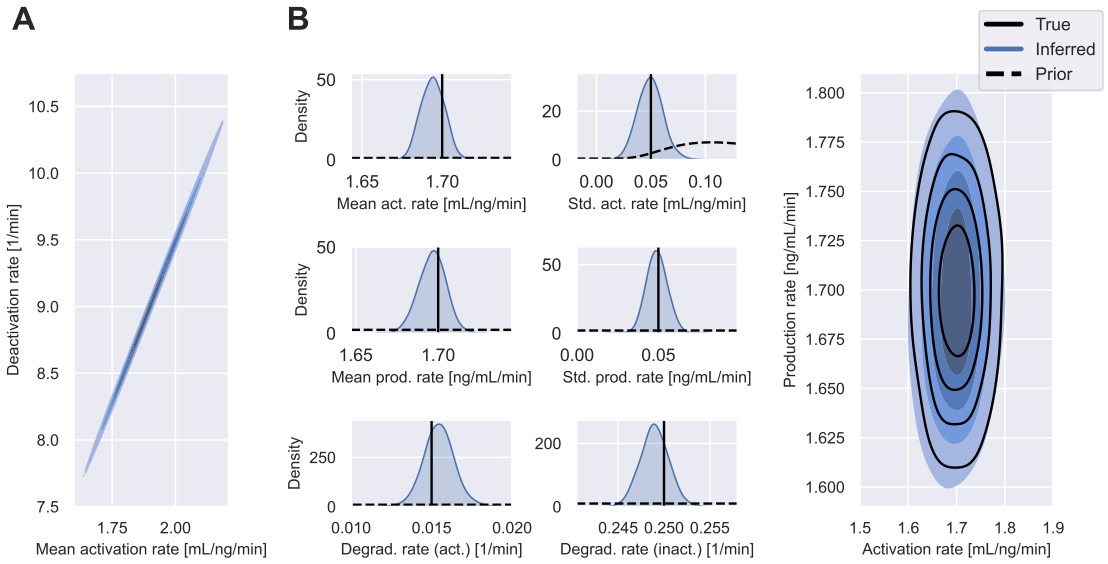


Figure 4.6: Inference results of EGF pathway model II. A: Shows the joint posterior distribution of the mean activation rate, $\mu_{k_{\text{on}}}$, and the deactivation rate, $\mu_{k_{\text{off}}}$, from Figure 4.5B. B: Shows the inferred posterior distribution and the inferred population distribution from a separate inference run, where we fix the deactivation rate to its data-generating value. All other inference settings, including data and priors, remain unchanged from the inference approach used to generate Figure 4.5.

log-posterior evaluation do not scale with the number of measured individuals, as demonstrated in Figure 4.7. The figure shows the evaluation time of the NLME log-posterior and its gradient (blue lines), defined in Equation 4.7, and the filter log-posterior and its gradient (red lines), defined in Equation 4.9, for the early cancer growth model and the EGF pathway model with increasing numbers of measured individuals. The filter log-posterior is defined using a Gaussian filter with varying numbers of simulated individuals. The gradients of the posteriors are automatically computed using `chi` (Augustin 2024). Details on the estimation of the evaluation times are presented in Appendix C.7.

The figure shows that the evaluation time of the NLME log-posterior scales linearly with the number of measured individuals, while the cost of the filter log-posterior remains constant. Instead of scaling with the number of measured individuals, the computational costs of the filter log-posterior are roughly proportional to the number of simulated individuals, as discussed in section 4.2. As a result, the speed up provided by Filter inference is of order $\mathcal{O}(N/S)$, where N and S denote the number of measured and simulated individuals, respectively. This reduces the log-posterior evaluation costs

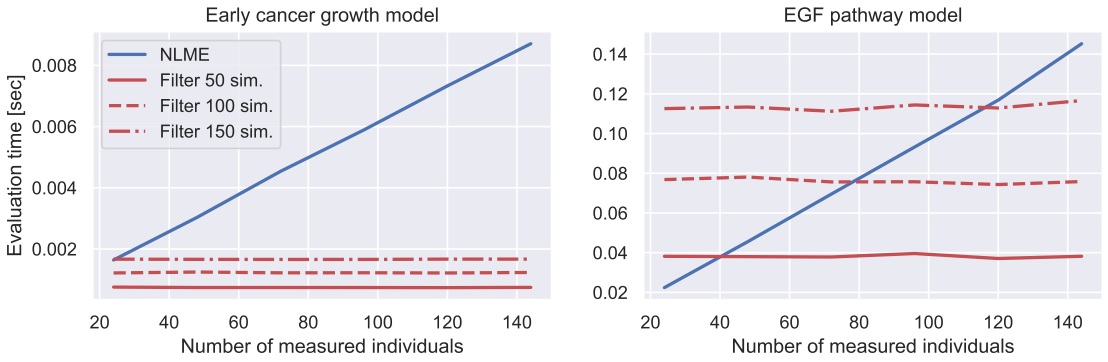


Figure 4.7: Computational costs of Filter inference and traditional NLME inference I - evaluation time of log-posterior. The figure shows the evaluation time of the traditional NLME log-posterior and its gradient, defined in Eq 4.7, (blue lines) and the Filter inference posterior and its gradient, defined in Eq 4.10, with a Gaussian filter and $S = 50$, $S = 100$ and $S = 150$ simulated individuals (red lines). The left and right panel illustrate the results for the early cancer growth model and the EGF pathway model, respectively.

34-fold for the early cancer growth model, and 13-fold for the EGF pathway model for datasets with 1000 snapshot measurements and filters with $S = 100$ simulated individuals. This cost reduction increases as the number of measured individuals grows.

However, the computational costs of inference are not exclusively determined by the evaluation time of the log-posterior – the total number of evaluations needed for convergence of the MCMC algorithm is also a determining factor. Since MCMC is a form of dependent sampling, samples are typically correlated, reducing the number of iid samples drawn from the posterior distribution (see subsection 2.2.6). In order to determine the total computational costs of the approaches, we thus need to compare their number of log-posterior evaluations needed for convergence.

Several metrics for the convergence assessment of MCMC samples exist, such as the $\hat{R}$-metric or the R^*-metric (Lambert and Vehtari 2022; Vehtari, Gelman, Simpson, et al. 2021). These metrics could be used to determine the number of log-posterior evaluations needed to reach a certain degree of convergence for different MCMC algorithms. An alternative approach to compare the computational costs of traditional NLME inference and Filter inference is to infer the posteriors using an MCMC algorithm that uses an initial calibration phase to adjust the number of log-posterior evaluations per MCMC step in order to maximise the convergence rate across inference problems. Such an algorithm is NUTS (Betancourt 2017; Hoffman et al. 2014). Using its calibration

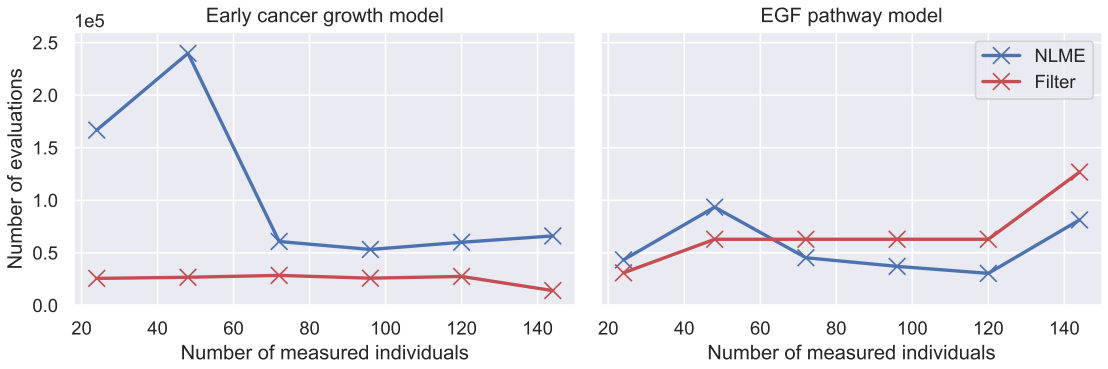


Figure 4.8: Computational costs of Filter inference and traditional NLME inference II - number of log-posterior evaluations. The figure shows the number of log-posterior evaluations of traditional NLME inference (blue lines) and Filter inference, with a Gaussian filter and $S = 100$ simulated individuals, (red lines) for the early cancer growth model and the EGF pathway model using varying sizes of snapshot datasets. Each log-posterior evaluation includes the evaluation of its gradient. The posterior distributions are inferred using 1000 MCMC iterations of NUTS after calibrating the algorithm for 500 iterations.

strategy, NUTS converges within 1000 MCMC iterations post calibration for the early cancer growth model and the EGF pathway model. We choose to estimate the total computational costs of traditional NLME inference and Filter inference using the latter approach, and run 1500 NUTS iterations for both, the early cancer growth model and the EGF pathway model for different datasets with varying numbers of measured individuals (see Figure 4.8). The initial 500 iterations of each inference run are used to calibrate the algorithm. Each evaluation includes the evaluation of the log-posterior and its gradient.

The figure shows that the number of log-posterior evaluations of traditional NLME inference and Filter inference are of the same order of magnitude for the investigated models. Each log-posterior evaluation includes the evaluation of the log-posterior gradient. For the early cancer growth model, NUTS requires fewer evaluations during Filter inference, while for the EGF pathway model NUTS evaluates the log-posterior less often during traditional NLME inference. Overall, the linear cost scaling of traditional NLME inference with the number of measured individuals, and the comparable total number of evaluations demonstrate that the benefit to using Filter inference scales linearly with the number of measured individuals, meaning, for large datasets typical in cell biology, Filter inference can be orders of magnitude faster than traditional NLME inference.

4.3.4 Sampling efficiency

Filter inference can reduce the costs of inference in both its stochastic and its deterministic form. In this section, we investigate the degree to which the deterministic version and the use of NUTS improves the efficiency of Filter inference.

To this end, we estimate the sampling efficiency of both approaches using the effective sample size (ESS) metric (**gelmanbda04**). The ESS estimates the number of iid samples drawn from each posterior dimension by an MCMC algorithm. Thus, the larger the ESS for a fixed number of log-posterior evaluations, the better the sampling efficiency of the algorithm. In Figure 4.9 we show the minimum ESS across dimensions, obtained from inferring the early cancer growth model posterior and the EGF pathway model posterior. We infer the posteriors twice: 1. using the MH algorithm and Filter inference with a stochastic posterior, defined in Algorithm 4.2; and 2. using NUTS and Filter inference with a deterministic posterior, defined in Algorithm 4.3.

The figure shows that the sampling efficiency is improved when using NUTS. After 35,000 iterations, the MH algorithm generated an ESS of approximately 600 when posterior samples were inferred from the cancer growth dataset with 90 measured individuals (see left panel in Figure 4.9), while NUTS generated an ESS of 932 after 1500 iterations, using the first 500 iterations for calibration. These 1500 iterations translate into 34,985 evaluations of the log-posterior and its gradient (see Table C.4). The majority of these evaluations (20,000 evaluations) are from the calibration phase and therefore do not contribute to the ESS. Using `chi`'s forward sensitivities, the evaluation costs for the considered problems approximately double, when the gradients are computed in addition to the log-posterior value (see Figure C.4). As a result, NUTS generated an ESS that is 1.5 times greater with approximately the same ($2 \times 14,985/35,000 \approx 1$) computational costs post calibration.

This efficiency advantage becomes more pronounced as the number of measured individuals increases. In particular, the ESS of the MH algorithm, despite extensive efforts to tune its hyperparameters (see Appendix C.8 for details), is smaller than 10 after 90,000 MH iterations for both the early cancer growth model and the EGF pathway model, when datasets with thousands of measurements are used for the inference (see

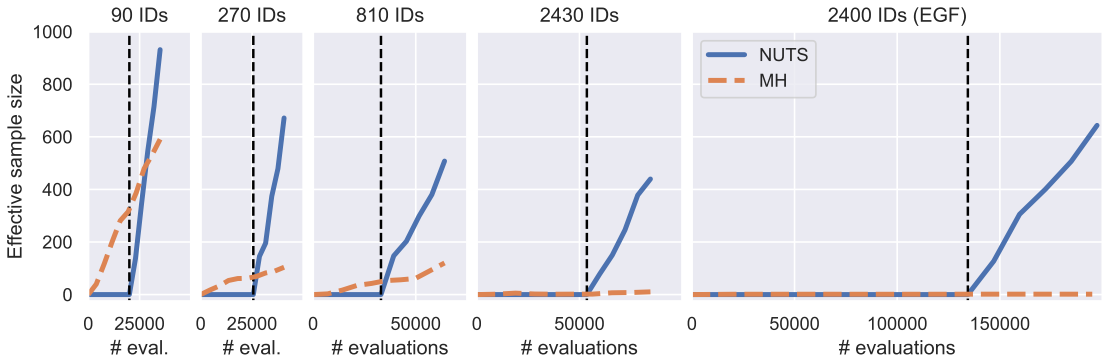


Figure 4.9: Sampling efficiency of Filter inference variants. The figure shows the minimum ESS across posterior dimensions as a function of log-posterior evaluations for different posterior distributions inferred with: 1. NUTS and the deterministic filter posterior (blue); and 2. MH and the stochastic filter posterior (orange). For NUTS, the number of evaluations include evaluations of the log-posterior gradient. For MH, the log-posterior gradient is not evaluated. Panels 1, 2, 3 and 4 show the minimum ESS of the cancer growth model posteriors from Figure 4.4 and panel 5 shows the EGF pathway model posteriors from Figure 4.6B.

panels 4 and 5 in Figure 4.9). At the same time, NUTS is able to generate an ESS of order 100 for all datasets and models by automatically tuning the number of evaluations per MCMC step during the calibration phase. The number of log-posterior evaluations across all inference runs are reported in Appendix C.4.

Our study indicates that Filter inference with the deterministic posterior and NUTS improves the sampling efficiency across inference problems relative to the MH algorithm and its stochastic variant. This improved efficiency is achieved, despite the increase of the posterior dimension from θ to $(\theta, \tilde{Y}, \tilde{\Psi})$. For example, for the cancer growth problem θ has 5 dimensions, while $(\theta, \tilde{Y}, \tilde{\Psi})$ has 805 dimensions. For more advanced gradient-free sampling methods than the MH algorithm or problems where the gradients cannot be accurately computed, this efficiency advantage may change.

Figure 4.9 also reveals that sampling from filter posteriors appears to become more challenging as the number of measured individuals increases. While NUTS is still able to achieve good sampling efficiencies, both MCMC algorithms (the MH algorithm and NUTS) become less efficient the more individuals are measured (see slope of ESS curves in Figure 4.9). To understand this behaviour, we investigate the relationship between Filter inference, ABC and BSL in the next section.

4.3.5 Relationship to ABC and BSL

In the section 4.2, we highlighted that the filter choice in Filter inference has similarities with the summary statistics choice in ABC or BSL. In this section, we illustrate this similarity and use it to shed light on the result from the previous section where we found that the deterministic filter posterior appears to be better suited for the inference from large numbers of snapshot measurement than the stochastic variant.

ABC is a hugely successful inference strategy across many fields of applications, such as population genetics (Sjödin et al. 2012), epidemiology (McKinley et al. 2018) and climate modelling (Holden et al. 2018). However, a common criticism of ABC is its dependence on manually chosen error margins which are used to quantify the distance between summary statistics of real and simulated measurements. Too large error margins reduce the quality of the inference results, while too narrow error margins lead to high rejection rates and, thus, to a poor sampling efficiency. To mitigate this challenge, methods for automatically tuning error margins have been developed (Prangle 2017; Silk et al. 2013).

In contrast to ABC, Filter inference does not require manually chosen error margins, despite some filters using summary statistics (see e.g. the Gaussian filter in Equation 4.11). We will denote filters using summary statistics henceforth as summary statistics-based (SS-based) filters. In Figure 4.10, we show that error margins arise naturally for SS-based filters, using Gaussian filters as an example.

The figure revisits the inference results from Figure 4.4 and shows the means and variances of the simulated measurements, $\tilde{Y}$, obtained at each iteration of the NUTS algorithm post calibration (see black markers). For simplicity, the figure shows the simulations only for one of the measured time points, here $t_5 = 0.6$. To be clear: each set of simulated measurements was used to define the $t_5 = 0.6$ contribution to the filter likelihood, $\sum_i \log p(y_{i5}|\tilde{Y}_5)$, in Equation 4.9 during the inference, and was sampled from the posterior distribution (Equation 4.10) by the NUTS algorithm.

The first panel illustrates the inference results for the dataset with 90 snapshot measurements. Panels 2, 3, and 4 correspond to the datasets with 270, 810 and 2430 snapshot measurements, respectively. As a reference for the simulated summary

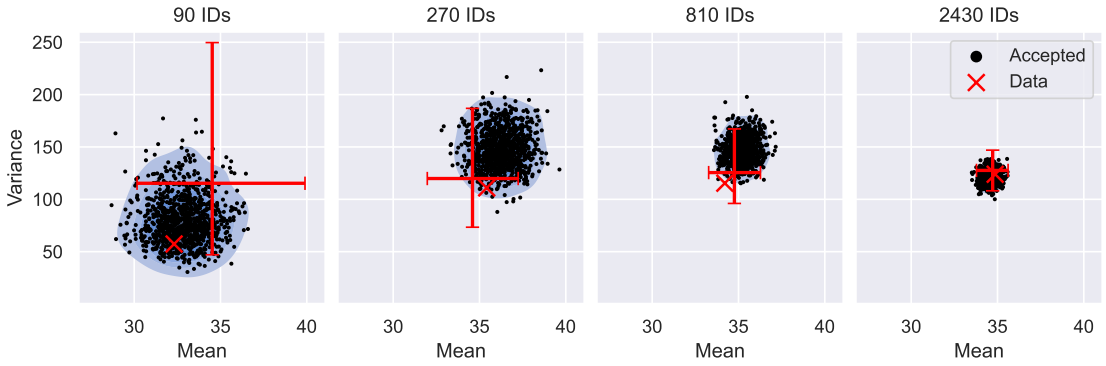


Figure 4.10: ABC interpretation of Filter inference. The figure shows the accepted means and variances of the Gaussian filter at $t = 0.6$ from the inference results in Figure 4.4, where Filter inference is performed on datasets with 90, 270, 810 and 2430 snapshot measurements of the cancer growth model. The accepted summary statistics are illustrated as black scatter points with the corresponding KDE plots shown in blue. The summary statistics of each dataset is illustrated by a red scatter point. The sample variation of the dataset summary statistics is represented by the 5th to 95th percentile of the summary statistic distribution (red bars), estimated from 1000 realisations of each dataset. The exact mean and variance of the data-generating distribution is close to the intersection of the bars.

statistics, we also illustrate the mean and the variance of the datasets in the panels (see red markers). These dataset summary statistics differ from the exact expectation value and the exact variance of the data-generating process as a result of sampling variance. The true mean and the true variance of the data-generating process are also visualised in the figure using red bars. The bars indicate the exact values and their sampling error, here illustrated using the 5th to 95th percentile range obtained from repeated sampling of each dataset. Recall that the dataset in panel 1 has only 15 measurements at $t_5 = 0.6$, while the datasets in the other panels have 45, 135, and 405 measurements at $t_5 = 0.6$, respectively. We estimate the true values of the data-generating process and their sampling errors by sampling 1000 realisations of the datasets. For each dataset, we then compute the means and the variances of the dataset realisations and plot the 5th to 95th percentile interval in the corresponding panel (red bars). The true mean and variance of the data-generating distribution are approximately equal to the median, i.e. the value marked by the intersection of the red bars. As the true data-generating mean and variance are independent of the estimation procedure, the bars intersect approximately at the same point across the panels.

The figure shows that Filter inference, while not explicitly defining error margins, only accepts simulated summary statistics during the inference that are within a certain distance from the data summary statistics (blue area). The magnitude of this distance scales with the number of measured individuals and is comparable to the 5th to 95th percentile interval of the data summary statistics across dataset sizes.

This observation suggests that Filter inference with a Gaussian filter is similar to a special variant of ABC which uses the mean and the variance as summary statistics and which automatically scales its error margins with the estimation error of the data summary statistics. This resemblance of ABC is a consequence of the filter construction (see Equation 4.11). The Gaussian filter likelihood assigns high likelihood only to simulated means and variances that are compatible with the data. When the dataset contains few observations, the filter likelihood is flat and permits large deviations from the mean and variance of the data, while datasets containing more observations become more restrictive. This leads to an automatic scaling of the error margin with the number of measurements, roughly equivalent to the uncertainty of the summary statistic estimates. In Appendix C.9, we provide a proof that in the limit $N \to \infty$, Filter inference with a Gaussian filter is equivalent to ABC based on the mean and variance with error margins tending to zero. In Appendix C.10, we further prove that this equivalence between Filter inference and ABC also extends to other SS-based filters, provided the filter likelihood has a unique and identifiable maximum and the maximum likelihood estimates (MLEs) converge to the summary statistics of the data. We will henceforth refer to these filters as sufficient filters.

Revisiting the results in Figure 4.9, Figure 4.10 also provides an explanation of why an increasing number of measured individuals reduces the ESS for the stochastic approach. Algorithm 4.2 constructs the filter likelihood by sampling simulated measurements, and thus summary statistics, for each proposal randomly, not taking the real measurements into account. The estimation error of these simulated summary statistics is determined by the number of simulated individuals, S, while the error margin in Filter inference, is set by the number of measured individuals, N. Rejections of proposals can therefore randomly occur when the estimation error of the simulated summary statistics is larger

than the error margin of the filter, even when the proposed model parameters are close or equal to the data-generating parameter values. In the experiments from Figure 4.9, we keep the number of simulated individuals fixed ($S = 100$). However, the number of measured individuals increases up to 2430, thus resulting in a low probability of sampling simulated summary statistics within the error margins of the Filter.

As a result, the scaling behaviour of the error margin in Filter inference explains why for a fixed number of simulated individuals, the rejection rate becomes larger as the number of measured individuals increases. Especially, the ESS of the inference using the stochastic variant of the filter posterior deteriorates quickly with the dataset size. In contrast, the deterministic filter posterior performs substantially better, suggesting that random rejections due to ancestral sampling can be avoided by giving the MCMC algorithm the responsibility of sampling all random variables involved in the posterior estimation.

A complementary way to understand the sampling efficiency of Filter inference is provided by the BSL literature. For BSL, low sampling efficiencies have been reported when too few realisations are used to reliably estimate the mean and the covariance of the simulated summary statistics (Frazier et al. 2022). Interpreting each simulated measurement as a 'summary statistic' of the data-generating process, Filter inference with a Gaussian filter is a BSL approach and therefore displays the same sampling behaviour.

4.3.6 Information loss and bias

In contrast to traditional NLME inference, Filter inference is an approximate inference approach. This approximation may result in information loss and bias. The potential for inaccurate inference results is common to all approximate methods, including ABC and BSL, and in this case comes from the filter approximation of the population-level log-likelihood (Equation 4.8).

Filters construct a noisy estimate of the likelihood from measurements of a small number of simulated individuals. The fewer individuals are simulated, the lower the costs of the log-likelihood evaluation. This provides an incentive to reduce the number of simulated individuals as much as possible. However, as illustrated in Figure 4.11, there is a

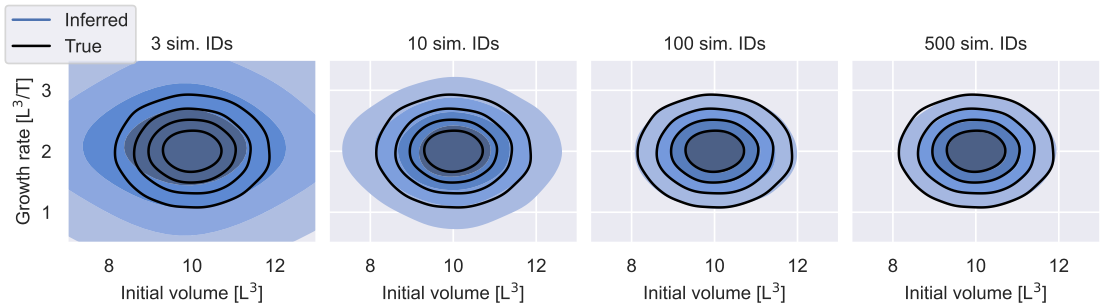


Figure 4.11: Information loss I - number of simulated individuals. The figure shows inferred population distributions from 2430 snapshot measurements of the early cancer growth model, using Filter inference with a Gaussian filter and $S = 3$, $S = 10$, $S = 100$ and $S = 500$ simulated individuals. The inferred distributions are visualised in blue. The different shades of blue indicate the bulk 20%, 40%, 60% and 80% probability regions. The data-generating distribution is illustrated by black contours.

trade-off between computational costs and the accuracy of the IIV estimation. The figure shows inference results for the cancer growth dataset with 2430 snapshot measurements using Filter inference with Gaussian filters with different numbers of simulated individuals. The fewer individuals are simulated, the more information contained in the data is lost. In this case the information loss manifests itself in an overestimation of the IIV.

In our experiments, we achieve reasonable inference results with $S = 100$ simulated individuals across models. However, the number of simulated individuals that achieves an optimal tradeoff between computational costs and information loss will likely vary between problems and modelling rationales.

A second source for information loss and bias is the choice of the filter itself. To illustrate this effect we modify the cancer growth model to include two patient subpopulations with different variants of cancer: an aggressive variant and a moderate variant. This is implemented by defining a covariate-dependent mean growth rate, $\mu_\lambda(\chi) = \mu_{\lambda,m} + \chi\,\Delta\mu_\lambda$, where $\chi = 0$ indicates patients with the moderate variant, and $\chi = 1$ indicates patients with the aggressive variant. This population model results in a multi-modal population distribution

$$p(\psi|\theta,\chi) = \mathcal{N}(y_0|\mu_{y_0},\sigma^2_{y_0})\,\mathcal{N}(\lambda|\mu_\lambda(\chi),\sigma^2_\lambda)\,\delta(\mu_\sigma - \sigma). \tag{4.17}$$

We synthesise two datasets: one dataset with snapshot measurements from 120 individuals; and one with snapshot measurements from 3000 individuals. In both

cases, half of the individuals have the aggressive cancer variant and the other half have the moderate variant. The data are synthesised with $(\mu_{y_0}, \sigma_{y_0}, \mu_{\lambda,m}, \Delta\mu_\lambda, \sigma_\lambda, \mu_\sigma) = (10, 1, 2, 2, 0.5, 0.8)$.

The inference results for different choices of filters and $S = 100$ simulated individuals are illustrated in Figure 4.12. Where tractable, traditional NLME inference is used to infer the exact posterior distribution. Otherwise, the data-generating distribution is used as a reference for the inference results. The figure shows that the quality of the results varies substantially with the choice of the filter. The Gaussian filter and the lognormal filter yield reasonable approximations of the overall individual-level variability, but are not able to resolve the multi-modal structure of the growth rate when only 120 patients are measured. For 3000 measured patients, both filters begin to distinguish the moderate and aggressive cancer growth subpopulations. In comparison, inference with a Gaussian mixture filter with two kernels resolves the multi-modal population structure for both numbers of measured individuals (see middle panel in Figure 4.12). Inference with a Gaussian KDE filter or a lognormal KDE filter similarly resolves the multi-modal population structure, but, here, the filter posteriors underestimate the IIV for both numbers of measured individuals.

The reasons for the observed information loss and biases are different for SS-based filters and KDE-based filters. Intuitively, it is not surprising that SS-based filters with a single mode, such as the Gaussian filter and the lognormal filter, may produce inaccurate inference results when the true measurement distribution is multi-modal. We nevertheless observe in Figure 4.12 that both filters resolve the multi-modal structure in the population distribution when 3000 individuals are measured.

To develop an understanding for this behaviour we refer to the ABC literaure: ABC also infers an approximate posterior distribution based on summary statistics. This posterior converges to the exact posterior distribution in the limit where 1. the error margin of the summary statistics goes to zero; 2. the summary statistics are sufficient; and 3. the number of simulated measurements is the same as the number of observed measurements (Joyce et al. 2008; Wilkinson 2013). Filter inference is equivalent to ABC for certain SS-based filters in the limit where $N \to \infty$ and the error margin of

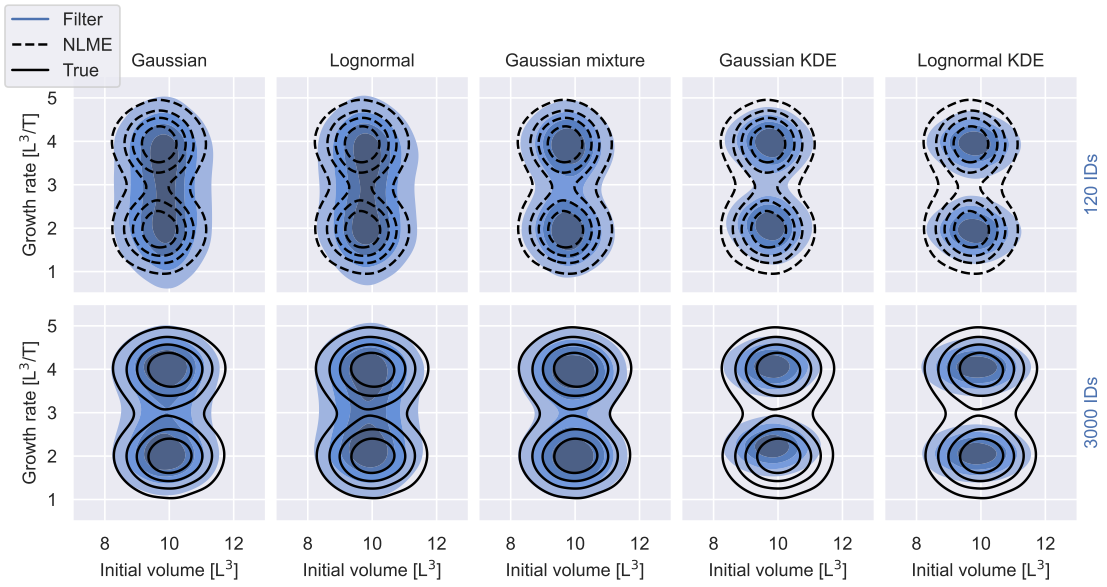


Figure 4.12: Information loss II - choice of filter. The figure shows inferred population distributions from 120 (top row) and 3000 (bottom row) snapshot measurements, using Filter inference with different filter choices and $S = 100$ simulated individuals. The inferred distributions, $\mathbb{E}_{\theta|\mathcal{D}}[p(\psi|\theta)]$, are visualised in blue. The different shades of blue indicate the bulk 20%, 40%, 60% and 80% probability regions. The data-generating distribution is illustrated by black solid lines. The posterior inferred with NLME inference is illustrated by black dashed lines.

ABC vanishes (see subsection 4.3.5). The filter posterior therefore also converges to the true posterior in an analogous limit where 1. the number of measured individuals goes to infinity; 2. the filters are sufficient; and 3. the number of simulated measurements is the same as the number of observed measurements. While we have no formal proof that the Gaussian filter or the lognormal filter are sufficient for the cancer growth model, the convergence of Filter inference for $N \to \infty$ may provide an explanation why the Gaussian and the lognormal filters start to resolve the multi-modal population structure for 3000 measured individuals, but not for 120 measured individuals.

We support this intuition by comparing the histogram over the simulated measurements, obtained during the inference, to the data-generating distribution in Figure 4.13. The closer the histogram approximates the data-generating distribution, the more accurate the inference results. The left panel shows the data-generating distribution (black) and the histogram over the simulations at $t = 0.6$ (blue) for the dataset with 3000 measured individuals, when a Gaussian filter is used for the inference. The histogram

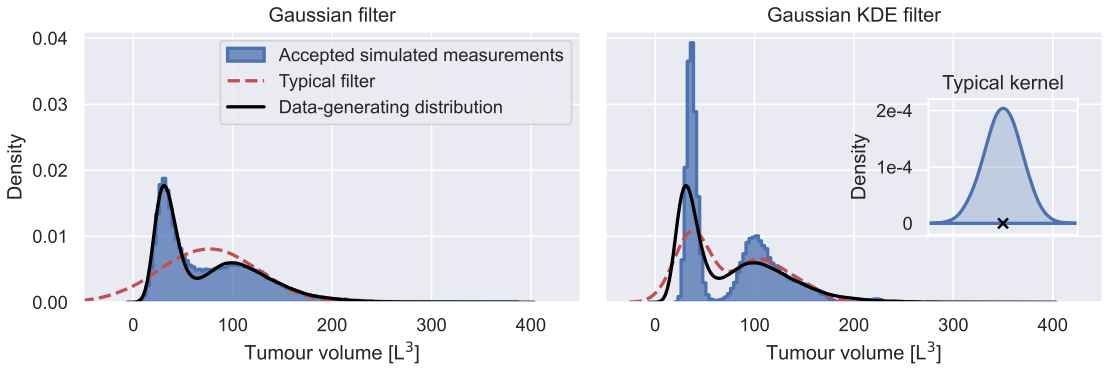


Figure 4.13: Accepted simulated measurements during Filter inference. The figure shows the histograms over all accepted simulated measurements at $t = 0.6$ (blue) during inference with a Gaussian filter (left panel) and a Gaussian KDE filter (right panel) from the bottom panel in Fig 4.12. The data-generating measurement distribution is illustrated in black. A typical realisation of each filter is illustrated in red. The inset figure in the right panel shows a typical kernel (blue) placed on a simulated measurement (black cross) during the construction of the Gaussian KDE filter. The simulated measurement is placed at a tumour volume of 350 for illustration purposes. The scale of the kernel is taken from the filter realisation.

over the measurements approximates the multi-modal data-generating distribution well, despite the use of the unimodal Gaussian filter (see red curve for a typical Gaussian filter sampled during the inference). This indicates that for this model the means and the variances at the measured time points are sufficient statistics for the inference.

KDE-based filters do not have the same convergence behaviour as SS-based filters, as illustrated in Figure 4.12. In fact, the posterior distributions appear to become less accurate for the Gaussian KDE filter and the lognormal KDE filter as the number of snapshot measurements increases. In the right panel of Figure 4.13, we repeat the analysis of the histogram over the simulations for the dataset with 3000 measured individuals for the Gaussian KDE filter. The panel shows that the histogram provides a less faithful approximation of the data-generating distribution than the histogram in the left panel, despite typical Gaussian KDE filters providing more accurate approximations of the data-generating distribution than the Gaussian filter (see red dashed lines).

To develop an understanding for this behaviour, note that the filter log-likelihood is maximised when the KL divergence between the filter and the observed measurement

distribution is minimised

$$\mathrm{KL}(q \,||\, p) = \sum_j \int \mathrm{d}y \, q_j(y) \left(\log q_j(y) - \log p(y|\tilde{Y}_j) \right) \approx \text{constant} - \sum_{ij} \log p(y_{ij}|\tilde{Y}_j).$$

Here, $\mathrm{KL}(q \,||\, p)$ denotes the KL divergence and q_j denotes the data-generating distribution at time t_j, which we approximate by the observed measurements on the right hand side. We identify the last term on the right as the negative log-likelihood of the filter (Equation 4.8). Consequently, maximising the filter likelihood is equivalent to minimising the KL divergence and, therefore, approximating the observed measurement distribution as closely as possible.

For KDE-based filters, the objective of closely approximating the measurement distribution leads to a mismatch between the observed measurement distribution and the distribution of simulated measurements. More specifically, the simulated measurement distribution will underrepresent areas of low probability density in the observed measurements (see right panel in Figure 4.13). This mismatch is a direct consequence of the filter construction. KDE filters are constructed by averaging the densities of S kernels centered at the simulated measurements (see Equations 4.14 and 4.15). Each kernel carries $1/S$ of the total probability density and has a finite width. In the right panel in Figure 4.13, typical kernels extend, for example, 50 tumour volume units in both directions and carry $1/100$ of the total probability density (see inset). As a result, simulated measurements do not have to occupy low density regions with less than $1/S$ probability density in order to still approximate those regions well. In fact, the finite probability density of the kernels makes it unfavourable to occupy regions of the observed measurement distribution with much less than $1/S$ probability density, resulting in a bias away from simulated measurements in low density regions. This explains why the histogram over the simulations underrepresents the low density regions of the data-generating distribution in the Figure 4.13). The absence of simulated measurements in low density regions results in an underestimation of the width of the simulated measurement distribution which, in turn, leads to an underestimation of the IIV (see Figure 4.12).

As the number of simulated measurements tends to infinity, the packaging of the probability density becomes more granular (see Equations 4.14 and 4.15) and the bias towards high density regions vanishes. If the observed measurement distribution is identical to the data-generating distribution, this implies convergence to the data-generating parameters. But, if the observed measurement distribution is not representative for the whole population, KDE-based filters will overfit the observed distribution, and thus Filter inference will underestimate the variability in the population and the uncertainty in the parameter estimates.

In practice, inference is performed on datasets with a finite number of measured individuals using a finite number of simulated individuals. In this context, SS-based filters appear to provide a better accuracy-cost trade-off, especially when informed summary statistic choices are possible, as the middle column of Figure 4.12 demonstrates. Here, we infer the population distribution from the synthesised datasets using a Gaussian mixture filter with $S = 100$ simulated individuals and two kernels. Two Gaussian kernels are able to represent the bimodal structure of the observed measurement distribution more faithfully, resulting in inference results with negligible information loss.

4.4 Conclusion

Filter inference is an efficient and scaleable inference approach for NLME models, enabling the study of variability from previously intractable datasets, for example, snapshot measurements of potentially thousands of individuals. In the context of MIPD, Filter inference enables the use of irregular and sparse measurements across a large number of patients, as might be expected to be available from clinical monitoring, for the improved quantification of inter-individual variability in terms of a population model. This population model can be used as prior information during MIPD.

However, besides its promising applications, Filter inference also introduces new challenges, such as the potential for information loss and bias, which currently can only be fully understood with the help of repeated synthetic data-generation and inference cycles. The efficiency and scalability of Filter inference may also depend on the availability of gradients, which can be difficult to obtain for large systems of differential equations.

4.5 Data and software availability

The data, models, and scripts used in this chapter are fully documented and hosted on GitHub (see repository: https://github.com/DavAug/filter-inference), making the presented results easily reproducible. A user-friendly API for Filter inference has been implemented in the open source Python package `chi` (Augustin 2024).

5

On the locality of information criteria and its consequences for MIPD

This chapter is being prepared for publication and extends earlier work published on bioRxiv:

DA, Wang K, Walz AC, Lambert B, Clerx M, Robinson M, Gavaghan D. Treatment response prediction: Is model selection unreliable?. *bioRxiv*. 2022.

Contents

5.1 Introduction . . . 178
5.2 Methods . . . 180
5.2.1 Model selection . . . 180
5.2.2 Structural uncertainty estimation . . . 185
5.2.3 Information criteria . . . 187
5.2.4 Information criteria in PKPD modelling . . . 188
5.2.5 Information criteria in PKPD modelling: A new perspective 192
5.3 Results & discussion . . . 193
5.3.1 MIPD of infliximab for treating Crohn's disease . . . 193
5.3.2 The locality of information criteria . . . 204
5.4 Conclusion . . . 207
5.5 Data and software availability . . . 209

5.1 Introduction

In the previous chapter (Chapter 4), we focused on the estimation of inter-individual treatment response variability using nonlinear mixed effects (NLME) inference. In the context of model-informed precision dosing (MIPD), this inter-individual variability (IIV) leads to parametric uncertainty which translates into predictive uncertainty for precision dosing (see subsection 2.2.7). The predictive uncertainty can be reduced by estimating parameter values from patient-specific treatment response measurements, but this estimation is specific to the chosen model structure and does not consider uncertainties about the best model to use for prediction.

Despite this, there is frequently uncertainty about the processes involved in the pharmacokinetics and pharmacodynamics (PKPD) of a drug. This is especially true in the early phases of drug development but can also remain a challenge beyond market approval (Ruberg et al. 2023). In some cases, this mechanistic uncertainty can lead to unforeseen adverse events in clinical practice, and even to post-market withdrawals (Koole et al. 2024; Mourya et al. 2024; Tanaka et al. 2024).

The most widely used approach to deal with mechanistic uncertainty is to propose multiple candidate models and to select the model for prediction that is most consistent with the available treatment response data. This consistency is typically quantified using information criteria. For example, Hartmann et al use the Akaike information criterion (AIC) to select a population PKPD model for BIIB059 – a monoclonal antibody developed for the treatment of a chronic autoimmune disease (Hartmann, Biliouris, Naik, et al. 2020). Similarly, Ebid et al use the AIC to select covariates for their follicle-stimulating hormone population PKPD model (Ebid et al. 2021). Other examples of the use of the AIC for model selection include a roblitinib study (Wilbaux et al. 2022), an entrectinib study (González-Sales et al. 2021), and a domagrozumab study (Wojciechowski et al. 2022).

Beyond the AIC, many other information criteria have been developed, e.g. (Gelman, Hwang, et al. 2014; Watanabe et al. 2010). All information criteria have in common

that they estimate the same quantity – the cross-entropy between a candidate model and the data-generating process. This estimation is subject to estimation errors. To avoid selecting incorrect models for prediction due to these errors, Akaike extended his information criteria framework in order to not only quantify the consistency of models with the available measurements, but also to estimate the residual uncertainty for the model selection (Akaike 1978) – an approach commonly referred to as structural uncertainty estimation.

The use of this structural uncertainty approach has recently seen an increasing trend in the PKPD modelling community. For example, Uster et al use AIC-based model weights to improve the predictive accuracy of MIPD for vancomycin (Uster et al. 2021), Concalves et al use AIC-based model weights to estimate the structural uncertainty of viral dynamic models (Goncalves et al. 2020), and Aranzana-Climent et al use model weights to predict clinical study outcomes from preclinical and clinical anti-infective data (Aranzana-Climent et al. 2024).

In this chapter, we study the use of information criteria in the context of MIPD using the treatment of Crohn's disease patients with infliximab as a use case. Our study focusses on the role of information criteria in estimating robust ranges of uncertainty for dose predictions. Here, we use 'robust' to mean ranges of uncertainty that include the correctly individualised dose with high probability. We evaluate this robustness in an idealised simulation setting where the correct dose for each patient is known.

Our findings support the use of structural uncertainty estimation over model selection for robust estimates of predictive uncertainty. However, we find that information criteria have important limitations in the context of MIPD. Some of these limitations are specific to the choice of the information criterion, others are generic to all information criteria. More specifically, popular information criteria, such as the AIC, can be biased towards models with fewer model parameters, regardless of their predictive robustness. This bias depends on the times and dosing regimens for which treatment responses are measured and can lead to premature exclusions of candidate models and to erroneous model predictions. We show that this bias can be traced back to the dynamic and typically nonlinear nature of treatment responses. We also find that a generic limitation of

information criteria is that they cannot distinguish the true data-generating process from models that are different but also consistent with the available measurements – even if those models predict inconsistent treatment responses for alternative dosing regimens. This indicates that model selection is conceptually the wrong framework for robust MIPD, and suggests that instead of choosing a model for prediction, a pool of candidate models consistent with the data and prior knowledge should be maintained to capture the full breadth of predictive uncertainty. We argue that this pool of candidate models should not be guided by parsimony, but by the consistency with the available measurements and prior knowledge.

The chapter is divided into three sections: Methods; Results & discussion; and Conclusion. In the Methods, we first review Akaike's framework for model selection and structural uncertainty estimation, and outline common information criteria. We then place Akaike's framework into the context of PKPD modelling. In the Results & discussion, we illustrate the strengths and limitations of information criteria for MIPD using the treatment of Crohn's disease as a use case. In the Conclusion, we summarise our findings and propose future directions for MIPD research.

5.2 Methods

We first review the general framework for model selection and structural uncertainty estimation based on information criteria. We then discuss this framework in the context of PKPD modelling and highlight limitations that we further explore in the Results & discussion. The data, models, and scripts used in this chapter are hosted on GitHub (https://github.com/DavAug/mipd-information-criteria). Models are implemented in the free and open-source Python package `chi` (Augustin 2024).

5.2.1 Model selection

In 1974, Hirotugo Akaike published a seminal paper on 'a new look at the statistical model identification', where he promoted the simple idea that models should be selected for prediction based on their ability to capture the data-generating process (Akaike

1972, 1974). To quantify this ability, he used a well established measure which assesses the similarity of distributions – the Kullback-Leibler (KL) divergence

$$\mathrm{KL}(q||m) = \int \mathrm{d}y\, q(y) \log q(y) - \int \mathrm{d}y\, q(y) \log p(y|\psi, m). \tag{5.1}$$

Here, $q(y)$ denotes the distribution of measurements from the data-generating process, and $p(y|\psi, m)$ denotes the candidate model of the measurement distribution. ψ denotes the model's parameters, and m denotes the model structure. We will discuss the meaning of a measurement distribution in the context of PKPD modelling in more detail in subsection 5.2.4.

The KL divergence measures the similarity of the model with the data-generating process, in the sense that its value decreases the more the model distribution resembles the data-generating distribution, reaching a value of zero only when the two distributions coincide. This makes it possible to find the best model description of the data-generating process by minimising the KL divergence.

To illustrate this approach, we show the KL divergence for different candidate models in Figure 5.1. We use the measurement process of a patient's drug concentration in the blood as a use case. For now, we consider the measurement to occur at a fixed point in time, so the data-generating process is not dynamic. For illustrative purposes, we presume to know the true distribution of possible measurements at this time point (see black distribution in the top panel). The mean of the distribution represents the blood concentration of the patient, and the width of the distribution reflects the limited measurement accuracy. Using the data-generating measurement distribution, we can calculate the KL divergence for different candidate models. For illustrative purposes, the figure does not consider different model structures, but simply varies the mean of the measurement distribution across the candidate models. However, the concept of using the KL divergence to identify the best model remains the same. We will illustrate the role of the model structure in more detail in section 5.3.

The top panel shows the probability densities of the data-generating process (black) and two arbitrarily chosen candidate models (blue and red). The bottom panel shows the KL divergences between all candidate models and the data-generating process.

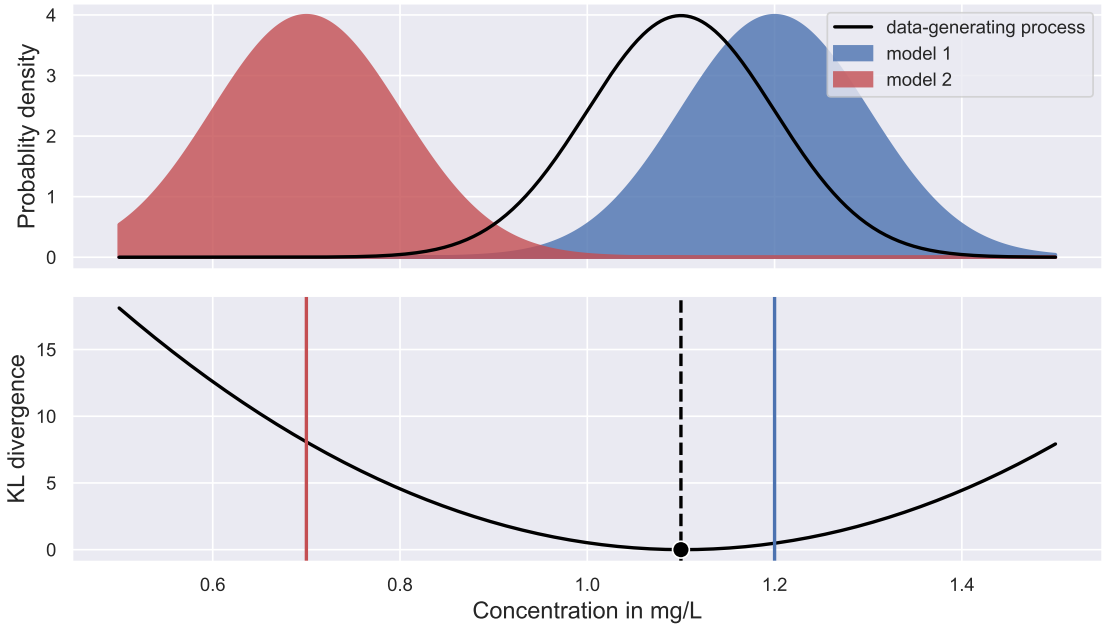


Figure 5.1: Model selection based on the KL divergence. The figure illustrates the values of the KL divergence between the data-generating distribution and different candidate models. The candidate models only differ in their means – the shape and variance are kept constant across models. The top panel shows the data-generating distribution in black, and two arbitrarily chosen candidate models in blue and in red. The KL divergences for all candidate models are shown in the bottom panel (see black curve). The values for the model equalling the data-generating distribution is highlighted using a dashed line, while the values for the highlighted candidate models are illustrated using blue and red lines.

The model parameters corresponding to the distributions highlighted in the top panel are illustrated using vertical lines.

The figure shows that the KL divergence varies across the candidate models (see bottom panel). For example, the highlighted model in red displays a larger KL divergence than the other highlighted model (blue), indicating that the blue model provides a better description of the data-generating process. This is supported by the probability densities in the top panel, showing that the overlap with the data-generating measurement distribution is, indeed, much greater for the blue candidate model than for the red one. The figure also shows that the KL divergence is zero only for one of the models – the model with a measurement mean identical to the data-generating process (see scatter point in bottom panel).

This suggests a simple solution to the structural uncertainty challenge: select the

model with a KL divergence equal to zero for prediction, as this model has the same measurement distribution as the data-generating process, and, therefore, captures its processes the best. The obvious challenge with this approach is that the data-generating distribution, $q(y)$, is typically unknown, making Equation 5.1 impossible to compute. However, Akaike realised that we can estimate parts of the KL divergence using only measurements of the data-generating process, overcoming this practical limitation (Akaike 1974).

More specifically, he noted that the second term of the KL divergence – the cross-entropy of the candidate model with the data-generating process, $S(q||m) = -\int \mathrm{d}y\, q(y) \log p(y|\psi, m)$ – can be estimated using the model's measurement likelihood

$$S(q||m) \approx -\frac{1}{n} \sum_{j=1}^{n} \log p(y_j|\psi, m), \tag{5.2}$$

where $\mathcal{D} = \{y_1, \ldots, y_n\}$ denotes n measurements taken of the data-generating process, and $p(y_j|\psi, m)$ denotes the likelihood of the jth measurement under the model. The first term of the KL divergence – the entropy of the data-generating process, $S(q) = -\int \mathrm{d}y\, q(y) \log q(y)$, – does not depend on the choice of the candidate model and can, therefore, be omitted for the purpose of model selection. This translates the model selection task from finding the candidate model with minimal KL divergence to finding the model with minimal cross-entropy.

Equation 5.2 provides an unbiased estimate of the cross-entropy when the measurements, used to estimate the cross-entropy, are not used to estimate the parameters of the model. If the measurements are used for both, the estimation of the model parameters and the estimation of the cross-entropy, the estimate defined by Equation 5.2 is biased and tends to overestimate the model's ability to describe the data-generating process – a bias that needs to be corrected. Information criteria, such as the AIC, are the most common approach to account for this bias (see subsection 5.2.3).

To illustrate Akaike's model selection framework, we repeat the experiment from Figure 5.1 and estimate the cross-entropy of the candidate models using two sets of measurements taken from the data-generating process (see Figure 5.2): 1. a dataset with $n = 30$ measurements (blue); and 2. a dataset with $n = 10$ measurements (red).

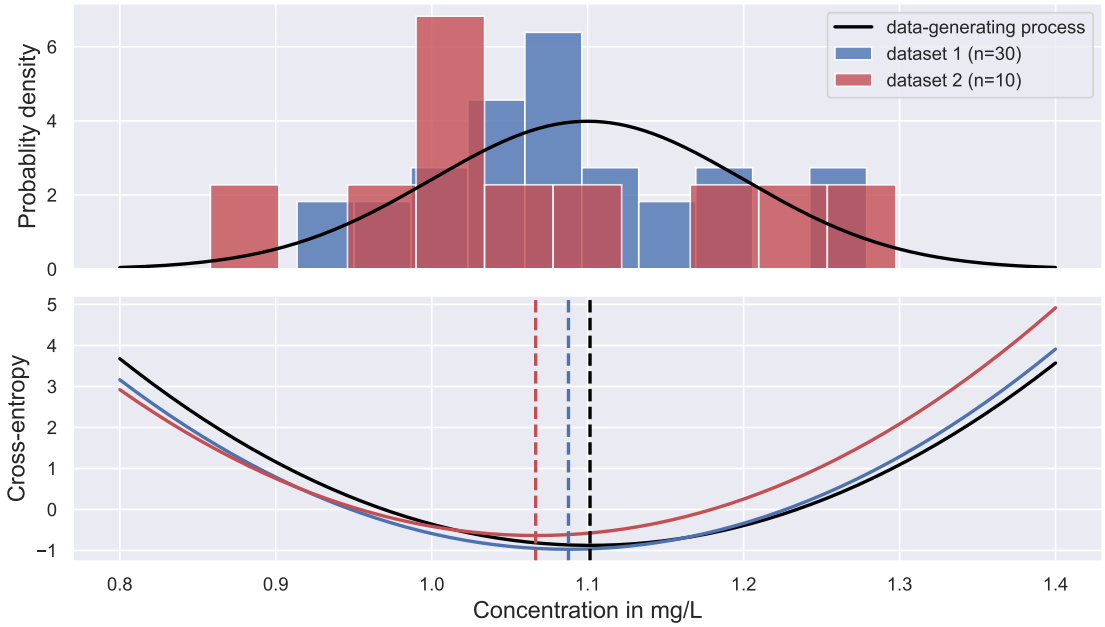


Figure 5.2: Model selection based on cross-entropy estimates. The figure illustrates the values of the cross-entropy between the data-generating distribution and different candidate models. The candidate models only differ in their measurement distribution means – the shape and variance are kept constant across models (see Figure 5.1). The top panel shows the data-generating distribution in black, and two datasets sampled from the data-generating distribution in blue and in red. The exact values of the cross entropy for all candidate models are shown in the bottom panel (see black curve). The values of the cross entropy estimated from the datasets are shown in blue and red. The models minimising the cross-entropies are highlighted using dashed lines.

The exact cross-entropies, calculated using the probability density of the data-generating process, are illustrated in the bottom panel for reference (see black line). The models that minimise the different cross-entropy estimates are illustrated using dashed lines.

The figure shows that the estimates are close to the exact values of the cross-entropy. However, the estimated values do depend on the realisation of the measurements, resulting in different model recommendations for different sets of measurements – the red dataset suggests, for example, that the model with mean 1.07 mg/L provides the best description of the data-generating process, while the blue dataset identifies the model with mean 1.09 mg/L as the best description. In reality, the data-generating distribution has a mean of 1.10 mg/L.

5.2.2 Structural uncertainty estimation

Akaike realised that the existence of such estimation errors causes problems for the robustness of model selection for prediction. He therefore proposed the construction of cross-entropy-based model weights that quantify the residual uncertainty about the best model to use for prediction (Akaike 1978):

$$w(m) = \frac{\mathrm{e}^{-n\,S(q||m)}}{\sum_{m=1}^{M} \mathrm{e}^{-n\,S(q||m)}}. \tag{5.3}$$

Here, n denotes the number of measurements in the dataset, and M denotes the number of candidate models. Note that, in this notation, we implicitly assume that there is a countable number of candidate models, such that we can simply enumerate them, $m = 1, \ldots, M$. This is in conflict with the example in Figure 5.2, where there is an infinite number of candidate models. However, in a typical PKPD modelling scenario, model structures do not vary continuously and Equation 5.3 applies.

Remarkably, Akaike's construction of the model weights was not rooted in strong theoretical arguments, but rather built on his intuition: 1. the model weights should zero-in on the model with the smallest KL divergence the more measurements are available;[1] and 2. in the case of a small number of measurements, all models with the potentially smallest KL divergence should receive substantial weights. Since then, several other arguments for Akaike's model weights have been brought forward, see e.g. (Cavanaugh et al. 2019), culminating in a recent unification of Akaike's model weights with the Bayesian approach to structural uncertainty estimation (Fong et al. 2020).

[1] To show that the model weights in Equation 2.57 singles out the candidate model with smallest KL divergence for a large number of measurements, note that the model weights can be expressed in terms of the KL divergence

$$w(m) = \frac{\mathrm{e}^{-n\,S(q||m)}}{\sum_{m=1}^{M} \mathrm{e}^{-n\,S(q||m)}} = \frac{\mathrm{e}^{n\,S(q)}\mathrm{e}^{-n\,S(q||m)}}{\mathrm{e}^{n\,S(q)} \sum_{m=1}^{M} \mathrm{e}^{-n\,S(q||m)}} = \frac{\mathrm{e}^{-n\,\mathrm{KL}(q||m)}}{\sum_{m=1}^{M} \mathrm{e}^{-n\,\mathrm{KL}(q||m)}}, \tag{5.4}$$

where, in the third equality, we use the definition of the KL divergence, $\mathrm{KL}(q||m) = S(q||m) - S(q)$ (see Equation 5.1). This expression converges to a delta-distribution for sufficiently many measurements, $\lim_{n\to\infty} w(m) = \delta_{m,\,m^*}$, assigning only the model with the minimal KL divergence, $m^* = \mathrm{argmin}_m \mathrm{KL}(q||m)$, with non-zero weight.

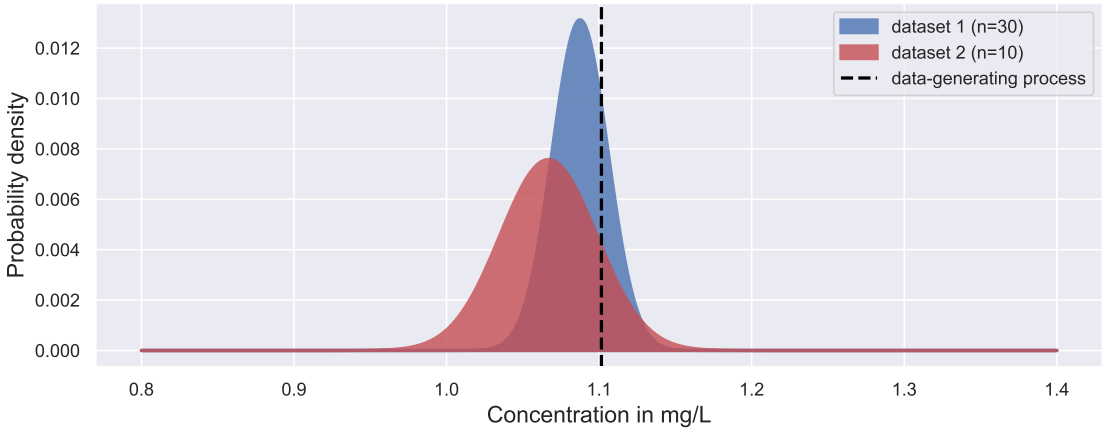


Figure 5.3: Estimating uncertainty using cross-entropy estimates. The figure illustrates the model weights of the candidate models for the two datasets from Figure 5.1. The candidate models only differ in their measurement distribution means – the shape and variance are kept constant across models.

More specifically, Fong and Holmes showed that Akaike's model weights can be interpreted as an approximate special case of the structural posterior distribution,

$$p(m|\mathcal{D}) \approx \frac{\mathrm{e}^{-n\,S(q||m)}\,p(m)}{\sum_{m=1}^{M}\mathrm{e}^{-n\,S(q||m)}\,p(m)}, \tag{5.5}$$

where $p(m)$ denotes the structural prior distribution, quantifying prior knowledge about feasible model structures (see subsection 2.2.10). This allows for a Bayesian inference workflow, already common for parameter estimation (Maier, Hartung, de Wiljes, et al. 2020; Oliver et al. 2023; Uster et al. 2021), where prior knowledge about feasible pharmacological and biological processes can be quantified and updated using available treatment response measurements. Akaike's model weights (Equation 5.3) are a special case of the approximate posterior distribution in the sense that they assign all model structures with equal prior probability, $p(m) = 1/M$.

Figure 5.3 illustrates Akaike's approach to uncertainty estimation using the use case from Figure 5.2, showing the estimates of the model weights for the different candidate models. Recall that, in this example, the models' measurement distributions only differ in their means – the shapes and variances are identical. The figure shows that Akaike's model weights assign substantial probability density to the model with the smallest KL divergence for both datasets (see dashed line). The weight distribution is wider for the

red dataset than than for the blue dataset. This is consistent with the intuition that more measurements contain more information about a data-generating process, illustrating that Akaike's model weights can provide a meaningful quantification of structural uncertainty.

5.2.3 Information criteria

Akaike's approach to model selection and structural uncertainty estimation, as we have introduced it so far, has one practical limitation – it assumes that the available measurements are only used to estimate the cross-entropy of the model (see Equation 5.2). But, in practice, it is more common to first fit models to the data in order to estimate their parameters, and then use the same data for the cross-entropy estimation. This biases estimates of the cross-entropy towards smaller values, resulting in a systematic overestimation of a model's ability to describe the data-generating process.

In his seminal paper from 1974, Akaike found that this bias is asymptotically proportional to the number of model parameters fitted to the measurements when the model parameters are estimated using maximum likelihood estimation (Akaike 1974). He used this insight to construct an estimator for the cross-entropy that is asymptotically unbiased in the limit of infinite measurements

$$S(q||m)_{aic} = -\frac{1}{n}\sum_{j=1}^{n} \log p(y_j|\psi^*, m) + k/n. \tag{5.6}$$

Here, ψ^* denotes the maximum likelihood estimates (MLEs) estimated from the measurements, $\mathcal{D}$, and k denotes the number of fitted parameters. This estimator is commonly referred to as Akaike information criterion (AIC).[2]

In the context of MIPD, the AIC is widely used to estimate cross-entropies. But, for parameter estimation, it is more common to use maximum a posteriori (MAP) estimation. In MAP estimation, the posterior distribution of the model parameters, $p(\psi|\mathcal{D}, m) \propto \prod_j^n p(y_j|\psi, m)\, p(\psi|m)$, is maximised instead of the likelihood. Here, the prior distribution, $p(\psi|m)$, quantifies prior knowledge or prior expectations for feasible

[2] In the literature, the more widely used notation for the AIC is

$$\text{AIC} = 2\log p(\mathcal{D}|\psi^*, m) - 2k,$$

which is equivalent to Equation 5.6 up to a constant normalisation factor.

parameter values, and the posterior distribution estimates the range of parameter values consistent with the measurements and the prior information. This improves the results of the parameter estimation when individual-specific measurements are limited and the likelihood is not informative for all model parameters (Maier, Hartung, de Wiljes, et al. 2020; Oliver et al. 2023; Sheiner and Beal 1982), but it also biases the cross-entropy estimates of the AIC.

Many other information criteria have been developed, see e.g. (Bürkner et al. 2021; Cavanaugh et al. 2019; Gelman, Hwang, et al. 2014; Vehtari, Gelman, and Gabry 2017). For example, an unbiased analogue of the AIC for Bayesian inference is the Watanabe-Akaike information criterion (WAIC) (Watanabe et al. 2010)

$$S(q||m)_{waic} = -\frac{1}{n}\sum_{j=1}^{n}\log p(y_j|\mathcal{D},m) + \frac{1}{n}\sum_{j=1}^{n}\mathrm{Var}\Big[\log p(y_j|\psi,m)\Big], \tag{5.7}$$

where $p(y|\mathcal{D},m) = \int \mathrm{d}\psi\, p(y|\psi,m)\, p(\psi|\mathcal{D},m)$ denotes the posterior predictive distribution of candidate model m. The posterior predictive distribution quantifies the range of measurements consistent with the measurement model and the estimated parametric uncertainty. The variance in the second term is calculated with respect to the posterior distribution of the model parameters, $p(\psi|\mathcal{D},m)$.

Similar to the AIC (Equation 5.6), the WAIC attempts to correct for the bias of the cross-entropy estimation by accounting for the degrees of freedom of the model. However, as it is not straightforward to count the degrees of freedom of Bayesian models – a parameter constrained by less prior information may be considered more 'free' than a parameter that is almost entirely determined by prior knowledge – the WAIC uses the variance of the measurement likelihood under the posterior distribution as a proxy for the degrees of freedom of a model.

5.2.4 Information criteria in PKPD modelling

Information criteria are also used to estimate cross-entropies in the context of PKPD modelling. But, the dynamical nature of treatment responses changes the meaning of the estimates, which has consequences for model selection and structural uncertainty estimation, as we will discuss in the remainder of this chapter.

PKPD models are (semi-)mechanistic or empirical descriptions of treatment response dynamics, typically based on systems of ordinary differential equations (Almoslem et al. 2022; Peters 2021; Yu et al. 2020). These descriptions formalise biomarkers or clinical readouts as functions of time and administered dosing regimens

$$\bar{y}(t, r, \psi, m). \tag{5.8}$$

Here, we use $\bar{y}$ to denote the quantity of interest, t to denote time, r to denote the dosing regimen, and ψ to denote the parameters of the model. m denotes the model structure, which, in this context, determines the treatment response mechanisms described by the model. This formalised relationship can be used to simulate treatment responses for different dosing regimens.

In order to fit model simulations to measurements, PKPD models are typically extended using a measurement noise model: $y = \bar{y} + \epsilon$. Here, y denotes measurements of the quantity of interest consistent with the model, and ϵ denotes deviations of measurements from the PKPD model simulation due to measurement noise. This measurement model represents treatment response measurements as a composite of two contributions: 1. the simulated treatment response; and 2. the realisation of the random measurement noise. Common noise models include normal noise models, and lognormal noise models (Lavielle 2014).

Figure 5.4 illustrates the composite nature of the measurement model using a 1-compartment PK model with indirect dose administration as a use case. The model is defined by two ordinary differential equations

$$\begin{aligned} \frac{\mathrm{d}a_d}{\mathrm{d}t} &= -k_a a_d + r(t), & a_d(t = 0, \psi, m^*) &= 0, \\ \frac{\mathrm{d}a_c}{\mathrm{d}t} &= k_a a_d - k_e a_c, & a_c(t = 0, \psi, m^*) &= 0, \end{aligned} \tag{5.9}$$

where a_d denotes the drug amount in the dose compartment, a_c denotes the drug amount in the central compartment, k_a denotes the absorption rate from the dose compartment into the central compartment, and k_e denotes the elimination rate of the drug from the

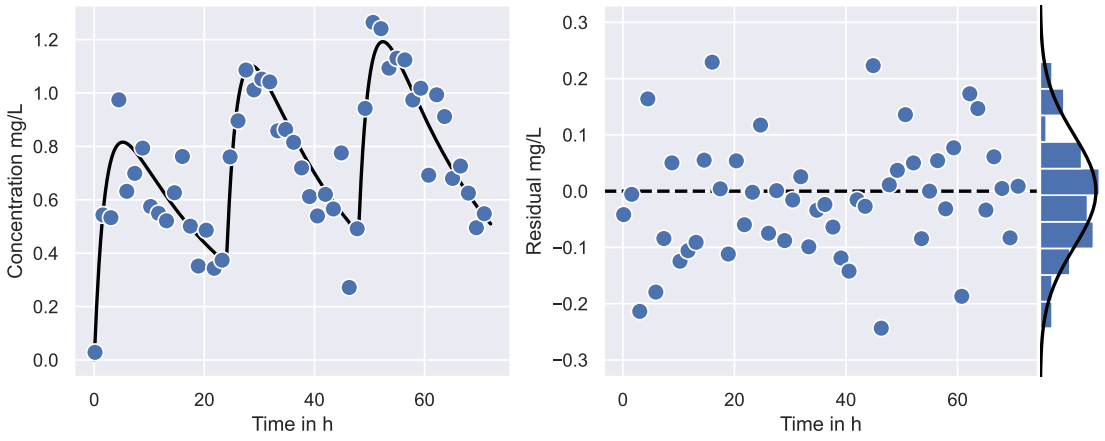


Figure 5.4: The measurement model. The left panel shows a 1-compartment PK model (black) fitted to blood concentration measurements (blue) following daily dose administrations. The right panel shows the differences between the measurements and the model simulation over time. The line of identity, where measurements equal the measured process, is indicated using a dashed line. The histogram over all residuals is shown in the right axis of the panel. The measurement noise model is illustrated in black.

system. $r(t)$ represents a time-dependent dose rate, defined by the dosing regimen.[3] m^* denotes the model structure. The clinical readout, illustrated in the left panel of the figure, is the drug concentration in the central compartment, $\bar{y} = a_c/v$, where v denotes the drug's volume of distribution (see black curve). Simulated blood concentration measurements are visualised using blue scatter points. Note that in clinical practice, the frequency of measurements is typically much lower than shown in Figure 5.4.

The right panel of the figure shows the residuals between the measured values and the treatment response simulations, $y_j - \bar{y}(t_j, r, \psi, m^*)$, over time. These residuals are interpreted under the measurement model as the realisations of the random variable, ϵ. Here, we use y_j to denote the measured value at time t_j. The measurement noise model – here, a normal distribution describing the distribution of the random

[3] Formally, the dosing regimen can be modelled in terms of a sequence of step functions

$$r(t) = \sum_{k=1}^{K} \frac{d_k}{\Delta_k} \Theta(t - t_k)\, \Theta(t_k + \Delta_k - t), \tag{5.10}$$

where K denotes the total number of dose administrations that define the dosing regimen. d_k denotes the kth dose administered at time t_k. The duration of the administration is denoted by Δ_k. $\Theta(x)$ represents the Heaviside step function, which is equal to 0 for $x \leq 0$ and is equal to 1 for $x > 0$. As a result, the dose rate, $r(t)$, assumes the constant value d_k/Δ_k during the kth dose administration. For periods outside dose administrations, the dose rate is zero.

variable, ϵ, – is illustrated on the right, vertical axis of the panel (see black curve). Aggregating the residuals across measurement time points recovers an approximate representation of the measurement noise distribution (see blue histogram). If the measurement model coincides with the data-generating process, the histogram over the residuals converges to the probability density of the measurement noise model as the number of measurements increases.

The measurement model defines a distribution of possible measurement values at each time point, t,

$$p(y|t, r, \psi, m), \tag{5.11}$$

where, by abuse of notation, we use ψ to denote both the parameters of the PKPD model and the parameters of the measurement noise model. Similarly, we use m to denote the combined model structure of the PKPD model and the measurement noise model. For example, in Figure 5.4, the structure of the measurement model is defined by the 1-compartment PK model and the normal measurement noise model, $p(y|t, r, \psi, m^*) = \mathcal{N}(y|a_c(t, r, \psi, m^*)/v, \sigma^2)$. The parameters of the model are the combined parameters of the two submodels: $\psi = (k_a, k_e, v, \sigma)$.

Using the measurement distribution, we can directly apply Akaike's framework for model selection and structural uncertainty estimation to PKPD modelling. In particular, we can estimate the cross-entropy of a measurement model with the data-generating process of a patient's treatment response using the measurement likelihood under the model

$$S_\rho(q||m) \approx -\frac{1}{n}\sum_{j=1}^{n} \log p(y_j|t_j, r, \psi, m). \tag{5.12}$$

Here, $\mathcal{D} = \{(y_1, t_1), \ldots, (y_n, t_n), r\}$ denotes n treatment response measurements under the dosing regimen r, and $p(y_j|t_j, r, \psi, m)$ denotes the likelihood of the jth measurement under the model. We will come back to the meaning of the subscript ρ below. This estimator is unbiased if the measurements are only used for the cross-entropy estimation. Otherwise, information criteria have to be used to account for bias (see subsection 5.2.3). This concludes the review of the existing methodology.

5.2.5 Information criteria in PKPD modelling: A new perspective

A challenge in PKPD modelling is that the definition of the cross-entropy (2nd term in Equation 5.1) does not straightforwardly extend to treatment response models, as now, measurement distributions depend on the time and the dosing regimen (see Equation 5.11). For example, one way to define the cross-entropy is to consider a fixed time and a fixed dosing regimen

$$S_{t,r}(q||m) = -\int \mathrm{d}y\, q(y|t,r) \log p(y|t,r,\psi,m). \tag{5.13}$$

Here, $q(y|t,r)$ denotes the data-generating process of the treatment response at time, t, under the dosing regimen, r. This definition is, arguably, the direct analogue to the previous definition (Equation 5.1). However, as already shown in Figures 5.2 and 5.3, the estimation of the cross-entropy requires several measurements of the data-generating process, and measuring treatment responses repeatedly at exactly the same time point is practically impossible. This definition is therefore not suited for model selection and structural uncertainty estimation.

An alternative choice is to define the cross-entropy globally for all times and dosing regimens: $S(q||m) = \int \mathrm{d}t\, \mathrm{d}r\, S_{t,r}(q||m)$. This definition of the cross-entropy is the only one that truly assesses whether a model can describe the data-generating process for any time and any dosing regimen. However, in practice, estimating the global cross-entropy from measurements requires measurements of the treatment response for all times and all dosing regimens. This is also an impractical definition of the cross-entropy.

A practical definition of the cross-entropy is a local version of the cross-entropy

$$S_\rho(q||m) = -\int \mathrm{d}t\, \mathrm{d}r\, \rho(t,r) \int S_{t,r}(q||m), \tag{5.14}$$

where $\rho(t,r)$ is a weight distribution, quantifying the relative contributions of different times and dosing regimens to the cross-entropy. $S_\rho(q||m)$ includes the point-wise and global definitions of the cross-entropy as special cases. In the former, ρ is a delta-distribution, assigning non-zero weight only to the considered time point and dosing regimen, while in the latter, ρ is a uniform distribution, assigning equal weight

to all times and dosing regimens. More importantly, the local cross-entropy is the quantity estimated by information criteria already used in PKPD modelling, provided ρ is set to the time and dosing regimen distribution of the measurements used for the estimation (see subsection 5.3.2 for more details).

ρ defines the locality of the cross-entropy and specifies the dynamical regimes considered for the comparison between the measurement model and the data-generating process. In the Results & discussion, we show that this locality can make the bias correction of popular information criteria invalid, which has consequences for the robustness of individualised dose predictions.

5.3 Results & discussion

To study the properties of information criteria for PKPD modelling in the context of MIPD, we consider a typical use case for precision dosing: the treatment of Crohn's disease (CD) with infliximab. We study the robustness of dose individualisations using two representative information criteria: 1. the AIC; and 2. the WAIC. We then interpret the observations in light of the locality of the cross-entropy and draw conclusions for the use of other information criteria in MIPD.

5.3.1 MIPD of infliximab for treating Crohn's disease

The study considers the treatment of CD using the monoclonal antibody drug infliximab. CD is a chronic autoimmune disease, characterised by chronic inflammation of the gastrointestinal tract (Roda et al. 2020). We use infliximab as a use case to illustrate the strengths and limitations of information criteria for MIPD. More specifically, we focus on the robustness of dose predictions and their uncertainty estimates.

To study the robustness of predictive uncertainty estimates, we simulate treatment responses of patients using a previously developed infliximab population PKPD model (Ternant et al. 2022) and predict individualised dosing regimens for these patients using a pool of candidate models. Models are selected from this pool for prediction based on information criteria. We consider the population PKPD model to be the 'unknown' data-generating process. In reality, the PKPD of infliximab is likely more complex, but

using this model as the ground truth, we can quantify the robustness of predictions by simulating the true treatment responses to the predicted dosing regimens. For robust predictions, the targeted treatment response is realised for a dosing regimen that is included with high probability in the predicted range of uncertainty.

The typical infliximab treatment strategy for adults is to administer intravenous (IV) doses of 5-10 mg/kg on weeks 0, 2, and 6 of the treatment, followed by a maintenance regimen of the same IV dose every 8 weeks (Sandborn et al. 2002). For children, the dose is more variable, but Clarkston et al find that an infliximab trough concentration of $\geq$29 mg/L at infusion 2 (week 2), and an infliximab trough concentration of $\geq$18 mg/L at infusion 3 (week 6) are strongly associated with improved early outcomes (Clarkston et al. 2019). The MIPD use case in this study is, therefore, to individualise dosages of the second infusion for simulated pediatric patients in order to meet a trough concentration of $\geq$18 mg/L at infusion 3. Measurements after the first infusion are used to inform this prediction. For clarity, we focus first on predicting an individualised dose for just one randomly selected patient, and assess the robustness of the predictions across patients afterwards.

We consider 4 candidate models for the prediction (see Figure 5.5): 1. a 1-compartment PK model; 2. a 2-compartment PK model with linear clearance; 3. a 2-compartment PK model with nonlinear clearance; and 4. a target-mediated drug disposition (TMDD) model. All models are considered in combination with a lognormal measurement noise model, but, in principle, combinations with other noise models could also be included. For formal model definitions, see Appendix D.1. The pool of candidate models reflects the variety of models used to describe the complex PK of monoclonal antibodies (Biliouris et al. 2018; Gibiansky et al. 2008; Hartmann, Biliouris, Naik, et al. 2020; Mager and Jusko 2001; Mager and Krzyzanski 2005; Sánchez-Hernández et al. 2020). For a review of infliximab PKPD models, see (Schräpel et al. 2021).

Note that, by design, the data-generating model structure (m=4) is among the candidate models (see Appendix D.2). This guarantees that any strengths and limitations of the predictive robustness can be attributed to the used information criterion. Note also that the prior distributions of the model parameters are based on the IIV displayed by

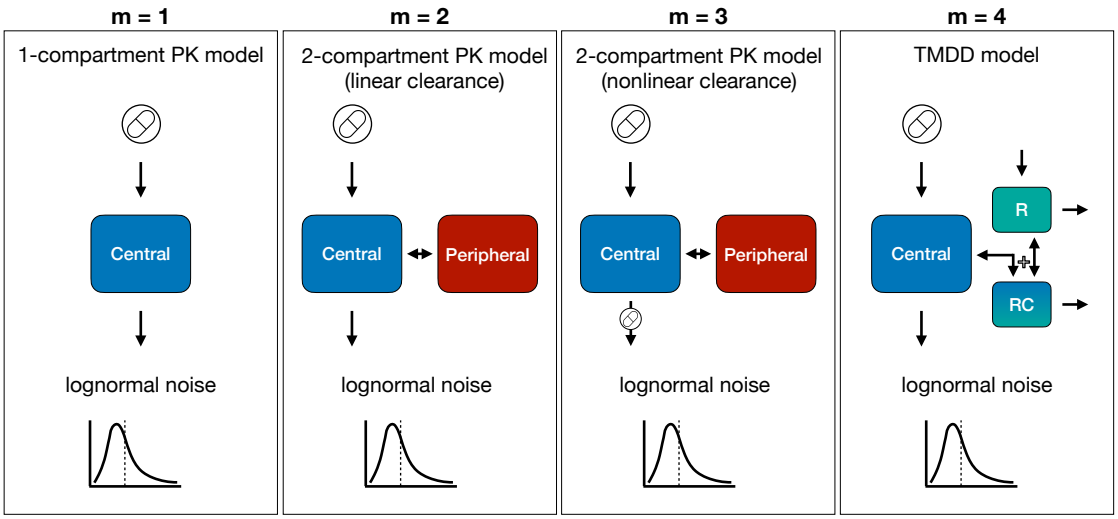


Figure 5.5: Pool of candidate models for infliximab PK. The figure shows candidate models for the PK of infliximab. Each candidate model consists of a mechanistic model and a measurement noise model.

the data-generating process (see Appendix D.12), making sure that also the assumptions for the model parameters are consistent with the data-generating process.

We begin the MIPD study using the AIC (Equation 5.6) for model selection and the estimation of structural uncertainty. In the context of model selection, we use the AIC to determine the model with the highest structural posterior probability mass and choose this model for prediction.[4] For the structural uncertainty estimation, we use the AIC to estimate the full structural posterior distribution (Equation 5.5) and consider the model predictions according to the estimated structural uncertainty. The structural prior distribution is loosely based on the prevalence of the model structures in the monoclonal antibody literature, and assigns the data-generating model structure with a high probability mass (see Table D.6). The model parameters are estimated using MAP estimation (see Appendix D.3 for details).

To evaluate the performance of the uncertainty estimation, we fit the models several times to different numbers of blood concentration measurements. More specifically,

[4]Note that model selection is more usually phrased in terms of choosing the model with the minimal cross-entropy (see subsection 5.2.1). In the absence of prior structural information, i.e. a uniform structural prior distribution, choosing the model with the highest structural posterior probability mass is equivalent to choosing the model with the smallest cross-entropy (see Equation 5.3). However, in this example we make use of prior information for the structural uncertainty estimation, in which case it is more consistent to choose the model with the highest structural probability mass.

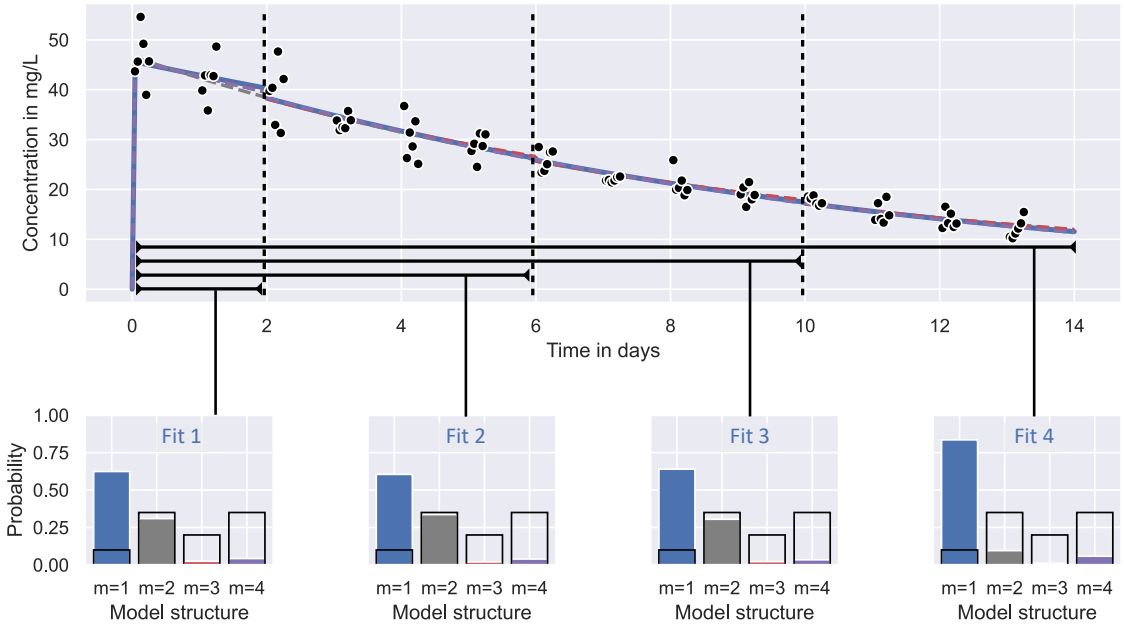


Figure 5.6: AIC-based estimates of the structural uncertainty. The figure shows simulated measurements of a patient's infliximab blood concentration in the top panel using scatter points. Fits of candidate models are illustrated using solid and dashed lines. Each model is fitted 4 times to the measurements: 1. after 2 treatment days; 2. after 6 treatment days; 3. after 10 treatment days; and 4. after 14 treatment days. The fit of the model with the highest estimated probability mass is illustrated using solid lines. The other fits are illustrated using dashed lines. The AIC-based estimates of the structural posterior distributions for each fit are shown in the bottom panel. The structural prior distribution is illustrated using black lines.

we take 6 measurements of the blood concentration each day for 2 weeks after the first infusion (175 mg IV) and fit the models to the data 4 times: 1. after 2 days ($n = 12$); 2. after 6 days ($n = 36$); 3. after 10 days ($n = 60$); 4. after 2 weeks ($n = 84$). The fits of the candidate models and the estimated structural uncertainties are illustrated in Figure 5.6.

The top panel of the figure shows the fits of the models using solid and dashed lines (m=1: blue, m=2: grey, m=3: red, and m=4: purple). The fit of the model with the highest estimated structural probability mass is illustrated using solid lines. The remaining fits are illustrated using dashed lines. The infliximab blood concentration measurements are illustrated using scatter points. The bottom panel shows the estimates of the structural posterior distribution. The structural prior distribution is illustrated using black lines.

The figure shows that the differences of the fits are visually difficult to detect, especially for the fits to all 84 blood concentration measurements after 2 weeks of monitoring. Nevertheless, the AIC-based estimates assign more probability mass to some candidate models than to others. In particular, the 1-compartment PK model (m=1) and the 2-compartment PK model with linear clearance (m=2) are assigned with substantially higher probability masses across the fits than the other two models. For all 4 fits, the 1-compartment PK model has the highest posterior probability mass. On day 14, the model is even estimated to describe the patient's treatment response with a probability of more than 80 %.

The overwhelming support for the 1-compartment PK model suggests that it provides the most consistent description of the patient's treatment response measurements. Notably, the 1-compartment PK model is not the data-generating model structure. To test whether this strong support nevertheless provides a robust estimate of the predictive uncertainty, we simulate the patient's treatment response to the dose individualisations predicted by the fitted models on day 14. We predict the doses by minimising the mean squared error (MSE) of each model's treatment response simulation to the targeted treatment response (18 mg/L) (see Appendix D.4 for details). For the optimisation we use Pints' implementation of the Nelder-Mead optimiser (Clerx, Robinson, et al. 2019).

Figure 5.7 shows the result of the dose individualisation. The left panel of the figure shows the treatment responses of the simulated patient to each dose (blue: m=1; grey: m=2; red: m=3; purple: m=4). The administered dosages are illustrated in the right panel of the figure. We only show the estimated probability masses of the doses under the structural uncertainty approach, where the dose probabilities are equal to the structural posterior probability masses of the candidate models (see Figure 5.6). The probability estimates under a model selection approach are not visualised, as here 100 % of the probability mass is assigned to the model with the highest structural probability. In this case this is candidate model m=1.

The figure shows that none of the predicted IV doses realise the targeted trough concentration at week 6 for the simulated patient. While the dose predicted by candidate model m=4 only narrowly misses the target, the 1-compartment PK model

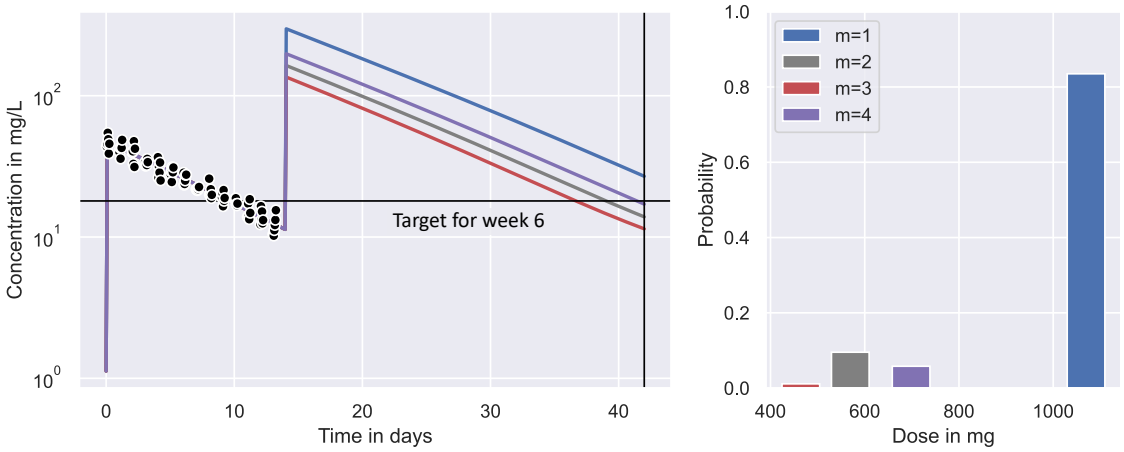


Figure 5.7: AIC-based predictive uncertainty estimation. The figure shows the treatment responses of the patient to the predicted IV doses. The doses are predicted using the inference results on the 14th treatment day in Figure 5.6 ($n = 84$). The left panel illustrates the measurements used for the inference using scatter points. The treatment responses of the patient to the different doses are illustrated using solid lines (blue: m=1, grey: m=2, red: m=3, purple: m=4). The targeted concentration level at day 42 is 18 mg/L. The right panel shows the predicted IV doses, weighted by the estimated uncertainty.

underestimates the treatment response of the patient substantially, resulting in a predicted dose that is more than 300 mg above the correctly individualised dose (correct dose: 734 mg, see Appendix D.7). This overestimation leads to a trough concentration of 26.9 mg/L which is well above the targeted concentration. For the purpose of the CD treatment, the elevated drug exposure may not pose a threat to the success of the treatment as it remains inside therapeutic range (Clarkston et al. 2019). But, the large deviation from the targeted treatment response shows that there are instances, in which the AIC fails to provide robust estimates of the predictive uncertainty. This applies to both, model selection and structural uncertainty estimation.

We analyse why the AIC can fail to provide robust uncertainty estimates in more detail in the next section (subsection 5.3.2). Now, we repeat the MIPD study for another information criterion – the WAIC (Equation 5.7). To this end, we fit to the blood concentrations of the simulated patient a second time – this time using Markov chain Monte Carlo (MCMC) sampling to estimate the posterior distribution (see Appendix D.5 for details). We use Pints' implementation of the Haario-Bardenet Adaptive Covariance Matrix MCMC algorithm for the inference (Clerx, Robinson, et al. 2019).

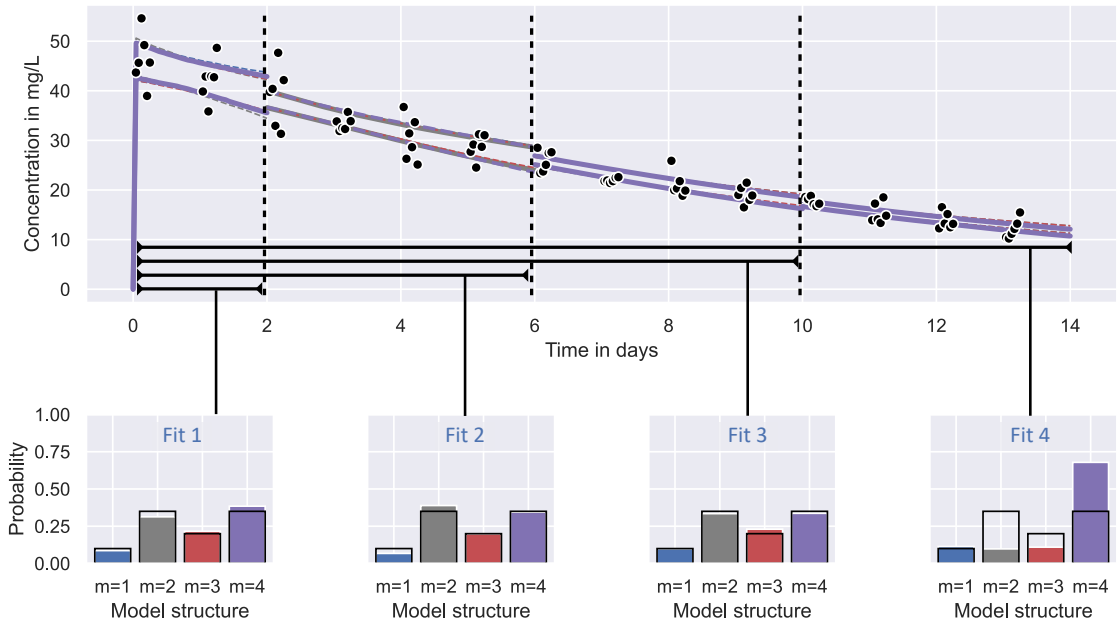


Figure 5.8: WAIC-based estimates of the structural uncertainty. The figure shows simulated measurements of a patient's infliximab blood concentration in the top panel using scatter points. Fits of candidate models are shown using lines, indicating the 5th to 95th percentile range of the noise-free posterior predictive distributions. Each model is fitted 4 times to the measurements: 1. after 2 treatment days; 2. after 6 treatment days; 3. after 10 treatment days; and 4. after 14 treatment days. The fit of the model with the highest probability is illustrated using solid lines. The other fits are illustrated using dashed lines. The WAIC-based estimates of the structural posterior distributions for each fit are shown in the bottom panel. The structural prior distribution is illustrated using black lines.

The new fits and structural uncertainty estimates are illustrated in Figure 5.8. Analogously to Figure 5.6, the top panel of the figure shows the fits to the measurements, and the bottom panel shows the structural posterior distributions. The main difference to the previous figure is that, now, each model fit contains a range of treatment responses due to the estimated parametric uncertainty. This range is indicated using the 5th to 95th percentile range of the posterior predictive distribution of each candidate model, $p(y|t, r, \mathcal{D}, m)$. For details of the estimation, see Appendix D.6.

The figure shows that, similar to Figure 5.6, the differences of the fits are visually difficult to detect. However, in contrast to before, those marginal differences are reflected by the structural uncertainty estimates. More specifically, on days 2, 6, and 10 the structural posterior distributions are almost identical to the prior distribution (see bottom panel). Only on day 14, the structural posterior distribution assigns a large fraction

of the posterior probability mass to the data-generating model structure (m=4). The results are in stark contrast to the previous results in Figure 5.6, where the AIC-based estimates show already on day 2 overwhelming support for the first candidate model (m=1) – a support that only becomes stronger as more measurements become available.

In this simulation study, we know that m=4 is the correct data-generating model structure. We nevertheless test the robustness of the WAIC-based estimates in the same way as before (Figure 5.7). We again predict doses by minimising the MSE using Pints' implementation of the Nelder-Mead optimiser (Clerx, Robinson, et al. 2019). However, this time we predict one dose for each set of parameters supported by the posterior distributions estimated on day 14. This constructs one dose distribution for each candidate model, $p(d|\mathcal{D}, m)$, where we use d to denote the individualised IV dose for infusion 2. The dose distributions reflect the uncertainty of the dose prediction due to the parametric uncertainty of the candidate models. Under a model selection approach, the predicted dose distribution is the one corresponding to the candidate model with the highest structural probability mass, here candidate model m=4 (see Figure 5.8). Under a structural uncertainty approach, the predicted dose distribution is the weighted average of the dose distributions across the candidate models, $p(d|\mathcal{D}) = \sum_m p(d|\mathcal{D}, m)\, p(m|\mathcal{D})$. This weighing represents the predictive uncertainty due to the structural uncertainty and is analogous to the AIC-based structural uncertainty approach in Figure 5.7. $p(d|\mathcal{D})$ is referred to as the posterior predictive dose distribution. For details on the estimation, see Appendix D.8.

The results of the dose individualisation are illustrated in Figure 5.9. Analogously to Figure 5.7, the figure shows the treatment responses of the simulated patient to several predicted doses (see left panel). More specifically, we simulate the treatment response of the patient to 5 000 doses randomly sampled from the predicted dose distributions. The treatment responses corresponding to the predictions of candidate model m=4 are illustrated using the 0.5th and 99.5th percentiles of the treatment response distribution at each time point (see purple lines). Similarly, the treatment responses corresponding to the posterior predictive dose distribution are illustrated using the percentiles of the treatment response distribution (light teal: 0.5th to 99.5th; middle

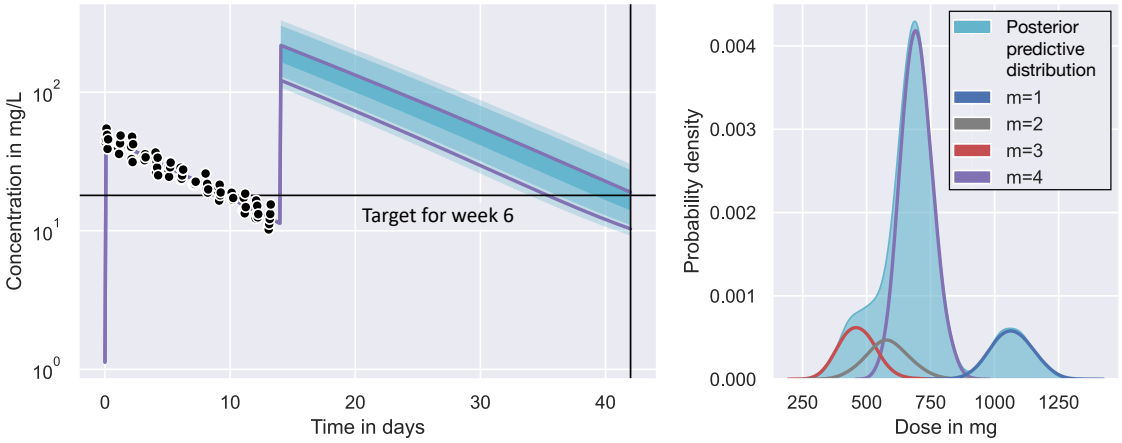


Figure 5.9: WAIC-based predictive uncertainty estimation. The figure shows the treatment responses of the patient to the predicted IV doses on week 2. The doses are predicted using the inference results on the 14th treatment day in Figure 5.8. The left panel illustrates the measurements used for the inference using scatter points. The treatment responses of the patient to the different doses are illustrated using solid lines and shades of teal. The solid lines illustrate the 0.5th to 99.5th percentile interval of the patient's treatment responses to the doses predicted by candidate model m=4. The shades of teal indicate the bulk 66 %, 95 %, and 99 % of the patient's treatment response to the doses predicted by the posterior predictive dose distribution. The targeted concentration level at day 42 is 18 mg/L. The right panel shows the posterior predictive dose distribution (teal) and its contributions from the other models (m=1: blue; m=2: grey; m=3: red; m=4: purple).

teal: 5th and 95th; dark teal: 17th and 83th; these percentiles correspond to the bulk 99 %, 95 %, and 66 % probability masses of the treatment response distribution). The individual treatment response distributions corresponding to the other candidate models are not shown. The right panel shows the posterior predictive dose distribution (teal) and its contributions from the candidate models (m=1: blue; m=2: grey; m=3: red; m=4: purple). The distribution predicted under the model selection approach the one corresponding to candidate model m=4.

The figure shows that both the dose distribution predicted by candidate model m=4 and the full posterior predictive dose distribution assign high probability densities to the correct dose for the simulated patient (correct dose: 734 mg). This indicates that there are instances in which both model selection and structural uncertainty estimation can provide robust estimates of the predictive uncertainty.

Notably, the figure also shows that the predictions obtained using the WAIC differ substantially from the ones obtained using the AIC. To establish whether the differences

between the AIC and the WAIC are systematic for the infliximab use case, we extend the MIPD study to a cohort of $N = 100$ simulated patients. Each patient is randomly drawn from Ternant et al's population model (Appendix D.2). Using the same methodology as before, we predict the doses for the patients multiple times. This time, we also use the fits from day 2, 6, and 10 in addition to the fits from day 14 to predict doses. This allows us to investigate the change of the predictive robustness with the availability of different numbers of measurements. We quantify the robustness of the predictions using the probability masses assigned to the correct dose of each patient, allowing for a 5 % error of the predictions.[5] For AIC-based model selection, this means that a model assigns 100 % probability to the correct dose interval if it predicts a dose within 5 % error of the correct dose, and zero probability otherwise. For the AIC-based structural uncertainty approach, the assignment is more granular, assigning the combined structural posterior probability mass of all predicted doses within the 5 % error interval to the correct dose interval. Similarly, for WAIC-based predictions, the probability of the correct dose interval is the cumulative probability assigned to the error interval. For example, the correct dose interval for the patient in Figure 5.7 is 734 mg $\pm$ 36.7 mg. The AIC-based structural uncertainty approach assigns 6 % probability mass to this range, while the WAIC-based model selection approach estimates 25 % probability mass. For details of the estimation, see Appendix D.9.

The results of the study are illustrated in Figure 5.10. The figure shows the histograms over the posterior predictive probability masses assigned to the correct dose intervals for each patient. The AIC-based estimates are shown in the top panel, and the WAIC-based estimates are shown in the bottom panel. The columns group the results based on the number of measurements used for estimation: column 1 (day 2 | $n = 12$); column 2 (day 6 | $n = 34$); column 3 (day 10 | $n = 60$); and column 4 (day 14 | $n = 84$).

The figure shows that the AIC-based estimates systematically fail to predict doses close to the correctly individualised dosages (see top panel). On day 2, both the model

[5] While chosen arbitrarily, this error tolerance reflects the limited value gain in clinical practice when doses are predicted with infinite precision. Small deviations will often not affect the treatment outcome. The error tolerance is also necessary for our study for technical reasons. The probability mass of any continuous random variable is zero. So we need to define an interval of doses instead of a single dose to obtain a meaningful quantification of the predictive robustness.

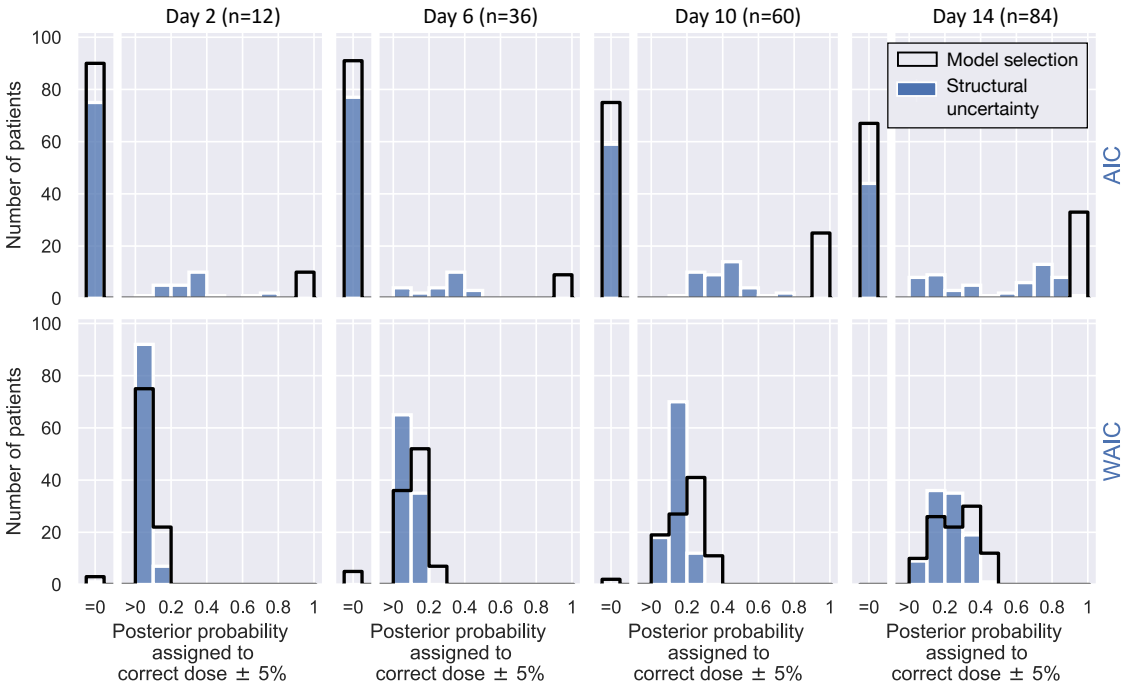


Figure 5.10: Predictive robustness of AIC and WAIC. The figure shows the results of the simulated infliximab cohort study, including $N = 100$ patients. Each panel shows the histogram over the probability masses assigned to the correct dose interval for each patient. The correct dose interval is defined as the 5 % error interval around the patient-specific correctly individualised dose. The results for the model selection approaches are illustrated using a black histograms. The results for the structural uncertainty approaches are illustrated using blue histograms. The top panel shows the results for the AIC-based estimates, and the bottom panel shows the results for the WAIC-based estimates. The columns group the results based on the numbers of measurements used for the estimation.

selection approach and the structural uncertainty approach assign 0 % probability mass to the correct dose interval for almost all patients in the cohort. The predictions become gradually more robust as more measurements become available. But, even after fitting to $n = 84$ treatment response measurements, the model selection approach still assigns 0 % probability mass to the correct dose interval for more than 60 patients. The structural uncertainty approach performs marginally better, failing to assign the correct dose interval with non-zero probability mass only for 44 patients. Overall, the AIC-based estimates of the predictive uncertainty are not a reliable indicator for the robustness of the predictions. This indicates that the AIC is not fit for purpose for precision dosing of infliximab.

The bottom panel shows that the WAIC-based approaches provide substantially more robust dose predictions across patients. Already on day 2, the correct dose intervals

are assigned with substantial probability masses for most patients. The model selection approach fails to predict the correct dose interval only for 3 out of 100 patients, while the structural uncertainty approach captures the correct dose interval for all patients. The predictions become more accurate as more measurements become available, assigning increasingly more probability mass to the correct dose intervals on days 6, 10, and 14. This shows that the WAIC-based approaches are consistent with the intuition that more measurements contain more information about the data-generating process. However, only the structural uncertainty approach is able to consistently assign the correct dose intervals with substantial probability masses for all patients, suggesting that, among the considered approaches, the WAIC-based structural uncertainty approach is the most robust approach for precision dosing of infliximab.

5.3.2 The locality of information criteria

In the previous section, we found that the MAP fits and the AIC fail to provide robust dose predictions for the infliximab use case, even when $n = 84$ treatment response measurements are used for the estimation. In contrast, the posterior predictive distribution and the WAIC estimated substantially more robust ranges of uncertainty. The differences in the predictive robustness are partly due to the differing approaches to the parameter estimation: MAP fits do not reflect parametric uncertainty in their predictions, while posterior predictive distributions do propagate parametric uncertainty to the model predictions. However, the differences in the structural uncertainty estimates for the randomly selected patient also suggest that the AIC and the WAIC provide inconsistent estimates of the models' cross-entropies (see Figures 5.6 and 5.8).

To investigate why the AIC and the WAIC provide different estimates of the cross-entropy, we revisit the results from Figure 5.6. Then, we used the AIC to estimate structural posterior distributions from different numbers of treatment response measurements. We now compare these estimates to the 'true' structural posterior distributions based on the exact cross-entropies of the candidate models (Equation 5.5). We calculate the exact cross-entropies using Equation 5.12 (see Appendix D.10 for details). If the AIC is unbiased, the structural posterior distributions based on the

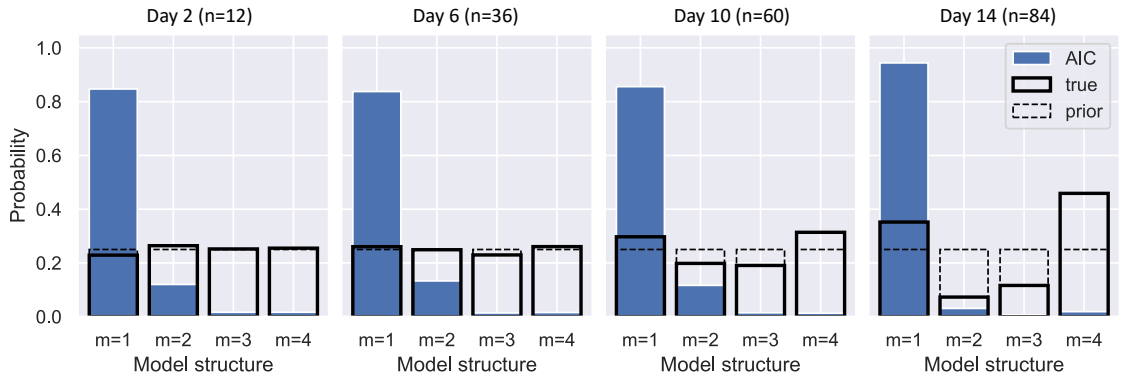


Figure 5.11: AIC-based estimates of the local cross-entropy The figure shows estimates of the structural posterior distribution using the fits and measurements from Figure 5.6. The AIC-based estimates are illustrated in blue. The estimates based on the exact cross-entropies ('true') are illustrated using black solid lines. The structural prior distribution is illustrated using dashed lines.

AIC should be in agreement with the ones based on the exact cross-entropies, up to idiosyncratic estimation errors. For ease of comparison, we assign, this time, equal structural prior probability masses to all candidate models.

The results of the estimation are shown in Figure 5.11. Each panel shows the AIC-based estimates of the structural posterior distribution from Figure 5.6, now estimated with uninformative structural prior distributions (see blue bars). The corresponding estimates based on the exact cross-entropies are illustrated using black bars (true).

The figure shows that the AIC-based estimates of the structural posterior distributions differ substantially from the ones based on the exact cross-entropies. The 'true' structural posterior distributions from day 2 and 6 assign all candidate models with similar probability masses. Only on day 10 and 14, the probability masses begin to be distributed non-uniformly. This indicates that the quality of the fits to the measurements only begins to differ across candidate models for later treatment days. This is in stark contrast to the AIC-based estimates which consistently assign the 1-compartment PK model with the majority of the probability mass, suggesting that the AIC estimates are biased.

This bias does not contradict Akaike's findings from 1972, where he proved that the AIC is an unbiased estimator of the cross-entropy (Akaike 1972). However, in his proof Akaike implicitly assumed the estimation of the global cross-entropy (see subsection 5.2.5). For the global cross-entropy, the degrees of freedom of candidate

models cannot change (with time or with the dosing regimen) and the number of model parameters can be used to correct for the bias of cross-entropy estimates (Equation 5.6). However, when measurements from only specific times and from specific dosing regimens are used for the cross-entropy estimation, information criteria estimate the local cross-entropy defined in Equation 5.14 (see Appendix D.11 for a proof). For this estimation, the AIC can be biased as evidenced by Figure 5.11.

The AIC can be biased because the parameter sensitivity of treatment response simulations varies in time and with the dosing regimen (Yu et al. 2020). This means that the effective number of degrees of freedom, influencing the fit of a model to measurements, can be smaller than the actual number of the model parameters. In those instances, the AIC penalises the estimates of the cross-entropy too much, introducing a bias towards models where the difference between the effective number of model parameters and the actual number of model parameters is minimal.

For example, in Figure 5.11, the exact cross-entropies on day 2 and day 6 indicate that all candidate models fit the available measurements equally well (see black lines). Nevertheless, the AIC-based estimates assign most of the probability mass to the 1-compartment PK model (m=1), followed by the two-compartment PK model with linear clearance with the second most probability mass. The data-generating TMDD model structure is assigned with negligible probability masses. This is because the 1-compartment PK model has only $k = 3$ model parameters, while the other models have $k = 5$ (m=2), and $k = 7$ (m=3 and m=4) model parameters. This means that cross-entropy estimates of the 1-compartment PK model receive the smallest bias corrections. According Akaike, varying bias corrections are justified by the scaling of the bias with the number of model parameters. But Figure 5.11 shows that the additional degrees of freedom do not actually allow the other model structures to fit better to the measurements from day 2 and 6. Consequently, the effective number of model parameters influencing the treatment response simulation of the candidate models is smaller than assumed by the AIC.

To support this argument, we use the same approach to also compare the WAIC-based estimates of the structural posterior distribution to the ones based on the exact

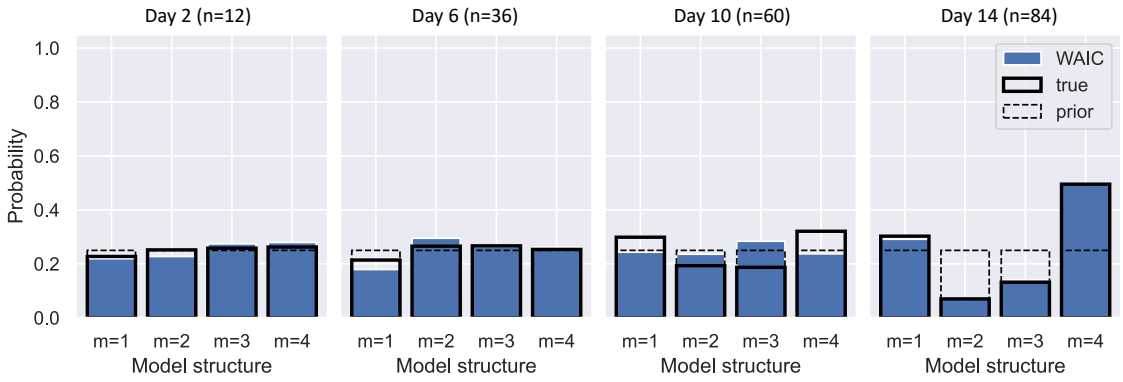


Figure 5.12: WAIC-based estimates of the local cross-entropy The figure shows estimates of the structural posterior distribution using the fits and measurements from Figure 5.8. The WAIC-based estimates are illustrated in blue. The estimates based on the exact cross-entropies ('true') are illustrated using the solid lines. The structural prior distribution is illustrated using dashed lines.

cross-entropies. The WAIC uses the variance of the log-likelihood under the parametric posterior distribution to estimate the effective number of degrees of freedom of a model (Equation 5.7). This effective number is used to account for the bias of the cross-entropy estimates, analogously to the AIC. Crucially, the bias correction of the WAIC is not fixed to the number of model parameters, but depends on the sensitivity of the treatment response simulations to the model parameters. Figure 5.12 shows the comparison of the WAIC-based estimates of the structural posterior distribution with the ones based on the exact cross-entropies.

The figure shows that the WAIC-based estimates are in agreement with the estimates based on the exact cross-entropies, up to idiosyncratic estimation errors. This indicates that there are information criteria that can provide unbiased estimates of the local cross-entropy, providing further evidence that bias corrections based on the total number of model parameters can be invalid in the context of MIPD.

5.4 Conclusion

In this chapter, we have shown that different information criteria have different strengths and limitations for predictive uncertainty estimation in the context of MIPD. In our study, we particularly focussed on the robustness of predicted dose individualisations

and their uncertainty estimates. We quantified this robustness using the probability masses assigned to the correctly individualised doses and their 5 % error intervals (see subsection 5.3.1).

We found that in an infliximab study across 100 simulated patients, AIC-based model selection and AIC-based structural uncertainty estimation failed to predict the correct dose intervals for more than 67 % and 44 % of the patients, even when $n = 84$ treatment response measurements were used for the estimation (see Figure 5.10). The model parameters were estimated using MAP estimation. In contrast, model selection and structural uncertainty estimation based on the WAIC provided substantially more robust estimates of the predictive uncertainty. Here, the parameters were estimated using the full posterior distribution. Already after two treatment days ($n = 12$), the model selection approach successfully predicted the correct dose intervals with high probability for 97 patients, while the structural uncertainty approach predicted the correct dose intervals for all patients. We showed that the differences between the AIC- and the WAIC-based approaches are, in part, a result of the locality of information criteria – information criteria do not assess the consistency of candidate models with the data-generating process globally. Instead, they determine the consistency locally for the times and dosing regimens for which measurements were available (see Equation 5.14). This caused biases of the AIC-based cross-entropy estimates for the infliximab use case due to time- and dosing regimen-varying sensitivities of treatment response simulations to model parameters. In contrast, the WAIC showed no signs of bias in our study.

Our results indicate that information criteria which use the total number of model parameters to account for the bias of estimates can nevertheless provide biased estimates of the cross-entropy in the context of MIPD. Examples include widely used information criteria, such as the AIC (Equation 5.6) and the Bayesian information criterion (BIC) (Neath et al. 2012; Schwarz 1978). In the context of MIPD, the bias 'correction' of such information criteria can bias structural uncertainty estimates towards favouring models where the difference between the effective number and the actual number of model parameters is minimal. The WAIC did not display such biases. This suggests

that information criteria which account for biases not based on the total number of model parameters may not be biased in a similar way.

Our results also suggest that, in the context of MIPD, structural uncertainty estimation is a fundamentally more robust approach for predicting individualised dosages than model selection. The locality of the cross-entropy makes it impossible to distinguish candidate models that display the same treatment response dynamics as the data-generating process at the times and dosing regimens for which measurements are available, even if those models predict very different treatment responses for alternative times and treatment strategies (see subsection 5.3.2). In those instances, model selection is particularly prone to selecting the wrong model structure due to idiosyncratic errors of the cross-entropy estimation. This indicates that model selection is conceptually the wrong framework for robust MIPD and suggests that instead of choosing a model for prediction, a pool of candidate models consistent with the data and prior knowledge should be maintained to capture the full breadth of predictive uncertainty.

While not studied in this chapter, our results also suggest that information criteria may display similar strengths and limitations for model selection and structural uncertainty estimation in NLME inference and population PKPD modelling. This is because the locality of cross-entropies applies also to population PKPD models. However, more research has to be carried out to understand the extent of the bias possible due to mismatches between bias corrections and the effective number of model parameters.

5.5 Data and software availability

The raw data, the data derived during the inference, model specifications (SBML format) and executable scripts (Jupyter notebooks) used to produce the results of this article are published on GitHub: https://github.com/DavAug/mipd-information-criteria.

6 Discussion

Contents

6.1 Main contributions 211
6.2 Future research 214

In this thesis we studied MIPD – an emerging technique for the prediction of patient-specific dosing regimens in clinical practice. MIPD can provide guidance and can help to successfully target desired treatment responses across patients despite substantial inter-individual variability. However, a major challenge for the responsible use of MIPD is the risk for model predictions to be erroneous – i.e. to provide dose recommendations that do not realise targeted treatment responses for to-be-treated patients. Such erroneous predictions can have severe consequences for the well-being of patients, including undertreatments that leave patients unnecessarily exposed to disease progression or other risk factors, and overtreatments that potentially lead to overexposure, elevated toxicity, and dangerous adverse events. This makes it impossible to blindly trust model predictions and complicates their use in clinical practice. In this thesis, we researched how we can minimise the risk for erroneous or wrong predictions as much as possible. In particular, we focussed on the robust estimation of predictive uncertainty.

6.1 Main contributions

One factor that is widely known to be key to both the accuracy of dose predictions and the ease of implementation of MIPD in clinical practice is the type and volume of data available for precision dosing (Darwich et al. 2017; Ribba, Bräm, et al. 2022). The more data are collected, the better dosing regimens can be individualised. However, practical constraints limit how much and what type of data may be available for MIPD. As a result, the trade-off between accuracy and practicality needs to be considered when applying MIPD approaches to medicines. But systematically studying this trade-off is complicated by the astronomical costs and resources needed for clinical trials. In Chapter 3, we developed a framework for clinical trial simulation that enables the resource-efficient optimisation of MIPD approaches before bringing them to the clinic.

Our framework emulates common challenges for successful precision dosing using a clinical trial model with five independent model components: 1. a mechanistic model; 2. a population model; 3. an inter-occasion model; 4. an execution model; and 5. a measurement model. Each model component captures a different complexity of clinical practice that challenges the successful individualisation of treatments, ranging from PKPD-related challenges, such as nonlinear and delayed treatment responses, to practical challenges, such as unintentional deviations from nominal dosing schedules (see subsection 3.2.1). The modularity of the model components simplifies the development process of the clinical trial model, allowing for independent updates of its components throughout the drug development pipeline. This makes it possible to iteratively improve the trial simulations and develop a promising companion MIPD tool as more understanding and information about the drug under trial becomes available (Polasek, Rayner, et al. 2019).

Simulating trials for precision dosing of warfarin, we found that different MIPD models have different strengths and limitations. These strengths and limitations can be generic to the methodology or specific to the model implementation. Modelling approaches that predict dosing regimens exclusively based on covariates of the treatment response variability are generically limited when unexplained IIV, IOV and EV are present (see section 3.3.5). But there were also challenges with monitoring-based MIPD

approaches. For example, we found that Deep RL models adopt a 'control over precision' treatment strategy, where doses are adjusted based on the feedback response from the monitoring data with limited foresight (see section 3.3.5). This lack of precision made the model reliant on indefinite monitoring in the warfarin use case and limited its ability to account for treatment response delays. The PKPD model achieved the highest degree of dosing regimen individualisation among the tested models. The model predicted fully individualised dosing regimens whose level of individualisation increased with the amount of available monitoring data. This indicates that across applications PKPD models are the most promising approach for precision dosing. However, PKPD models are more susceptible to model misspecifications than the other two approaches (see section 3.3.5), necessitating a careful evaluation of the predictive uncertainty prior to MIPD.

The predictive uncertainty of PKPD models is determined by two contributions: 1. parametric uncertainty; and 2. structural uncertainty. Parametric uncertainty arises in the context of MIPD as a consequence of inter-individual treatment response variability. This IIV-induced parametric uncertainty can be estimated using NLME inference, translating IIV into parametric prior distributions which can inform parameter values during MIPD. The quality of the NLME-based IIV estimation is limited by the number of patients whose measurements are used for the inference. However, the NLME inference quickly becomes prohibitively expensive when the number of measured patients increases (Hasenauer et al. 2011). This shortcoming is particularly limiting when patients can only be sparsely monitored, since such sparse measurements are relatively uninformative about the individual treatment response dynamics across patients, requiring large numbers of measurements for good inference results. At the same time, realising the collection of measurements across a large number of patients limits the number of measurements that can feasibly be collected in clinical practice.

In Chapter 4, we developed a new inference method for the estimation of suitable prior distributions for MIPD from measurements across a large number of patients. We call this inference approach 'filter inference'. Filter inference reduces the computational costs of estimating NLME model parameters from measurements originating from a large number of individuals. This new method makes it computationally feasible

to estimate suitable prior distributions for MIPD from sparsely collected data across a large number of patients, making, for example, the improved estimation of inter-individual variability from measurements collected as a byproduct of post-market safety monitoring, at least in principle, possible.

With filter inference, we extended Hasenauer et al's inference approach (Hasenauer et al. 2011). We showed that their simulation-based KDE approximation of the likelihood can be generalised to approximating the likelihood with any distribution based on simulated measurements. The choice of the distribution acts as an information filter for the comparison between simulated and real measurements and influences the quality of the inference results. We showed that choosing distributions or *filters*, taking into account the nature of the problem, could improve the inference results. The filter choice in filter inference has similarities with the choice of summary statistics in ABC and BSL.

In Chapter 5, we focussed our studies on the other contribution to predictive uncertainty of PKPD models: structural uncertainty. The most widely used approach to deal with structural uncertainty is to propose multiple candidate models and to select the model for prediction that is most consistent with available treatment response data. This consistency is typically quantified using information criteria, such as the AIC (Akaike 1974). Alternatively, information criteria can be used to estimate structural posterior distributions to quantify the residual uncertainty for the model selection (Akaike 1978; Fong et al. 2020) – an approach commonly referred to as structural uncertainty estimation.

Using the treatment of Crohn's disease patients with infliximab as a use case, we studied the robustness of dose predictions and their uncertainty estimates for both, model selection and structural uncertainty estimation. Here, we used 'robust' to mean estimated ranges of uncertainty that included correctly individualised doses with high probability (see subsection 5.3.1). We found that information criteria have important limitations in the context of MIPD. Some of these limitations are specific to the choice of the information criterion, others are generic to all information criteria. More specifically, our results suggested that information criteria which use the total number of model parameters to correct for the bias of cross-entropy estimates can provide biased

estimates of the structural uncertainty. Such information criteria include the AIC and the BIC, which are currently the most popular information criteria in the PKPD modelling community. Other information criteria, such as the WAIC, did not play similar biases.

Our results also suggested that, in the context of MIPD, model selection is conceptually the wrong framework for robust MIPD. We showed that information criteria estimate the cross-entropy only locally, making it impossible to distinguish candidate models with the same treatment response dynamics as the data-generating process at the measured times and dosing regimens from the true data-generating process, even if those models predict very different treatment responses for alternative times and treatment strategies (see subsection 5.3.2). In those instances, model selection is particularly prone to selecting the wrong model structure due to idiosyncratic errors of the cross-entropy estimation. This indicates that a pool of candidate models consistent with the data and prior knowledge should be maintained to capture the full breadth of predictive uncertainty.

6.2 Future research

Countless research avenues for improving the robustness of predictive uncertainty estimates in MIPD remain to be explored. In this section, we name a few directly relating to the research performed in this thesis.

The potential of the clinical trial simulation framework developed in Chapter 3 has remained largely underutilised. We used the framework to demonstrate the benefits of simulating clinical trials for understanding the strengths and limitations of existing approaches to MIPD, but we did not use it to systematically develop an optimal approach to warfarin precision dosing (or any other medicine). In the context of warfarin treatment, 'optimality' of the treatment may be quantified based on the metrics used in Figure 3.7: 1. the maintenance INR; 2. the peak INR; 3. the time in the therapeutic range. In addition to optimising these metrics related to the treatment outcome, the simulation framework could be used to improve the data-efficiency of MIPD approaches, making MIPD more suited for clinical practice. The ultimate test for the value gain of simulating

clinical trials also remained unexplored: validating the results of simulated clinical trials by the analogous results of real clinical trials.

In Chapter 4, we developed a scalable approach to NLME inference that utilises filters to make the estimation of NLME model parameter values more efficient. Besides its promising applications, the filters of filter inference also introduce new challenges, such as the potential for information loss and bias, which currently can only be fully understood with the help of repeated synthetic data-generation and inference cycles. An in-depth study – theoretical and/or empirical – of the properties of different filters remains outstanding. In addition to better understanding filter inference, a limitation of both NLME inference and filter inference is that they estimate model parameter by fitting to the available measurements of all patients at the same time. In a post-market monitoring setting, where measurements may be collected on a rolling basis, this creates a substantial computational overhead for the parameter estimation. Alternative approaches to NLME inference could be developed which estimate the parameter values of NLME models iteratively as more measurements become available. Such iterative estimates of posterior distributions are already possible in non-hierarchical Bayesian inference, but are complicated in hierarchical Bayesian inference by the contributions from the individual-level posterior distributions (see subsection 2.2.8).

In Chapter 5, we found that information criteria have different strengths and limitations for estimating predictive uncertainty robustly. In particular, we found that the AIC and the BIC can be biased, while the WAIC did not show any signs of bias. A systematic study of information criteria in the context of MIPD remains outstanding. In addition, our results suggested that information criteria may display similar strengths and limitations for model selection and structural uncertainty estimation in NLME inference and population PKPD modelling. However, to understand the extent of the bias possible due to mismatches between bias corrections and the effective number of model parameters, more research needs to be carried out.

Finally, a general limitation of the structural uncertainty approach studied in this thesis is the assumption that suitable candidate models are being considered (Yao et al. 2018). If none of the considered candidate models approximate the data-generating

process, the estimated predictive uncertainty will not be robust. Metrics could be studied that indicate when none of the candidate models are supported by the available data – the current approach is limited to distributing all probability mass of the structural posterior distributions to candidate models, even if none of them fit the data. Such metrics could reduce the risk for underestimating predictive uncertainty. Similar metrics are already used in machine learning for a different purpose: the data-driven inference of interpretable models by means of evolutionary optimisation (Cranmer 2023).

Appendices

A

Software development: A Python package for treatment response modelling

This supplementary chapter is based on my publication

> **DA**. Chi: A Python package for treatment response modelling. *Journal of Open Source Software*. 2024.

Contents

A.1 Introduction . . . 218
A.2 Methods . . . 219
A.3 Results & discussion . . . 222
A.3.1 Simulating treatment responses . . . 222
A.3.2 Optimising dosing regimens . . . 224
A.3.3 Estimating parametric uncertainty . . . 226
A.3.4 Estimating structural uncertainty . . . 228
A.4 Conclusion . . . 229
A.5 Data and software availability . . . 230

A.1 Introduction

In this chapter, we outline the features of a FOSS Python library for mechanistic treatment response modelling which we developed as a part of this thesis for the purpose of MIPD. We call this library 'Chi' (χ) (Augustin 2024), and it is freely

available at https://github.com/DavAug/chi. Chi provides all functionality needed for mechanistic approaches to MIPD, and it serves as a free tool for the development of publicly shareable treatment response models. Chi's functionality is documented at https://chi.readthedocs.io. Chi has a modular design and interfaces with other open-source libraries, such as Pints (Clerx, Robinson, et al. 2019), Myokit (Clerx, Collins, et al. 2016), and ArviZ (R. Kumar et al. 2019). This makes its components easy to study and its inference methods easy to scrutinise. The author's hope is that Chi will contribute to a responsible use of MIPD in clinical practice and that it facilitates the community-backed development of better MIPD solutions.

A.2 Methods

Chi was developed for MIPD. As a result, Chi supports all modelling aspects described in preceding chapters, including the simulation of PKPD models, PBPK models, and QSP models under arbitrary dosing regimens (subsections 2.2.1 and 2.2.2), as well as the optimisation of those dosing regimens with respect to desired treatment responses (subsection 2.2.3). Chi also supports the estimation of model parameters using maximum likelihood estimation (subsection 2.2.4) or Bayesian inference (sections 2.2.5 and 2.2.6), the simulation of NLME models and the inference of their parameters (subsections 2.2.7 and 2.2.8), as well as the estimation of information criteria and structural posterior distributions (subsections 2.2.9 and 2.2.10).

To support this wide range of functionality in a maintainable way, Chi has been developed making use of small, independent modules that can be easily tested (Khorikov 2020), updated, and extended. These modules can be composed together to implement arbitrarily complex treatment response models, likelihoods, and posterior distributions.

More specifically, each module is defined by a base class with a defined set of methods, specifying the interfaces between the different modules. Each base class can then be extended to implement functionality specific to a particular use case, while preserving the behaviour of the other modules. Chi comes with several of such extensions already pre-implemented. For example, extensions of the measurement noise model base class – the `chi.ErrorModel` – include the `chi.GaussianErrorModel` and the

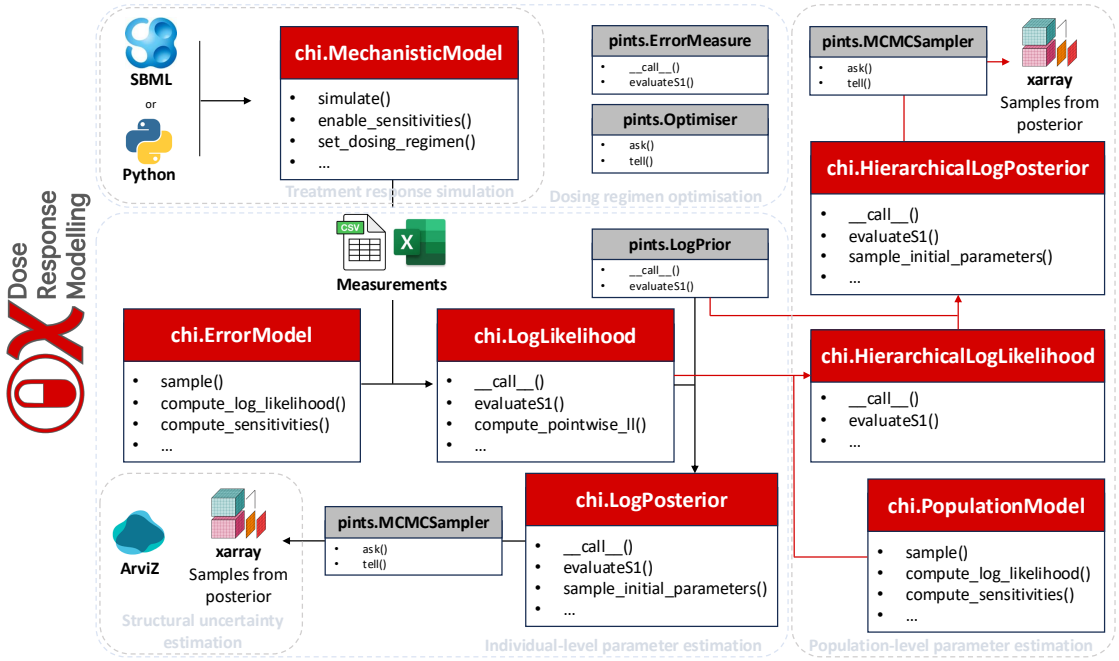


Figure A.1: Architecture of Chi. The figure shows how Chi's modules interface with each other and other open-source libraries. Chi's modules are illustrated in red. Other modules are illustrated in grey, or indicated by the corresponding logo of the Python package. Each box represents a base class which defines a family of classes, containing several extension for different use cases, e.g. an extension of the chi.ErrorModel is the chi.LogNormalErrorModel. All classes needed for a particular task are indicated using grey dashed boxes. For more details, see https://chi.readthedocs.io/en/latest/api.

`chi.LogNormalErrorModel`, implementing a normal measurement noise model and lognormal measurement noise model, respectively. However, even if extensions for a particular use case should not yet exist, the modular architecture makes it easy to implement extensions either by creating requests on GitHub or by implementing them oneself.

Figure A.1 illustrates this modular architecture of Chi, showing its most important base classes, together with some of their key methods (see red boxes). Interfaces with other open-source software are indicated by logos of the corresponding Python packages. Interfaces with Pints – Chi's FOSS inference library of choice (Clerx, Robinson, et al. 2019) – are indicated using grey boxes. We will discuss some conrete code examples in the next section (section A.3).

The figure shows that some use cases require the use of only a small number of classes, while others are more involved, building on several modules. For example, treatment responses can be simulated using just one of Chi's classes – the `chi.MechanisticModel`

and its extensions (see top-left corner in Figure A.1). Mechanistic models can be implemented either using the systems biology markup language (SBML) (Hucka et al. 2003) or using Python code. SBML is a widely established format for specifying differential equation-based mechanistic models and, in Chi, SBML automates the implementation of dosing regimens and parameter sensitivities. However, SBML does constrain the type of possible models. The option to use any Python code for model implementation overcomes this limitation, enabling modellers to also experiment with models which have known analytic expressions or which are not easily represented by differential equations. For more details, see https://chi.readthedocs.io/en/latest/getting_started.

Other tasks can require interfacing with third-party libraries, such as Pints (Clerx, Robinson, et al. 2019). For example, dosing regimen optimisation in Chi makes use of two of Pints' classes, in addition to the `chi.MechanisticModel` (see top-left corner in Figure A.1). The `chi.MechanisticModel` instance simulates the treatment response for different dosing regimens, while the `pints.ErrorMeasure` instance defines the objective for the dosing regimen optimisation. One objective discussed in subsection 2.2.3 is, for example, the minimisation of the mean squared error (MSE) between the simulated treatment response and a target treatment response. This objective can be implemented by extending the `pints.ErrorMeasure` to a MSE as a function of the dosing regimen. The optimal dosing regimen can be found using one of Pints' optimisation algorithms (`pints.Optimiser`).

The most common use case of Chi is the estimation of model parameters from data. To this end, Chi makes it easy to define individual-level log-likelihoods based on measurements, instances of a `chi.MechanisticModel`, and instances of `chi.ErrorModels` – one error model for each measured model output (see `chi.LogLikelihood`). This log-likelihood can be optimised using a `pints.Optimiser` to find the MLEs, or it can be combined with a `pints.LogPrior` to enable the inference of a posterior distribution. Samples from the posterior distribution (`chi.LogPosterior`) can be generated using one of Pints' MCMC algorithms (`pints.MCMCSampler`). Note that that all of Chi's modules in Figure A.1 are differentiable (see methods: `enable_sensitivities()`; `compute_sensitivities()`; and `evaluateS1()`), making the use of gradient-based

inference methods out-of-the-box possible. Such gradient-based methods are particularly important for the inference of NLME model parameters (see Chapter 4).

To infer the parameters of NLME models, several individual-level likelihoods – one for each measured individual – can be combined together with a `chi.PopulationModel` to define a `chi.HierarchicalLogLikelihood`. This hierarchical likelihood can be combined with a `pints.LogPrior` to define a hierarchical posterior distribution from which samples can be generated using a `pints.MCMCSampler`.

Although not featured in Figure A.1, the definition of (hierarchical) log-likelihoods and (hierarchical) log-posteriors rarely requires following the above described steps manually, as Chi provides a convenience class – the `chi.ProblemModellingController` – which automates much of the model building pipeline. More specifically, the controller can be used to automatically define log-posteriors based on a mechanistic model, error models, and measurements. These measurements can be provided in a spreadsheet format, readable by `pandas.DataFrames`, such as CSV or Excel. Using this format, the modelling controller automatically extracts measurements, dosing regimens (if present), and covariates (if present) to construct the posterior distribution. For more details, see https://chi.readthedocs.io/en/latest/getting_started.

A.3 Results & discussion

In this section, we outline some of the code examples used to generate the illustrations in section 2.2. However, we will not display the full breadth of Chi's functionality, as this is already evidenced by the full body of work of this thesis. In particular, we would like to invite the reader to investigate the accompanying GitHub repositories of each research chapter for demonstrations of Chi's use for MIPD (see sections 3.5, 4.5, 5.5).

A.3.1 Simulating treatment responses

In this section, we outline the simulation of treatment responses under different dosing regimens. More specifically, we discuss the code used to simulate the drug concentration time courses illustrated in the right panel of Figure 2.5. The model used for the

simulations is a 1-compartment PK model, and we focus on the time course with 12 h-infusions (see red line).

Code example 1: Simulation of treatment responses

```
import chi
import chi.library
import numpy as np

# Define model
model = chi.library.ModelLibrary().one_compartment_pk_model()
model.set_administration(compartment='central', direct=False)
model.set_outputs(['central.drug_concentration'])

# Define parameters
parameters = [
    0,    # Initial drug amount in central compartment
    0,    # Initial drug amount in dose compartment
    10,   # Volume of distribution in L
    2,    # Absorption rate in 1/h
    0.1   # Elimination rate in 1/h
]

# Define dosing regimen
dose = 5        # in mg
period = 24     # in h
duration = 12   # in h
model.set_dosing_regimen(
    dose=dose, start=0, period=period, duration=duration)

# Simulate concentration
times = np.linspace(start=0, stop=96, num=500)
out = model.simulate(parameters=parameters, times=times)
```

The code example shows the definition of the `chi.MechanisticModel` instance – `model` – in the first code block, after the import statements. Here, we use the predefined 1-compartment PK model from the `chi.library.ModelLibrary` for simplicity. For details on how to implement mechanistic models using SBML or Python code, we refer to https://chi.readthedocs.io.

The next line shows the specification of the route of administration for future dosing regimens – here, an indirect administration to the central compartment through a dose compartment. We also specify the output of the model simulation using the `model.set_outputs` method. Note that the naming of compartments, parameters, and variables is determined by the model definition. Here, the model denotes the drug concentration in the central compartment as '`central.drug_concentration`'.

The next two code blocks define the parameters of the model and the dosing regimen used for simulation. The units of the inputs are chosen according to the model definition. Chi does not automatically convert between units.

In the final code block, the treatment response is simulated. The model output is returned for 500 evenly spread out time points between the start of the simulation (0 h) and the end of the simulation (96 h). The output of the model is returned as a `np.ndarray` of shape `(n_outputs, n_times)`, here (1, 500).

A.3.2 Optimising dosing regimens

In this section, we demonstrate the optimisation of dosing regimens. More specifically, we discuss the code used to optimise a dosing regimen, where the same dose is administered every 24 hours, similar to Figure 2.7. This leaves only one parameter for the optimisation. The code example can be straightforwardly generalised to optimising multiple degrees of freedom of the dosing regimen.

In contrast to Figure 2.7, we continue to use the 1-compartment PK model from the previous code example (Code example 1). First, we define the objective for the optimisation by extending the `pints.ErrorMeasure` base class. We then define an instance of this class, and finally use a `pints.Optimiser` to find the optimal daily dose with respect to the target treatment response.

Code example 2: Optimisation of dosing regimens

```
import pints

# Extend pints.ErrorMeasure to define objective of optimisation
class MSE(pints.ErrorMeasure):
    def __init__(self, model, parameters):
        super(MSE, self).__init__()
        self._model = model
        self._parameters = parameters

        # Define target treatment response
        self._times = np.arange(start=0, stop=96, num=500)
        self._target = 0.42    # in mg/L

    def __call__(self, parameters):
        # Get input parameter
        daily_dose = parameters[0]    # in mg
```

```
        # Simulate treatment response
        self._model.set_dosing_regimen(dose=daily_dose, period=24)
        try:
            out = self._model.simulate(
                parameters=self._parameters, times=self._times)
        except Exception as e:
            print(f'Simulation error occured: {e}.')
            return np.infty

        # Compute MSE
        mse = np.mean((out - self._target)**2)

        return mse

    def n_parameters(self):
        # Here, same dose every day, so only one parameter.
        return 1


# Define objective function (using variables from Code example 1)
objective_function = MSE(model, parameters)

# Find optimal dosing regimen
starting_dose = [10]  # in mg
optimal_dose, mse_score = pints.optimise(
    objective_function, x0=starting_dose, method=pints.PSO)
```

The code example shows the definition of the objective of the optimisation in the first code block, after the import statements. We choose this objective to be the minimisation of the MSE with respect to a target treatment response of 0.42 mg/L.

The MSE class defines the objective using three methods: the `__init__` method; the `__call__` method; and the `n_parameters` method. The `__init__` method defines the model used for simulation (`self._model`: the `chi.MechanisticModel` instance; and `self._parameters`: the model parameters) and the target treatment response (`self._target`: the target drug concentration; and `self._times`: the times at which the simulated treatment response is compared to the target), and is executed upon initialisation of the class.

The `__call__` method formalises the objective of the optimisation, defining the function that the `pints.Optimiser` will try to optimise. More specifically, the optimiser will try to find the set of `__call__`-input parameters (`parameters`) that minimise its return value. Here, the input parameters are just one parameter – the dose amount – and the output of the `__call__` method is the MSE of the simulated treatment response

with respect to the target treatment response. Note that the simulation of the treatment response is wrapped in a `try-except` statement to catch simulation errors during the optimisation. Such simulation errors can occur for extreme dose amounts – typically negative or very large dose amounts –, making the simulation unstable. But, instead of letting those simulation errors terminate the execution, we catch them and return a value that guarantees a rejection of the proposed input parameters (`np.infty`, i.e. ∞).

The `n_parameters` method returns the number of parameters that are being optimised (`len(parameters)`). Here, only one parameter is optimised – the dose amount.

The final two code blocks instantiate the objective function, using the model and the parameters defined in Code example 1, and run the optimisation, using Pints' implementation of the particle swarm optimiser (`pints.PSO`) (Kennedy et al. 1995). The optimisation begins from an arbitrarily chosen starting point (`x0`) and iteratively finds the dose that minimises the MSE using the PSO algorithm. Note that the numerical optimisation is not guaranteed to find the global optimum of the objective function (see subsection 2.2.6). It can, therefore, be helpful to run the optimisation several times from different starting points to establish the robustness of the result.

A.3.3 Estimating parametric uncertainty

In this section, we demonstrate the estimation of model parameters using Bayesian inference. More specifically, we discuss the code used to infer the posterior distributions in Figure 2.13. We focus on the red posterior distribution in the figure.

Code example 3: Estimation of parametric uncertainty

```
# Define measurements (usually imported from CSV or Excel)
times = np.array([0.16, 0.24, 0.32, 0.72, 0.8, 0.88]) * 24  # in h
measurements = [0.96, 0.91, 0.54, 0.62, 0.28, 0.30]  # in mg/L
model.set_dosing_regimen(dose=15, period=24)

# Define log-likelihood
log_likelihood = chi.LogLikelihood(
    mechanistic_model=model,
    error_model=chi.GaussianErrorModel(),
    times=times,
    observations=measurements)
log_likelihood.fix_parameters({
    'central.drug_amount': 0,     # in mg
    'dose.drug_amount': 0,        # in mg
```

```
    'dose.absorption_rate': 2,    # in 1/h
    'Sigma': 0.1})                # in mg/L

# Define posterior distribution
log_prior = pints.ComposedLogPrior(
    pints.GaussianLogPrior(mean=14.3, sd=2),    # v in L
    pints.GaussianLogPrior(mean=0.5, sd=0.1))   # k_e in 1/h
log_posterior = chi.LogPosterior(log_likelihood, log_prior)

# Run inference
n_iterations = 10000  # Number MCMC iterations
controller = chi.SamplingController(log_posterior)
controller.set_sampler(pints.HaarioBardenetACMC)
posterior = controller.run(n_iterations, log_to_screen=True)
```

The code example shows the definition of the posterior distribution in three steps. First, we define the measurements. These measurements are typically be based on physical measurements and, perhaps, imported from a CSV file or an Excel spreadsheet. Here, they are made up. Note that for inference, the dosing regimen is also determined by the measurements, which is why we set the dosing regimen in the first code block.

Second, we define the log-likelihood using the 1-compartment PK model from Code example 1. We also define the measurement noise model for the inference using an extension of the `chi.ErrorModel` – the `chi.GaussianErrorModel`. After defining the log-likelihood, we fix a few of the model parameters, excluding them from the parameter estimation. This is optional and, in this case, is only done to showcase the functionality of fixing parameter values.

Third, we define the log-posterior using the log-likelihood and an extension of the `pints.LogPrior` – the `pints.ComposedLogPrior`. The `pints.ComposedLogPrior` defines the prior distribution one parameter dimension at a time. The order of the parameters is determined by the model. Here, the first parameter of the model is the volume of distribution and the second parameter is the elimination rate. All other model parameters have been fixed in the previous step. To check the order of model parameters, the log-likelihood has a method to return the parameter names, `log_likelihood.get_parameter_names()`.

This log-posterior defines the logarithm of the posterior distribution up to its constant, parameter-independent term (see Equation 2.22). MCMC algorithms can, thus, be used to generate correlated samples from the posterior distribution (see

subsection 2.2.6). This is done in the last block of the code example, where we use the `chi.SamplingController` to execute the inference. In this case, we use the `pints.HaarioBardenetACMC` algorithm – an extenstion of the `pints.MCMCSampler` base class – to sample from the posterior distribution. We run the algorithm for 10 000 iterations. The convergence of the histogram of the samples to the steady state distribution can be investigated using ArviZ, e.g. using `az.plot_traces(posterior)` or using `az.summary(posterior)` (R. Kumar et al. 2019).

A.3.4 Estimating structural uncertainty

In this section, we demonstrate the estimation of structural uncertainty using information criteria. More specifically, we discuss the code used to estimate the PSIS-LOO cross-entropy estimator, similar to Figure 2.28, and extend it to sketch out the estimation of entropy-based structural posterior distributions (Equation 2.65). For simplicity, we continue from the inference results in the previous code example (Code example 3), where we inferred the posterior distribution of the 1-compartment PK model. For the cross-entropy estimation, we make use of AviZ's implementation of the PSIS-LOO estimator (R. Kumar et al. 2019).

Cross-entropy estimators in AviZ estimate the cross-entropy using samples from the posterior distribution and the point-wise log-likelihood contributions from each measurement. To facilitate the collection of the relevant inputs, Chi provides a convenience function, called `chi.compute_pointwise_loglikelihood()`, which calculates the point-wise log-likelihoods and aggregates the data in the correct format for AviZ's estimators.

Code example 4: Estimation of structural uncertainty

```
# Complement posterior samples by pointwise log-likelihoods
warmup = 5000  # Should be determined based on trace plots
samples = posterior.sel(draw=slice(warmup, n_iterations))
results = chi.compute_pointwise_loglikelihood(
    log_likelihood=log_likelihood,
    posterior_samples=samples,
    per_individual=False,
    return_inference_data=True)

# Estimate cross-entropy
```

```
n_measurements = len(measurements)
estimate = -az.loo(inference_data).elpd_loo / n_measurements

# Sketch of structural uncertainty estimation
prior = np.array([
    0.1,  # 1-compartment PK model
    0.4,  # Model 2
    0.5   # Model 3
])
cross_entropy = np.array([
    estimate,  # 1-compartment PK model
    -0.6,      # Model 2
    -0.3       # Model 3
])
posterior = \
    np.exp(-n_measurements * cross_entropy) * prior \
    / np.sum(np.exp(-n_measurements * cross_entropy) * prior)
```

The code example shows that the cross-entropy can be estimates using Chi and ArviZ in just a few lines. In the first code block, we use Chi's convenience function to calculate the point-wise contributions of each measurement to the log-likelihood for all samples from the posterior distribution. Note that we exclude the warm-up iterations of the MCMC algorithm from this calculation.

In the next code block, we estimate the cross-entropy using ArviZ's `az.loo` function, which implements the PSIS-LOO estimator. Note that the function estimates the unnormalised cross-entropy ($-nS$). We therefore normalise the estimate for clarity.

In the final block of the code example, we sketch out the estimation of the entropy-based structural posterior distribution (Equation 2.65). This is only a sketch, because we only estimate the cross-entropy of the 1-compartment PK model. The other cross-entropies are made up for illustrative purposes.

This concludes the results section. For more code examples, tutorials, and a detailed API documentation, we refer to https://chi.readthedocs.io.

A.4 Conclusion

Chi is a FOSS Python package for treatment response modelling that provides all functionality needed for mechanistic approaches to MIPD. In particular, Chi can be used to simulate treatment responses, to optimise dosing regimens, to estimate model parameters, and to estimate information criteria and structural uncertainty.

Some limitations of Chi are that some functionalities still remain to be implemented, including error models for censored measurements and cross-entropy estimation of population models.

A.5 Data and software availability

The data, models, and scripts used in this chapter are fully documented and hosted on GitHub (see repository: https://github.com/DavAug/chi). Chi is a fully free and open-source Python package (Augustin 2024). For install instructions, tutorials, and a detailed API documentation, please see https://chi.readthedocs.io.

B

Supporting information for Chapter 3

Contents

B.1 Warfarin clinical trial model . . . 232
B.1.1 The mechanistic model . . . 232
B.1.2 Implementation of the dosing regimen . . . 232
B.1.3 Simulation of the prothrombin time test . . . 233
B.1.4 The population model . . . 233
B.2 The Regression model . . . 269
B.2.1 Background . . . 269
B.2.2 Implementation . . . 269
B.2.3 Dose prediction . . . 270
B.2.4 Training . . . 271
B.3 The Deep RL model . . . 272
B.3.1 Background . . . 272
B.3.2 Implementation . . . 276
B.3.3 Dose prediction . . . 278
B.3.4 Training . . . 278
B.4 The PKPD model . . . 280
B.4.1 Background . . . 280
B.4.2 Implementation . . . 285
B.4.3 Dose prediction . . . 287
B.4.4 Fitting . . . 289
B.5 Clinical trial simulation . . . 301
B.6 Supplementary tables . . . 305

B.1 Warfarin clinical trial model

B.1.1 The mechanistic model

We model the PKPD of warfarin using Wajima et al's QSP model defined in Table B.1. The table shows the 53 differential equations and 52 algebraic equations of the model, describing the dynamics of warfarin, vitamin K and different coagulation factors. The parameters and variables are shown in verbose names for easier comprehesion of the large number of equations. For example, $\mathrm{d}a_d/\mathrm{d}t = -k_a a_d + r$ from Equation 3.6 is displayed as

$$\frac{d}{d\mathrm{t}}\mathrm{drug_amount} = -\mathrm{absorption_rate} * \mathrm{drug_amount} + \mathrm{dose_rate}.$$

For a detailed discussion of the model we refer to (Wajima et al. 2009). To simplify the implementation of the model, we provide its SBML specification on GitHub (https://github.com/DavAug/mipd-warfarin), which can be used to simulate the model, for example using `chi` (Augustin 2024).

B.1.2 Implementation of the dosing regimen

We implement the dosing regimen using a time-dependent dose rate defined by a sequence of K dose administrations

$$r(t) = \sum_{k=1}^{K} \frac{d_k}{\Delta_k} \Theta(t - t_k)\, \Theta(t_k + \Delta_k - t), \tag{B.1}$$

in accordance with subsection 2.2.2. d_k denotes the dose administered at time t_k. The duration of the administration is denoted by Δ_k. $\Theta(x)$ denotes the Heaviside step function, which is equal to 0 for $x \leq 0$ and is equal to 1 for $x > 0$. As a result, the dose rate, $r(t)$, assumes the constant value d_k/Δ_k during the kth dose administration. For periods outside dose administrations, the dose rate is zero.

For all simulations in this article, the oral dose of warfarin is approximated by a dose duration of $\Delta_k = 0.01\,\mathrm{h}$.

B.1.3 Simulation of the prothrombin time test

The prothrombin time (PT) test quantifies the clotting time of blood. We simulate the PT test following (Wajima et al. 2009): First, a blood sample is taken from the virtual patient, characterised by the states of the model (see Table B.2). The sample is then prepared for the PT test by diluting it $1:2$ with a thromboplastin reagent. We implement this dilution by setting the thrombomodulin amount to 0 and by dividing all state values by a factor of 3. Lastly, we set all external input rates and endogenous production rates to 0 to reflect the *in vitro* nature of the test (see parameters marked by (†) in Table B.3).

The test is initiated by setting the tissue factor concentration to 300 nmol/L. The nonzero tissue factor concentration triggers the extrinsic activation pathway and leads to an increase in the fibrin concentration over time, which in real blood results in clot formation. In the model, clot formation is assumed to occur when the area under the fibrin concentration curve (fibrin AUC) reaches a threshold of 1500 nmols/L. The time elapsed between tissue factor exposure and clot formation defines the PT time. The INR of a blood sample is given by the PT normalised by a reference PT. In our experiments, we use a reference PT of 11.8 s, which we adopted from (Wajima et al. 2009).

Wajima et al's original model does not directly calculate the fibrin AUC (see Table B.1), requiring a post-simulation integration of the fibrin concentration time curve to determine the fibrin AUC. To avoid this complication of the PT test simulation, we complement the model with an auxiliary ODE to integrate the fibrin concentration curve over time during the model simulation

$$\frac{d}{d\mathrm{t}}\mathrm{fibrin_area_under_the_curve} = \mathrm{fibrin_concentration}.$$

This makes it possible to efficiently simulate the fibrin AUC together with the rest of the model using a differential equation solver. The initial value of the fibrin AUC is set to 0.

B.1.4 The population model

We model the treatment response variability by combining the works of Hartmann et al and Hamberg et al (Hamberg, Wadelius, et al. 2010; Hartmann, Biliouris, Lesko, et al. 2016, 2020). The first component of the population model is adopted from Hartmann

et al (Hartmann, Biliouris, Lesko, et al. 2016), who capture IIV using variability in the production rates of prothrombin, protein S, protein C and coagulation factors V, VII, IX, X, XI, XII and XIII (see parameters marked by (†) in Table B.3). The rates are assumed to be normally distributed across individuals

$$p(\psi_\dagger|\theta_{\psi_\dagger}) = \mathcal{N}(\psi_\dagger|\mu_{\psi_\dagger}, \sigma^2_{\psi_\dagger}), \tag{B.2}$$

where $\psi_\dagger$ denotes the production rates. $\mu_{\psi_\dagger}$ and $\sigma_{\psi_\dagger}$ denote the mean and the standard deviation of $\psi_\dagger$ in the population. The standard deviation of the parameters is assumed to be equal to $20\,\%$ of their mean value, $\sigma_{\psi_\dagger} = 0.2\mu_{\psi_\dagger}$.

The second component of the population model is a modification of Hartmann et al's covariate model, published in (Hartmann, Biliouris, Lesko, et al. 2020), based on the work of Hamberg et al (Hamberg, Friberg, et al. 2013). Hartmann et al's model introduces VKORC1 as a covariate of warfarin's EC50 variability and CYP2C9 as a covariate of warfarin's clearance. The typical EC50 is reduced by $22\,\%$ relative to the GG VKORC1 genotype for each A allele

$$\mu_{c_{50}}(v_1, v_2) \;=\; c_{v_1} + c_{v_2}, \tag{B.3}$$

where $v_1, v_2 \in \{G, A\}$ denote the VKORC1 alleles and $c_A = 0.66\, c_G$. Hamberg et al use a structurally identical model for the explained EC50 variability, but also model unexplained variability using random deviations from the typical value, $c_{50} = \mu_{c_{50}}\, \mathrm{e}^{\eta_{c_{50}}}$. $\eta_{c_{50}}$ is a Gaussian random variable with zero mean and variance $\sigma^2_{c_{50}}$. This defines a lognormal distribution of EC50 values in the population

$$p(c_{50}|\theta_{c_{50}}, \chi_{c_{50}}) = \mathrm{LN}\left(c_{50} \,|\, \mu_{c_{50}}(v_1, v_2), \sigma_{c_{50}}\right). \tag{B.4}$$

In our model, we adopt this population model for the EC50 parameter.

The typical clearance in Hartmann et al's model is reduced by $30\,\%$ for the *2*2 CYP2C9 genotype and by $80\,\%$ for the *3*3 genotype relative to the *1*1 genotype (Hartmann, Biliouris, Lesko, et al. 2020). In Hamberg et al's model, the clearance contributions of the CYP2C9 alleles are modelled additively (Hamberg, Friberg, et al. 2013). In addition, the clearance is reduced with the age of the patient. We adopt

Hamberg et al's model structure and use the relationship between the clearance and the elimination rate (clearance $=$ volume $*$ elimination_rate) to define the typical elimination rate as a function of the CYP2C9 alleles and the age

$$\mu_{k_e}(c_1, c_2, t_\star) = (k_{c_1} + k_{c_2})\,(1 - \tanh(r_{age}(t_\star - 71)))\,. \tag{B.5}$$

$c_1, c_2 \in \{*1, *2, *3\}$ denote the CYP2C9 alleles, $t_\star$ denotes the age and r_{age} denotes the reduction rate of the elimination rate with age. Analogous to the EC50, we use a lognormal distribution to model the unexplained variability of the elimination rate

$$p(k_e|\theta_{k_e}, \chi_{k_e}) = \mathrm{LN}\left(k_e \,|\, \mu_{k_e}(c_1, c_2, t_\star), \sigma_{k_e}\right). \tag{B.6}$$

We also adopt Hamberg et al's model of the variability of the volume of distribution, v, which describes the IIV using a lognormal distribution with no additional subpopulation structure

$$p(v|\theta_v) = \mathrm{LN}\left(v \,|\, \mu_v, \sigma_v\right). \tag{B.7}$$

The third component of the population model further extends the covariate model and implements the reduced ability of VKORC to perform two of its primary functions in the coagulation network when polymorphisms in VKORC1 are present (Rost et al. 2004): 1. the promotion of vitamin K conversion to vitamin K hydroquinone (VKH2); and 2. the promotion of vitamin K epoxide (VKO) conversion to vitamin K. We implement this effect by applying the EC50 covariate model to the conversion rates

$$\begin{aligned} \mu_{r_{\mathsf{vk}\to\mathsf{vkh2}}}(v_1, v_2) &= r_{\mathsf{vk}\to\mathsf{vkh2},v_1} + r_{\mathsf{vk}\to\mathsf{vkh2},v_2} \\ \mu_{r_{\mathsf{vko}\to\mathsf{vk}}}(v_1, v_2) &= r_{\mathsf{vko}\to\mathsf{vk},v_1} + r_{\mathsf{vko}\to\mathsf{vk},v_2}. \end{aligned} \tag{B.8}$$

The unexplained variability of the conversion rates in the population is captured using lognormal distributions

$$\begin{aligned} p(r_{\mathsf{vk}\to\mathsf{vkh2}}|\theta_{r_{\mathsf{vk}\to\mathsf{vkh2}}}, \chi_{r_{\mathsf{vk}\to\mathsf{vkh2}}}) &= \mathrm{LN}\left(r_{\mathsf{vk}\to\mathsf{vkh2}} \,|\, \mu_{r_{\mathsf{vk}\to\mathsf{vkh2}}}(v_1, v_2), \sigma_{r_{\mathsf{vk}\to\mathsf{vkh2}}}\right) \\ p(r_{\mathsf{vko}\to\mathsf{vk}}|\theta_{r_{\mathsf{vko}\to\mathsf{vk}}}, \chi_{r_{\mathsf{vko}\to\mathsf{vk}}}) &= \mathrm{LN}\left(r_{\mathsf{vko}\to\mathsf{vk}} \,|\, \mu_{r_{\mathsf{vko}\to\mathsf{vk}}}(v_1, v_2), \sigma_{r_{\mathsf{vko}\to\mathsf{vk}}}\right). \end{aligned} \tag{B.9}$$

The remaining parameters of the model are assumed to display no IIV.

In summary, the population model of the warfarin clinical trial model describes IIV using variability in the EC50, the elimination rate, the vitamin K conversion rates, the volume of distribution as well the production rates of different coagulation factors

$$\begin{aligned} p(\psi|\theta,\chi) &= p(c_{50}|\theta_{c_{50}},\chi_{c_{50}})\, p(k_e|\theta_{k_e},\chi_{k_e})\, p(r_{\mathsf{vk}\to\mathsf{vkh2}}|\theta_{r_{\mathsf{vk}\to\mathsf{vkh2}}},\chi_{r_{\mathsf{vk}\to\mathsf{vkh2}}}) \\ &\quad p(r_{\mathsf{vko}\to\mathsf{vk}}|\theta_{r_{\mathsf{vko}\to\mathsf{vk}}},\chi_{r_{\mathsf{vko}\to\mathsf{vk}}})\, p(v|\theta_v) \prod_{\dagger} p(\psi_\dagger|\theta_{\psi_\dagger}) \prod_k \delta(\mu_{\psi_k}-\psi_k). \end{aligned} \tag{B.10}$$

All remaining parameters, here indexed by k, display no IIV. The parameters of the model, θ, are the parameters of the individual marginal distributions. The covariates of the model are the VKORC1 alleles, the CYP2C9 alleles and the age, $\chi = (v_1, v_2, c_1, c_2, t_\star)$.

The full set of parameter values used for the simulation is provided in a CSV file on GitHub (https://github.com/DavAug/mipd-warfarin). The parameter values are largely adopted from Hartmann et al and Hamberg et al (Hamberg, Wadelius, et al. 2010; Hartmann, Biliouris, Lesko, et al. 2016, 2020). However, IIV in the conversion rate variability was not previously captured by the models, leaving the model parameters in Equation B.9 unspecified.

To calibrate the parameters of the conversion rate distributions, we begin by first setting the typical conversion rates for individuals with the GG VKORC1 genotype to the conversion rates provided by Hartmann et al (Hartmann, Biliouris, Lesko, et al. 2020). The reduction of the typical rates with VKORC1 polymorphisms and the scale parameter of the distribution is set to the values of the EC50 distribution. Simulating the baseline INR without warfarin treatment for 1000 random individuals shows that this initial set of parameter values produces INRs around 1, indicating reasonable simulation results (see black line in Figure B.1). However, for some individuals, the baseline INR displays values > 2, which we identify as unrealistically high. To avoid too high baseline INRs, we reduce the effect of VKORC1 polymorphisms to a total reduction of the typical conversion rates of $20\,\%$ relative to the GG genotype. This results in a baseline INR distribution with less extreme values (see blue histogram in Figure B.1). It should be noted that we did not have access to real baseline INR measurements for this calibration. Calibrating the model to real measurements will likely improve the parameter estimates.

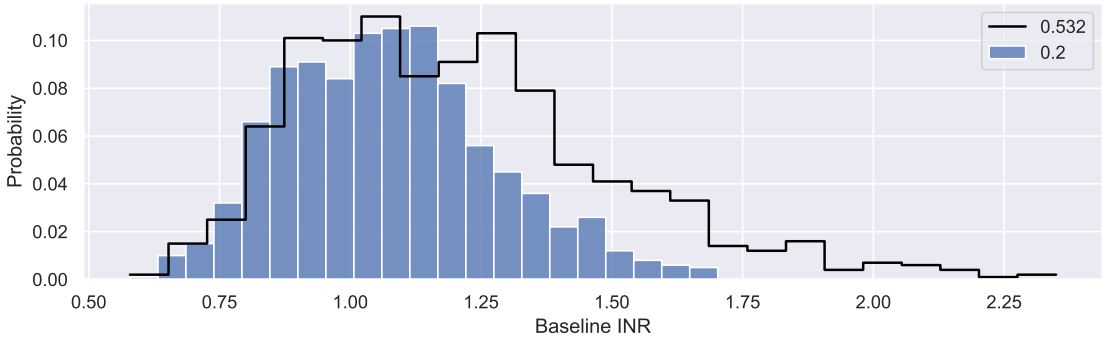


Figure B.1: Calibration of baseline INR distribution. The figure shows the histogram over baseline INR values for 1000 virtual patients simulated with the warfarin clinical trial model before (black) and after (blue) the calibration. Before the calibration the model assumes a $53.2\,\%$ reduction of the vitamin K conversion rate for individuals with the AA VKORC1 genotype relative to individuals with the GG VKORC1 genotype. The calibrated model assumes reductions of $20\,\%$.

In a second calibration step, we adjust the G VKORC1 allele contribution to the EC50 and the scale parameter of the EC50 distribution. This is necessary because: 1. adopting Hamberg et al's value for the volume of distribution shifts the simulated warfarin concentrations relative to Hartmann et al's model which is not accounted for when keeping the original c_G value in Equation B.3; and 2. the detailed description of warfarin's mode of action in the clinical trial model distinguishes variability contributions from the EC50 and the vitamin K conversion rates to the overall treatment response variability. This reduces the variability contribution from the EC50 variability relative to Hamberg et al's model.

To calibrate the two parameters, we simulate the maintenance warfarin dose distribution and compare it to clinical maintenance doses reported in the literature (IWPC 2009). A challenge with this comparison is that individuals in the clinical dataset have varying INR values. For the comparison, we therefore focus on individuals with INRs inside the therapeutic range. To generate a maintenance dose distribution with the clinical trial model, we simulate the treatment response of 1000 individuals treated with warfarin for 56 days. We start the treatment for all individuals with the same dose ($10\,\mathrm{mg}$). We then measure the INR every 7 days and adjust the dose using linear extrapolation ($d' = d\,y^*/y$), when the INR is outside the therapeutic range. d denotes the current dose, d' denotes the adjusted dose, y denotes the INR measurement and

$y^* = 2.5$ denotes the target INR. If the measurement is inside the therapeutic range, the dose is not adjusted. We classify an individual as 'having reached maintenance treatment' if the individual's INR measurement is inside the therapeutic range for the last 3 weeks of the simulation.

In Figure B.2, we show the clinical maintenance dose distribution (black curve) and the simulated maintenance dose distribution (blue histogram) after the model calibration. The two distributions are in good agreement.

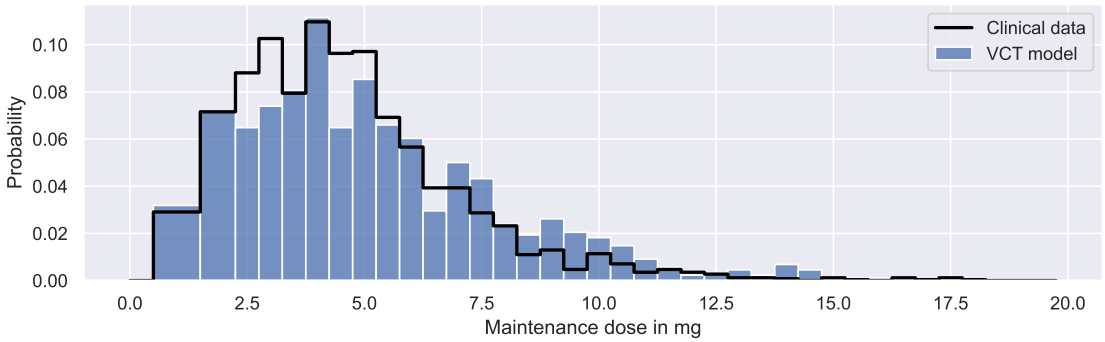


Figure B.2: Clinical maintenance dose distribution and simulated maintenance dose distribution. The figure shows the histogram over warfarin maintenance doses administered in clinical practice (black) and during the clinical trial simulation (blue). The clinical data contains maintenance dosages from 2543 individuals with INRs inside the TR. The data simulated with the clinical trial mdoel contains maintenance dosages from 846 virtual patients with INRs inside the TR.

Table B.1: Definition of the mechanistic model

$$\frac{d}{dt}\text{coagulation_factor_ii_amount}$$
$$= \text{central_size} * \text{production_rate_ii} * \text{vitamin_k_hydroquinone_concentration} - \text{central_size} * \text{degradation_rate_ii} * \text{coagulation_factor_ii_concentration}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_av_ax_complex_ii} * \text{av_ax_complex_concentration} * \text{coagulation_factor_ii_concentration}}{\text{half_maximal_effect_concentration_av_ax_complex_ii} + \text{av_ax_complex_concentration}}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_ax_ii} * \text{activated_coagulation_factor_x_concentration} * \text{coagulation_factor_ii_concentration}}{\text{half_maximal_effect_concentration_ax_ii} + \text{activated_coagulation_factor_x_concentration}}$$

$$\frac{d}{dt}\text{activated_coagulation_factor_ii_amount}$$
$$= \frac{\text{central_size} * \text{maximal_activation_rate_av_ax_complex_ii} * \text{av_ax_complex_concentration} * \text{coagulation_factor_ii_concentration}}{\text{half_maximal_effect_concentration_av_ax_complex_ii} + \text{av_ax_complex_concentration}}$$
$$+ \frac{\text{central_size} * \text{maximal_activation_rate_ax_ii} * \text{activated_coagulation_factor_x_concentration} * \text{coagulation_factor_ii_concentration}}{\text{half_maximal_effect_concentration_ax_ii} + \text{activated_coagulation_factor_x_concentration}}$$
$$- \text{central_size} * \text{degradation_rate_ii_activated} * \text{activated_coagulation_factor_ii_concentration} - \text{central_size}$$
$$* \text{complex_formation_rate_aii_atiii_heparin_complex} * \text{activated_coagulation_factor_ii_concentration} * \text{atiii_heparin_complex_concentration}$$
$$- \text{central_size} * \text{complex_formation_rate_aii_tmod} * \text{activated_coagulation_factor_ii_concentration} * \text{thrombomodulin_concentration}$$

$$\frac{d}{dt}\text{coagulation_factor_v_amount}$$
$$= \text{central_size} * \text{production_rate_v} - \text{central_size} * \text{degradation_rate_v} * \text{coagulation_factor_v_concentration}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_thrombin_v} * \text{activated_coagulation_factor_ii_concentration} * \text{coagulation_factor_v_concentration}}{\text{half_maximal_effect_concentration_thrombin_v} + \text{activated_coagulation_factor_ii_concentration}}$$

$$\frac{d}{dt}\text{activated_coagulation_factor_v_amount}$$
$$= \frac{\text{central_size} * \text{maximal_activation_rate_thrombin_v} * \text{activated_coagulation_factor_ii_concentration} * \text{coagulation_factor_v_concentration}}{\text{half_maximal_effect_concentration_thrombin_v} + \text{activated_coagulation_factor_ii_concentration}}$$
$$- \text{central_size} * \text{degradation_rate_v_activated} * \text{activated_coagulation_factor_v_concentration}$$
$$- \frac{\text{central_size} * \text{maximal_degradation_rate_apc_ps_complex_av} * \text{apc_ps_complex_concentration} * \text{activated_coagulation_factor_v_concentration}}{\text{half_maximal_effect_concentration_apc_ps_complex_av} + \text{apc_ps_complex_concentration}}$$
$$- \text{central_size} * \text{complex_formation_rate_av_ax} * \text{activated_coagulation_factor_v_concentration} * \text{activated_coagulation_factor_x_concentration}$$

$$\frac{d}{dt}\text{coagulation_factor_vii_amount}$$
$$= \text{central_size} * \text{production_rate_vii} * \text{vitamin_k_hydroquinone_concentration} - \text{central_size} * \text{degradation_rate_vii} * \text{coagulation_factor_vii_concentration}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_thrombin_vii} * \text{activated_coagulation_factor_ii_concentration} * \text{coagulation_factor_vii_concentration}}{\text{half_maximal_effect_concentration_thrombin_vii} + \text{activated_coagulation_factor_ii_concentration}}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_aix_vii} * \text{activated_coagulation_factor_ix_concentration} * \text{coagulation_factor_vii_concentration}}{\text{half_maximal_effect_concentration_aix_vii} + \text{activated_coagulation_factor_ix_concentration}}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_ax_vii} * \text{activated_coagulation_factor_x_concentration} * \text{coagulation_factor_vii_concentration}}{\text{half_maximal_effect_concentration_ax_vii} + \text{activated_coagulation_factor_x_concentration}}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_avii_tf_complex_vii} * \text{avii_tf_complex_concentration} * \text{coagulation_factor_vii_concentration}}{\text{half_maximal_effect_concentration_avii_tf_complex_vii} + \text{avii_tf_complex_concentration}}$$
$$- \text{central_size} * \text{complex_formation_rate_vii_tf} * \text{coagulation_factor_vii_concentration} * \text{tissue_factor_concentration}$$

$$\frac{d}{dt}\text{activated_coagulation_factor_vii_amount}$$
$$= \frac{\text{central_size} * \text{maximal_activation_rate_thrombin_vii} * \text{activated_coagulation_factor_ii_concentration} * \text{coagulation_factor_vii_concentration}}{\text{half_maximal_effect_concentration_thrombin_vii} + \text{activated_coagulation_factor_ii_concentration}}$$
$$+ \frac{\text{central_size} * \text{maximal_activation_rate_aix_vii} * \text{activated_coagulation_factor_ix_concentration} * \text{coagulation_factor_vii_concentration}}{\text{half_maximal_effect_concentration_aix_vii} + \text{activated_coagulation_factor_ix_concentration}}$$
$$+ \frac{\text{central_size} * \text{maximal_activation_rate_ax_vii} * \text{activated_coagulation_factor_x_concentration} * \text{coagulation_factor_vii_concentration}}{\text{half_maximal_effect_concentration_ax_vii} + \text{activated_coagulation_factor_x_concentration}}$$
$$+ \frac{\text{central_size} * \text{maximal_activation_rate_avii_tf_complex_vii} * \text{avii_tf_complex_concentration} * \text{coagulation_factor_vii_concentration}}{\text{half_maximal_effect_concentration_avii_tf_complex_vii} + \text{avii_tf_complex_concentration}}$$
$$- \text{central_size} * \text{degradation_rate_vii_activated} * \text{activated_coagulation_factor_vii_concentration} - \text{central_size}$$
$$* \text{complex_formation_rate_avii_tf} * \text{activated_coagulation_factor_vii_concentration} * \text{tissue_factor_concentration}$$

$$\frac{d}{dt}\text{coagulation_factor_viii_amount}$$
$$= \text{central_size} * \text{production_rate_viii} - \text{central_size} * \text{degradation_rate_viii} * \text{coagulation_factor_viii_concentration}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_thrombin_viii} * \text{activated_coagulation_factor_ii_concentration} * \text{coagulation_factor_viii_concentration}}{\text{half_maximal_effect_concentration_thrombin_viii} + \text{activated_coagulation_factor_ii_concentration}}$$

$$\frac{d}{dt}\text{activated_coagulation_factor_viii_amount}$$
$$= \frac{\text{central_size} * \text{maximal_activation_rate_thrombin_viii} * \text{activated_coagulation_factor_ii_concentration} * \text{coagulation_factor_viii_concentration}}{\text{half_maximal_effect_concentration_thrombin_viii} + \text{activated_coagulation_factor_ii_concentration}}$$
$$- \text{central_size} * \text{degradation_rate_viii_activated} * \text{activated_coagulation_factor_viii_concentration}$$
$$- \frac{\text{central_size} * \text{maximal_degradation_rate_apc_ps_complex_aviii} * \text{apc_ps_complex_concentration} * \text{activated_coagulation_factor_viii_concentration}}{\text{half_maximal_effect_concentration_apc_ps_complex_aviii} + \text{apc_ps_complex_concentration}}$$
$$- \text{central_size} * \text{complex_formation_rate_aix_aviii} * \text{activated_coagulation_factor_ix_concentration} * \text{activated_coagulation_factor_viii_concentration}$$

$$\frac{d}{dt}\text{coagulation_factor_ix_amount}$$
$$= \text{central_size} * \text{production_rate_ix} * \text{vitamin_k_hydroquinone_concentration} - \text{central_size} * \text{degradation_rate_ix} * \text{coagulation_factor_ix_concentration}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_axi_ix} * \text{activated_coagulation_factor_xi_concentration} * \text{coagulation_factor_ix_concentration}}{\text{half_maximal_effect_concentration_axi_ix} + \text{activated_coagulation_factor_xi_concentration}}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_avii_tf_complex_ix} * \text{avii_tf_complex_concentration} * \text{coagulation_factor_ix_concentration}}{\text{half_maximal_effect_concentration_avii_tf_complex_ix} + \text{avii_tf_complex_concentration}}$$

$$\frac{d}{dt}\text{activated_coagulation_factor_ix_amount}$$
$$= \frac{\text{central_size} * \text{maximal_activation_rate_axi_ix} * \text{activated_coagulation_factor_xi_concentration} * \text{coagulation_factor_ix_concentration}}{\text{half_maximal_effect_concentration_axi_ix} + \text{activated_coagulation_factor_xi_concentration}}$$
$$+ \frac{\text{central_size} * \text{maximal_activation_rate_avii_tf_complex_ix} * \text{avii_tf_complex_concentration} * \text{coagulation_factor_ix_concentration}}{\text{half_maximal_effect_concentration_avii_tf_complex_ix} + \text{avii_tf_complex_concentration}} - \text{central_size}$$
$$* \text{degradation_rate_ix_activated} * \text{activated_coagulation_factor_ix_concentration} - \text{central_size} * \text{complex_formation_rate_aix_aviii}$$
$$* \text{activated_coagulation_factor_ix_concentration} * \text{activated_coagulation_factor_viii_concentration} - \text{central_size}$$
$$* \text{complex_formation_rate_aix_atiii_heparin_complex} * \text{activated_coagulation_factor_ix_concentration} * \text{atiii_heparin_complex_concentration}$$

$$\frac{d}{dt}\text{coagulation_factor_x_amount}$$
$$= \text{central_size} * \text{production_rate_x} * \text{vitamin_k_hydroquinone_concentration} - \text{central_size} * \text{degradation_rate_x} * \text{coagulation_factor_x_concentration}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_avii_x} * \text{activated_coagulation_factor_vii_concentration} * \text{coagulation_factor_x_concentration}}{\text{half_maximal_effect_concentration_avii_x} + \text{activated_coagulation_factor_vii_concentration}}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_avii_tf_complex_x} * \text{avii_tf_complex_concentration} * \text{coagulation_factor_x_concentration}}{\text{half_maximal_effect_concentration_avii_tf_complex_x} + \text{avii_tf_complex_concentration}}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_aix_x} * \text{activated_coagulation_factor_ix_concentration} * \text{coagulation_factor_x_concentration}}{\text{half_maximal_effect_concentration_aix_x} + \text{activated_coagulation_factor_ix_concentration}}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_aix_aviii_complex_x} * \text{aix_aviii_complex_concentration} * \text{coagulation_factor_x_concentration}}{\text{half_maximal_effect_concentration_aix_aviii_complex_x} + \text{aix_aviii_complex_concentration}}$$

$$\frac{d}{dt}\text{activated_coagulation_factor_x_amount}$$
$$= -\text{central_size} * \text{complex_formation_rate_av_ax} * \text{activated_coagulation_factor_v_concentration} * \text{activated_coagulation_factor_x_concentration}$$
$$+ \frac{\text{central_size} * \text{maximal_activation_rate_avii_x} * \text{activated_coagulation_factor_vii_concentration} * \text{coagulation_factor_x_concentration}}{\text{half_maximal_effect_concentration_avii_x} + \text{activated_coagulation_factor_vii_concentration}}$$
$$+ \frac{\text{central_size} * \text{maximal_activation_rate_avii_tf_complex_x} * \text{avii_tf_complex_concentration} * \text{coagulation_factor_x_concentration}}{\text{half_maximal_effect_concentration_avii_tf_complex_x} + \text{avii_tf_complex_concentration}}$$
$$+ \frac{\text{central_size} * \text{maximal_activation_rate_aix_x} * \text{activated_coagulation_factor_ix_concentration} * \text{coagulation_factor_x_concentration}}{\text{half_maximal_effect_concentration_aix_x} + \text{activated_coagulation_factor_ix_concentration}}$$
$$+ \frac{\text{central_size} * \text{maximal_activation_rate_aix_aviii_complex_x} * \text{aix_aviii_complex_concentration} * \text{coagulation_factor_x_concentration}}{\text{half_maximal_effect_concentration_aix_aviii_complex_x} + \text{aix_aviii_complex_concentration}}$$
$$- \text{central_size} * \text{degradation_rate_x_activated} * \text{activated_coagulation_factor_x_concentration} - \text{central_size}$$
$$* \text{complex_formation_rate_ax_atiii_heparin_complex} * \text{activated_coagulation_factor_x_concentration} * \text{atiii_heparin_complex_concentration}$$
$$- \text{central_size} * \text{complex_formation_rate_ax_tfpi} * \text{activated_coagulation_factor_x_concentration} * \text{tissue_factor_pathway_inhibitor_concentration}$$

$$\frac{d}{dt}\text{coagulation_factor_xi_amount}$$
$$= \text{central_size} * \text{production_rate_xi} - \text{central_size} * \text{degradation_rate_xi} * \text{coagulation_factor_xi_concentration}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_thrombin_xi} * \text{activated_coagulation_factor_ii_concentration} * \text{coagulation_factor_xi_concentration}}{\text{half_maximal_effect_concentration_thrombin_xi} + \text{activated_coagulation_factor_ii_concentration}}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_axii_xi} * \text{activated_coagulation_factor_xii_concentration} * \text{coagulation_factor_xi_concentration}}{\text{half_maximal_effect_concentration_axii_xi} + \text{activated_coagulation_factor_xii_concentration}}$$

$$\frac{d}{dt}\text{activated_coagulation_factor_xi_amount} = \frac{\text{central_size} * \text{maximal_activation_rate_thrombin_xi} * \text{activated_coagulation_factor_ii_concentration} * \text{coagulation_factor_xi_concentration}}{\text{half_maximal_effect_concentration_thrombin_xi} + \text{activated_coagulation_factor_ii_concentration}} + \frac{\text{central_size} * \text{maximal_activation_rate_axii_xi} * \text{activated_coagulation_factor_xii_concentration} * \text{coagulation_factor_xi_concentration}}{\text{half_maximal_effect_concentration_axii_xi} + \text{activated_coagulation_factor_xii_concentration}} - \text{central_size} * \text{degradation_rate_xi_activated} * \text{activated_coagulation_factor_xi_concentration}$$

$$\frac{d}{dt}\text{coagulation_factor_xii_amount} = \text{central_size} * \text{production_rate_xii} - \text{central_size} * \text{degradation_rate_xii} * \text{coagulation_factor_xii_concentration} - \frac{\text{central_size} * \text{maximal_activation_rate_ca_xii} * \text{contact_system_activator_concentration} * \text{coagulation_factor_xii_concentration}}{\text{half_maximal_effect_concentration_ca_xii} + \text{contact_system_activator_concentration}} - \frac{\text{central_size} * \text{maximal_activation_rate_kallikrein_xii} * \text{kallikrein_concentration} * \text{coagulation_factor_xii_concentration}}{\text{half_maximal_effect_concentration_kallikrein_xii} + \text{kallikrein_concentration}}$$

$$\frac{d}{dt}\text{activated_coagulation_factor_xii_amount} = \frac{\text{central_size} * \text{maximal_activation_rate_ca_xii} * \text{contact_system_activator_concentration} * \text{coagulation_factor_xii_concentration}}{\text{half_maximal_effect_concentration_ca_xii} + \text{contact_system_activator_concentration}} + \frac{\text{central_size} * \text{maximal_activation_rate_kallikrein_xii} * \text{kallikrein_concentration} * \text{coagulation_factor_xii_concentration}}{\text{half_maximal_effect_concentration_kallikrein_xii} + \text{kallikrein_concentration}} - \text{central_size} * \text{degradation_rate_xii_activated} * \text{activated_coagulation_factor_xii_concentration}$$

$$\frac{d}{dt}\text{coagulation_factor_xiii_amount} = \text{central_size} * \text{production_rate_xiii} - \text{central_size} * \text{degradation_rate_xiii} * \text{coagulation_factor_xiii_concentration} - \frac{\text{central_size} * \text{maximal_activation_rate_thrombin_xiii} * \text{activated_coagulation_factor_ii_concentration} * \text{coagulation_factor_xiii_concentration}}{\text{half_maximal_effect_concentration_thrombin_xiii} + \text{activated_coagulation_factor_ii_concentration}}$$

$$\frac{d}{dt}\text{activated_coagulation_factor_xiii_amount} = \frac{\text{central_size} * \text{maximal_activation_rate_thrombin_xiii} * \text{activated_coagulation_factor_ii_concentration} * \text{coagulation_factor_xiii_concentration}}{\text{half_maximal_effect_concentration_thrombin_xiii} + \text{activated_coagulation_factor_ii_concentration}} - \text{central_size} * \text{degradation_rate_xiii_activated} * \text{activated_coagulation_factor_xiii_concentration}$$

$$\frac{d}{dt}\text{contact_system_activator_amount} = -\text{central_size} * \text{degradation_rate_ca} * \text{contact_system_activator_concentration}$$

$$\begin{aligned}
&\frac{d}{dt}\text{fibrin_amount} \\
&= \frac{\text{central_size} * \text{maximal_conversion_rate_thrombin_fibrinogen} * \text{activated_coagulation_factor_ii_concentration} * \text{fibrinogen_concentration}}{\text{half_maximal_effect_concentration_thrombin_fibrinogen} + \text{activated_coagulation_factor_ii_concentration}} \\
&\quad - \text{central_size} * \text{degradation_rate_fibrin} * \text{fibrin_concentration} \\
&\quad - \frac{\text{central_size} * \text{maximal_conversion_rate_axiii_fibrin} * \text{activated_coagulation_factor_xiii_concentration} * \text{fibrin_concentration}}{\text{half_maximal_effect_concentration_axiii_fibrin} + \text{activated_coagulation_factor_xiii_concentration}} \\
&\quad - \frac{\text{central_size} * \text{maximal_degradation_rate_plasmin_fibrin} * \text{plasmin_concentration} * \text{fibrin_concentration}}{\text{half_maximal_effect_concentration_plasmin_fibrin} + \text{plasmin_concentration}}
\end{aligned}$$

$$\begin{aligned}
&\frac{d}{dt}\text{cross_linked_fibrin_amount} \\
&= \frac{\text{central_size} * \text{maximal_conversion_rate_axiii_fibrin} * \text{activated_coagulation_factor_xiii_concentration} * \text{fibrin_concentration}}{\text{half_maximal_effect_concentration_axiii_fibrin} + \text{activated_coagulation_factor_xiii_concentration}} \\
&\quad - \text{central_size} * \text{degradation_rate_xf} * \text{cross_linked_fibrin_concentration} \\
&\quad - \frac{\text{central_size} * \text{maximal_degradation_rate_plasmin_xf} * \text{plasmin_concentration} * \text{cross_linked_fibrin_concentration}}{\text{half_maximal_effect_concentration_plasmin_xf} + \text{plasmin_concentration}} \\
&\quad - \frac{\text{central_size} * \text{maximal_degradation_rate_apc_ps_complex_xf} * \text{apc_ps_complex_concentration} * \text{cross_linked_fibrin_concentration}}{\text{half_maximal_effect_concentration_apc_ps_complex_xf} + \text{apc_ps_complex_concentration}}
\end{aligned}$$

$$\begin{aligned}
&\frac{d}{dt}\text{fibrin_degradation_product_amount} \\
&= \text{central_size} * \text{degradation_rate_fibrinogen} * \text{fibrinogen_concentration} \\
&\quad + \frac{\text{central_size} * \text{maximal_degradation_rate_plasmin_fibrinogen} * \text{plasmin_concentration} * \text{fibrinogen_concentration}}{\text{half_maximal_effect_concentration_plasmin_fibrinogen} + \text{plasmin_concentration}} \\
&\quad + \text{central_size} * \text{degradation_rate_fibrin} * \text{fibrin_concentration} \\
&\quad + \frac{\text{central_size} * \text{maximal_degradation_rate_plasmin_fibrin} * \text{plasmin_concentration} * \text{fibrin_concentration}}{\text{half_maximal_effect_concentration_plasmin_fibrin} + \text{plasmin_concentration}} \\
&\quad - \text{central_size} * \text{degradation_rate_fdp} * \text{fibrin_degradation_product_concentration}
\end{aligned}$$

$$\frac{d}{dt}\text{d_dimer_amount}$$
$$= \text{central_size} * \text{degradation_rate_xf} * \text{cross_linked_fibrin_concentration}$$
$$+ \frac{\text{central_size} * \text{maximal_degradation_rate_plasmin_xf} * \text{plasmin_concentration} * \text{cross_linked_fibrin_concentration}}{\text{half_maximal_effect_concentration_plasmin_xf} + \text{plasmin_concentration}}$$
$$+ \frac{\text{central_size} * \text{maximal_degradation_rate_apc_ps_complex_xf} * \text{apc_ps_complex_concentration} * \text{cross_linked_fibrin_concentration}}{\text{half_maximal_effect_concentration_apc_ps_complex_xf} + \text{apc_ps_complex_concentration}}$$
$$- \text{central_size} * \text{degradation_rate_d_dimer} * \text{d_dimer_concentration}$$

$$\frac{d}{dt}\text{fibrinogen_amount}$$
$$= \text{central_size} * \text{production_rate_fibrinogen} - \text{central_size} * \text{degradation_rate_fibrinogen} * \text{fibrinogen_concentration}$$
$$- \frac{\text{central_size} * \text{maximal_conversion_rate_thrombin_fibrinogen} * \text{activated_coagulation_factor_ii_concentration} * \text{fibrinogen_concentration}}{\text{half_maximal_effect_concentration_thrombin_fibrinogen} + \text{activated_coagulation_factor_ii_concentration}}$$
$$- \frac{\text{central_size} * \text{maximal_degradation_rate_plasmin_fibrinogen} * \text{plasmin_concentration} * \text{fibrinogen_concentration}}{\text{half_maximal_effect_concentration_plasmin_fibrinogen} + \text{plasmin_concentration}}$$

$$\frac{d}{dt}\text{kallikrein_amount}$$
$$= \frac{\text{central_size} * \text{maximal_conversion_rate_axii_prekallikrein} * \text{activated_coagulation_factor_xii_concentration} * \text{prekallikrein_concentration}}{\text{half_maximal_effect_concentration_axii_prekallikrein} + \text{activated_coagulation_factor_xii_concentration}}$$
$$- \text{central_size} * \text{degradation_rate_kallikrein} * \text{kallikrein_concentration}$$

$$\frac{d}{dt}\text{prekallikrein_amount}$$
$$= \text{central_size} * \text{production_rate_prekallikrein} - \text{central_size} * \text{degradation_rate_prekallikrein} * \text{prekallikrein_concentration}$$
$$- \frac{\text{central_size} * \text{maximal_conversion_rate_axii_prekallikrein} * \text{activated_coagulation_factor_xii_concentration} * \text{prekallikrein_concentration}}{\text{half_maximal_effect_concentration_axii_prekallikrein} + \text{activated_coagulation_factor_xii_concentration}}$$

$$\frac{d}{dt}\text{plasmin_amount} = \frac{\text{central_size} * \text{maximal_conversion_rate_thrombin_plasminogen} * \text{activated_coagulation_factor_ii_concentration} * \text{plasminogen_concentration}}{\text{half_maximal_effect_concentration_thrombin_plasminogen} + \text{activated_coagulation_factor_ii_concentration}} + \frac{\text{central_size} * \text{maximal_conversion_rate_fibrin_plasminogen} * \text{fibrin_concentration} * \text{plasminogen_concentration}}{\text{half_maximal_effect_concentration_fibrin_plasminogen} + \text{fibrin_concentration}} + \frac{\text{central_size} * \text{maximal_conversion_rate_apc_ps_complex_plasminogen} * \text{apc_ps_complex_concentration} * \text{plasminogen_concentration}}{\text{half_maximal_effect_concentration_apc_ps_complex_plasminogen} + \text{apc_ps_complex_concentration}} - \text{central_size} * \text{degradation_rate_plasmin} * \text{plasmin_concentration}$$

$$\frac{d}{dt}\text{plasminogen_amount} = \text{central_size} * \text{production_rate_plasminogen} - \text{central_size} * \text{degradation_rate_plasminogen} * \text{plasminogen_concentration} - \frac{\text{central_size} * \text{maximal_conversion_rate_thrombin_plasminogen} * \text{activated_coagulation_factor_ii_concentration} * \text{plasminogen_concentration}}{\text{half_maximal_effect_concentration_thrombin_plasminogen} + \text{activated_coagulation_factor_ii_concentration}} - \frac{\text{central_size} * \text{maximal_conversion_rate_fibrin_plasminogen} * \text{fibrin_concentration} * \text{plasminogen_concentration}}{\text{half_maximal_effect_concentration_fibrin_plasminogen} + \text{fibrin_concentration}} - \frac{\text{central_size} * \text{maximal_conversion_rate_apc_ps_complex_plasminogen} * \text{apc_ps_complex_concentration} * \text{plasminogen_concentration}}{\text{half_maximal_effect_concentration_apc_ps_complex_plasminogen} + \text{apc_ps_complex_concentration}}$$

$$\frac{d}{dt}\text{protein_c_amount} = \text{central_size} * \text{production_rate_pc} * \text{vitamin_k_hydroquinone_concentration} - \text{central_size} * \text{degradation_rate_pc} * \text{protein_c_concentration} - \frac{\text{central_size} * \text{maximal_activation_rate_aii_tmod_complex_pc} * \text{aii_tmod_complex_concentration} * \text{protein_c_concentration}}{\text{half_maximal_effect_concentration_aii_tmod_complex_pc} + \text{aii_tmod_complex_concentration}}$$

$$\frac{d}{dt}\text{activated_protein_c_amount} = \frac{\text{central_size} * \text{maximal_activation_rate_aii_tmod_complex_pc} * \text{aii_tmod_complex_concentration} * \text{protein_c_concentration}}{\text{half_maximal_effect_concentration_aii_tmod_complex_pc} + \text{aii_tmod_complex_concentration}} - \text{central_size} * \text{degradation_rate_pc_activated} * \text{activated_protein_c_concentration} - \text{central_size} * \text{complex_formation_rate_apc_ps} * \text{activated_protein_c_concentration} * \text{protein_s_concentration}$$

$$\frac{d}{dt}\text{protein_s_amount} = -\text{central_size} * \text{complex_formation_rate_apc_ps} * \text{activated_protein_c_concentration} * \text{protein_s_concentration} + \text{central_size} * \text{production_rate_ps} * \text{vitamin_k_hydroquinone_concentration} - \text{central_size} * \text{degradation_rate_ps} * \text{protein_s_concentration}$$

$$\frac{d}{dt}\text{thrombomodulin_amount} = -\text{central_size} * \text{complex_formation_rate_aii_tmod} * \text{activated_coagulation_factor_ii_concentration} * \text{thrombomodulin_concentration} + \text{central_size} * \text{production_rate_tmod} - \text{central_size} * \text{degradation_rate_tmod} * \text{thrombomodulin_concentration}$$

$$\frac{d}{dt}\text{tissue_factor_amount} = -\text{central_size} * \text{complex_formation_rate_vii_tf} * \text{coagulation_factor_vii_concentration} * \text{tissue_factor_concentration} - \text{central_size} * \text{complex_formation_rate_avii_tf} * \text{activated_coagulation_factor_vii_concentration} * \text{tissue_factor_concentration} - \text{central_size} * \text{degradation_rate_tissue_factor} * \text{tissue_factor_concentration}$$

$$\frac{d}{dt}\text{tissue_factor_pathway_inhibitor_amount} = -\text{central_size} * \text{complex_formation_rate_ax_tfpi} * \text{activated_coagulation_factor_x_concentration} * \text{tissue_factor_pathway_inhibitor_concentration} + \text{central_size} * \text{production_rate_tfpi} - \text{central_size} * \text{degradation_rate_tfpi} * \text{tissue_factor_pathway_inhibitor_concentration}$$

$$\begin{aligned}\frac{d}{dt}\text{vitamin_k_amount} &= \text{central_size} * \text{input_rate_vk} - \text{central_size} * \text{elimination_rate_vk} * \text{vitamin_k_concentration} - \text{central_size} * \text{conversion_rate_vk_vkh2} \\ &\quad * \text{vitamin_k_concentration} * \left(1.0 - \frac{\text{maximal_inhibitory_effect_warfarin} * \text{warfarin_concentration}^{\text{gamma}}}{\text{half_maximal_effect_concentration_warfarin}^{\text{gamma}} + \text{warfarin_concentration}^{\text{gamma}}}\right) \\ &\quad + \text{central_size} * \text{conversion_rate_vko_vk} * \text{vitamin_k_epoxide_concentration} \\ &\quad * \left(1.0 - \frac{\text{maximal_inhibitory_effect_warfarin} * \text{warfarin_concentration}^{\text{gamma}}}{\text{half_maximal_effect_concentration_warfarin}^{\text{gamma}} + \text{warfarin_concentration}^{\text{gamma}}}\right) \\ &\quad - \text{central_size} * \text{transition_rate_vk_central_to_peripheral} * \text{vitamin_k_concentration} \\ &\quad + \text{peripheral_vitamin_k_size} * \text{transition_rate_vk_peripheral_to_central} * \text{vitamin_k_peripheral_concentration}\end{aligned}$$

$$\frac{d}{dt}\text{vitamin_k_peripheral_amount} = \text{central_size} * \text{transition_rate_vk_central_to_peripheral} * \text{vitamin_k_concentration} - \text{peripheral_vitamin_k_size} * \text{transition_rate_vk_peripheral_to_central} * \text{vitamin_k_peripheral_concentration}$$

$$\frac{d}{dt}\text{vitamin_k_epoxide_amount} = \text{central_size} * \text{conversion_rate_vkh2_vko} * \text{vitamin_k_hydroquinone_concentration} - \text{central_size} * \text{conversion_rate_vko_vk} * \text{vitamin_k_epoxide_concentration} * \left(1.0 - \frac{\text{maximal_inhibitory_effect_warfarin} * \text{warfarin_concentration}^{\text{gamma}}}{\text{half_maximal_effect_concentration_warfarin}^{\text{gamma}} + \text{warfarin_concentration}^{\text{gamma}}}\right)$$

$$\frac{d}{dt}\text{vitamin_k_hydroquinone_amount} = \text{central_size} * \text{conversion_rate_vk_vkh2} * \text{vitamin_k_concentration} * \left(1.0 - \frac{\text{maximal_inhibitory_effect_warfarin} * \text{warfarin_concentration}^{\text{gamma}}}{\text{half_maximal_effect_concentration_warfarin}^{\text{gamma}} + \text{warfarin_concentration}^{\text{gamma}}}\right) - \text{central_size} * \text{conversion_rate_vkh2_vko} * \text{vitamin_k_hydroquinone_concentration}$$

$$\frac{d}{dt}\text{aii_atiii_heparin_complex_amount} = \text{central_size} * \text{complex_formation_rate_aii_atiii_heparin_complex} * \text{activated_coagulation_factor_ii_concentration} * \text{atiii_heparin_complex_concentration}$$

$$\frac{d}{dt}\text{aii_at_complex_amount} = \text{central_size} * \text{degradation_rate_ii_activated} * \text{activated_coagulation_factor_ii_concentration} - \text{central_size} * \text{degradation_rate_tat} * \text{aii_at_complex_concentration}$$

$$\frac{d}{dt}\text{aii_tmod_complex_amount} = \text{central_size} * \text{complex_formation_rate_aii_tmod} * \text{activated_coagulation_factor_ii_concentration} * \text{thrombomodulin_concentration} - \text{central_size} * \text{degradation_rate_aii_tmod_complex} * \text{aii_tmod_complex_concentration}$$

$$\frac{d}{dt}\text{av_ax_complex_amount} = \text{central_size} * \text{complex_formation_rate_av_ax} * \text{activated_coagulation_factor_v_concentration} * \text{activated_coagulation_factor_x_concentration} - \text{central_size} * \text{degradation_rate_av_ax_complex} * \text{av_ax_complex_concentration} - \frac{\text{central_size} * \text{maximal_degradation_rate_apc_ps_complex_av_ax_complex} * \text{apc_ps_complex_concentration} * \text{av_ax_complex_concentration}}{\text{half_maximal_effect_concentration_apc_ps_complex_av_ax_complex} + \text{apc_ps_complex_concentration}}$$

$$\frac{d}{dt}\text{vii_tf_complex_amount}$$
$$= \text{central_size} * \text{complex_formation_rate_vii_tf} * \text{coagulation_factor_vii_concentration}$$
$$* \text{tissue_factor_concentration} - \text{central_size} * \text{degradation_rate_vii_tf_complex} * \text{vii_tf_complex_concentration}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_ax_vii_tf_complex} * \text{activated_coagulation_factor_x_concentration} * \text{vii_tf_complex_concentration}}{\text{half_maximal_effect_concentration_ax_vii_tf_complex} + \text{activated_coagulation_factor_x_concentration}}$$
$$- \frac{\text{central_size} * \text{maximal_activation_rate_tf_vii_tf_complex} * \text{tissue_factor_concentration} * \text{vii_tf_complex_concentration}}{\text{half_maximal_effect_concentration_tf_vii_tf_complex} + \text{tissue_factor_concentration}}$$

$$\frac{d}{dt}\text{avii_tf_complex_amount}$$
$$= \frac{\text{central_size} * \text{maximal_activation_rate_ax_vii_tf_complex} * \text{activated_coagulation_factor_x_concentration} * \text{vii_tf_complex_concentration}}{\text{half_maximal_effect_concentration_ax_vii_tf_complex} + \text{activated_coagulation_factor_x_concentration}}$$
$$+ \frac{\text{central_size} * \text{maximal_activation_rate_tf_vii_tf_complex} * \text{tissue_factor_concentration} * \text{vii_tf_complex_concentration}}{\text{half_maximal_effect_concentration_tf_vii_tf_complex} + \text{tissue_factor_concentration}}$$
$$+ \text{central_size} * \text{complex_formation_rate_avii_tf} * \text{activated_coagulation_factor_vii_concentration}$$
$$* \text{tissue_factor_concentration} - \text{central_size} * \text{degradation_rate_avii_tf_complex} * \text{avii_tf_complex_concentration}$$
$$- \text{central_size} * \text{complex_formation_rate_avii_tf_ax_tfpi} * \text{avii_tf_complex_concentration} * \text{ax_tfpi_complex_concentration}$$

$$\frac{d}{dt}\text{avii_tf_ax_tfpi_complex_amount} = \text{central_size} * \text{complex_formation_rate_avii_tf_ax_tfpi}$$
$$* \text{avii_tf_complex_concentration} * \text{ax_tfpi_complex_concentration} - \text{central_size}$$
$$* \text{degradation_rate_avii_tf_ax_tfpi_complex} * \text{avii_tf_ax_tfpi_complex_concentration}$$

$$\frac{d}{dt}\text{aix_aviii_complex_amount} = \text{central_size} * \text{complex_formation_rate_aix_aviii} * \text{activated_coagulation_factor_ix_concentration}$$
$$* \text{activated_coagulation_factor_viii_concentration} - \text{central_size}$$
$$* \text{degradation_rate_aix_aviii_complex} * \text{aix_aviii_complex_concentration}$$

$$\frac{d}{dt}\text{aix_atiii_heparin_complex_amount} = \text{central_size} * \text{complex_formation_rate_aix_atiii_heparin_complex}$$
$$* \text{activated_coagulation_factor_ix_concentration} * \text{atiii_heparin_complex_concentration}$$

$$\frac{d}{dt}\text{ax_atiii_heparin_complex_complex_amount} = \text{central_size} * \text{complex_formation_rate_ax_atiii_heparin_complex} * \text{activated_coagulation_factor_x_concentration} * \text{atiii_heparin_complex_concentration}$$

$$\frac{d}{dt}\text{ax_tfpi_complex_amount} = -\text{central_size} * \text{complex_formation_rate_avii_tf_ax_tfpi} * \text{avii_tf_complex_concentration} * \text{ax_tfpi_complex_concentration} + \text{central_size} * \text{complex_formation_rate_ax_tfpi} * \text{activated_coagulation_factor_x_concentration} * \text{tissue_factor_pathway_inhibitor_concentration} - \text{central_size} * \text{degradation_rate_ax_tfpi_complex} * \text{ax_tfpi_complex_concentration}$$

$$\frac{d}{dt}\text{apc_ps_complex_amount} = \text{central_size} * \text{complex_formation_rate_apc_ps} * \text{activated_protein_c_concentration} * \text{protein_s_concentration} - \text{central_size} * \text{degradation_rate_apc_ps_complex} * \text{apc_ps_complex_concentration}$$

$$\frac{d}{dt}\text{atiii_heparin_complex_amount} = -\text{central_size} * \text{elimination_rate_atiii_heparin_complex} * \text{atiii_heparin_complex_concentration} - \text{central_size} * \text{complex_formation_rate_aii_atiii_heparin_complex} * \text{activated_coagulation_factor_ii_concentration} * \text{atiii_heparin_complex_concentration} - \text{central_size} * \text{complex_formation_rate_aix_atiii_heparin_complex} * \text{activated_coagulation_factor_ix_concentration} * \text{atiii_heparin_complex_concentration} - \text{central_size} * \text{complex_formation_rate_ax_atiii_heparin_complex} * \text{activated_coagulation_factor_x_concentration} * \text{atiii_heparin_complex_concentration}$$

$$\frac{d}{dt}\text{drug_amount} = -\text{absorption_rate} * \text{drug_amount} + \text{dose_rate}$$

$$\frac{d}{dt}\text{warfarin_amount} = -\text{central_warfarin_size} * \text{elimination_rate_warfarin} * \text{warfarin_concentration} + \text{absorption_rate} * \text{drug_amount}$$

$$\text{coagulation_factor_ii_concentration} = \frac{\text{coagulation_factor_ii_amount}}{\text{central_size}}$$

$$\text{activated_coagulation_factor_ii_concentration} = \frac{\text{activated_coagulation_factor_ii_amount}}{\text{central_size}}$$

$$\text{coagulation_factor_v_concentration} = \frac{\text{coagulation_factor_v_amount}}{\text{central_size}}$$

$$\text{activated_coagulation_factor_v_concentration} = \frac{\text{activated_coagulation_factor_v_amount}}{\text{central_size}}$$

$$\text{coagulation_factor_vii_concentration} = \frac{\text{coagulation_factor_vii_amount}}{\text{central_size}}$$

$$\text{activated_coagulation_factor_vii_concentration} = \frac{\text{activated_coagulation_factor_vii_amount}}{\text{central_size}}$$

$$\text{coagulation_factor_viii_concentration} = \frac{\text{coagulation_factor_viii_amount}}{\text{central_size}}$$

$$\text{activated_coagulation_factor_viii_concentration} = \frac{\text{activated_coagulation_factor_viii_amount}}{\text{central_size}}$$

$$\text{coagulation_factor_ix_concentration} = \frac{\text{coagulation_factor_ix_amount}}{\text{central_size}}$$

$$\text{activated_coagulation_factor_ix_concentration} = \frac{\text{activated_coagulation_factor_ix_amount}}{\text{central_size}}$$

$$\text{coagulation_factor_x_concentration} = \frac{\text{coagulation_factor_x_amount}}{\text{central_size}}$$

$$\text{activated_coagulation_factor_x_concentration} = \frac{\text{activated_coagulation_factor_x_amount}}{\text{central_size}}$$

$$\text{coagulation_factor_xi_concentration} = \frac{\text{coagulation_factor_xi_amount}}{\text{central_size}}$$

$$\text{activated_coagulation_factor_xi_concentration} = \frac{\text{activated_coagulation_factor_xi_amount}}{\text{central_size}}$$

$$\text{coagulation_factor_xii_concentration} = \frac{\text{coagulation_factor_xii_amount}}{\text{central_size}}$$

$$\text{activated_coagulation_factor_xii_concentration} = \frac{\text{activated_coagulation_factor_xii_amount}}{\text{central_size}}$$

$$\text{coagulation_factor_xiii_concentration} = \frac{\text{coagulation_factor_xiii_amount}}{\text{central_size}}$$

$$\text{activated_coagulation_factor_xiii_concentration} = \frac{\text{activated_coagulation_factor_xiii_amount}}{\text{central_size}}$$

$$\text{contact_system_activator_concentration} = \frac{\text{contact_system_activator_amount}}{\text{central_size}}$$

$$\text{fibrin_concentration} = \frac{\text{fibrin_amount}}{\text{central_size}}$$

$$\text{cross_linked_fibrin_concentration} = \frac{\text{cross_linked_fibrin_amount}}{\text{central_size}}$$

$$\text{fibrin_degradation_product_concentration} = \frac{\text{fibrin_degradation_product_amount}}{\text{central_size}}$$

$$\text{d_dimer_concentration} = \frac{\text{d_dimer_amount}}{\text{central_size}}$$

$$\text{fibrinogen_concentration} = \frac{\text{fibrinogen_amount}}{\text{central_size}}$$

$$\text{kallikrein_concentration} = \frac{\text{kallikrein_amount}}{\text{central_size}}$$

$$\text{prekallikrein_concentration} = \frac{\text{prekallikrein_amount}}{\text{central_size}}$$

$$\text{plasmin_concentration} = \frac{\text{plasmin_amount}}{\text{central_size}}$$

$$\text{plasminogen_concentration} = \frac{\text{plasminogen_amount}}{\text{central_size}}$$

$$\text{protein_c_concentration} = \frac{\text{protein_c_amount}}{\text{central_size}}$$

$$\text{activated_protein_c_concentration} = \frac{\text{activated_protein_c_amount}}{\text{central_size}}$$

$$\text{protein_s_concentration} = \frac{\text{protein_s_amount}}{\text{central_size}}$$

$$\text{thrombomodulin_concentration} = \frac{\text{thrombomodulin_amount}}{\text{central_size}}$$

$$\text{tissue_factor_concentration} = \frac{\text{tissue_factor_amount}}{\text{central_size}}$$

$$\text{tissue_factor_pathway_inhibitor_concentration} = \frac{\text{tissue_factor_pathway_inhibitor_amount}}{\text{central_size}}$$

$$\text{vitamin_k_concentration} = \frac{\text{vitamin_k_amount}}{\text{central_size}}$$

$$\text{vitamin_k_peripheral_concentration} = \frac{\text{vitamin_k_peripheral_amount}}{\text{peripheral_vitamin_k_size}}$$

$$\text{vitamin_k_epoxide_concentration} = \frac{\text{vitamin_k_epoxide_amount}}{\text{central_size}}$$

$$\text{vitamin_k_hydroquinone_concentration} = \frac{\text{vitamin_k_hydroquinone_amount}}{\text{central_size}}$$

$$\text{aii_atiii_heparin_complex_concentration} = \frac{\text{aii_atiii_heparin_complex_amount}}{\text{central_size}}$$

$$\text{aii_at_complex_concentration} = \frac{\text{aii_at_complex_amount}}{\text{central_size}}$$

$$\text{aii_tmod_complex_concentration} = \frac{\text{aii_tmod_complex_amount}}{\text{central_size}}$$

$$\text{av_ax_complex_concentration} = \frac{\text{av_ax_complex_amount}}{\text{central_size}}$$

$$\text{vii_tf_complex_concentration} = \frac{\text{vii_tf_complex_amount}}{\text{central_size}}$$

$$\text{avii_tf_complex_concentration} = \frac{\text{avii_tf_complex_amount}}{\text{central_size}}$$

$$\text{avii_tf_ax_tfpi_complex_concentration} = \frac{\text{avii_tf_ax_tfpi_complex_amount}}{\text{central_size}}$$

$$\text{aix_aviii_complex_concentration} = \frac{\text{aix_aviii_complex_amount}}{\text{central_size}}$$

$$\text{aix_atiii_heparin_complex_concentration} = \frac{\text{aix_atiii_heparin_complex_amount}}{\text{central_size}}$$

$$\text{ax_atiii_heparin_complex_complex_concentration} = \frac{\text{ax_atiii_heparin_complex_complex_amount}}{\text{central_size}}$$

$$\text{ax_tfpi_complex_concentration} = \frac{\text{ax_tfpi_complex_amount}}{\text{central_size}}$$

$$\text{apc_ps_complex_concentration} = \frac{\text{apc_ps_complex_amount}}{\text{central_size}}$$

$$\text{atiii_heparin_complex_concentration} = \frac{\text{atiii_heparin_complex_amount}}{\text{central_size}}$$

$$\text{warfarin_concentration} = \frac{\text{warfarin_amount}}{\text{central_warfarin_size}}$$

Table B.2: States of the mechanistic model

atiii_heparin_complex_amount

coagulation_factor_ii_amount

activated_coagulation_factor_ii_amount

aii_at_complex_amount

aii_atiii_heparin_complex_amount

thrombomodulin_amount

aii_tmod_complex_amount

coagulation_factor_v_amount

activated_coagulation_factor_v_amount

activated_coagulation_factor_x_amount

av_ax_complex_amount

coagulation_factor_vii_amount

activated_coagulation_factor_vii_amount

tissue_factor_amount

vii_tf_complex_amount

avii_tf_complex_amount

ax_tfpi_complex_amount

avii_tf_ax_tfpi_complex_amount

coagulation_factor_viii_amount

activated_coagulation_factor_viii_amount

coagulation_factor_ix_amount

activated_coagulation_factor_ix_amount

aix_aviii_complex_amount

aix_atiii_heparin_complex_amount

coagulation_factor_x_amount

ax_atiii_heparin_complex_complex_amount

tissue_factor_pathway_inhibitor_amount

coagulation_factor_xi_amount

activated_coagulation_factor_xi_amount

coagulation_factor_xii_amount

activated_coagulation_factor_xii_amount

coagulation_factor_xiii_amount

activated_coagulation_factor_xiii_amount

contact_system_activator_amount

fibrinogen_amount

fibrin_degradation_product_amount

fibrin_amount

cross_linked_fibrin_amount

d_dimer_amount

prekallikrein_amount

kallikrein_amount

plasminogen_amount

plasmin_amount

protein_c_amount

activated_protein_c_amount

protein_s_amount

apc_ps_complex_amount

vitamin_k_amount

vitamin_k_hydroquinone_amount

vitamin_k_epoxide_amount

vitamin_k_peripheral_amount

warfarin_amount

drug_amount

Table B.3: Parameters of the mechanistic model (Parameters with the same name as a state refer to the initial condition of the state)

degradation_rate_ii
degradation_rate_ii_activated
half_maximal_effect_concentration_av_ax_complex_ii
half_maximal_effect_concentration_ax_ii
maximal_activation_rate_av_ax_complex_ii
maximal_activation_rate_ax_ii
production_rate_ii (†)
degradation_rate_v
degradation_rate_v_activated
half_maximal_effect_concentration_thrombin_v
half_maximal_effect_concentration_apc_ps_complex_av
maximal_activation_rate_thrombin_v
maximal_degradation_rate_apc_ps_complex_av
production_rate_v (††)
degradation_rate_vii
degradation_rate_vii_activated
half_maximal_effect_concentration_thrombin_vii
half_maximal_effect_concentration_aix_vii
half_maximal_effect_concentration_ax_vii

half_maximal_effect_concentration_avii_tf_complex_vii

maximal_activation_rate_thrombin_vii

maximal_activation_rate_aix_vii

maximal_activation_rate_ax_vii

maximal_activation_rate_avii_tf_complex_vii

production_rate_vii (†)

degradation_rate_viii

degradation_rate_viii_activated

half_maximal_effect_concentration_thrombin_viii

half_maximal_effect_concentration_apc_ps_complex_aviii

maximal_activation_rate_thrombin_viii

maximal_degradation_rate_apc_ps_complex_aviii

production_rate_viii (†)

degradation_rate_ix

degradation_rate_ix_activated

half_maximal_effect_concentration_avii_tf_complex_ix

half_maximal_effect_concentration_axi_ix

maximal_activation_rate_avii_tf_complex_ix

maximal_activation_rate_axi_ix

production_rate_ix (†)

degradation_rate_x

degradation_rate_x_activated

half_maximal_effect_concentration_avii_x

half_maximal_effect_concentration_avii_tf_complex_x

half_maximal_effect_concentration_aix_x

half_maximal_effect_concentration_aix_aviii_complex_x

maximal_activation_rate_avii_x

maximal_activation_rate_avii_tf_complex_x

maximal_activation_rate_aix_x

maximal_activation_rate_aix_aviii_complex_x

production_rate_x (†)

degradation_rate_xi

degradation_rate_xi_activated

half_maximal_effect_concentration_axii_xi

half_maximal_effect_concentration_thrombin_xi

maximal_activation_rate_axii_xi

maximal_activation_rate_thrombin_xi

production_rate_xi (††)

degradation_rate_xii

degradation_rate_xii_activated

half_maximal_effect_concentration_ca_xii

half_maximal_effect_concentration_kallikrein_xii

maximal_activation_rate_ca_xii

maximal_activation_rate_kallikrein_xii

production_rate_xii (††)

degradation_rate_xiii

degradation_rate_xiii_activated

half_maximal_effect_concentration_thrombin_xiii

maximal_activation_rate_thrombin_xiii

production_rate_xiii (††)

degradation_rate_ca

degradation_rate_fibrinogen

degradation_rate_fibrin

degradation_rate_xf

half_maximal_effect_concentration_axiii_fibrin

half_maximal_effect_concentration_thrombin_fibrinogen

half_maximal_effect_concentration_plasmin_fibrinogen

half_maximal_effect_concentration_plasmin_fibrin

half_maximal_effect_concentration_plasmin_xf

half_maximal_effect_concentration_apc_ps_complex_xf

maximal_conversion_rate_axiii_fibrin

maximal_conversion_rate_thrombin_fibrinogen

maximal_degradation_rate_plasmin_fibrinogen

maximal_degradation_rate_plasmin_fibrin

maximal_degradation_rate_plasmin_xf

maximal_degradation_rate_apc_ps_complex_xf

production_rate_fibrinogen (†)

degradation_rate_fdp

degradation_rate_d_dimer

degradation_rate_kallikrein

degradation_rate_prekallikrein

half_maximal_effect_concentration_axii_prekallikrein

maximal_conversion_rate_axii_prekallikrein

production_rate_prekallikrein (†)

degradation_rate_plasminogen

degradation_rate_plasmin

half_maximal_effect_concentration_apc_ps_complex_plasminogen

half_maximal_effect_concentration_fibrin_plasminogen

half_maximal_effect_concentration_thrombin_plasminogen

maximal_conversion_rate_apc_ps_complex_plasminogen

maximal_conversion_rate_fibrin_plasminogen

maximal_conversion_rate_thrombin_plasminogen

production_rate_plasminogen (†)

degradation_rate_pc

degradation_rate_pc_activated

half_maximal_effect_concentration_aii_tmod_complex_pc

maximal_activation_rate_aii_tmod_complex_pc

production_rate_pc (†)

degradation_rate_ps

production_rate_ps (†)

degradation_rate_tmod

production_rate_tmod (†)

degradation_rate_tissue_factor

conversion_rate_vk_vkh2

conversion_rate_vkh2_vko

conversion_rate_vko_vk

elimination_rate_vk

half_maximal_effect_concentration_warfarin

maximal_inhibitory_effect_warfarin

transition_rate_vk_central_to_peripheral (†)

transition_rate_vk_peripheral_to_central (†)

input_rate_vk (†)

elimination_rate_warfarin

gamma

degradation_rate_tfpi

production_rate_tfpi (†)

degradation_rate_tat

complex_formation_rate_aii_atiii_heparin_complex

complex_formation_rate_aii_tmod

degradation_rate_aii_tmod_complex

complex_formation_rate_av_ax

degradation_rate_av_ax_complex

half_maximal_effect_concentration_apc_ps_complex_av_ax_complex

maximal_degradation_rate_apc_ps_complex_av_ax_complex

complex_formation_rate_vii_tf

degradation_rate_vii_tf_complex

half_maximal_effect_concentration_ax_vii_tf_complex

half_maximal_effect_concentration_tf_vii_tf_complex

maximal_activation_rate_ax_vii_tf_complex

maximal_activation_rate_tf_vii_tf_complex

complex_formation_rate_avii_tf

degradation_rate_avii_tf_complex

complex_formation_rate_aix_atiii_heparin_complex

complex_formation_rate_aix_aviii

degradation_rate_aix_aviii_complex

complex_formation_rate_ax_atiii_heparin_complex

complex_formation_rate_ax_tfpi

degradation_rate_ax_tfpi_complex

complex_formation_rate_apc_ps

degradation_rate_apc_ps_complex

complex_formation_rate_avii_tf_ax_tfpi

degradation_rate_avii_tf_ax_tfpi_complex

elimination_rate_atiii_heparin_complex

atiii_heparin_complex_amount

coagulation_factor_ii_amount

activated_coagulation_factor_ii_amount

aii_at_complex_amount

aii_atiii_heparin_complex_amount

thrombomodulin_amount

aii_tmod_complex_amount

coagulation_factor_v_amount

activated_coagulation_factor_v_amount

activated_coagulation_factor_x_amount

av_ax_complex_amount

coagulation_factor_vii_amount

activated_coagulation_factor_vii_amount

tissue_factor_amount

vii_tf_complex_amount

avii_tf_complex_amount

ax_tfpi_complex_amount

avii_tf_ax_tfpi_complex_amount

coagulation_factor_viii_amount

activated_coagulation_factor_viii_amount

coagulation_factor_ix_amount

activated_coagulation_factor_ix_amount

aix_aviii_complex_amount

aix_atiii_heparin_complex_amount

coagulation_factor_x_amount

ax_atiii_heparin_complex_complex_amount

tissue_factor_pathway_inhibitor_amount

coagulation_factor_xi_amount

activated_coagulation_factor_xi_amount

coagulation_factor_xii_amount

activated_coagulation_factor_xii_amount

coagulation_factor_xiii_amount

activated_coagulation_factor_xiii_amount

contact_system_activator_amount

fibrinogen_amount

fibrin_degradation_product_amount

fibrin_amount

cross_linked_fibrin_amount

d_dimer_amount

prekallikrein_amount

kallikrein_amount

plasminogen_amount

plasmin_amount

protein_c_amount

activated_protein_c_amount

protein_s_amount

apc_ps_complex_amount

vitamin_k_amount

vitamin_k_hydroquinone_amount

vitamin_k_epoxide_amount

vitamin_k_peripheral_amount (†)

warfarin_amount

drug_amount (†)

B.2 The Regression model

B.2.1 Background

This MIPD method assumes that inter-individual differences of warfarin maintenance doses can be predicted based on covariates of the treatment response variability

$$d^* = d^*(\chi, y^*). \tag{B.11}$$

d^* denotes the maintenance dose, χ denotes the covariates and y^* denotes the target treatment response. This assumed functional relationship between d^*, χ and y^* is unknown, but may be learned from data, e.g. using a neural network (Hornik et al. 1989).

B.2.2 Implementation

In our implementation of the regression model, we approximate the function d^* with a neural network implemented in `PyTorch` (Paszke et al. 2019)

$$d^*(\chi, y^*) \approx f_\theta(\chi, y^*), \tag{B.12}$$

where f denotes the neural network and θ denotes its parameters. The architecture of the network is a modelling choice. Our network is composed of three fully connected layers with nonlinear activations

$$f_\theta(\chi, y^*) = f_{\theta_3}^{(3)}\Big(f_{\theta_2}^{(2)}\Big(f_{\theta_1}^{(1)}(\chi, y^*)\Big)\Big). \tag{B.13}$$

The parameters of f are the parameters of the individual layers, $\theta = (\theta_1, \theta_2, \theta_3)$. Layer 1 takes as input the covariates and the target treatment response, resulting in five input values, $\mathbf{x} = (a_G, a_{*1}, a_{*2}, t_\star, y^*)$. Here, a_G denotes the number of G VKORC1 alleles, a_{*1} and a_{*2} denote the number of $*1$ and $*2$ CYP2C9 alleles and $t_\star$ denotes the age. The first 4 elements of $\mathbf{x}$ are given by the covariates, χ, of each patient. The fifth value is the INR value measured under maintenance treatment. The output of layer 1 are 1024 intermediate (hidden) features obtained by calculating the matrix product of the fully connected layer, $\mathbf{W}\mathbf{x}$, and applying the rectified linear unit (ReLU) activation function (Agarap 2018) to the output

$$f^{(1)}(\chi, y^*) = \mathrm{ReLU}(\mathbf{W}\mathbf{x}). \tag{B.14}$$

Here, $\mathbf{W}$, denotes the weights of the fully connected layer and is of shape (1024, 5). The learnable parameters of layer 1, θ_1, are the elements of the weight matrix and the parameters of the ReLU activation function.

Layer 2 is also a fully connected layer with ReLU activation and takes as input the output from layer 1

$$f^{(2)}(f^{(1)}) = \text{ReLU}(\mathbf{W}f^{(1)}). \tag{B.15}$$

The weight matrix of layer 2 is of shape (1024, 1024), resulting in an output of 1024 intermediate (hidden) features, similar to layer 1. The parameters of layer 2, θ_2, are the elements of the weight matrix and the parameters of the ReLU activation function.

Layer 3 is the final layer of the network, taking the features from layer 2 as input

$$f^{(3)}(f^{(2)}) = \text{Sigmoid}(\mathbf{W}f^{(2)}). \tag{B.16}$$

The weight matrix of layer 3 is of shape (1, 1024). In contrast to layer 1 and 2, the final layer has a sigmoid activation, constraining the output values of the network to the interval between 0 and 1. The parameters of layer 3, θ_3, are the elements of the weight matrix and the parameters of the sigmoid activation function.

For computational stability, both the input and the output of the network are normalised. The allele counts, (a_G, a_{*1}, a_{*2}), are normalised by the maximum allele count; here, 2. The age, $t_\star$, is centered around 51 and normalised by 15, approximating the typical age and the typical standard deviation of the age in a population. The target INR, y^*, is centered at 2.5 and normalised by 2.5, motivated by the therapeutic range of warfarin. The output of the network is already normalised to 0 and 1.

B.2.3 Dose prediction

For maintenance dose predictions, the model output needs to be translated into dosages of warfarin. We do this by assuming a maximally feasible maintenance dose of 30 mg. This makes it possible to interpret the network output as the predicted maintenance dose in units of the maximal maintenance dose, such that predicted doses can be

obtained by multiplying the model output with 30 mg. This constrains the model predictions to doses between 0 mg and 30 mg.

For predictions during the MIPD trial, the network output is further constrained to doses that can be administered in terms of combinations of commercially available warfarin tablets, namely 0 mg, 1 mg and any dose ≥ 2 mg in steps of 0.5 mg. The continuous output of the model is mapped to these dosages by rounding the output to the nearest commercially available dose.

B.2.4 Training

We train the model on the simulated clinical phase III data, introduced in subsection 3.3.1. The data includes monitoring data, covariates and warfarin dosing regimens for 1000 patients simulated with the warfarin trial model (see Appendix B.1). The data used for training are the covariates, the INR measurements, and the maintenance warfarin dosages at the end of the trial. We split the data randomly into a train set ($90\,\%$) and into a test set ($10\,\%$). The inputs and the dosages are normalised as outlined above. The model is trained for 1000 epochs to minimise the mean squared error objective function using Adam. We use a batch size of 128 and a learning rate of 10^{-4}. The learning rate is reduced by a factor of 2 every time the objective function does not improve for 10 consecutive epochs until a minimum learning rate of 10^{-7} is reached. The results of the training are shown in Figure B.3.

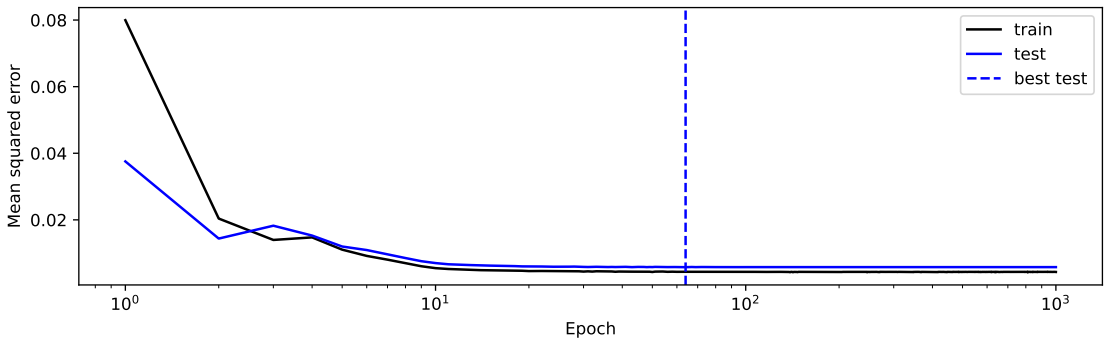


Figure B.3: Training of model 1: The figure shows the mean squared error of the model predictions on the train set (black) and the test set (blue). The epoch with the lowest mean squared error on the test set is indicated by a dashed line.

The figure shows the objective function evaluated on the training dataset and the test dataset. The epoch with the smallest test error is highlighted with a dashed line. We use this 'best test error' model for MIPD in the remainder of the article.

B.3 The Deep RL model

B.3.1 Background

This MIPD method is a reinforcement learning approach which assumes that dose decisions can be modelled as a Markov decision process defined by a state space, $\mathcal{S}$, an action space, $\mathcal{A}$, and stochastic transition dynamics, $p(s_{j+1}|s_j, a_j)$ (Zadeh et al. 2023). $s_j \in \mathcal{S}$ denotes the state at time $t_j = j\Delta t$, where Δt is the temporal resolution of the model, and $a_j \in \mathcal{A}$ denotes the action at time t_j. In our case, we define the state of an individual by the individual's covariates and INR monitoring measurement, $s_j = (\chi, y_j)$. The actions are the different warfarin doses that may be administered, $a_j = d_j$, and the transition dynamics are the treatment response dynamics following a dose administration. We choose a temporal resolution of the model of one day, $\Delta t = 1$ day, since warfarin doses are administered daily. The Markov decision process assumption makes it possible to derive optimal dose decisions based only on the current treatment response of an individual (Bellman 1966), as discussed in more detail below.

In reinforcement learning optimality is defined using a reward function, $r(s)$. The reward function is a modelling choice and quantifies the desirability of a state. In our case, we define the reward function using the negative mean squared error of the INR measurement, y, to the target treatment response, y^*,

$$r(s) = -\left(y - y^*\right)^2. \tag{B.17}$$

We reward INR measurements close to the target INR, and penalise INR measurements away from the target INR.

Using the reward function, we can retrospectively quantify the 'return' of a sequence of actions, $(a_0, \ldots, a_n)$, using the accumulation of the rewards associated with the

states, $(s_1, \ldots, s_{n+1})$, that followed the actions

$$R(a_0, \ldots, a_n, s_1, \ldots, s_{n+1}) = \sum_{j=1}^{n+1} r(s_j). \tag{B.18}$$

In reinforcement learning, this retrospective evaluation is converted into a prospective evaluation using a policy, $\pi(a|s)$, which formalises the way actions are chosen. The policy defines the probability of taking action a given that the system is currently in state s. This makes it possible to estimate the future return or 'value' of choosing an action a when being in state s, assuming that all following actions are chosen according to the policy

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{j=1}^{\infty} \gamma^{j-1} r(s_j) \,\middle|\, s, a\right] \tag{B.19}$$

We use j to reference the time relative to the current state. For the prospective evaluation, it is customary to discount the contributions from states further into the future with a discount factor, $\gamma < 1$, as those states are associated with a larger degree of uncertainty. The expectation is taken with respect to all possible future states and actions, distributed according to the transition dynamics and the policy. Note that the Markov decision process assumption significantly simplifies this expectation, as neither the transition dynamics nor the policy need to take past states and actions into account.

The action-value function, $Q^{\pi}(s, a)$, provides a simple scoring system for actions – the action with the largest function value has the largest expected future return. The value of the function is also referred to as the Q-value. The optimal policy, π^*, is defined as the policy that always chooses the action with the largest expected future return

$$\pi^*(a|s) = \begin{cases} 1 & \text{for} \quad \underset{a}{\operatorname{argmax}}\, Q^{\pi^*}(s, a) \\ 0 & \text{else.} \end{cases} \tag{B.20}$$

This optimal policy is independent of time, previous actions and previous states – a consequence of the Markov property of the decision process. In the context of MIPD, this means that reinforcement learning can be used to define doses as a function of the current state of an individual

$$d_j = f(\chi, y_j), \tag{B.21}$$

where the function f is independent of time, previous treatment response measurements and previous dose administrations. Under the Markov assumption, these doses optimally target the desired treatment response, Equation B.17.

While the action-value function, Q^{π^*}, cannot be computed exactly without knowing the transition dynamics, (Watkins et al. 1992) showed that Q^{π^*} can be learned from data iteratively without a detailed understanding of the dynamics of the system

$$Q_{n+1}(s_k, a_k) = (1-\alpha)Q_n(s_k, a_k) + \alpha(r(s'_k) + \gamma \max_{a'_k} Q_n(s'_k, a'_k)). \tag{B.22}$$

This learning algorithm is known as Q-learning. n denotes the iteration, $\alpha < 1$ denotes the learning rate, s_k denotes an observed state of the system, a_k denotes the action taken and $r(s'_k)$ denotes the reward associated with the state observed after taking action a_k. For a sufficiently large dataset, $\mathcal{D} = \{(s_k, a_k, s'_k)\}$, the estimate, Q_n, converges to the exact action-value function associated with the optimal policy, $\lim_{n\to\infty} Q_n = Q^{\pi^*}$.

However, a challenge with Equation B.22 is that each iteration only updates the Q-value of the observed state-action pair, (s_k, a_k), making the learning algorithm slow and incapable of generalising Q-values beyond the state-action pairs contained in the dataset. To address these limitations, (Mnih et al. 2013) introduced the deep Q network (DQN) modelling approach, in which the action-value function is approximated using a neural network

$$Q_n(s, a) \approx f_\theta(s, a). \tag{B.23}$$

f denotes the neural network and θ denotes its parameters. This converts the learning task to a function approximation task, where each state-action pair in the dataset informs the parameters of the model. As a result, the learning algorithm becomes more efficient and more capable of generalising Q-values to unobserved state-action pairs.

The network is trained by translating the Q-learning algorithm into an objective function. (Mnih et al. 2013) do this using the temporal difference error

$$\Delta_k(\theta) = \left| r(s'_k) + \gamma \max_{a'_k} f_\theta(s'_k, a'_k) - f_\theta(s_k, a_k) \right|. \tag{B.24}$$

The temporal difference error is the difference between the current Q-value estimate in Equation B.22, $Q_n(s_k, a_k)$, and the update, $r(s'_k) + \gamma \max_{a'_k} Q_n(s'_k, a'_k)$, where

we replaced the action-value function by the neural network approximation. The temporal difference error vanishes for the action-value function associated with the optimal policy, Q^{π^*}. By minimising the temporal difference error, a neural network can be trained to approximate Q^{π^*}.

In deep Q network (DQN) modelling, the optimal action-value function, $Q^{\pi^*}(s,a)$, is approximated using a neural network

$$Q^{\pi^*}(s,a) \approx f_\theta(s,a). \tag{B.25}$$

This network is learned from data. f denotes the neural network and θ denotes its parameters. The network is trained using a carefully constructed objective function which is motivated by the Q-learning algorithm (Mnih et al. 2013). More specifically, the objective function is the temporal difference error

$$\Delta(\theta) = \Big| r(s') + \gamma \max_{a'} f_\theta(s',a') - f_\theta(s,a) \Big|, \tag{B.26}$$

which is evaluated using data tuples consisting of a state, s, a chosen action, a, and the subsequently observed state, s'.

To motivate the temporal difference error, recall that $Q^\pi(s,a)$ quantifies the expected return when being in state s and choosing action a, while all subsequent actions are chosen according to the policy $\pi(a|s)$ (see Equation B.19). The optimal policy, $\pi^*(a|s)$, is defined as the policy that always chooses the action with the maximal value of the action-value function. In **??**, we discussed that the Q-learning algorithm (Equation B.22) finds the optimal action-value function, $Q^{\pi^*}(s,a)$, iteratively by updating the current action-value function (under some other policy) using the observed reward, $r(s')$, and the predicted maximal action-value under the current policy from s', $\max_{a'} Q_n(s',a'))$. More specifically, the current function value, $Q_n(s,a)$, is updated to $Q_{n+1}(s,a) = (1-\alpha)Q_n(s,a) + \alpha(r(s') + \gamma \max_{a'} Q_n(s',a'))$. Here, α denotes the learning rate. I.e. the target value of the action-value function is the moving target $r(s') + \gamma \max_{a'} Q_n(s',a')$, which for sufficiently many observed states and rewards converges to the optimal action-value function. The difference between this target value and the current action-value function, $Q_n(s,a)$, is referred to as the temporal difference error (Equation B.26).

By construction, the expectation of the temporal difference error is minimised only for the optimal policy, $\pi^*(a|s)$. As a result, we can train a neural network to approximate the optimal action-value function, Q^{π^*}, by minimising the temporal difference error.

B.3.2 Implementation

In our implementation of the DQN model, we closely follow Mnih et al and use a neural network to predict the Q-values for all possible actions simultaneously

$$\mathbf{Q}^{\pi^*}(s) \approx f_\theta(s), \tag{B.27}$$

where $\mathbf{Q}^{\pi^*}(s)$ is a vector containing the Q-values when being in state s, $\mathbf{Q}^{\pi^*}(s) = [Q^{\pi^*}(s, a_1), \ldots, Q^{\pi^*}(s, a_n)]$ (Mnih et al. 2013). In our case, the actions, $(a_1, \ldots, a_n)$, are the daily warfarin dosages that can be administered as a combination of commercially available warfarin tablets, namely 0 mg, 1 mg and any dose ≥ 2 mg in 0.5 mg increments. We constrain dosages to a maximum of 30 mg, resulting in a total of $n = 58$ possible dosages.

We implement the neural network in `PyTorch` (Paszke et al. 2019) using four fully connected layers with nonlinear activation functions

$$f_\theta(s) = f^{(4)}_{\theta_4}\Big(f^{(3)}_{\theta_3}\Big(f^{(2)}_{\theta_2}\Big(f^{(1)}_{\theta_1}(\chi, y)\Big)\Big)\Big). \tag{B.28}$$

The parameters of f are the parameters of the individual layers, $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$. Layer 1 takes as input the state of the system, i.e. an individual's covariates and the treatment response measurement, encoded using five input values, $\mathbf{x} = (a_G, a_{*1}, a_{*2}, t_\star, y)$, where a_G denotes the number of G VKORC1 alleles, a_{*1} and a_{*2} denote the number of $*1$ and $*2$ CYP2C9 alleles and $t_\star$ denotes the age. The first 4 values in $\mathbf{x}$ are given by the covariates, χ. The output of layer 1 are 256 intermediate (hidden) features obtained by calculating the matrix product of the fully connected layer, $\mathbf{W}\mathbf{x}$, and applying the ReLU activation function to the output

$$f^{(1)}(\chi, y^*) = \mathrm{ReLU}(\mathbf{W}\mathbf{x}). \tag{B.29}$$

Here, $\mathbf{W}$, denotes the weights of the fully connected layer and is of shape (256, 5). The learnable parameters of layer 1, θ_1, are the elements of the weight matrix and the parameters of the ReLU activation function.

Layer 2 also uses a ReLU activation and takes as inputs the intermediate features of layer 1

$$f^{(2)}(f^{(1)}) = \text{ReLU}(\mathbf{W}f^{(1)}). \tag{B.30}$$

The weight matrix of layer 2 is of shape (128, 256), resulting in an output of 128 intermediate features. The parameters of layer 2, θ_2, are the elements of the weight matrix and the parameters of the ReLU activation function.

Layer 3 has a similar architecture as layer 2 but takes as input the outputs from layer 2

$$f^{(3)}(f^{(2)}) = \text{ReLU}(\mathbf{W}f^{(2)}). \tag{B.31}$$

The weight matrix of layer 3 is of shape (64, 128), resulting in an output of 64 intermediate features. The parameters of layer 3, θ_3, are the elements of the weight matrix and the parameters of the ReLU activation function.

Layer 4 is the final layer of the network, taking the intermediate features from layer 3 as input

$$f^{(4)}(f^{(3)}) = \mathbf{W}f^{(3)}. \tag{B.32}$$

The weight matrix of layer 4 is of shape (58, 64), resulting in 58 outputs of the network. These 58 outputs are the predicted Q-values for the 58 possible warfarin dosages. In contrast to the other layers, the final layer uses no activation function.

For computational stability the input to the network is normalised. The allele counts, (a_G, a_{*1}, a_{*2}), are normalised by the maximum allele count, 2. The age, $t_\star$, is centered around 51 and normalised by 15, approximating the typical age and the typical standard deviation of the age in a population. The INR measurement, y, is centered at 2.5 and normalised by 2.5, motivated by the therapeutic range of warfarin.

B.3.3 Dose prediction

For dose predictions, the predicted Q-values need to be translated into dosages of warfarin. We do this following the optimal policy (see Equation B.20) by administering the dose with the maximal Q-value. During the MIPD trial, we predict individualised dosing regimens iteratively using the current INR measurement and the covariates of an individual.

B.3.4 Training

Following Zadeh et al, we train the model not directly on trial data, but on treatment responses simulated with a PKPD model (Zadeh et al. 2023). The PKPD model is fitted to trial data before the training.

The training on emulated treatment responses is necessary because standard DQN models are trained online (Ribba, Bräm, et al. 2022), making it challenging to use data from completed clinical trials for training. Offline reinforcement learning is an active field of research (Agarwal et al. 2020) not considered in this chapter. The PKPD model we choose for the simulation is Hamberg et al's model (Hamberg, Wadelius, et al. 2010), which we also use for the PKPD model-based MIPD approach (see subsection 3.2.5). For details on the inference of the PKPD model parameters, we refer to section B.4.

We train the model for 1500 epochs with a learning rate of 10^{-4}, a batch size of 128 and a discount factor of $\gamma = 0.9$. The data used for training are state-action-next state tuples simulated with the PKPD model. To simulate the data, the VKORC1 and CYP2C9 genotypes of individuals are drawn uniformly from the possible variants. The age is drawn from a lognormal distribution with median 51 and scale 0.15. The treatment response is then simulated for each individual for 30 days. The dosing regimens are determined by the DQN model using daily monitoring measurements and the covariates. The model administers dosages following an ϵ-greedy policy with $\epsilon = 0.05$, which balances exploration and exploitation during the training. This data-generation results in a total of 30 INR measurements and 30 dose administrations for each individual.

We train the model using experience replay with a buffer size of 30,000 data points (L.-J. Lin 1992). The buffer memorises the treatment response data of 1000 simulated

individuals. In each epoch, we train the model on the full buffer. At the start of the training, we fill the buffer, simulating 1000 individuals treated by the DQN model with the initial (random) policy. Every 20 epochs, we randomly replace the data of 100 individuals by the data obtained from 100 new individuals treated by the model with the most recent policy. This enables an online learning of the DQN model, while still benefiting from experience (L.-J. Lin 1992). To stabilise the training, we use Double Q-learning (Van Hasselt et al. 2016), where two DQN models are trained in tandem. One model is used to predict the Q value of the chosen action and to find the next action with the maximal Q value during the temporal difference error evaluation (see Eq B.26). The other network is used to evaluate the Q value for the next action. The second network is not trained during this iteration. After each epoch, the role of the two networks is exchanged. This stabilises the training of the DQN model (Van Hasselt et al. 2016).

The results of the training are shown in Figure B.4. The top panel of the figure shows the average temporal difference error of the two models for each epoch. The bottom panel shows the average reward in the experience buffer. The model's policy converges within the first 750 epochs of the training, reaching an average temporal difference error close to zero.

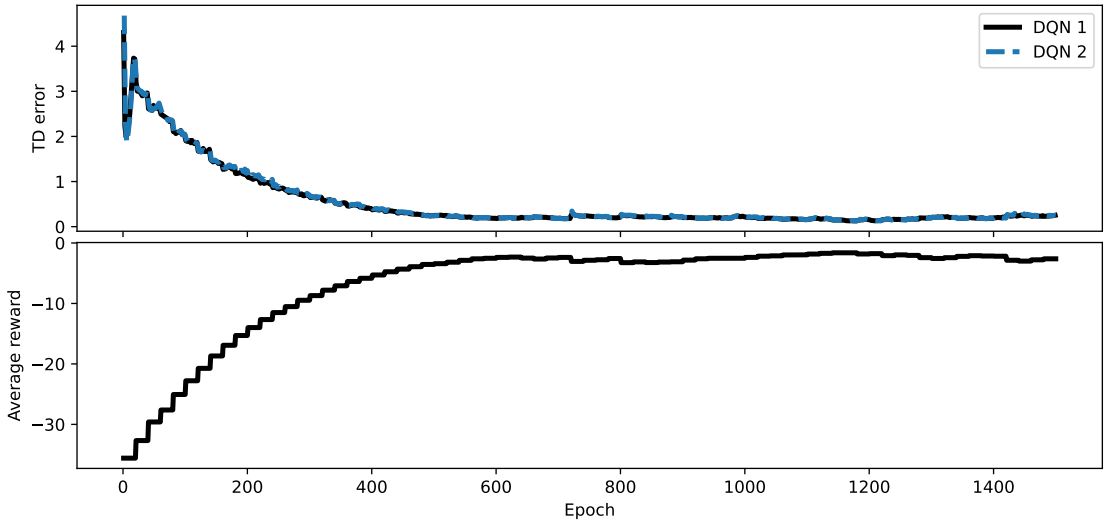


Figure B.4: Training of model 2: The top panel shows the temporal difference error of the two DQN models used during Double Q-learning. The bottom panel shows the average experience buffer reward.

B.4 The PKPD model

B.4.1 Background

This MIPD method uses pharmacokinetic and pharmacodynamic (PKPD) modelling to predict individualised dosing regimens. PKPD models are (semi-)mechanistic or empirical descriptions of treatment response dynamics, formalising biomarkers or clinical readouts as functions of time and the dosing regimen

$$\bar{y}(t, r, \psi). \tag{B.33}$$

$\bar{y}$ denotes the quantity of interest, t denotes the time, r denotes the dosing regimen and ψ denotes additional model parameters. This formalised relationship can be used to predict treatment responses (see section 2.2).

Inter-individual variability

Analogously to the clinical trial model introduced in subsection 3.2.1, PKPD models can be extended to describe variability in the treatment response using a population model

$$p(\psi|\theta, \chi), \tag{B.34}$$

where each ψ in the distribution represents an individual. θ denotes the parameters of the population distribution and χ denotes the covariates of the IIV. The treatment response for an individual characterised by the parameters, ψ, is predicted with the PKPD model (see Equation B.33). The PKPD model together with the population model is referred to as nonlinear mixed effects (NLME) model (see subsection 2.2.7).

To illustrate the effect of varying parameter values on treatment responses, we show INR simulations of Hamberg et al's NLME model in Figure B.5 (Hamberg, Wadelius, et al. 2010). For illustrative purposes, we only consider variability in the EC50 across individuals. The remaining parameter values are fixed. The model incorporates single nucleotide polymorphisms of the VKORC1 genotype as covariates of the EC50 variability (see Appendix B.4.2 for the details of the model).

The figure shows the population distribution of the EC50 parameter values in terms of three subpopulations, each containing individuals with identical VKORC1 genotypes

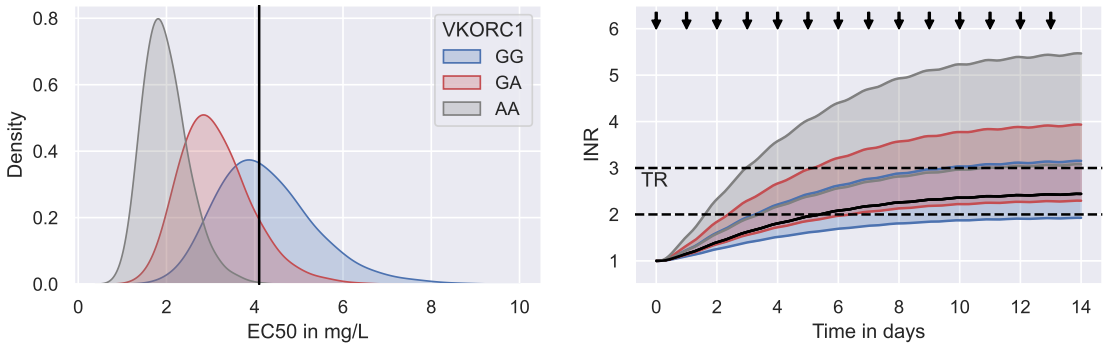


Figure B.5: Inter-individual variability of warfarin EC50 and treatment response. The figure shows Hamberg et al's model of the warfarin EC50 variability across patients (see left panel) and the predicted INR response variability when daily doses of $7.5\,\text{mg}$ warfarin are administered to all individuals. The colours indicate individuals with the same VKORC1 genotypes: GG (blue), GA (red) and AA (grey). The coloured areas in the right panel indicate the 5th to 95th percentile interval of the INR response across individuals with the same VKORC1 genotype. The black solid line represents a typical patient with the GG genotype. The dose administrations are indicated by black arrows in the right panel. The therapeutic range (TR) is indicated by dashed black lines.

(see left panel). The corresponding distribution of treatment response simulations are visualised in the right panel. In the simulations, all individuals are treated with the same warfarin dosing regimen ($7.5\,\text{mg}$ every 24 hours). Notably, Hamberg et al's model predicts substantial IIV that is not explained by the VKORC1 genotype, such that individuals with, for example, the GG genotype have EC50 values ranging from $1\,\text{mg/L}$ to $9\,\text{mg/L}$. A typical individual with the GG genotype is illustrated using a black solid line.

Large treatment response variability implies that also optimal dosing regimens will vary between individuals (see Figure B.6). The top left panel shows the treatment response across individuals treated with dosing regimens, that are optimised to achieve a minial MSE to a target INR of 2.5. For each VKORC1 genotype, the dosing regimen of 1000 individuals is optimised. The dose distributions of the first 6 doses (on day 1 to 6 of the treatment) are visualised in the right panel. Each subpanel shows the dose distribution across individuals with the different VKORC1 genotypes: GG (blue); GA (red); and AA (grey). The distribution of maintenance doses is illustrated in the bottom left panel.

The figure shows that optimising dosing regimens for individuals reduces the predicted treatment response variability (see top left panel), achieving TTRs of $\geq 86\,\%$ within the

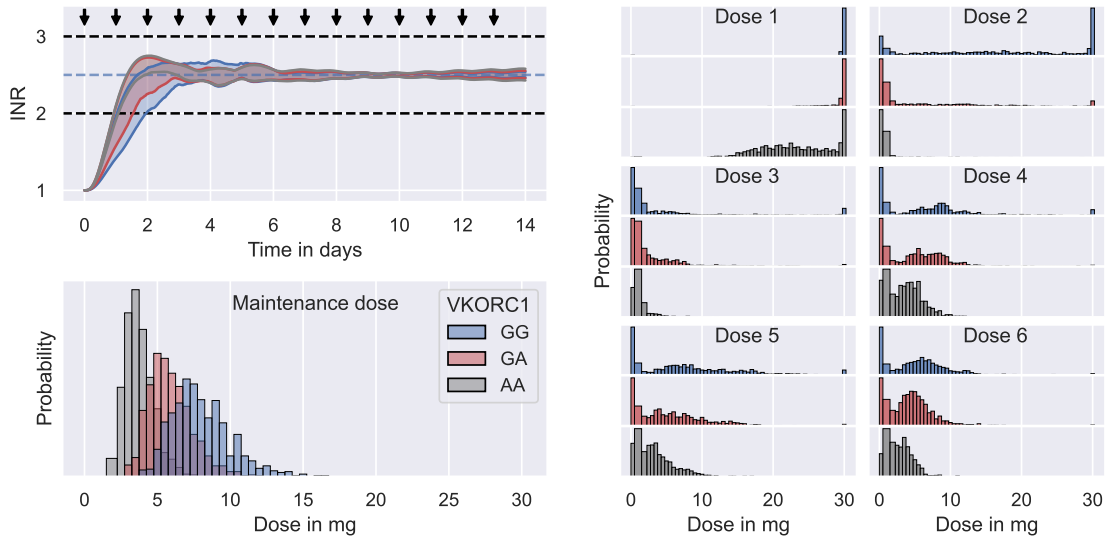


Figure B.6: Variability of optimal dosing regimen. The figure shows predicted treatment responses (top left panel) of 3000 individuals to dosing regimens that minimise the MSE to a target INR of 2.5. The treatment response distributions are illustrated by the 5th to 95th INR percentile intervals across individuals with the same VKORC1 genotype: GG (blue); GA (red); and AA (grey). Each genotype group is represented by 1000 individuals. The distribution of optimal maintenance dosages is illustrated in the bottom left panel. The distributions of optimal induction dosages on days 0 to 5 are illustrated in the 6 panels on the right. The dose administrations are indicated by black arrows in the top left panel. The therapeutic range (black) and the target value (blue) are indicated by dashed lines.

first 14 treatment days for most individuals. This improved homogeneity in the treatment response is achieved by varying dosing regimens across individuals. Maintenance dosages for individuals with the AA VKORC1 genotype range from $2\,$mg to $7.5\,$mg, while dosages for individuals with the GG genotype span a range of $\geq 10\,$mg (see bottom left figure). This variability is even more pronounced during the induction phase of the treatment (see right 6 panels).

Covariate-based dosing regimen optimisation

Figure B.6 shows that dosing regimen optimisation can account for variability in the predicted treatment response when the EC50s of treated individuals are known. In practice, the EC50 and, more generally, the other model parameters, ψ, that best predict an individual's treatment response are, however, unknown. One approach to alleviate this problem is to use NLME models fitted to clinical trial data to suggest

model parameters for individuals based on covariates of the IIV (Hamberg and Wadelius 2014; Hamberg, Wadelius, et al. 2010). In particular, we can predict a set of typical parameter values for an individual with the covariates, χ, using the distribution of model parameters, $p(\psi|\theta, \chi)$. In this chapter, we define typical parameter values as the median of the parameter distribution. Other choices such as the mean or the values with maximum probability density (MAP) are also possible.

In Figure B.7, we illustrate this covariate-based dosing regimen optimisation using Hamberg et al's NLME model. The left panel shows the typical EC50 values for individuals with the GG (blue), GA (red) and AA (grey) VKORC1 genotypes predicted by Hamberg et al's NLME model (see solid lines). The dashed lines illustrate the predicted EC50 distributions by the model. The middle panel shows the optimal dosing regimens associated with the typical parameters, obtained by minimising the MSE of the predicted treatment response to the target INR of 2.5. These optimal dosing regimens are able to partially account for the treatment response variability across individuals (see right panel), such that the majority of the predicted treatment responses is inside the TR.

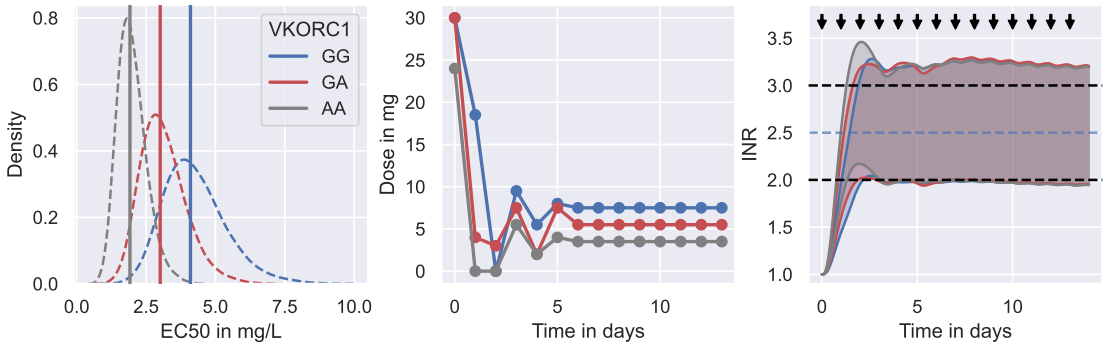


Figure B.7: Covariate-based dosing regimen optimisation. The left panel shows typical EC50 values for individuals with the GG (blue), GA (red) and AA (grey) VKORC1 genotypes predicted by Hamberg et al's NLME model as solid lines. The distributions of EC50 values across individuals are indicated by dashed lines. The middle panel illustrates the dosing regimens that minimise the MSE of the predicted treatment response for the typical EC50 values to a target INR of 2.5. The right panel shows the treatment responses of 3000 individuals (1000 for each genotype). Each individual is treated with the optimal dosing regimen predicted for the typical EC50 (see middle panel). The treatment response distributions are illustrated by the 5th to 95th INR percentile intervals across individuals with the same VKORC1 genotype. The dose administrations are indicated by black arrows in the top left panel. The therapeutic range (black) and the target value (blue) are indicated by dashed lines.

Bayesian dosing regimen optimisation

Covariate-based dosing regimen optimisation is able to partially account for treatment response variability by predicting typical model parameters based on covariates of the IIV. These predictions rely on the calibration of the NLME model, with no mechanism to adjust predictions if the predicted treatment response deviates from the observed treatment response due to residual uncertainty in the model parameters, model misspecification or IOV of the individual's treatment response.

A complementary approach that addresses these shortcomings is Bayesian dosing regimen optimisation (Maier, Hartung, de Wiljes, et al. 2020). Bayesian dosing regimen optimisation uses feedback of the treatment response in form of monitoring measurements to iteratively update the model to best describe the treatment response

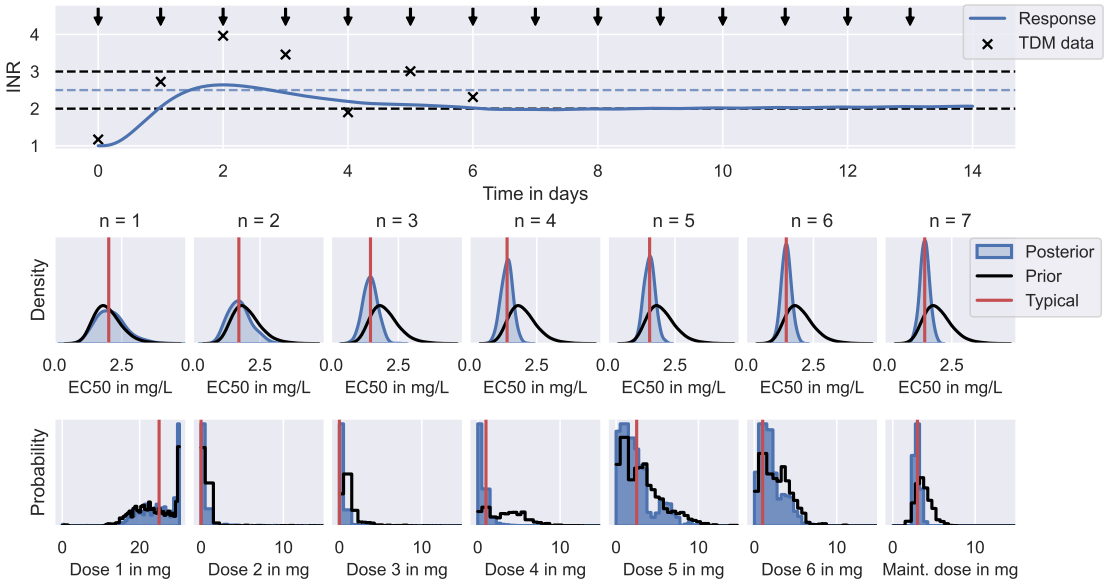


Figure B.8: Bayesian dosing regimen optimisation. The figure illustrates the dosing regimen individualisation for an individual with the AA VKORC1 genotype using Bayesian dosing regimen optimisation. The predicted treatment response (blue line) and the monitoring measurements (black crosses) are illustrated in the top panel. The dose administrations are indicated using black arrows. The therapeutic range (black) and the target value (blue) are indicated using dashed lines. The posterior distributions (blue) inferred from the measurements on each day are illustrated in the middle panel. n denotes the number of measurements available for each parameter estimation. The typical parameter value of the posterior distribution is illustrated in red. The bottom panel shows the optimal dose distributions associated with the posteriors (blue), the typical parameters and the prior (black).

of an individual, making dosing regimen adjustments and more personalised predictions possible (see subsection 2.2.5).

We illustrate this concept of Bayesian dosing regimen optimisation using Hamberg et al's NLME model as an example in Figure B.8. The figure shows the predicted treatment response (blue) for an individual with the AA VKORC1 genotype treated with an iteratively optimised dosing regimen (see top panel). The monitoring measurements of the INR are collected before each dose administration on days 0 to 6, and are illustrated using black scatter points. From day 6 onwards, the dose amount remains unchanged.

B.4.2 Implementation

In our implementation of the PKPD modelling approach, we use Hamberg et al's PKPD model to simulate the warfarin treatment response dynamics (Hamberg, Wadelius, et al. 2010). The model is schematically illustrated in Figure C.1.

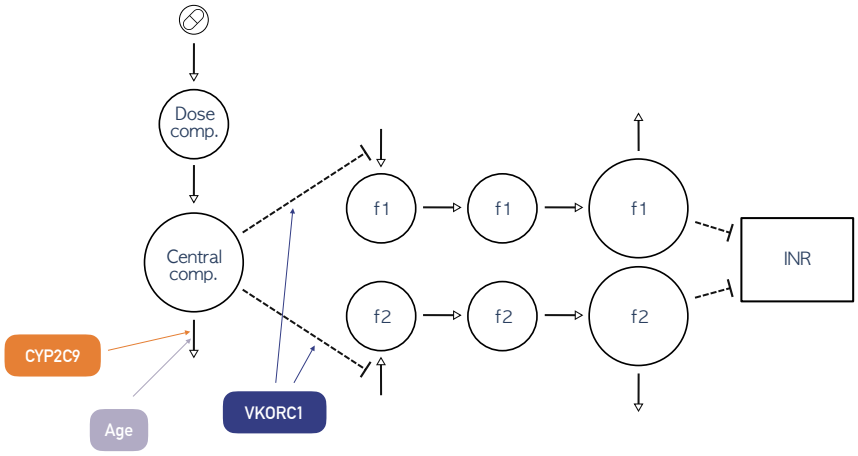


Figure B.9: Hamberg et al's model of the warfarin treatment response. The figure shows a diagrammatic representation of Hamberg et al's model of the INR response to warfarin treatment. The diagram illustrates the PK model component (left side) and the PD model component (right side). The drug (see pill in the top left corner) is administered to a dose compartment from which it moves into a central compartment. From the central compartment warfarin is cleared with a rate that is influenced by the CYP2C9 genotype and the age of an individual. The drug concentration in the central compartment inhibits the production of two (effective) coagulation factors, f_1 and f_2. The inhibitory effect of the drug is influenced by the VKORC1 genotype. Each coagulation factor is produced in one compartment, moved to an intermediary compartment and is cleared in a third compartment. The coagulation factor concentrations in the final compartments determine the INR. The lower the coagulation factor concentrations, the larger the INR.

The model provides a semi-mechanistic description of the INR response to warfarin treatment, parametrised by 10 pharmacologically-interpretable parameters, $\psi = (\bar{y}_0, \bar{y}_{max}, k_a, k_e, \kappa, \gamma, c_{50}, k_1, k_2, v)$, (see Equations 2.4, 2.5, and 2.6).

This semi-mechanistic description is complemented by a population model which captures the variability of treatment response across individuals. More specifically, the elimination rates, the half maximal inhibitory effect concentrations, and the effective volumes of distribution are assumed to vary across patients. This variation is described in the traditional NLME picture individual-specific random deviations from a set of typical parameter values

$$k_e = \mu_{k_e} \, \mathrm{e}^{\eta_{k_e}}, \quad c_{50} = \mu_{c_{50}} \, \mathrm{e}^{\eta_{c_{50}}}, \quad v = \mu_v \, \mathrm{e}^{\eta_v}, \tag{B.35}$$

where $(\mu_{k_e}, \mu_{c_{50}}, \mu_v)$ denote the typical values and $(\eta_{k_e}, \eta_{c_{50}}, \eta_v)$ denote the individual-specific random deviations. For $\eta < 0$, the parameter of an individual is smaller than the typical value and for $\eta > 0$, it is larger than the typical value. The model assumes that the η follow Gaussian distributions centered at zero with variance σ^2, e.g. $\mathcal{N}(\eta_{k_e}|0, \sigma^2_{k_e})$. The remaining parameters are fixed across individuals.

The model further assumes that the variability of k_e and c_{50} can be partially explained by 3 covariates: 1. the CYP2C9 genotype (one of *1/*1, *1/*2, *1/*3, *2/*2, *2/*3 and *3/*3); 2. the VKORC1 genotype (one of GG, GA and AA); and 3. the age. In particular, the typical elimination rate and the typical half maximal inhibitory effect concentration are modelled as functions of the covariates $\chi = (c_1, c_2, v_1, v_2, t_\star)$

$$\begin{aligned} \mu_{k_e}(c_1, c_2, t_\star) &= (k_{c_1} + k_{c_2}) \, (1 - \tanh(r_{age}(t_\star - 71))) \\ \mu_{c_{50}}(v_1, v_2) &= c_{v_1} + c_{v_2}, \end{aligned} \tag{B.36}$$

where $c_1, c_2 \in \{*1, *2, *3\}$ denote the genotypes of the CYP2C9 alleles, $v_1, v_2 \in \{\mathrm{G}, \mathrm{A}\}$ denote the genotypes of the VKORC1 alleles and $t_\star$ denotes the age in years. The $\tanh$-function is a modification of Hamberg et al's original model that we introduced to avoid non-positive elimination rates, $\mu_{k_e} \leq 0$, for $t_\star \gg 71$. Equations B.35

and B.36 define the distribution of the individual-level parameters in the population conditional on the covariates

$$\begin{aligned} p(\psi|\theta,\chi) &= \mathrm{LN}\left(k_e \,|\, \mu_{k_e}(c_1,c_2,t_\star),\sigma_{k_e}\right)\,\mathrm{LN}\left(c_{50} \,|\, \mu_{c_{50}}(v_1,v_2),\sigma_{c_{50}}\right) \\ &\quad \mathrm{LN}\left(v \,|\, \mu_v,\sigma_v\right)\,\delta(\mu_{\bar{y}_0}-\bar{y}_0)\,\delta(\mu_{\bar{y}_{max}}-\bar{y}_{max})\,\delta(\mu_{k_a}-k_a) \\ &\quad \delta(\mu_\kappa-\kappa)\,\delta(\mu_\gamma-\gamma)\,\delta(\mu_{k_1}-k_1)\,\delta(\mu_{k_2}-k_2), \end{aligned} \tag{B.37}$$

where $\theta = (k_{*1}, k_{*2}, k_{*3}, r_{age}, \sigma_{k_e}, c_G, c_A, \sigma_{c_{50}}, \mu_v, \sigma_v, \mu_{\bar{y}_0}, \mu_{\bar{y}_{max}}, \mu_{k_a}, \mu_\kappa, \mu_\gamma, \mu_{k_1}, \mu_{k_2})$ denotes the parameters of the population model, $\mathrm{LN}(x|\mu,\sigma) = \mathrm{e}^{(\log\mu-\log x)^2/2\sigma^2}/x\sigma\sqrt{2\pi}$ denotes a lognormal distribution, and $\delta(\mu - x)$ denotes a Dirac delta distribution.

In their original study, (Hamberg, Wadelius, et al. 2010) fixed the parameter values of the absorption rate, the baseline INR, and the maximum INR increase to $\mu_{k_a} = 2\,1/\mathrm{h}$, $\mu_{\bar{y}_0} = 1$, and $\mu_{\bar{y}_{max}} = 20$, respectively. In our study, we fix the absorption rate and the maximum INR shift to the same values. However, we do not fix the baseline INR. Instead, we replace the pooled baseline INR population model in Equation B.38 by a lognormal covariate population model

$$\mathrm{LN}\left(\bar{y}_0 \,|\, \mu_{\bar{y}_0}(v_1,v_2),\sigma_{\bar{y}_0}\right) \quad \text{with} \quad \mu_{\bar{y}_0}(v_1,v_2) = \bar{y}_{0,v_1} + \bar{y}_{0,v_2}. \tag{B.38}$$

We do this in order to account for IIV in the baseline INR (see Figure B.1).

B.4.3 Dose prediction

Individualised dosing regimens are predicted based on the covariates and the monitoring data using Bayesian dosing regimen optimisation (see Figure B.8). The model is first fitted to an individual's data. In a second step, the fitted model is used to predict an optimised dosing regimen.

To fit the model, we use Bayesian inference to estimate a distribution of individual-level parameters compatible with both, the population model, $p(\psi|\chi,\theta)$, and an individual's data, $\mathcal{D}_i = ((y_{i1},t_{i1}),\ldots,(y_{in},t_{in}),r_i,\chi_i)$. i labels the data specific to an individual and n denotes the number of monitoring measurements. To this end, we use the population model, $p(\psi|\chi_i,\theta)$, to define the prior distribution for the inference

$$\log p(\psi|\mathcal{D}_i,\theta) = \sum_{j=1}^{n} \log p(y_{ij}|t_{ij},r_i,\psi) + \log p(\psi|\chi_i,\theta) + \text{constant}. \tag{B.39}$$

The population-level parameters are estimated from the simulated clinical trial data, available prior to the MIPD (see subsection B.4.4).

The first term in Equation B.39 is the likelihood of the measurements under the model, which is defined using a measurement distribution, $p(y|t, r, \psi)$. The measurement distribution is an extension of the PKPD model output, defined in Equation B.33, accounting for discrepancies between measurements, y, and the model output, $\bar{y}$. In this chapter, we use a lognormal measurement distribution, $p(y|t, r, \psi) = \mathrm{LN}(y|\bar{y}(t, r, \psi), \sigma)$, to capture the measurement noise. For more details about the inference approach, we refer back to (see section 2.2).

We implement the PKPD model, the error model, and the posterior distribution in `chi` (Augustin 2024) and use the adaptive covariance matrix Markov Chain Monte Carlo sampling algorithm implemented in `pints` (Clerx, Robinson, et al. 2019) to sample from the posterior distribution. We run the algorithm for 10,000 iterations and discard the first 5000 iterations as tuning phase for the algorithm. We estimate the convergence of the marginal posterior distributions using the $\hat{R}$-statistic. All posteriors during the MIPD trial reach $\hat{R} \leq 1.01$. The results of the inference, as well as data and code to reproduce the results are hosted on GitHub (https://github.com/DavAug/mipd-warfarin).

To predict individualised dosing regimens, we minimise the mean squared error of the PKPD model prediction to the target INR value, $y^* = 2.5$, using n evaluation time points

$$\mathrm{MSE}(r_i) = \frac{1}{n} \sum_{j=1}^{n} \left(\bar{y}(t_j, r_i, \psi_i) - \bar{y}^* \right)^2. \tag{B.40}$$

The time points of the evaluation are a modelling choice. In this article, we choose $n = 5000$ equidistant time points starting at $t_1 = 0\,\mathrm{h}$ and ending at $t_{5000} = 50 \times 24\,\mathrm{h}$ since the start of the treatment, ensuring that the treatment response remains close to the target value even after the 19-day MIPD trial. The parameters, ψ_i, used to predict the treatment response are specific to each individual and are estimated from the posterior distribution, defined in Equation B.39. In this article, we approximate ψ_i by the typical value of the posterior distribution, i.e. $\psi_i = \mathrm{median}[\{\psi_{is}\}]$, where ψ_{is} denotes a sample generated using the MCMC sampling algorithm, after the tuning phase.

The parameters that can be optimised during the dosing regimen optimisation are the dosages, the administration times and the dose durations (see Equation 2.7). In this article, we constrain dosing regimens to daily bolus dosages, which we implement by setting the administration times to $t_k = (k-1) \times 24\,\text{h}$ and the dose durations to $\Delta_k = 0.01\,\text{h}$. To ensure that the optimisation converges to a single maintenance dose (as opposed to a maintenance dose pattern that may require different doses on different days), doses following the 19th treatment day are set to the 19th dose, i.e. $d_k = d_{19}$ for $k \geq 19$. This leaves 19 parameters for the optimisation – the first 19 warfarin dosages.

We optimise these 19 dosages using a sliding window approach: 1. doses already administered are fixed to the administered values; 2. the next 7 doses are optimised; 3. the remaining doses are set to the last dose in the optimisation window. At the start of the treatment, no doses have been administered and the first 7 doses are optimised. Doses 8 to 19 are set to the dose amount of dose 7. On the next day, the first dose is fixed, and doses 2 to 8 are optimised. The remaining doses are set to the dose amount of dose 8. This procedure continues until dose 14, where the condition of setting all doses following dose 19 reduces the window size to 6 days, such that on the last day of the trial only the 19th dose is optimised.

The mean squared error objective function is implemented using `chi` (Augustin 2024) and `pints` (Clerx, Robinson, et al. 2019). We minimise the objective function using the covariance matrix adaptation-evolution strategy (CMA-ES) optimisation algorithm implemented in `pints`.

B.4.4 Fitting

We fit the model to clinical phase I, II and III data simulated in subsection 3.3.1. The data includes monitoring data, covariates and warfarin dosing regimens of a total of 1160 simulated patients. We calibrate the model using hierarchical Bayesian inference.

Calibration to trial phase I data

We first calibrate the model parameters to the data from trial phase I. The data contains warfarin concentration measurements of 60 individuals (see left panel in Figure 3.5).

Denoting the data as $\mathcal{D}_1 = (\mathcal{D}_{1\,1}, \ldots, \mathcal{D}_{1\,60})$, where $\mathcal{D}_{1i}$ denotes the data of individual i analogously to subsection B.4.3, we can define a hierarchical posterior for the population model parameters using Bayes rule

$$\begin{aligned} \log p(\theta, \Psi_1 | \mathcal{D}_1) &= \sum_i \log p(\mathcal{D}_{1i} | \psi_{1i}) + \sum_i \log p(\psi_{1i} | \theta, \chi_{1i}) + \log p(\theta) \\ &\quad + \text{constant}. \end{aligned} \tag{B.41}$$

$\Psi_1 = (\psi_{1\,1}, \ldots, \psi_{1\,60})$ denotes the individual-level parameters of the individuals in the trial. The focus of the calibration is the posterior distribution of the population-level parameters, θ, obtained by marginalising over the individual-level parameters, $p(\theta|\mathcal{D}_1) = \int \mathrm{d}\Psi_1\, p(\theta, \Psi_1|\mathcal{D}_1)$. For an in-depth review of Bayesian inference of NLME models, we refer to the methods section in (Augustin, Lambert, K. Wang, et al. 2023).

The first term in Equation B.41 quantifies the likelihood of individual-level parameters using a measurement distribution analogously to subsection B.4.3. We model the error of the simulated warfarin concentrations with a lognormal error model. The scale parameter of the error model, σ, is pooled across individuals during the inference.

The data from trial I does not contain INR measurements and can therefore only be used to calibrate the PK component of the model. This leaves 8 parameters for the calibration: $(k_{*1}, k_{*2}, k_{*3}, r_{age}, \sigma_{k_e}, \mu_v, \sigma_v, \mu_\sigma)$, where we use μ_σ to denote the pooled scale parameter of the measurement distribution.

At the start of the calibration, we assume minimal prior knowledge about the model parameters to mimic a typical PKPD modelling scenario. We define weakly informative marginal priors whose responsibility is only to constrain parameter values

to biologically meaningful ranges

$$\begin{aligned}
\log k_{*1} &\sim \mathcal{N}(-3, 0.1) \\
\sigma_{k_e} &\sim \mathrm{LN}(-1, 0.3) \\
\frac{k_{*1} - k_{*2}}{k_{*2}} &\sim \mathcal{U}(0, 1) \\
\frac{k_{*1} - k_{*3}}{k_{*2}} &\sim \mathcal{U}(0, 1) \\
r_{age} &\sim \mathcal{N}(0, 0.01) \\
\log \mu_v &\sim \mathcal{N}(2.7, 0.1) \\
\sigma_v &\sim \mathrm{LN}(-1, 0.3) \\
\mu_\sigma &\sim \mathrm{LN}(-1, 0.3).
\end{aligned} \tag{B.42}$$

We construct the prior distribution, $p(\theta)$, by multiplying the marginal distributions.

We implement the PKPD model, the likelihood, the population model, the prior distribution and the posterior distribution in `chi` (Augustin 2024). Samples from the posterior distribution are generated using the No-U-Turn sampler (NUTS) implemented in `pints` (Clerx, Robinson, et al. 2019). We run the algorithm for 1500 iterations and discard the first 500 iterations as warmup. We estimate the convergence of the marginal posterior distributions using the $\hat{R}$-statistic. All marginal posteriors have $\hat{R} \leq 1.01$. We show the inferred posterior distribution in Figure B.10. The results of the inference, as well as data and code to reproduce the results are hosted on GitHub (https://github.com/DavAug/mipd-warfarin).

Figure B.10 shows the results of the calibration. The marginal prior distributions are illustrated in black (solid lines) and the posterior distributions are illustrated in blue. Approximations of the posterior distributions, explained and used in section B.4.4, are illustrated in black (dashed lines).

The goodness-of-fit is visualised in the top panel of Figure B.11. The left panel shows the concentration measurements used for the calibration. Measurements taken from the same individual are connected using a solid line. The nominal administration time of the warfarin dose is indicated using a blue arrow. The middle panel shows the quality of the individual fits. Each scatter point represents the fit of the PKPD

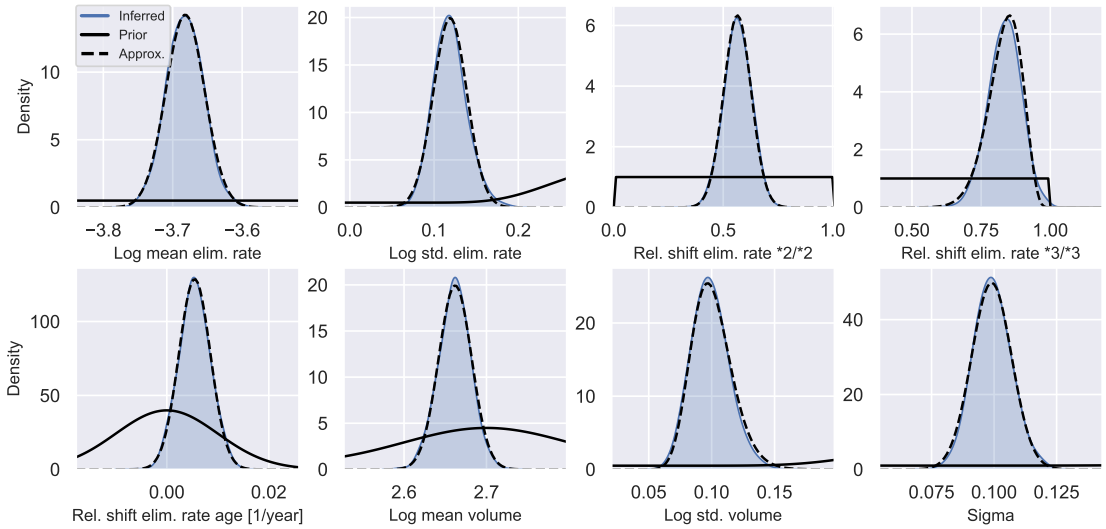


Figure B.10: Marginal posteriors after calibration to trial phase I data. The figure shows the marginal posteriors of the model parameters after calibrating Hamberg et al's model to the phase I trial data. The priors are illustrated in black (solid lines) and the inferred posteriors are illustrated in blue using kernel density estimates. The approximations of the posteriors are illustrated by black dashed lines.

model, $\bar{y}(t, r, \psi)$, to a measurement. To illustrate the uncertainty of the model fits, we sample 1000 sets of model parameters for each individual from the posterior distribution (Equation B.41) and simulate the PKPD model output for each parameter set, e.g. for individual 1, we sample from $p(\psi_{11}|\mathcal{D}_1) = \int \mathrm{d}\theta\, \mathrm{d}\psi_{12} \ldots \mathrm{d}\psi_{160}\, p(\theta, \Psi_1|\mathcal{D}_1)$ and solve $\bar{y}(t, r_{11}, \psi_{11})$ for each sample of ψ_{11}. The median PKPD model output for each fit is illustrated by a black scatter point. The uncertainty of the fit is illustrated using an error bar, representing the 5th to 95th percentile interval of the output distribution. Model outputs identical to measurements lie on the diagonal line (blue). The right panel shows the quality of the fits on a population level. This time PKPD model outputs are simulated using the typical parameters in the population, e.g. for individual 1, we sample 1000 sets of population parameters from the posterior distribution, $p(\theta|\mathcal{D}_1)$, and use Equation B.36 and the the individual's covariates, χ_{11}, to calculate the typical parameters, (μ_{k_e}, μ_v). For each set of typical parameters, we simulate the PKPD model output. The median model output and percentile interval are determined analogously to the middle panel. Individuals with different CYP2C9 genotypes are highlighted in different colours: *1*1 (blue); *1*2 (orange); *1*3 (green); *2*2 (red); *2*3 (purple);

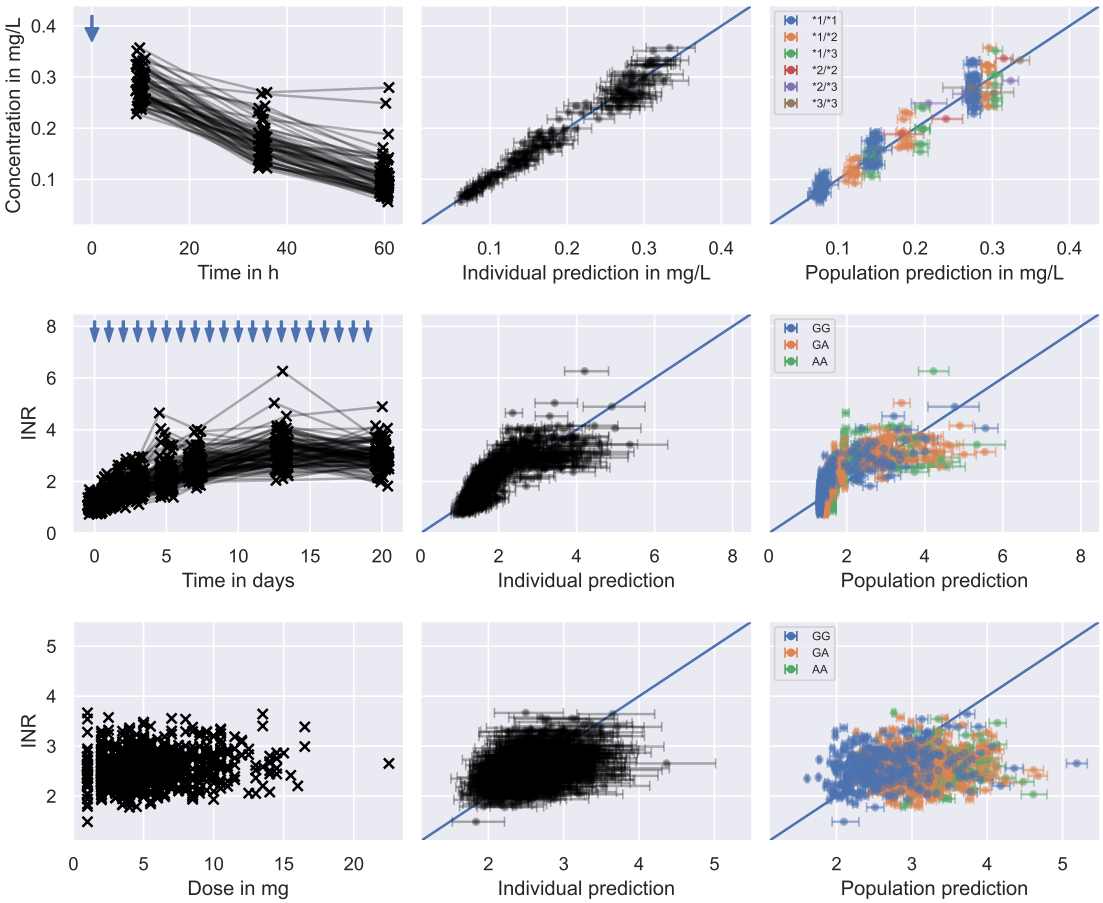


Figure B.11: Goodness of fit. The figure shows the goodness-of-fit to the trial data: phase I (top row); phase II (middle row); phase III (bottom row). In the columns shows the measurements used for the calibration of the model. Nominal administration times are indicated using blue arrows. The middle columns show the individual-level fits to the measurements. Each scatter point represents the fit of the PKPD model, $\bar{y}(t, r, \psi)$, to a measurement. We sample 1000 sets of model parameters for each individual from the posterior distribution and simulate the PKPD model output for each parameter set, e.g. for individual 1 in trial I, we sample from $p(\psi_{1\,1}|\mathcal{D}_1) = \int \mathrm{d}\theta\, \mathrm{d}\psi_{1\,2} \ldots \mathrm{d}\psi_{1\,60}\, p(\theta, \Psi_1|\mathcal{D}_1)$ and solve $\bar{y}(t, r_{1\,1}, \psi_{1\,1})$ for each sample of $\psi_{1\,1}$. The median PKPD model output for each fit is illustrated by the black scatter point. The error bars illustrate the 5th to 95th percentile interval of the output distribution. Model outputs identical to measurements lie on the diagonal line (blue). The right columns show the population-level fits to the measurements. We simulate the PKPD model outputs using 1000 sets of the typical parameters sampled from the posterior, e.g. for individual 1 in trial 1, we sample 1000 sets of population parameters from the posterior distribution, $p(\theta|\mathcal{D}_1)$, and use Equation B.36 and the the individual's covariates, $\chi_{1\,1}$, to calculate the typical parameters. For each set of typical parameters, we simulate the PKPD model output. The median model output is illustrated by the scatter point and the 5th to 95th percentile interval are illustrated by the error bars.

*3*3 (brown). The age differences between individuals are not highlighted in the figure, but taken into account in the computation of the typical parameters.

The middle panel shows that the individual-level residuals are symmetrically distributed around the line of identity (blue), giving no indication of model misspecification. The residual error also seems to increase with the model output, which supports the lognormal error model choice. Similarly the residuals with respect to the typical treatment response are symmetrically distributed around the line of identity (see right panel), showing that the typical values of the model parameters are a good representation of the typical treatment response in the population.

Calibration to trial phase II data

After the calibration to the phase I data, we continue the model parameter calibration using the data from trial phase II. The data contains INR measurements of 100 individuals (see middle panel in Figure 3.5). Denoting the data as $\mathcal{D}_2 = (\mathcal{D}_{2\,1}, \ldots, \mathcal{D}_{2\,100})$, where $\mathcal{D}_{2i}$ denotes the data of individual i, we define the hierarchical posterior for the population model parameters analogously to Equation B.41 using Bayes rule

$$\begin{aligned} \log p(\theta, \Psi_2 | \mathcal{D}_1, \mathcal{D}_2) \quad &= \quad \sum_i \log p(\mathcal{D}_{2i} | \psi_{2i}) + \sum_i \log p(\psi_{2i} | \theta, \chi_{2i}) + \log p(\theta | \mathcal{D}_1) \\ &\quad + \text{constant}. \end{aligned} \tag{B.43}$$

$\Psi_2 = (\psi_{2\,1}, \ldots, \psi_{2\,100})$ denotes the individual-level parameters of the individuals in the trial. The likelihood of the individual-level parameters is defined using the measurement distribution of the INR measurements defined in subsection B.4.3. The scale parameter of the lognormal measurement distribution, σ, is pooled across individuals.

The prior distribution of the model parameters is informed by the inference results from the previous section. We set the marginal prior distributions of the PK-related model parameters to the marginal posterior distributions from the phase I calibration in Figure B.10. For computational efficiency, we approximate the inferred distributions (blue) using parametrised distributions (dashed lines). We calibrate the parameters of the distributions manually, using empirical means and standard deviations of the samples

as a staring point. The resulting prior distributions for the PK parameters are

$$\begin{aligned}
\log k_{*1} &\sim \mathcal{N}(-3.682, 0.028) \\
\sigma_{k_e} &\sim \mathcal{N}(0.119, 0.020) \\
\frac{k_{*1} - k_{*2}}{k_{*2}} &\sim \mathcal{N}(0.565, 0.063) \\
\frac{k_{*1} - k_{*3}}{k_{*2}} &\sim \text{Beta}(30, 6) \\
r_{age} &\sim \mathcal{N}(0.00546, 0.0031) \\
\log \mu_v &\sim \mathcal{N}(2.662, 0.020) \\
\sigma_v &\sim \text{LN}(-2.31, 0.16).
\end{aligned}$$

The prior information of the remaining parameters is assumed to be minimal. We define weakly informative marginal distributions only to ensure that parameters assume values in meaningful biological ranges

$$\begin{aligned}
\log \bar{y}_{0,G} &\sim \mathcal{N}(0, 0.5) \\
\sigma_{\bar{y}_0} &\sim \text{LN}(-1, 2) \\
\bar{y}_{0,A}/\bar{y}_{0,G} &\sim \text{LN}(0, 1) \\
\log c_G &\sim \mathcal{N}(1.41, 0.5) \\
\sigma_{c_{50}} &\sim \text{LN}(-1, 0.5) \\
\frac{c_G - c_A}{c_G} &\sim \mathcal{U}(0, 1) \\
\mu_{k_1} &\sim \mathcal{N}(0.2, 0.05) \\
\mu_{k_2} &\sim \mathcal{N}(0.02, 0.005) \\
\mu_\sigma &\sim \text{LN}(-1, 0.3).
\end{aligned}$$

We fix the potency of warfarin to $\gamma = 1.15$, following (Hamberg, Wadelius, et al. 2010).

We infer the posterior following the same approach as in the previous section (see section B.4.4), using NUTS and 1500 iterations. All marginal posteriors have $\hat{R} \leq 1.01$. We show the inferred posterior distribution in Figure B.12. The results of the inference, as well as data and code to reproduce the results are hosted on GitHub (https://github.com/DavAug/mipd-warfarin).

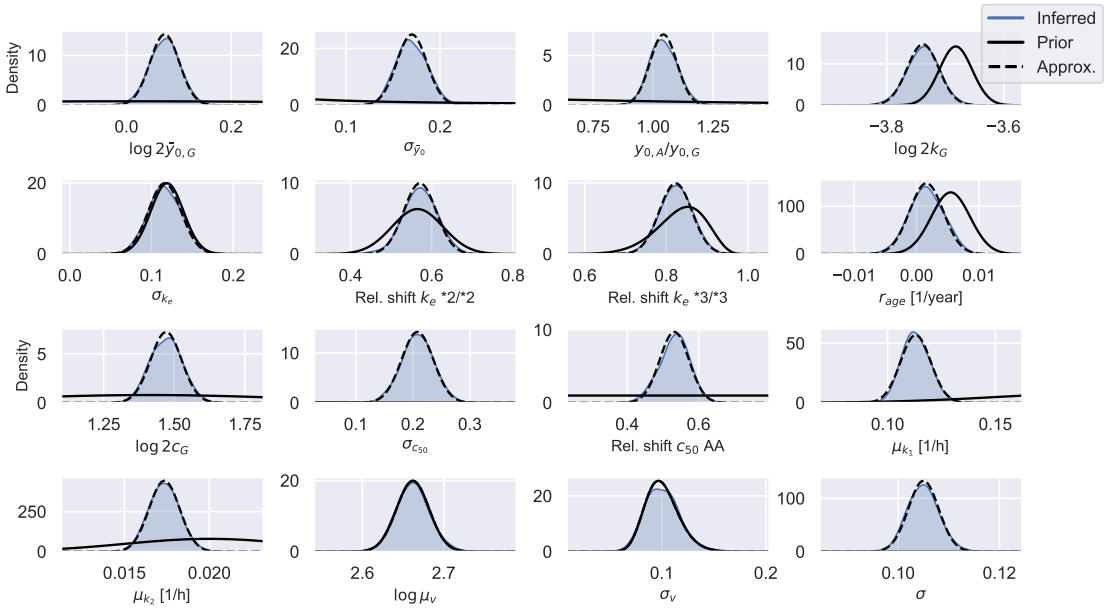


Figure B.12: Marginal posteriors after calibration to trial phase II data. The figure shows the marginal posteriors of the model parameters after calibrating the modified Hamberg et al model to the phase II trial data. The priors are illustrated in black (solid lines) and the inferred posteriors are illustrated in blue using kernel density estimates. The approximations of the posteriors are illustrated by black dashed lines.

Figure B.12 shows the results of the calibration. The marginal prior distributions are illustrated in black (solid lines) and the posterior distributions are illustrated in blue. Approximations of the posterior distributions, used in section B.4.4, are illustrated in black (dashed lines).

The goodness-of-fit is visualised in the middle row of Figure B.11. The residuals of the individual-level fits in the middle column indicate no sign of model misspecification. The asymmetric distribution of the residuals in the right column suggests however that the typical parameter values do not represent the typical treatment response. In particular, predicted typical INRs in the therapeutic window appear to underestimate the measured INRs, as indicated by the scew of the residual distribution towards values above the line of identity for INRs between 2 and 3. This mismatch between typical model parameters and typical treatment responses does not necessarily imply a misspecification of the population model, but may simply be the consequence of the nonlinear relationship between the PKPD model output and the model parameters.

Calibration to trial phase III data

We conclude the inference by calibrating the model parameters to the data from trial phase III. The data contains INR measurements of 1000 individuals under maintenance treatment (see right panel in Figure 3.5). Denoting the data with $\mathcal{D}_3 = (\mathcal{D}_{3\,1}, \ldots, \mathcal{D}_{3\,1000})$, where $\mathcal{D}_{3i}$ denotes the data of individual i, we define the hierarchical posterior for the population model parameters using Bayes rule

$$\log p(\theta, \Psi_3 | \mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3) = \tag{B.44}$$
$$\sum_i \log p(\mathcal{D}_{3i} | \psi_{3i}) + \sum_i \log p(\psi_{3i} | \theta, \chi_{3i}) + \log p(\theta | \mathcal{D}_1, \mathcal{D}_2) + \text{constant}.$$

$\Psi_3 = (\psi_{3\,1}, \ldots, \psi_{3\,1000})$ denotes the individual-level parameters of the individuals in the trial. For efficiency, we define the INR measurement distribution using the analytically-computed steady state of the PKPD model.

In steady state, all rate of change equations of the PKPD model in Section subsection B.4.2 vanish. For a constant dose rate, $\bar{r} =$ constant, we can use this to derive an analytic expression for the steady state of the warfarin concentration

$$c_{ss} = \frac{\bar{r}}{v k_e}. \tag{B.45}$$

Similarly, we can derive a steady state expression for the INR

$$\bar{y}_{ss} = \bar{y}_0 + \bar{y}_{max}\, \kappa \frac{c_{ss}^\gamma}{c_{50}^\gamma + c_{ss}^\gamma}. \tag{B.46}$$

Thus, we can efficiently simulate the INR under maintenance treatment using these expressions, assuming a dose rate that is constant in time.

However, the daily warfarin doses administered during the trial do not give rise to constant dose rates. The steady state expressions can therefore only approximate the treatment response. The exact warfarin concentrations fluctuate around the steady state value computed with the average daily dose rate, $\bar{r} = d^*/24\,\text{h}$. Despite these fluctuations in the warfarin concentration, the approximation error for the INR simulations can be neglected for the inference, as the treatment response delay dampens the effect of the fluctuations on the INR (see Figure 2.6), resulting in an effectively constant INR value under maintenance treatment.

For the inference, we fix the scale parameters of the measurement distribution, the elimination rate distribution, the EC50 distribution and the volume of distribution distribution to the typical values from the phase II calibration: $\mu_\sigma = 0.105$; $\sigma_{k_e} = 0.116$; $\sigma_{c_{50}} = 0.208$; $\sigma_v = 0.101$ (see Figure B.12). This is because snapshot measurements make it difficult to distinguish contributions from inter-individual variability and measurement noise to the observed treatment response variability (Augustin, Lambert, K. Wang, et al. 2023). The maintenance treatment data also makes it difficult to distinguish variability in the treatment response from variability in the baseline INR. We therefore also fix the baseline INR parameters to the typical values from Figure B.12: $\log 2\bar{y}_{0,G} = 0.073$; $\sigma_{\bar{y}_0} = 0.170$; $\bar{y}_{0,A}/\bar{y}_{0,G} = 1.043$. This leaves 7 parameters for the calibration: $(k_{*1}, k_{*2}, k_{*3}, r_{age}, c_G, c_A, \mu_v)$.

The prior distribution of the model parameters is set to the marginal posterior distributions from the phase II calibration in Figure B.12

$$\begin{aligned}
\log 2k_{*1} &\sim \mathcal{N}(-3.787, 0.027) \\
\frac{k_{*1} - k_{*2}}{k_{*2}} &\sim \mathcal{N}(0.571, 0.040) \\
\frac{k_{*1} - k_{*3}}{k_{*2}} &\sim \mathcal{N}(0.823, 0.040) \\
r_{age} &\sim \mathcal{N}(0.00157, 0.0027) \\
\log 2c_G &\sim \mathcal{N}(1.471, 0.055) \\
\frac{c_G - c_A}{c_G} &\sim \mathcal{N}(0.532, 0.041) \\
\log \mu_v &\sim \mathcal{N}(2.662, 0.020).
\end{aligned}$$

We infer the posterior following the same approach as in the previous sections (see section B.4.4), using NUTS and 1500 iterations. All marginal posteriors have $\hat{R} \leq 1.01$. We show the inferred posterior distribution in Figure B.13. The results of the inference, as well as data and code to reproduce the results are hosted on GitHub (https://github.com/DavAug/mipd-warfarin).

The figure shows the final result of the model calibration. The marginal prior distributions are illustrated in black (solid lines) and the posterior distributions are

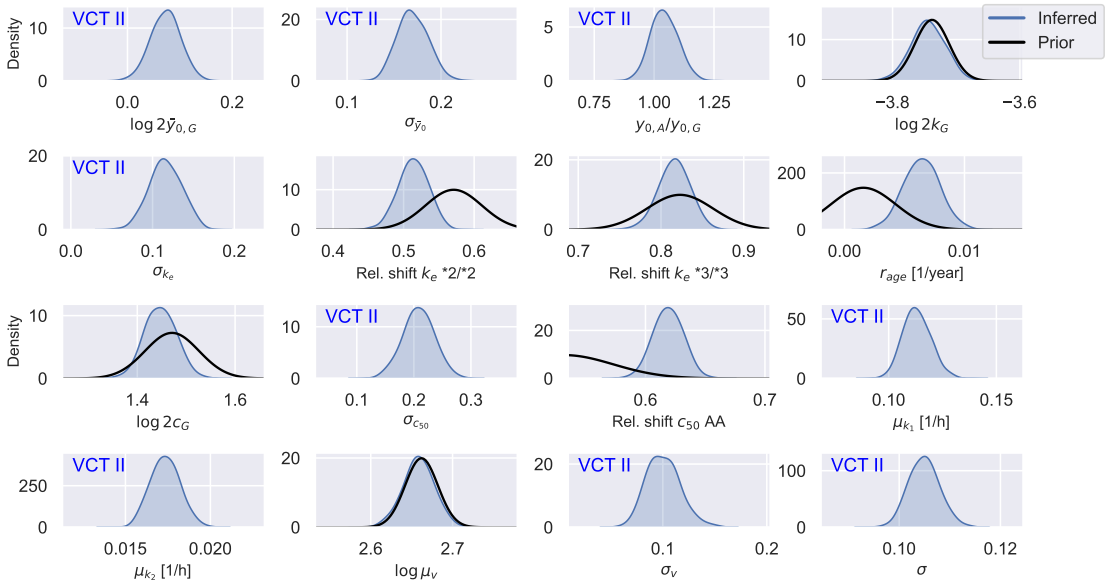


Figure B.13: Marginal posteriors after calibration to trial phase III data. The figure shows the marginal posteriors of the model parameters after calibrating the modified Hamberg et al model to the phase III trial data. The priors are illustrated in black (solid lines) and the inferred posteriors are illustrated in blue using kernel density estimates. The approximations of the posteriors are illustrated by black dashed lines.

illustrated in blue. Posteriors not updated since the calibration to the trial phase II data are labelled with (VCT II).

The goodness-of-fit is visualised in the bottom panel of Figure B.11. Similar to the inference results from trial phase II (see middle row), the residuals of the individual-level fits in the middle column indicate no sign of model misspecification, while the asymmetric distribution of the residuals in the right column suggests that the typical parameter values do not represent the typical treatment response.

Prediction of parameter priors for MIPD

We predict individualised prior distributions used in Equation B.39 for MIPD (see subsection B.4.3) using the calibrated model. As illustrated in Fig Figure B.13, the result of the calibration is a distribution of population-level parameters, $p(\theta|\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3)$, compatible with the clinical trial data. The spread of the distribution indicates that there is residual uncertainty about the model parameters.

To account for this residual uncertainty, we average the population model, $p(\psi|\theta, \chi)$,

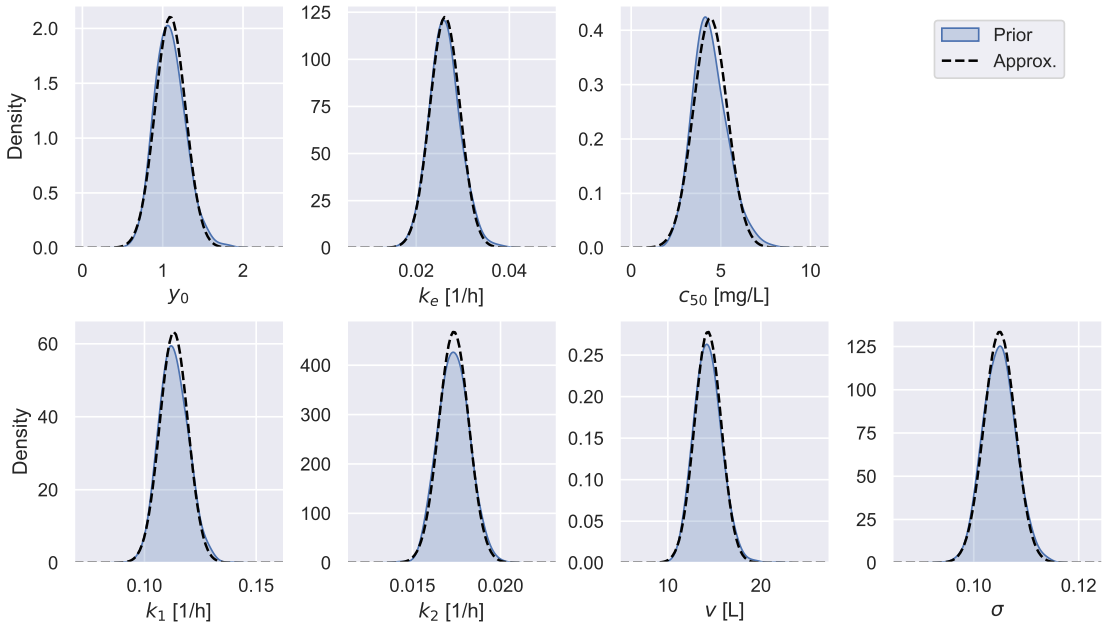


Figure B.14: Individualised prior distribution. The figure shows the prior distribution used during the MIPD trial for individual with the covariates $\chi = (GG, *1*1, 71)$. The inferred prior distributions, $p(\psi|\chi, \mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3)$, are visualised in blue. Gaussian approximations of the prior distributions are illustrated in black.

over the posterior distribution

$$p(\psi|\chi, \mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3) = \mathbb{E}_{\theta|\mathcal{D}_1,\mathcal{D}_2,\mathcal{D}_3}\Big[p(\psi|\theta, \chi)\Big]. \tag{B.47}$$

In the Bayesian inference literature, $p(\psi|\chi, \mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3)$, may be referred to as the posterior predictive distribution. An example individualised prior for an individual with the covariates $\chi = (GG, *1*1, 71)$ is illustrated in Figure B.14.

The figure shows the individualised prior distribution constructed from 1000 samples of $p(\psi|\chi, \mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3)$ (blue). Each ψ is generated by first sampling a set of population parameters from $p(\theta|\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3)$ and subsequently sampling the ψ from $p(\psi|\theta, \chi)$ using the sampled θ and the covariates of the individual. For more efficient inference during the MIPD trial, we again approximate the probability densities using parametrised distributions. All marginal distributions are well approximated using Gaussian distributions (black). The prior distributions of all individuals in the MIPD trial, as well as data and code to reproduce the results are hosted on GitHub (https://github.com/DavAug/mipd-warfarin).

B.5 Clinical trial simulation

Algorithm B.1: Clinical trial phase I.

Input : 1. Mechanistic model: $\bar{y}(t, r, \psi)$;
2. Population model: $p(\psi|\theta, \chi)$;
3. Execution model: $p(\Delta t|\tau)$;
4. Measurement model: $p(y|\bar{y}, \psi)$;
5. Covariates of virtual patients: $\chi_1, \ldots, \chi_{60}$;
6. Nominal dosing regimen: $r(t) = d\,\Theta(t)\,\Theta(\Delta - t)/\Delta$ with $d = 10\,\text{mg}$, $\Delta = 0.01\,\text{h}$;
7. Nominal measurement times: $t_1 = 10\,\text{h}$, $t_2 = 35\,\text{h}$, $t_3 = 60\,\text{h}$

Output : Virtual clinical trial data $\mathcal{D}$.

$\mathcal{D} = [\,]$
for $i \leftarrow 1$ **to** 60 **do**
  $\psi_i \sim p(\psi|\theta, \chi_i)$ // Sample parameters of virtual patient
  $\Delta t_0 \sim p(\Delta t|\tau)$ // Sample delay of administration
  $r' = d\,\Theta(t - \Delta t_0)\,\Theta(\Delta t_0 + \Delta - t)/\Delta$ // Update dosing regimen
  $\mathcal{D}_i = [\,]$
  for $j \leftarrow 1$ **to** 3 **do**
    $\Delta t_j \sim p(\Delta t|\tau)$ // Sample delay of measurement
    $\bar{y}_{ij} = \bar{y}(t_j + \Delta t_j, r', \psi_i)$ // Simulate concentration
    $y_{ij} \sim p(y|\bar{y}_{ij}, \psi_i)$ // Sample measurement
    $\mathcal{D}_i$.append$((y_{ij}, t_j))$ // Store measurement with nominal time
  end for
  $\mathcal{D}$.append$((\mathcal{D}_i, \chi_i, r))$ // Store data with nominal dosing regimen
end for
return $\mathcal{D}$

Algorithm B.2: Clinical trial phase II.

Input : 1. Mechanistic model: $\bar{y}(t, r, \psi)$;
2. Population model: $p(\psi|\theta, \chi)$;
3. Inter-occasion model: $p(\eta|t)$;
4. Execution model: $p(\Delta t|\tau)$;
5. Measurement model: $p(y|\bar{y}, \psi)$;
6. Covariates of virtual patients: $\chi_1, \ldots, \chi_{100}$;
7. Nominal measurement times: $t_1 = 0\,\text{h}$, $t_2 = 24\,\text{h}$, $t_3 = 48\,\text{h}$, $t_4 = 72\,\text{h}$, $t_5 = 96\,\text{h}$, $t_6 = 120\,\text{h}$, $t_7 = 144\,\text{h}$, $t_8 = 168\,\text{h}$

Output : Simulated clinical trial data $\mathcal{D}$.

$\mathcal{D} = [\,]$
measurement_times $= (t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8)$
dose_adjustment_times $= (t_4, t_5, t_6, t_7)$
for $i \leftarrow 1$ **to** 100 **do**
  $\psi_i \sim p(\psi|\theta, \chi_i)$ // Sample parameters of simulated patient
  $r'_i = r_i$ // Initialise dosing regimen
  $\mathcal{D}_i = [\,]$
  for $j \leftarrow 1$ **to** 20 **do**
    $\eta_j \sim p(\eta|t_j)$ // Sample inter-occasion variability
    $\psi_{ij} = \psi_i\,\eta_j$ // Update model parameters
    $\Delta t_j \sim p(\Delta t|\tau)$ // Sample random delay
    if $t_j \in$ *measurement_times* **then**
      $\bar{y}_{ij} = \bar{y}(\Delta t_j, r'_i, \psi_{ij})$ // Simulate INR starting from t_j (see l. 25)
      $y_{ij} \sim p(y|\bar{y}_{ij}, \psi_{ij})$ // Sample measurement
      $\mathcal{D}_i$.append$((y_{ij}, t_j))$ // Store measurement with nominal time
    end if
    if $j \leq 3$ **then**
      $d'_{ij} = 10 - 2.5(j-1)$ // Get dose 1, 2 or 3
    else if $\tilde{t}_j \in$ *dose_adjustment_times* **then**
      $d'_{ij} =$ updateDose(y_{ij}) // Adjust dose using heuristic
    end if
    $r_i = r_i$.update$(d_{ij} = d'_{ij})$ // Update nominal dosing regimen
    $r'_i = d'_{ij}\,\Theta(t - \Delta t_j)\,\Theta(\Delta t_j + \Delta - t)/\Delta$ // Define actual dosing regimen
    $\psi_0 = \bar{y}(t = 24, r'_i, \psi_{ij})$.getState() // Simulate state of model at next day
    $\psi_i = \psi_i$.setState(ψ_0) // Update state to shift reference time to t_j
  end for
  $\mathcal{D}$.append$((\mathcal{D}_i, \chi_i, r))$ // Store data with nominal dosing regimen
end for
return $\mathcal{D}$

Algorithm B.3: Clinical trial phase III.

Input : 1. Mechanistic model: $\bar{y}(t, r, \psi)$;
2. Population model: $p(\psi|\theta, \chi)$;
3. Inter-occasion model: $p(\eta|t)$;
4. Execution model: $p(\Delta t|\tau)$;
5. Measurement model: $p(y|\bar{y}, \psi)$;
6. Covariates of virtual patients: $\chi_1, \ldots, \chi_{1000}$;
7. Nominal regimen: $r_i(t) = \sum_{j=1}^{55} d_{ij}\, \Theta(t - \tilde{t}_j)\, \Theta(\tilde{t}_j + \Delta - t)/\Delta$ with $\tilde{t}_j = 24(j-1)\,\text{h}$;
8. Nominal measurement times: $\{t_j = 24(j-1)\,\text{h}, \text{for}\, j = 1, \ldots, 55\}$

Output : Simulated clinical trial data $\mathcal{D}$.

$\mathcal{D} = [\,]$
measurement_times $= (t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12})$
dose_adjustment_times $= (t_4, t_5, t_6, t_7, t_8, t_9)$
for $i \leftarrow 1$ **to** 1000 **do**
  $\psi_i \sim p(\psi|\theta, \chi_i)$ // Sample parameters of virtual patient
  $r'_i = r_i$ // Initialise dosing regimen
  $\mathcal{D}_i = [\,]$
  for $j \leftarrow 1$ **to** 55 **do**
    $\eta_j \sim p(\eta|t_j)$ // Sample inter-occasion variability
    $\psi_{ij} = \psi_i\, \eta_j$ // Update model parameters
    $\Delta t_j \sim p(\Delta t|\tau)$ // Sample measurement and administration delay
    if $t_j \in$ *measurement_times* **then**
      $\bar{y}_{ij} = \bar{y}(\Delta t_j, r'_i, \psi_{ij})$ // Simulate INR starting from t_j (see l. 25)
      $y_{ij} \sim p(y|\bar{y}_{ij}, \psi_{ij})$ // Sample measurement
    end if
    if $j \leq 3$ **then**
      $d'_{ij} = 10 - 2.5(j-1)$ // Get dose 1, 2 or 3
    else if $\tilde{t}_j \in$ *dose_adjustment_times* **then**
      $d'_{ij} = \text{updateDose}(y_{ij})$ // Adjust dose using heuristic
    end if
    $r_i = r_i.\text{update}(d_{ij} = d'_{ij})$ // Update nominal dosing regimen
    $r'_i = d'_{ij}\, \Theta(t - \Delta t_j)\, \Theta(\Delta t_j + \Delta - t)/\Delta$ // Define actual dosing regimen
    $\psi_0 = \bar{y}(t = 24, r'_i, \psi_{ij}).\text{getState}()$ // Simulate state of model at next day
    $\psi_i = \psi_i.\text{setState}(\psi_0)$ // Update model state to shift reference time to t_j
  end for
  $\mathcal{D}_i.\text{append}((y_{i\,12}, t_{12}))$ // Store final measurement with nominal time
  $\mathcal{D}.\text{append}((\mathcal{D}_i, \chi_i, r))$ // Store data with nominal dosing regimen
end for
return $\mathcal{D}$

Algorithm B.4: MIPD trial.

Input : 1. MIPD model: $d(\mathcal{D}, \chi, r)$;
2. Mechanistic model: $\bar{y}(t, r, \psi)$;
3. Population model: $p(\psi|\theta, \chi)$;
4. Inter-occasion model: $p(\eta|t)$;
5. Execution model: $p(\Delta t|\tau)$;
6. Measurement model: $p(y|\bar{y}, \psi)$;
7. Covariates of virtual patients: $\chi_1, \ldots, \chi_{1000}$;
8. Nominal regimen: $r_i(t) = \sum_{j=1}^{20} d_{ij}\, \Theta(t - \tilde{t}_j)\, \Theta(\tilde{t}_j + \Delta - t)/\Delta$ with $\tilde{t}_j = 24(j-1)$ h

Output : Simulated clinical trial data $\mathcal{D}$.

1 $\mathcal{D} = [\,]$
2 **for** $i \leftarrow 1$ **to** 1000 **do**
3 $\quad \psi_i \sim p(\psi|\theta, \chi_i)$ // Sample parameters of virtual patient
4 $\quad r'_i = r_i$ // Initialise dosing regimen
5 $\quad \mathcal{D}_i = [\,]$
6 $\quad$ **for** $j \leftarrow 1$ **to** 20 **do**
7 $\quad\quad \eta_j \sim p(\eta|t_j)$ // Sample inter-occasion variability
8 $\quad\quad \psi_{ij} = \psi_i\, \eta_j$ // Update model parameters
9 $\quad\quad \Delta t_j \sim p(\Delta t|\tau)$ // Sample measurement and administration delay
10
11 $\quad\quad \bar{y}_{ij} = \bar{y}(\Delta t_j, r'_i, \psi_{ij})$ // Simulate INR starting from t_j (see l. 25)
12 $\quad\quad y_{ij} \sim p(y|\bar{y}_{ij}, \psi_{ij})$ // Sample measurement
13 $\quad\quad \mathcal{D}_i$.append$((y_{ij}, t_j))$ // Store measurement with nominal time
14
15 $\quad\quad d'_{ij} = d(\mathcal{D}_i, \chi_i, r_i)$ // Predict next dose using MIPD model
16 $\quad\quad r_i = r_i$.update$(d_{ij} = d'_{ij})$ // Update nominal dosing regimen
17 $\quad\quad r'_i = d'_{ij}\, \Theta(t - \Delta t_j)\, \Theta(\Delta t_j + \Delta - t)/\Delta$ // Define actual dosing regimen
18
19 $\quad\quad \psi_0 = \bar{y}(t = 24, r'_i, \psi_{ij})$.getState() // Simulate state of model at next day
20 $\quad\quad \psi_i = \psi_i$.setState$(\psi_0)$ // Update state to shift reference time to t_j
21 $\quad$ **end for**
22 $\quad \mathcal{D}$.append$((\mathcal{D}_i, \chi_i, r))$ // Store data with nominal dosing regimen
23 **end for**
24 **return** $\mathcal{D}$

B.6 Supplementary tables

	Phase I	Phase II	Phase III
Number of patients	60	100	1000
VKORC1 GG	N/A	36 %	36.5 %
VKORC1 GA	N/A	49 %	48.5 %
VKORC1 AA	N/A	15 %	15 %
CYPC29 *1*1	62 %	63 %	66.1 %
CYPC29 *1*2	20 %	19 %	18.4 %
CYPC29 *1*3	13 %	13 %	12.3 %
CYPC29 *2*2	2 %	2 %	1.4 %
CYPC29 *2*3	2 %	2 %	1.2 %
CYPC29 *3*3	1 %	1 %	0.6 %
Age	69 (58 - 79)	65 (56 - 76)	50 (40 - 64)

Table B.4: Demographics of the simulated clinical trials. The age distribution of each cohort is represented by the median and the 5th to 95th percentile interval.

C

Supporting information for Chapter 4

Contents

C.1 Optimised implementation of filter inference 306
C.2 Estimation of early cancer growth model parameters 307
C.3 Estimation of variance estimation error 307
C.4 IIV-noise distinguishability 308
C.5 Estimating the average of a population distribution over the posterior distribution 310
C.6 Estimation of EGF pathway model parameters 311
C.7 Estimation of the evaluation time 312
C.8 Hyperparameter tuning of MH algorithm 313
C.9 Equivalence of filter inference with a Gaussian filter and ABC based on the mean and variance 314
C.10 Equivalence of filter inference with identifiable summary statistics-based filters and ABC based on the same summary statistics . . . 315
C.11 Supplementary tables and figures 316

C.1 Optimised implementation of filter inference

The implementation of filter inference can be optimised such that it solves the time series model only once for each simulated individual by leveraging the fact that numerical differential equation solvers can efficiently evaluate a system of differential at any number of time points. The optimised implementation, therefore, solves the time series model for each individual only once up front, computing $\bar{y}(\psi_s, t)$ for all measured time points

in one time series model solve. This removes the unnecessary costs of integrating the time series model for each measured time point separately.

C.2 Estimation of early cancer growth model parameters

The parameters of the early cancer growth model are estimated with pints' implementation of NUTS, unless otherwise specified. MCMC chains are run for 1500 iterations, where the first 500 iterations are used for calibration. The convergence is assessed using the $\hat{R}$-statistic from 3 MCMC chains, initialised at randomly sampled points from the prior distribution. The prior distribution for all inference runs is

$$\begin{aligned} \mu_{y_0} &\sim \mathcal{N}(9, 9) \\ \sigma_{y_0} &\sim \mathrm{LN}(2, 1) \\ \mu_{\lambda} &\sim \mathcal{N}(5, 9) \\ \sigma_{\lambda} &\sim \mathrm{LN}(0.5, 1) \\ \mu_{\sigma} &\sim \mathrm{LN}(0.8, 0.1). \end{aligned}$$

Detailed information on the convergence can be found in Table C.1.

	90 IDs (NLME)	90 IDs	270 IDs	810 IDs	2430 IDs
$\hat{R}$ of μ_{y_0}	1.00	1.00	1.00	1.00	1.00
$\hat{R}$ of σ_{y_0}	1.00	1.00	1.00	1.00	1.00
$\hat{R}$ of μ_{λ}	1.00	1.00	1.00	1.00	1.00
$\hat{R}$ of σ_{λ}	1.00	1.00	1.00	1.00	1.00
$\hat{R}$ of μ_{σ}	1.00	1.00	1.00	1.00	1.00
# divergences	0	0	0	0	0

Table C.1: Convergence statistics of MCMC chains during the parameter estimation of the early cancer growth model.

C.3 Estimation of variance estimation error

We estimate the variance at time point t_j using

$$\hat{\sigma}_j^2 = \frac{1}{n_j - 1} \sum_{i=1}^{n_j} (y_{ij} - \hat{\mu}_j)^2 ,$$

where n_j denotes the number of measurements at t_j and $\hat{\mu}_j = \sum_i y_{ij}/n_j$ denotes their mean. Assuming that y_{ij} are drawn from a Gaussian distribution $\mathcal{N}(y|\mu_j, \sigma_j^2)$ with mean μ_j and variance σ_j^2, the normalised variance estimator follows a χ^2-distribution with $n_j - 1$ degrees of freedom

$$\frac{n_j - 1}{\sigma_j^2}\hat{\sigma}_j^2 \sim \chi^2_{n_j-1}.$$

The variance of a $\chi^2_{n_j-1}$ random variable is $2(n_j - 1)$. As a result, we can estimate the standard deviation of $\hat{\sigma}_j^2$ using

$$\sqrt{\mathrm{Var}\left[\hat{\sigma}_j^2\right]} = \sigma_j^2\sqrt{\frac{2}{n_j - 1}} \approx \hat{\sigma}_j^2\sqrt{\frac{2}{n_j - 1}}.$$

At $t = 0$, the measurement distribution in Figure 4.3 happens to be Gaussian, and the above estimate is unbiased.

C.4 IIV-noise distinguishability

To understand the IIV-noise distinguishability when estimating parameters from snapshot measurements using NLME inference and filter inference, we generate snapshot datasets from the early cancer growth model analogously to subsection 4.3.1, but with increased numbers of measured individuals per time point. In particular, we generate datasets with $N = 90$, $N = 270$, $N = 810$ and $N = 2430$ individuals and estimate the parameters using NLME inference and filter inference with a Gaussian filter and $S = 100$ simulated individuals. The inference results are depicted in Figure C.1.

The figure shows that the uncertainty of the population mean estimates and the population standard deviation estimates decreases as the number of measured individuals increases, demonstrating that the estimation of IIV becomes more accurate as more individuals are measured. However, it also shows that for most of the datasets the posterior distributions of the noise parameter, μ_σ, do not substantially differ from the prior distribution. As argued in subsection 4.3.1, this is because the prior distribution, $p(\mu_\sigma)$, focuses on values that give rise to noise magnitudes that are all compatible with the variability of the measurements, such that variability contributions from IIV and noise can only be distinguished if they give rise to different shapes of the measurement

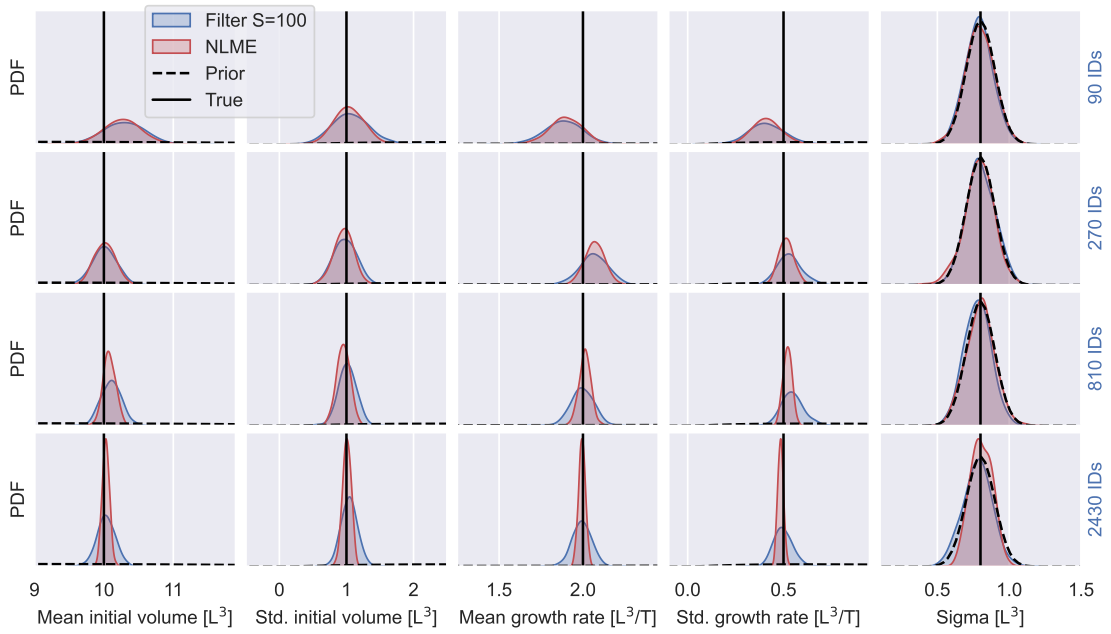


Figure C.1: IIV-noise identifiability of early cancer growth model I. The figure shows inference results of the early cancer model using NLME inference and filter inference with a Gaussian filter and $S = 100$ simulated individuals. The parameters are estimated from snapshot datasets with measurements from $N = 90$, $N = 270$, $N = 810$, $N = 2430$ individuals. The measurements are generated for 6 time points between 0 and 0.6 as described in subsection 4.3.1. The first row shows the posterior distributions inferred from measurements of $N = 90$ individuals. The second, third and fourth row analogously show the results for $N = 270$, $N = 810$ and $N = 2430$. The filter posteriors are illustrated in blue and the NLME posteriors are illustrated in red. The prior distributions are illustrated by black dashed lines and the data-generating parameter values are depicted by solid black lines.

distribution, $p(y|\theta, t)$, and sufficiently many measurements are available to resolve such distributional differences. Figure C.2 shows the measurement distribution for different samples from the posterior distribution in the top row of Fig Figure C.1, demonstrating that the shape of the distribution does change for different contributions from IIV and noise, while the variance of the distribution remains the same.

Such distributional differences cannot be resolved by $N = 90$, $N = 270$ and $N = 810$ snapshot measurements, as demonstrated by columns 1-3 in Fig C.1. However, using snapshot measurements from $N = 2430$, NLME inference begins to have sufficient statistical power to resolve differences between IIV and noise, resulting in an update of $p(\mu_\sigma|\mathcal{D})$ relative to the prior (see bottom right corner of Figure C.1). Filter inference with a Gaussian filter and $S = 100$ simulated individuals is not able to distinguish IIV

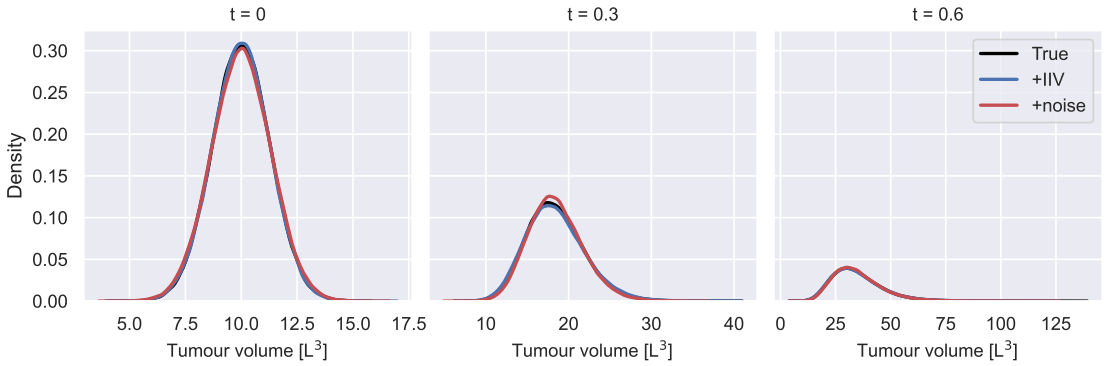


Figure C.2: Early cancer growth measurement distribution. The figure is an extension of Fig C.1 and shows the measurement distribution $p(y|\theta, t)$ of the early cancer growth model at $t = 0$, $t = 0.3$ and $t = 0.6$ for three different sets of parameter values. The measurement distribution corresponding to the data-generating parameters $\theta = (\mu_{y_0}, \sigma_{y_0}, \mu_\lambda, \sigma_\lambda, \mu_\sigma) = (10, 1, 2, 0.5, 0.8)$ is shown in black. A measurement distribution that overestimates the IIV contributions to the measurement variability, $\theta = (10, 1.195, 2, 0.5, 0.47)$, is illustrated in blue, and a measurement distribution that overestimates the noise contributions to the measurement variability, $\theta = (10, 0.7, 2, 0.5, 1.127)$, is illustrated in red.

and noise, even when $N = 2430$ individuals are measured. This is a consequence of the filter approximation of the measurement distribution which only uses simulations of $S = 100$ individuals. While $N = 2430$ measurements start to resolve distributional differences between IIV and noise contributions, $S = 100$ simulated individuals are insufficient to do so. Increasing the number of simulated individuals alleviates this limitation, as demonstrated in Figure C.3, where we illustrate posterior distributions inferred from $N = 60\,000$ snapshot measurements using filter inference with Gaussian filters and $S = 5000$ simulated individuals.

C.5 Estimating the average of a population distribution over the posterior distribution

Formally, the average of the population distribution over the posterior distribution, also known as the posterior predictive population distribution, is defined as

$$p(\psi|\mathcal{D}) = \mathbb{E}_{\theta|\mathcal{D}}[p(\psi|\theta)] = \int \mathrm{d}\theta\, p(\psi|\theta)\, p(\theta|\mathcal{D}).$$

While the integral is generally difficult, or impossible, to compute analytically, the above equation implicitly defines a joint distribution of the individual-level parameters

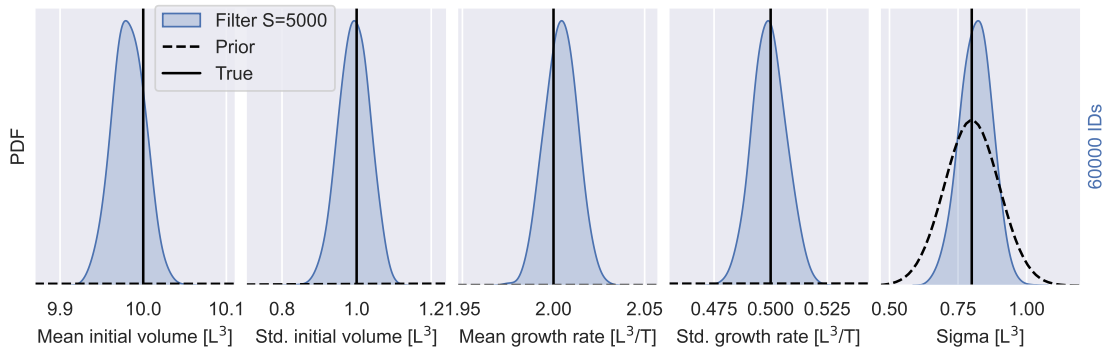


Figure C.3: IIV-noise identifiability of early cancer growth model II. The figure is an extension of Fig C.1 and shows filter inference results for the early cancer growth model from 60 000 snapshot measurements using Gaussian filters with $S = 5000$ simulated individuals. The prior distributions are illustrated by black dashed lines and the data-generating parameter values are depicted by solid black lines.

and the population-level parameters, $p(\psi, \theta|\mathcal{D}) = p(\psi|\theta)\, p(\theta|\mathcal{D})$. To sample from $p(\psi|\mathcal{D})$, we can sample from $p(\psi, \theta|\mathcal{D})$ and marginalise over the sampled population-level parameters afterwards. In particular, we first sample a θ from $p(\theta|\mathcal{D})$, and then draw a ψ from $p(\psi|\theta)$ using the sampled θ. The histogram over the sampled ψ converges to the posterior predictive population distribution as the number of samples tends to infinity. In practice, samples from $p(\theta|\mathcal{D})$ are generated by sampling with replacement from the MCMC posterior samples.

C.6 Estimation of EGF pathway model parameters

The parameters of the EGF pathway model are estimated with pints' implementation of NUTS, unless otherwise specified. MCMC chains are run for 1500 iterations, where the first 500 iterations are used for calibration. The convergence is assessed using the $\hat{R}$-statistic from 3 MCMC chains, initialised at randomly sampled points from the prior

distribution. The prior distribution for all inference runs is

$$\begin{aligned}
\mu_p &\sim \mathcal{N}(2, 0.25) \\
\sigma_p &\sim \mathrm{LN}(-2, 0.5) \\
\mu_{k_{\mathrm{on}}} &\sim \mathcal{N}(2, 0.25) \\
\sigma_{k_{\mathrm{on}}} &\sim \mathrm{LN}(-2, 0.5) \\
\mu_{k_{\mathrm{off}}} &\sim \mathcal{N}(10, 4) \\
\mu_{k_{\mathrm{deg},r}} &\sim \mathcal{N}(0.3, 0.0025) \\
\mu_{k_{\mathrm{deg},a}} &\sim \mathcal{N}(0.02, 0.000025).
\end{aligned}$$

Detailed information on the convergence can be found in Table C.2.

	$\hat{R}$
μ_p	1.00
σ_p	1.00
$\mu_{k_{\mathrm{on}}}$	1.00
$\sigma_{k_{\mathrm{on}}}$	1.00
$\mu_{k_{\mathrm{off}}}$	1.00
$\mu_{k_{\mathrm{deg},r}}$	1.00
$\mu_{k_{\mathrm{deg},a}}$	1.00
# divergences	0

Table C.2: Convergence statistics of MCMC chains during the parameter estimation of the EGF pathway model.

C.7 Estimation of the evaluation time

The evaluation times of the NLME log-posterior and the filter log-posterior in Figure 4.7 are estimated by evaluating the log-posteriors for each dataset 10 times at fixed parameter values on a MacBook Pro, 2.8 GHz Quad-Core Intel Core i7 processor with 16 GB 2133 MHz LPDDR3 memory. The execution time of each evaluation is measured using the Python package `timeit`. The minimum execution time of the 10 repeats is reported in the figure. In this way, we are able to find the optimal execution time in Python, accounting for Python's unpredictable overhead of execution time.

C.8 Hyperparameter tuning of MH algorithm

The hyperparameter of the MH algorithm is the step size of the proposal step. More precisely the MH algorithm proposes new parameters by sampling from a multivariate Gaussian distribution centred at the last accepted proposal. The covariance matrix of this distribution sets the scale of the step size and is the hyperparameter of the algorithm. To find a good set of step sizes we performed a grid search over several step sizes (see Table C.3).

Scale factor	90 IDs	270 IDs	810 IDs	2430 IDs	2400 (EGF)
1.4	71	**24**	4	3.3	3.14
1.2	71	15	7	3.6	**3.32**
1	**115**	12	**11**	3.3	3.04
0.8	63	16	6	3.2	3.22
0.6	82	7	6	3.6	3.13
0.4	63	8	6	**3.8**	3.08
0.2	52	5	5	3.4	3.29

Table C.3: Grid search results: ESS of MH. We ran each MH algorithm with 3 chains for 5000 iterations and computed the ESS after discarding the first 2500 iterations. The reference variances of the filter posteriors are reported in Table C.5 and Table C.6.

In particular, we used the variances of the filter posterior distributions inferred using NUTS in Figure C.1 and Figure 4.6 as a starting point – information one would usually not have available prior to the inference. This allowed us to carry out the grid search more efficiently, as the marginal variances of the target distribution provide a good starting point for the diagonal of the covariance matrix of the MH proposal distribution. For each problem we ran the MH algorithm 7 times with 3 chains for 5000 MCMC iterations with diagonal covariance matrices whose diagonal was set to x times the variances of the target distribution. We performed the grid search for $x \in \{1.4, 1.2, 1, 0.8, 0.6, 0.4, 0.2\}$. The covariance matrix with the largest ESS was chosen across for 3 for the inference. All chains where randomly initialised in the vicinity of the data-generating parameters and the first 2500 iterations were discarded before the ESS computation.

C.9 Equivalence of filter inference with a Gaussian filter and ABC based on the mean and variance

Let $Y_1, \dots Y_N$ be N i.i.d. random, real-valued variables drawn from the data-generating distribution $q(y)$. Let q have nonzero variance, $\mathrm{Var}[Y] > 0$. Let $\tilde{Y}_1, \dots \tilde{Y}_S$ be S i.i.d. random variables drawn from the model $p(y|\theta)$. Let further $\mathcal{K}_\varepsilon(X - \tilde{X})$ be a kernel with an error margin ε, used in ABC to quantify the distance between the data summary statistic $X = X(Y_1, \dots Y_N)$ and the simulated summary statistic $\tilde{X} = X(\tilde{Y}_1, \dots \tilde{Y}_S)$. Let the kernel converge to a Dirac delta distribution up to a proportionality factor as the error margin goes to zero, $\lim_{\varepsilon \to 0} \mathcal{K}_\varepsilon(X - \tilde{X}) \propto \delta(X - \tilde{X})$. Let further $p(y|\tilde{\mu}, \tilde{\sigma}^2) = \mathrm{e}^{-(y-\tilde{\mu})^2/2\tilde{\sigma}^2} / \sqrt{2\pi\tilde{\sigma}^2}$ denote a Gaussian filter and $p(\mathcal{D}|\tilde{\mu}, \tilde{\sigma}^2) = \prod_{i=1}^N p(Y_i|\tilde{\mu}, \tilde{\sigma}^2)$ denote its likelihood, where $\mu = \sum_{i=1}^N Y_i/N$ denotes the mean estimator and $\sigma^2 = \sum_{i=1}^N (Y_i - \mu)^2/(N-1)$ denotes the variance estimator. Then, a Gaussian filter converges to a Dirac delta distribution between (μ, σ^2) and $(\tilde{\mu}, \tilde{\sigma}^2)$ up to a proportionality factor as the number of measurements goes to infinity, $\lim_{N\to\infty} p(\mathcal{D}|\tilde{\mu}, \tilde{\sigma}^2) \propto \delta(\mu - \tilde{\mu})\, \delta(\sigma^2 - \tilde{\sigma}^2)$. As a result, filter inference with a Gaussian filter is equivalent to ABC based on the mean and variance, $\mathcal{K}_\varepsilon(\mu - \tilde{\mu})\, \mathcal{K}_\varepsilon(\sigma^2 - \tilde{\sigma}^2)$, in the limit $N \to \infty$ and $\varepsilon \to 0$.

Proof: The maximum likelihood estimators (MLEs) of the Gaussian filter likelihood are

$$\mu_* = \frac{1}{N} \sum_{i=1}^N Y_i \quad \text{and} \quad \sigma_*^2 = \sum_{i=1}^N (Y_i - \mu)^2/N,$$

where $\mu_* = \mu$ and $\sigma_*^2 = \frac{N-1}{N}\sigma^2$. The MLEs are the unique maximum of the Gaussian filter likelihood. The curvature of the log-likelihood at the MLEs is

$$\begin{aligned}
\frac{\partial^2}{\partial\tilde{\mu}^2} \log p(\mathcal{D}|\tilde{\mu}, \tilde{\sigma}^2)\Big|_{\tilde{\mu}=\mu_*, \tilde{\sigma}^2=\sigma_*^2} &= -\frac{N}{\sigma_*^2} \\
\frac{\partial^2}{\partial\tilde{\sigma}^2} \log p(\mathcal{D}|\tilde{\mu}, \tilde{\sigma}^2)\Big|_{\tilde{\mu}=\mu_*, \tilde{\sigma}^2=\sigma_*^2} &= \frac{1-N}{\sigma_*^4} \\
\frac{\partial^2}{\partial\tilde{\mu}\,\partial\tilde{\sigma}} \log p(\mathcal{D}|\tilde{\mu}, \tilde{\sigma}^2)\Big|_{\tilde{\mu}=\mu_*, \tilde{\sigma}^2=\sigma_*^2} &= 0.
\end{aligned}$$

In the limit $N \to \infty$, the curvature of the log-likelihood at the MLEs tends to infinity. Due to the monotonicity of the logarithm this implies that the curvature of the likelihood

tends to infinity at the MLEs, proving that in the limit $N \to \infty$ a Gaussian filter likelihood is proportional to a Dirac delta distribution, $\lim_{N\to\infty} \prod_{i=1}^{N} p(\mathcal{D}|\tilde{\mu}, \tilde{\sigma}^2) \propto \delta(\mu_* - \tilde{\mu})\, \delta(\sigma_*^2 - \tilde{\sigma}^2)$. In this limit, the maximum likelihood estimator of the variance coincides with the variance estimator, $\lim_{N\to\infty} \sigma_*^2 = \sigma^2$, proving the equivalence between filter inference with a Gaussian filter and ABC based on the mean and the variance in the limit $N \to \infty$ and $\varepsilon \to 0$.

C.10 Equivalence of filter inference with identifiable summary statistics-based filters and ABC based on the same summary statistics

Let $Y_1, \dots Y_N$ be N i.i.d. random, real-valued variables drawn from the data-generating distribution $q(y)$. Let q have nonzero variance, $\mathrm{Var}[Y] > 0$. Let $\tilde{Y}_1, \dots \tilde{Y}_S$ be S i.i.d. random, real-valued variables drawn from the model $p(y|\theta)$. Let further $\mathcal{K}_\varepsilon(X - \tilde{X})$ be a kernel with an error margin ε, used in ABC to quantify the distance between the data summary statistic $X = X(Y_1, \dots Y_N)$ and the simulated summary statistic $\tilde{X} = X(\tilde{Y}_1, \dots \tilde{Y}_S)$. Let the kernel converge to a Dirac delta distribution up to a proportionality factor as the error margin goes to zero, $\lim_{\varepsilon\to 0} \mathcal{K}_\varepsilon(X - \tilde{X}) \propto \delta(X - \tilde{X})$. Let further $p(y|\tilde{X}_1, \dots, \tilde{X}_K)$ denote a filter defined by K summary statistics. Let $p(\mathcal{D}|\tilde{X}_1, \dots, \tilde{X}_K) = \prod_{i=1}^{N} p(Y_i|\tilde{X}_1, \dots, \tilde{X}_K)$ denote its likelihood. Let $p(y|\tilde{X}_1, \dots, \tilde{X}_K)$ be a *summary statistics-based filter*, when the maximum likelihood estimates of the filter likelihood converge to the summary statistics of the data $X_1, \dots, X_K$ as $N \to \infty$. Let $p(y|\tilde{X}_1, \dots, \tilde{X}_K)$ be an *identifiable filter*, when the filter likelihood has a unique maximum, whose curvature tends to infinity in the limit $N \to \infty$. Then, an identifiable summary statistics-based filter converges to a Dirac delta distribution between the data summary statistics and the simulated summary statistics up to a proportionality factor as the number of measurements goes to infinity, $\lim_{N\to\infty} p(\mathcal{D}|\tilde{X}_1, \dots, \tilde{X}_K) \propto \prod_{k=1}^{K} \delta(X_k - \tilde{X}_k)$. As a result, filter inference with an identifiable summary statistics-based filter is equivalent to ABC based on the same summary statistics, $\prod_{k=1}^{K} \mathcal{K}_\varepsilon(X_k - \tilde{X}_k)$, in the limit $N \to \infty$ and $\varepsilon \to 0$.

Proof: The MLEs of a summary statistics-based filter recover the summary statistics of the data in the limit $N \to \infty$. The MLEs of an identifiable filter are the unique maximum of the filter likelihood, whose curvature tends to infinity as $N \to \infty$. As a result, the filter likelihood must be proportional to a Dirac delta distribution at the summary statistics of the data in the limit $N \to \infty$, $\lim_{N\to\infty} p(\mathcal{D}|\tilde{X}_1, \ldots, \tilde{X}_K) \propto \prod_{k=1}^{K} \delta(X_k - \tilde{X}_k)$. The kernel from ABC also converges to a Dirac delta distribution up to a proportionality factor when the error margin goes to zero, $\lim_{\varepsilon\to 0} \prod_{k=1}^{K} \mathcal{K}_\varepsilon(X_k - \tilde{X}_k) = \prod_{k=1}^{K} \delta(X_k - \tilde{X}_k)$. As a result, filter inference with an identifiable summary statistics-based filter is equivalent to ABC based on the same summary statistics in the limit $N \to \infty$ and $\varepsilon \to 0$.

C.11 Supplementary tables and figures

	90 IDs	270 IDs	810 IDs	2430 IDs	2400 IDs (EGF)
NUTS (warm-up)	20,000	25,626	33,059	53,589	134,413
NUTS (sampling)	14,985	14,985	30,969	30,969	62,937
NUTS (total)	34,985	40,611	64,028	84,558	197,350
Metropolis-Hastings	90,000	90,000	90,000	90,000	180,000

Table C.4: Number of log-posterior evaluations. The table shows the total number of times the log-posterior is evaluated during the inferences in Figure 4.9.

	90 IDs	270 IDs	810 IDs	2430 IDs
$\mathrm{Var}[\mu_{y_0}]$	0.086	0.036	0.02	0.017
$\mathrm{Var}[\sigma_{y_0}]$	0.012	0.0084	0.0046	0.0032
$\mathrm{Var}[\mu_\lambda]$	0.072	0.036	0.018	0.0014
$\mathrm{Var}[\sigma_\lambda]$	0.0082	0.00456	0.0033	0.0024
$\mathrm{Var}[\mu_\sigma]$	0.0089	0.1176	0.0082	0.0096

Table C.5: Variances of filter posteriors: Early cancer model.

	Variance
μ_p	0.00005
σ_p	0.0001
$\mu_{k_{\text{on}}}$	0.00006
$\sigma_{k_{\text{on}}}$	0.00004
$\mu_{k_{\text{deg},r}}$	0.0000008
$\mu_{k_{\text{deg},a}}$	0.000002

Table C.6: Variances of filter posterior: EGF pathway model.

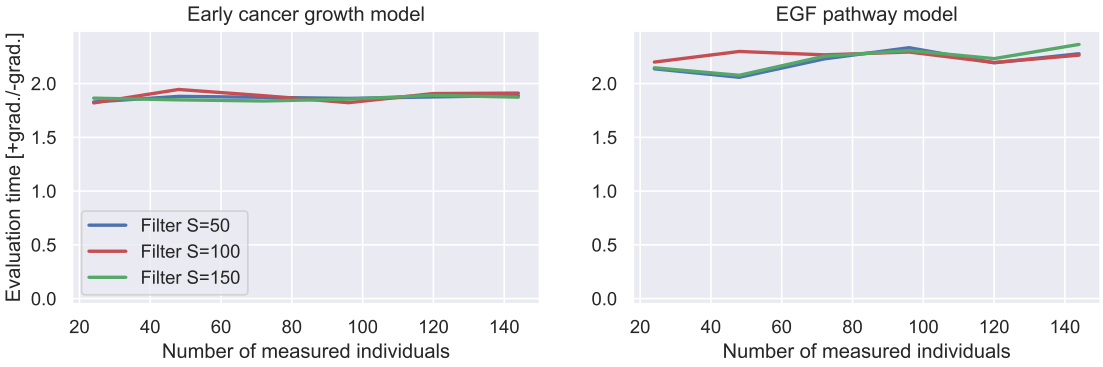


Figure C.4: Computational costs of log-posterior evaluation with and without gradients. The figure is an extension of Figure 4.7 and shows the evaluation time of the filter log-posterior with gradients in units of the evaluation time of the filter log-posterior without gradients for different numbers of measured individuals. The evaluation times are estimated according to section C.7. The left panel shows the results for the early cancer growth model and the right panel the results for the EGF pathway model for $S = 50$ (blue), $S = 100$ (red) and $S = 150$ (green) simulated individuals.

D

Supporting information – Chapter 5

Contents

D.1 Infliximab candidate models . . . 318
D.1.1 1-compartment PK model (m=1) . . . 318
D.1.2 2-compartment PK model with linear clearance (m=2) . . . 319
D.1.3 2-compartment PK model with nonlinear clearance (m=3) 320
D.1.4 TMDD model (m=4) . . . 320
D.2 Data-generating population model for infliximab . . . 321
D.3 MAP estimation . . . 322
D.4 Dose optimisation using MAP estimates . . . 322
D.5 Estimation of posterior distributions . . . 322
D.6 Estimation of the posterior predictive distributions in Figure 5.8 . 323
D.7 Correctly individualised dose for simulated patient . . . 324
D.8 Estimation of posterior predictive dose distribution . . . 324
D.9 Estimation of probability assigned to dose interval . . . 325
D.10 Estimation of exact cross-entropies . . . 326
D.11 Proof of convergence to the local cross-entropy . . . 327
D.12 Supplementary tables . . . 329

D.1 Infliximab candidate models

D.1.1 1-compartment PK model (m=1)

The mechanistic model is defined as follows:

$$\frac{\mathrm{d}a_c}{\mathrm{d}t} = -k_e a_c + r(t), \qquad a_c(t=0, r, \psi, m=1) = 0. \tag{D.1}$$

Here, a_c models the drug amount in the blood. k_e denotes the elimination rate of the drug from the system. $r(t)$ represents a time-dependent dose rate. We make the common choice to parametrise the elimination of the drug by the clearance, $k_{Cl} = vk_e$, where v denotes the drug's volume of distribution.

The measured quantity of interest is the drug concentration in the blood, $c = a_c/v$. The measurement noise model is a lognormal distribution, defining the measurement distribution as

$$p(y|t, r, \psi, m = 1) = \frac{1}{\sqrt{2\pi\sigma^2}} \frac{1}{y} \mathrm{e}^{-\left(\log y - \log c(t,r,\psi,m=1)\right)^2/2\sigma^2}, \tag{D.2}$$

where $\psi = (k_{Cl}, v, \sigma)$ denotes the model parameters.

D.1.2 2-compartment PK model with linear clearance (m=2)

The mechanistic model is defined as follows:

$$\begin{aligned} \frac{\mathrm{d}a_c}{\mathrm{d}t} &= -k_e\, a_c + k_{cp}\, a_p - k_{pc}\, a_c, & a_c(t = 0, r, \psi, m = 2) &= 0, \\ \frac{\mathrm{d}a_p}{\mathrm{d}t} &= -k_{cp}\, a_p + k_{pc}\, a_c, & a_p(t = 0, r, \psi, m = 2) &= 0, \end{aligned} \tag{D.3}$$

where a_c denotes the drug amount in the central compartment, and a_p denotes the drug amount in the peripheral compartment. k_e denotes the elimination rate of the drug, k_{pc} denotes the transition rate from the central compartment to the peripheral compartment, and k_{cp} denotes the transition rate from the peripheral compartment to the central compartment. $r(t)$ represents a time-dependent dose rate. We make the common choice to parametrise the elimination of the drug by the clearance, $k_{Cl} = vk_e$, where v denotes the drug's volume of distribution.

The measured quantity of interest is the drug concentration in the blood, $c = a_c/v$. The measurement noise model is a lognormal distribution, defining the measurement distribution as

$$p(y|t, r, \psi, m = 2) = \frac{1}{\sqrt{2\pi\sigma^2}} \frac{1}{y} \mathrm{e}^{-\left(\log y - \log c(t,r,\psi,m=2)\right)^2/2\sigma^2}, \tag{D.4}$$

where $\psi = (k_{Cl}, k_{cp}, k_{pc}, v, \sigma)$ denotes the model parameters.

D.1.3 2-compartment PK model with nonlinear clearance (m=3)

The mechanistic model is defined as follows:

$$\begin{aligned} \frac{\mathrm{d}a_c}{\mathrm{d}t} &= -k_{Cl}(c,\psi)\,a_c/v + k_{cp}\,a_p - k_{pc}\,a_c, & a_c(t=0,r,\psi,m=3) &= 0, \\ \frac{\mathrm{d}a_p}{\mathrm{d}t} &= -k_{cp}\,a_p + k_{pc}\,a_c, & a_p(t=0,r,\psi,m=3) &= 0. \end{aligned} \tag{D.5}$$

a_c denotes the drug amount in the central compartment, and a_p denotes the drug amount in the peripheral compartment. v denotes the drug's volume of distribution, k_{pc} denotes the transition rate from the central compartment to the peripheral compartment, and k_{cp} denotes the transition rate from the peripheral compartment to the central compartment. $r(t)$ represents a time-dependent dose rate.

$k_e(c,\psi)$ denotes the nonlinear clearance

$$k_{Cl}(c,\psi) = k_0 + \frac{\Delta}{c_{50}+c}, \tag{D.6}$$

where k_0 denotes the base clearance, Δ denotes the maximal increase of the clearance, and c_{50} denotes the drug concentration at which the increase is half maximal.

The measured quantity of interest is the drug concentration in the blood, $c = a_c/v$. The measurement noise model is a lognormal distribution, defining the measurement distribution as

$$p(y|t,r,\psi,m=3) = \frac{1}{\sqrt{2\pi\sigma^2}}\frac{1}{y}\mathrm{e}^{-\left(\log y - \log c(t,r,\psi,m=3)\right)^2/2\sigma^2}, \tag{D.7}$$

where $\psi = (k_0, \Delta, c_{50}, k_{cp}, k_{pc}, v, \sigma)$ denotes the model parameters.

D.1.4 TMDD model (m=4)

The mechanistic model is defined as follows:

$$\begin{aligned} \frac{\mathrm{d}c}{\mathrm{d}t} &= -k_e c - k_f c\,c_r + k_{ds}c_{rc} + r(t)/v, & c(t=0,r,\psi,m=4) &= 0, \\ \frac{\mathrm{d}c_r}{\mathrm{d}t} &= k_s - k_{dg}c_r - k_f c\,c_r + k_{ds}c_{rc}, & c_r(t=0,r,\psi,m=4) &= c_{r,0}, \\ \frac{\mathrm{d}c_{rc}}{\mathrm{d}t} &= -k_{int}c_{rc} + k_f c\,c_r - k_{ds}c_{rc}, & c_{rc}(t=0,\psi) &= 0. \end{aligned} \tag{D.8}$$

c denotes the free drug concentration in the blood, c_r denotes the target receptor concentration, and c_{rc} denotes the drug-target complex concentration. k_e and v denote

the elimination rate and the volume of distribution of the drug. k_s and k_{dg} denote the synthesis rate and the degradation rate of the target receptor. $c_{r,0}$ denotes the initial receptor concentration. k_f and k_{ds} denote the formation rate and the disassociation rate of the complex. k_{int} denotes the internalisation rate of the complex. The system is assumed to be in equilibrium at $t = 0$, allowing us to set $k_s = k_{dg}c_{r,0}$. We also make the common choice to parametrise the elimination of the drug by the clearance, $k_{cl} = vk_e$, and assume that k_f and k_{ds} are orders of magnitude faster than the other rates, in accordance to the quasi equilibrium approximation of Ternant et al (Ternant et al. 2022). More specifically, we set $k_f = 50k_{dg}$ and $k_{ds}/k_f = K_d$ to implement this approximation. The analytic expression for the quasi equilibrium approximation was found to be unstable in the differential equation solver of Chi (Augustin 2024).

The measured quantity of interest is the drug concentration in the blood, $c = a_c/v$. The measurement noise model is a lognormal distribution, defining the measurement distribution as

$$p(y|t, r, \psi, m = 4) = \frac{1}{\sqrt{2\pi\sigma^2}} \frac{1}{y} \mathrm{e}^{-\left(\log y - \log c(t,r,\psi,m=4)\right)^2/2\sigma^2}, \tag{D.9}$$

where $\psi = (k_{cl}, v, c_{r,0}, K_d, k_{int}, k_{dg}, \sigma)$ denotes the model parameters.

D.2 Data-generating population model for infliximab

The individual-level data-generating model, $p(y|t, r, \psi, m = 4)$, is defined in Appendix D.1.4.

The data-generating population model, describing the inter-individual variability of model parameters, is defined as

$$\begin{aligned} p(\psi|\theta) \quad = \quad & \mathrm{LN}(k_{cl}|\mu_{k_{cl}}, \sigma_{k_{cl}})\, \mathrm{LN}(v|\mu_v, \sigma_v)\, \mathrm{LN}(c_{r,0}|\mu_{c_{r,0}}, \sigma_{c_{r,0}})\, \mathrm{LN}(K_d|\mu_{K_d}, \sigma_{K_d}) \\ & \mathrm{LN}(k_{int}|\mu_{k_{int}}, \sigma_{k_{int}})\, \mathrm{LN}(k_{dg}|\mu_{k_{dg}}, \sigma_{k_{dg}})\, \mathrm{LN}(\sigma|\mu_\sigma, \sigma_\sigma), \end{aligned} \tag{D.10}$$

where $\mathrm{LN}(x|\mu, \sigma) = \exp(-(\log x - \log \mu)^2/2\sigma^2)/x\sqrt{2\pi\sigma^2}$ denotes the probability density of a lognormal distribution with median μ and scale σ. The data-generating model parameters are adopted from Ternant et al (Ternant et al. 2022), and are defined in Appendix D.1.

D.3 MAP estimation

We implement the posterior distributions of each candidate model in Chi Augustin 2024. For each candidate model, the log-posterior distribution is defined by the log-likelihood of the measurements under the model and the log-prior distribution of the model parameters

$$\log p(\psi|\mathcal{D}, m) = \frac{1}{n}\sum_{j=1}^{n} \log p(y_j|t_j, r, \psi, m) + \log p(\psi|m) + \text{constant}. \quad \text{(D.11)}$$

The measurement distributions of the candidate models are defined in Appendix D.1. The prior distributions for the parameters of each candidate model are defined in Tables D.2, D.3, D.4, and D.5. We optimise the posterior distributions to find the MAP estimates using pints' implementation of the particle swarm optimiser (PSO) (Clerx, Robinson, et al. 2019). The parameter estimates are available on https://github.com/DavAug/mipd-information-criteria).

D.4 Dose optimisation using MAP estimates

To predict individualised IV doses for treatment day 14, we minimise the mean squared error of each model's drug concentration simulation to the targeted trough concentration, $\bar{y}^* = 18\,\text{mg/L}$

$$\text{MSE}(d) = \left(\bar{y}(t = 14, r, \psi^*) - \bar{y}^*\right)^2, \quad \text{(D.12)}$$

where the dosing regimen, $r(t, d)$, administers a fixed IV dose on day 0, and a free dose, d on day 14. We optimise the MSE using the Nelder-Mead optimiser. The dose estimates are available on https://github.com/DavAug/mipd-information-criteria.

D.5 Estimation of posterior distributions

We implement the posterior distributions of each candidate model in Chi Augustin 2024, analogously to Appendix D.3: For each candidate model, the log-posterior distribution is defined by the log-likelihood of the measurements under the model and the log-prior distribution of the model parameters

$$\log p(\psi|\mathcal{D}, m) = \frac{1}{n}\sum_{j=1}^{n} \log p(y_j|t_j, r, \psi, m) + \log p(\psi|m) + \text{constant}. \quad \text{(D.13)}$$

The measurement distributions of the candidate models are defined in Appendix D.1. The prior distributions for the parameters of each candidate model are defined in Tables D.2, D.3, D.4, and D.5.

We estimate the posterior distributions using MCMC sampling. In particular, we use Pint's implementation of the Haario-Bardenet Adaptive Covariance Matrix MCMC algorithm (Clerx, Robinson, et al. 2019). For each iteration of the algorithm, one correlated sample from the posterior distribution is drawn. We run the algorithm for 10 000 iterations, discard the first 5 000 iterations as warm up for the (automated) hyperparameter tuning of the algorithm, and assess the convergence of the historgrams over the correlated samples to the posterior distribution using ArviZ's implementation of the $\hat{R}$-statistic. For models, the $\hat{R}$-statistic estimates a value of ≤ 1.01 for all model parameters, suggesting convergence of the histograms.

The samples drawn from the posterior distributions of all models, as well as the corresponding $\hat{R}$-statistics are available on https://github.com/DavAug/mipd-information-criteria.

D.6 Estimation of the posterior predictive distributions in Figure 5.8

We estimate posterior predictive distributions using the samples from the posterior distributions. More specifically, we estimate the integral over the posterior distribution by an average over its samples

$$p(y|t,r,\mathcal{D},m) = \int \mathrm{d}\psi\, p(y|t,r,\psi,m)\, p(\psi|\mathcal{D},m) \approx \frac{1}{S}\sum_{s=1}^{S} p(y|t,r,\psi_s,m), \quad \text{(D.14)}$$

where ψ_s denotes the sth sample of S samples drawn from the posterior distribution, $p(\psi|\mathcal{D},m)$. For details on the sampling from posterior distributions, see Appendix D.5.

In Figure 5.8, we focus on the range of treatment responses consistent with the parametric uncertainty (as opposed to the range of measurements consistent with the parametric uncertainty). We therefore estimate the posterior predictive distribution using a noise-free measurement model, $p(y|t,r,\psi,m) = \delta(y - \bar{y}(t,r,\psi,m))$.

D.7 Correctly individualised dose for simulated patient

We determine the correct dose that realises the targeted trough concentration of 18 mg/L for the simulated patient using an analogous approach to the dose prediction for the MAP fits (see Appendix D.5). The only difference is that we use the data-generating model with the patient-specific model parameters for the treatment response simulation.

We find that an IV dose of 734 mg on day 14 realises the targeted trough concentration, see https://github.com/DavAug/mipd-information-criteria.

D.8 Estimation of posterior predictive dose distribution

We estimate the posterior predictive dose distribution, $p(d|\mathcal{D}) = \sum_m p(m|\mathcal{D})\, p(d|\mathcal{D}, m)$, by repeatedly sampling a model structure from the structural posterior distribution, $p(m|\mathcal{D})$. We then predict a dose that meets the targeted trough concentration on week 6 for each sampled model structure. More specifically, for each sampled model structure, we also randomly draw a set of parameters from the corresponding parametric posterior distribution and minimise the MSE between the treatment response simulation and the targeted trough concentration, as described in section D.4.

The main difference to the MAP predictions is that the model structures are randomly sampled from structural posterior distribution to reflect the support of each candidate model, and that the model parameters are not fixed to the MAP estimates, but randomly sampled from the posterior distribution.

In Figure 5.9, we sample a model structure from the structural posterior distribution, and optimise the dose for a randomly sampled set of model parameters, 5 000 times in order to estimate the posterior predictive dose distribution.

For scripts, implementing this approach, and the estimated posterior predictive dose distribution, see https://github.com/DavAug/mipd-information-criteria.

D.9 Estimation of probability assigned to dose interval

Formally, the probability assigned to a dose interval, $[d_{min}, d_{max}]$, is defined as

$$p([d_{min}, d_{max}]) = \int_{d_{min}}^{d_{max}} \mathrm{d}d\, p(d|\mathcal{D}), \tag{D.15}$$

where $p(d|\mathcal{D})$ is the posterior predictive dose distribution.

In the WAIC-based structural uncertainty approach, the posterior predictive distribution is given by the weighted average of the individual dose distributions of the candidate models

$$p(d|\mathcal{D}) = \sum_m p(d|\mathcal{D}, m)\, p(m|\mathcal{D}), \tag{D.16}$$

where $p(d|\mathcal{D}, m)$ denotes the dose distribution of candidate model, m, and $p(m|\mathcal{D})$ denotes the WAIC-based structural posterior distribution.

In the WAIC-based model selection approach, the structural uncertainty is neglected, and the model with the maximum structural posterior probability is selected for prediction. This reduces the above expression to

$$p(d|\mathcal{D}) = \sum_m p(d|\mathcal{D}, m)\, \delta_{m,m^*}, \tag{D.17}$$

where $m^* = \text{argmax}\, p(m|\mathcal{D})$.

In the AIC-based estimation approach, Equation D.16 also reduces to simpler expressions. In particular, for the AIC-based structural uncertainty approach, each candidate model predicts only one dose

$$p(d|\mathcal{D}) = \sum_m \delta_{d,d_m}\, p(m|\mathcal{D}), \tag{D.18}$$

where d_m denotes the dose predicted by model m.

For the AIC-based model selection approach, the expression for posterior predictive probability reduces further to

$$p(d|\mathcal{D}) = \sum_m \delta_{d,d_m}\, \delta_{m,m^*}, \tag{D.19}$$

where $m^* = \text{argmax}\, p(m|\mathcal{D})$.

In practice, the probability of dose intervals can be estimated using sampling: Sampling S doses from a posterior predictive dose distribution, $d_1, \ldots, d_S$, we can estimate the probability of the interval by calculating the fraction of sampled doses in the interval. For increasing sample sizes, this estimator converges to the probability of the interval

$$\lim_{S\to\infty} \frac{\sum_s \theta(d_s - d_{min})\,\theta(d_{max} - d_s)}{S} = p([d_{min}, d_{max}]), \tag{D.20}$$

where $\theta(x)$ denotes the Heaviside step function which assumes a value of 1 for $x > 0$, and is otherwise zero.

For the results in Figure 5.10, we sample $S = 1000$ doses from each posterior predictive distribution to estimate the probabilities of the 5 % error interval around the correct dose for each patient. The methodology for sampling from the posterior predictive dose distribution is the same as in previous section and documented in Appendix D.4 and Appendix D.8.

D.10 Estimation of exact cross-entropies

To estimate the exact cross-entropies of MAP fits with the data-generating process, we use Equation 5.2

$$S_\rho(q||m) \approx -\frac{1}{nS}\sum_{s=1}^{S}\sum_{j=1}^{n} \log p(y_{js}|t_j, r, \psi^*, m). \tag{D.21}$$

Here, we use ψ^* to denote the MAP estimates, estimated from measurements $\mathcal{D} = \{(y_1, t_1), \ldots, (y_n, t_n), r\}$. The measurements used to estimate the cross-entropy, $\mathcal{D}^* = \{(y_{11}, t_1), \ldots, (y_{1S}, t_1), \ldots, (y_{n1}, t_1), \ldots, (y_{nS}, t_1), r\}$ are sampled at the same time points as the measurements used for the parameter estimation. S denotes the number of measurements drawn at each time point.

To estimate the exact cross-entropies of posterior predictive distributions with the data-generating process, we use analogously

$$S_\rho(q||m) \approx -\frac{1}{nS}\sum_{s=1}^{S}\sum_{j=1}^{n} \log p(y_{js}|t_j, r, \mathcal{D}, m), \tag{D.22}$$

where we estimate the posterior predictive distribution using Equation D.14.

For the estimations in Figures 5.11 and 5.12, we use $S = 1000$ at each time point for the cross-entropy estimation (see https://github.com/DavAug/mipd-information-criteria).

D.11 Proof of convergence to the local cross-entropy

To prove that (unbiased) information criteria converge to the local cross-entropy for a specific choice of the locality measure, ρ, we consider the simplest unbiased estimator (Equation 5.2)

$$\hat{S}_\rho(q||m) = -\frac{1}{n}\sum_{j=1}^{n} \log p(y_j|t_j, r_j, \psi, m), \tag{D.23}$$

where the measurements $\mathcal{D} = \{(y_1, t_1, r_1), \dots, (y_n, t_n, r_n)\}$ are only used for the cross-entropy estimation. The expectation value of this estimator is

$$\begin{aligned}
\mathbb{E}\left[\hat{S}_\rho(q||m)\right] &= -\frac{1}{n}\sum_{j=1}^{n} \mathbb{E}\Big[\log p(y|t_j, r_j, \psi, m)\Big] \\
&= -\frac{1}{n}\sum_{j=1}^{n} \int \mathrm{d}y\, q(y|t_j, r_j) \log p(y|t_j, r_j, \psi, m) \\
&= \frac{1}{n}\sum_{j=1}^{n} S_{t_j, r_j}(q||m) \\
&= \frac{1}{n}\sum_{j=1}^{n} \int \mathrm{d}t\, \mathrm{d}r\, \delta(t - t_j)\, \delta(r - r_j)\, S_{t,r}(q||m) \\
&= S_\rho(q||m) \quad \text{with} \quad \rho(r, t) = \frac{1}{n}\sum_{j=1}^{n} \delta(t - t_j)\, \delta(r - r_j).
\end{aligned} \tag{D.24}$$

In the 3rd equality, we use the definition of the point-wise cross-entropy (Equation 5.13) and in the 5th equality, we use the definition of the local cross-entropy (Equation 5.14).

This shows that the expectation value of $\hat{S}_\rho(q||m)$ is the local cross-entropy where the locality measure, ρ, is equal to the distribution of measurement times and dosing regimens used for the cross-entropy estimation.

The variance of this estimator is

$$\mathrm{Var}\left[\hat{S}_\rho(q||m)\right] = -\frac{1}{n}\sum_{j=1}^{n} \mathrm{Var}\Big[\log p(y|t_j, r_j, \psi, m)\Big]. \tag{D.25}$$

This shows that the variance of the estimator convergence to 0 in the limit $n \to \infty$, provided the variance of the point-wise log-likelihoods grows slower than linearly in n.

This slow growth of the variance with the number of measurements is a typical assumption of information criteria (Akaike 1972, 1974; Bürkner et al. 2021; Watanabe et al. 2010).

This concludes the proof. Note that this does not imply that all information criteria are unbiased estimators of the local cross-entropy, as argued in subsection 5.3.2.

D.12 Supplementary tables

Parameter X	Median μ_x	Scale σ_x
k_{cl} in L/day	0.24	0.31
v in L	5.6	0.21
$c_{r,0}$ in mg/L	0.069	0.092
K_d in mg/L	0.067	0.15
k_{int} in 1/day	0.024	0.19
k_{dg} in 1/day	10.5	0.17
σ in dimensionless	0.1	0.1

Table D.1: Data-generating parameters of population model.

Parameter	Prior distribution
k_{cl} in L/day	$\mathrm{LN}(\cdot\|0.24, 0.31)$
v in L	$\mathrm{LN}(\cdot\|5.6, 0.21)$
σ in dimensionless	$\mathrm{LN}(\cdot\|0.1, 0.1)$

Table D.2: Parametric prior distribution for candidate model m=1.

Parameter	Prior distribution
k_{cl} in L/day	$\mathrm{LN}(\cdot\|0.24, 0.31)$
k_{cp} in 1/day	$\mathrm{LN}(\cdot\|0.04, 0.3)$
k_{pc} in 1/day	$\mathrm{LN}(\cdot\|0.04, 0.3)$
v in L	$\mathrm{LN}(\cdot\|5.6, 0.21)$
σ in dimensionless	$\mathrm{LN}(\cdot\|0.1, 0.1)$

Table D.3: Parametric prior distribution for candidate model m=2.

Parameter	Prior distribution
k_0 in L/day	$\mathrm{LN}(\cdot\|0.12, 0.31)$
Δ in mg/day	$\mathrm{LN}(\cdot\|1.2, 0.31)$
c_{50} in mg/L	$\mathrm{LN}(\cdot\|10, 0.3)$
k_{cp} in 1/day	$\mathrm{LN}(\cdot\|0.04, 0.3)$
k_{pc} in 1/day	$\mathrm{LN}(\cdot\|0.04, 0.3)$
v in L	$\mathrm{LN}(\cdot\|5.6, 0.21)$
σ in dimensionless	$\mathrm{LN}(\cdot\|0.1, 0.1)$

Table D.4: Parametric prior distribution for candidate model m=3.

Parameter	Prior distribution
k_{cl} in L/day	$\mathrm{LN}(\cdot\|0.24, 0.31)$
v in L	$\mathrm{LN}(\cdot\|5.6, 0.21)$
$c_{r,0}$ in mg/L	$\mathrm{LN}(\cdot\|0.069, 0.092)$
K_d in mg/L	$\mathrm{LN}(\cdot\|0.067, 0.15)$
k_{int} in 1/day	$\mathrm{LN}(\cdot\|0.024, 0.19)$
k_{dg} in 1/day	$\mathrm{LN}(\cdot\|10.5, 0.17)$
σ in dimensionless	$\mathrm{LN}(\cdot\|0.1, 0.1)$

Table D.5: Parametric prior distribution for candidate model m=4.

Candidate model	Prior probability
m=1	0.1
m=2	0.35
m=3	0.2
m=4	0.35

Table D.6: Structural prior distribution.

References

Abbas, Abul K, Andrew H Lichtman, and Shiv Pillai (2021). *Cellular and Molecular Immunology*. Elsevier Health Sciences.

Abrantes, João A et al. (2019). "Handling interoccasion variability in model-based dose individualization using therapeutic drug monitoring data". In: *British Journal of Clinical pharmacology* 85.6, pp. 1326–1336.

Adan, Aysun et al. (2017). "Flow cytometry: basic principles and applications". In: *Critical Reviews in Biotechnology* 37.2, pp. 163–176.

Agarap, Abien Fred (2018). "Deep learning using rectified linear units (relu)". In: *ArXiv preprint arXiv:1803.08375*.

Agarwal, Rishabh, Dale Schuurmans, and Mohammad Norouzi (2020). "An optimistic perspective on offline reinforcement learning". In: *International Conference on Machine Learning*. PMLR, pp. 104–114.

Agbinya, Johnson I (2022). "Markov chain and its applications". In: *Applied Data Analytics-Principles and Applications*. River Publishers, pp. 1–15.

Akaike, Hirotugu (1972). "Information theory and an extension of the maximum likelihood principle". In: *Proceedings of the 2nd International Symposium for Information Theory*. Supplementary to Problems of Control and Information Theory, pp. 267–281.

— (1974). "A new look at the statistical model identification". In: *IEEE Transactions on Automatic Control* 19.6, pp. 716–723.

— (1978). "On the Likelihood of a Time Series Model". In: *Journal of the Royal Statistical Society. Series D (The Statistician)* 27.3/4, pp. 217–235.

Almoslem, Mohammed et al. (2022). "Kinetics of Drug Action: A PKPD Approach". In: *The ADME Encyclopedia: A Comprehensive Guide on Biopharmacy and Pharmacokinetics*. Springer, pp. 601–613.

Amari, Shun-ichi (1993). "Backpropagation and stochastic gradient descent method". In: *Neurocomputing* 5.4-5, pp. 185–196.

An, Guohua (2017). "Small-molecule compounds exhibiting target-mediated drug disposition (TMDD): a minireview". In: *The Journal of Clinical Pharmacology* 57.2, pp. 137–150.

Anderson, Jeffrey L et al. (2012). "A randomized and clinical effectiveness trial comparing two pharmacogenetic algorithms and standard care for individualizing warfarin dosing (CoumaGen-II)". In: *Circulation* 125.16, pp. 1997–2005.

Aranzana-Climent, Vincent et al. (2024). "Integration of individual preclinical and clinical anti-infective PKPD data to predict clinical study outcomes". In: *Clinical and Translational Science* 17.7, e13870.

Augustin, David (2024). "Chi: A Python package for treatment response modelling". In: *Journal of Open Source Software* 9.94, p. 5925.

Augustin, David, Ben Lambert, Martin Robinson, et al. (2023). "Simulating clinical trials for model-informed precision dosing: Using warfarin treatment as a use case". In: *Frontiers in Pharmacology* 14.

Augustin, David, Ben Lambert, Ken Wang, et al. (2023). "Filter inference: A scalable nonlinear mixed effects inference approach for snapshot time series data". In: *PLOS Computational Biology* 19.5, e1011135.

Augustin, David, Antje-Christine Walz, et al. (2022). "Treatment response prediction: Is model selection unreliable?" In: *BioRxiv*, pp. 2022–03.

Azer, Karim et al. (2021). "History and future perspectives on the discipline of quantitative systems pharmacology modeling and its applications". In: *Frontiers in Physiology* 12, p. 637999.

Baird, Leemon (1995). "Residual algorithms: Reinforcement learning with function approximation". In: *Machine Learning Proceedings 1995*. Elsevier, pp. 30–37.

Bauer, Robert J. (2019). "NONMEM Tutorial Part I: Description of Commands and Options, With Simple Examples of Population Analysis". In: *CPT: Pharmacometrics & Systems Pharmacology* 8.8, pp. 525–537.

Beaumont, Mark A. (2019). "Approximate Bayesian Computation". In: *Annual Review of Statistics and Its Application* 6.1, pp. 379–403.

Bellman, Richard (1966). "Dynamic programming". In: *Science* 153.3731, pp. 34–37.

Betancourt, Michael (2017). "A conceptual introduction to Hamiltonian Monte Carlo". In: *ArXiv preprint arXiv:1701.02434*.

Biliouris, K et al. (2018). "A pre-clinical quantitative model predicts the pharmacokinetics/pharmacodynamics of an anti-BDCA2 monoclonal antibody in humans". In: *Journal of Pharmacokinetics and Pharmacodynamics* 45, pp. 817–827.

Bischoff, KB and RL Dedrick (1968). "Thiopental pharmacokinetics". In: *Journal of Pharmaceutical Sciences* 57.8, pp. 1346–1351.

Bivard, Andrew, Longting Lin, and Mark W Parsonsb (2013). "Review of stroke thrombolytics". In: *Journal of Stroke* 15.2, p. 90.

Blei, David M, Alp Kucukelbir, and Jon D McAuliffe (2017). "Variational inference: A review for statisticians". In: *Journal of the American Statistical Association* 112.518, pp. 859–877.

Boyd, Stephen P and Lieven Vandenberghe (2004). *Convex optimization*. Cambridge University Press.

Braak, Cajo JF Ter (2006). "A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces". In: *Statistics and Computing* 16, pp. 239–249.

Broeker, A et al. (2019). "Towards precision dosing of vancomycin: a systematic evaluation of pharmacometric models for Bayesian forecasting". In: *Clinical Microbiology and Infection* 25.10, 1286–e1.

Bronte, Giuseppe et al. (2014). "Are erlotinib and gefitinib interchangeable, opposite or complementary for non-small cell lung cancer treatment? Biological, pharmacological and clinical aspects". In: *Critical Reviews in Oncology/Hematology* 89.2, pp. 300–313.

Brooks, Steve et al. (2011). *Handbook of Markov Chain Monte Carlo*. CRC press.

Browning, Alexander P, Niloufar Ansari, et al. (2022). "Identifying cell-to-cell variability in internalization using flow cytometry". In: *Journal of the Royal Society Interface* 19.190, p. 20220019.

Browning, Alexander P, Christopher Drovandi, et al. (2022). "Efficient inference and identifiability analysis for differential equation models with random parameters". In: *PLOS Computational Biology* 18.11, e1010734.

Buatois, Simon et al. (2018). "Comparison of Model Averaging and Model Selection in Dose Finding Trials Analyzed by Nonlinear Mixed Effect Models". In: *The AAPS Journal* 20.3, p. 56.

Bürkner, Paul-Christian, Jonah Gabry, and Aki Vehtari (2021). "Efficient leave-one-out cross-validation for Bayesian non-factorized normal and Student-t models". In: *Computational Statistics* 36.2, pp. 1243–1261.

Burnham, Kenneth P. and David R. Anderson (2002). *Model selection and multimodel inference*. Second. A practical information-theoretic approach. Springer-Verlag, New York, pp. xxvi+488.

Cavanaugh, Joseph E. and Andrew A. Neath (2019). "The Akaike information criterion: Background, derivation, properties, application, interpretation, and refinements". In: *WIREs Computational Statistics* 11.3, e1460.

Chib, Siddhartha and Edward Greenberg (1995). "Understanding the metropolis-hastings algorithm". In: *The American Statistician* 49.4, pp. 327–335.

Clarkston, Kathryn et al. (2019). "Development of infliximab target concentrations during induction in pediatric Crohn disease patients". In: *Journal of Pediatric Gastroenterology and Nutrition* 69.1, pp. 68–74.

Clerx, Michael, David Augustin, et al. (2022). "Derivative-based Inference for Cell and Channel Electrophysiology Models". In: *2022 Computing in Cardiology (CinC)*. Vol. 498. IEEE, pp. 1–4.

Clerx, Michael, Pieter Collins, et al. (2016). "Myokit: A simple interface to cardiac cellular electrophysiology". In: *Progress in Biophysics and Molecular Biology* 120.1–3, pp. 100–114.

Clerx, Michael, Martin Robinson, et al. (2019). "Probabilistic Inference on Noisy Time Series (PINTS)". In: *Journal of Open Research Software* 7.1, p. 23.

Cowles, Mary Kathryn and Bradley P Carlin (1996). "Markov chain Monte Carlo convergence diagnostics: a comparative review". In: *Journal of the American Statistical Association* 91.434, pp. 883–904.

Cox, David Roxbee (2006). *Principles of statistical inference*. Cambridge University Press.

Cranmer, Miles (2023). "Interpretable Machine Learning for the Physical Sciences". PhD thesis. Princeton University.

Creswell, Richard et al. (2020). "Using flexible noise models to avoid noise model misspecification in inference of differential equation time series models". In: *arXiv preprint arXiv:2011.04854*.

Csajka, Chantal and Davide Verotta (2006). "Pharmacokinetic–pharmacodynamic modelling: history and perspectives". In: *Journal of Pharmacokinetics and Pharmacodynamics* 33, pp. 227–279.

Darwich, A S et al. (2017). "Why Has Model-Informed Precision Dosing Not Yet Become Common Clinical Reality? Lessons From the Past and a Roadmap for the Future". In: *Clinical Pharmacology & Therapeutics* 101.5, pp. 646–656.

Dirks, Nathanael L and Bernd Meibohm (2010). "Population pharmacokinetics of therapeutic monoclonal antibodies". In: *Clinical Pharmacokinetics* 49, pp. 633–659.

Dixit, Purushottam D et al. (2019). "Maximum entropy framework for inference of cell population heterogeneity in signaling networks". In: *BioRxiv*, p. 137513.

— (2020). "Maximum entropy framework for predictive inference of cell population heterogeneity and responses in signaling networks". In: *Cell Systems* 10.2, pp. 204–212.

Draper, David (1995). "Assessment and Propagation of Model Uncertainty". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 57.1, pp. 45–97.

Drovandi, Christopher et al. (2022). "Population Calibration using Likelihood-Free Bayesian Inference". In: *ArXiv preprint arXiv:2202.01962*.

Ebid, Abdel Hameed IM et al. (2021). "Population PK-PD-PD Modeling of Recombinant Follicle Stimulating Hormone in In Vitro Fertilization/Intracytoplasmic Sperm Injection: Implications on Dosing and Timing of Gonadotrophin Therapy". In: *The Journal of Clinical Pharmacology* 61.5, pp. 700–713.

Faelens, Ruben et al. (2021). "Model-informed precision dosing during infliximab induction therapy reduces variability in exposure and endoscopic improvement between patients with ulcerative colitis". In: *Pharmaceutics* 13.10, p. 1623.

FDA (2010). *COUMADIN® TABLETS (Warfarin Sodium Tablets, USP) Crystalline COUMADIN® FOR INJECTION (Warfarin Sodium for Injection, USP)*. https://www.accessdata.fda.gov/drugsatfda_docs/label/2010/009218s108lbl.pdf. Accessed: 2022-10-03.

— (2011). *Coumadin full prescribing information*. https://www.accessdata.fda.gov/drugsatfda_docs/label/2011/009218s107lbl.pdf. Accessed: 2022-10-03.

Fernandez-Martinez, Juan L and Zulima Fernandez-Muniz (2020). "The curse of dimensionality in inverse problems". In: *Journal of Computational and Applied Mathematics* 369, p. 112571.

Fong, E and C C Holmes (2020). "On the marginal likelihood and cross-validation". In: *Biometrika* 107.2, pp. 489–496.

Frazier, David T et al. (2022). "Bayesian inference using synthetic likelihood: asymptotics and adjustments". In: *Journal of the American Statistical Association* just-accepted, pp. 1–28.

Friel, Nial and Jason Wyse (2012). "Estimating the evidence–a review". In: *Statistica Neerlandica* 66.3, pp. 288–308.

Gage, Brian F et al. (2008). "Use of pharmacogenetic and clinical factors to predict the therapeutic dose of warfarin". In: *Clinical Pharmacology & Therapeutics* 84.3, pp. 326–331.

Gaon, Maor and Ronen Brafman (2020). "Reinforcement learning with non-markovian rewards". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04, pp. 3980–3987.

Gastmans, Heleen et al. (2022). "Systematic comparison of hospital-wide Standard and Model-based therapeutic drug monitoring of vancomycin in adults". In: *Pharmaceutics* 14.7, p. 1459.

Gelman, Andrew, John B Carlin, et al. (1995). *Bayesian data analysis*. Chapman and Hall/CRC.

Gelman, Andrew, Jessica Hwang, and Aki Vehtari (2014). "Understanding predictive information criteria for Bayesian models". In: *Statistics and Computing* 24.6, pp. 997–1016.

Gelman, Andrew, Aki Vehtari, et al. (2020). "Bayesian workflow". In: *ArXiv preprint arXiv:2011.01808*.

Gibiansky, Leonid et al. (2008). "Approximations of the target-mediated drug disposition model and identifiability of model parameters". In: *Journal of Pharmacokinetics and Pharmacodynamics* 35, pp. 573–591.

Gill, Katherine L et al. (2016). "Potential sources of inter-subject variability in monoclonal antibody pharmacokinetics". In: *Clinical Pharmacokinetics* 55.7, pp. 789–805.

Gillette, James R, Jerry R Mitchell, and Bernard B Brodie (1974). "Biochemical mechanisms of drug toxicity". In: *Annual Review of Pharmacology* 14.1, pp. 271–288.

Glassman, Patrick M and Vladimir R Muzykantov (2019). "Pharmacokinetic and pharmacodynamic properties of drug delivery systems". In: *Journal of Pharmacology and Experimental Therapeutics* 370.3, pp. 570–580.

Goncalves, Antonio et al. (2020). "Model Averaging in Viral Dynamic Models". In: *The AAPS Journal* 22.2, p. 48.

Gong, Inna Y et al. (2011). "Prospective evaluation of a pharmacogenetics-guided warfarin loading and maintenance dose regimen for initiation of therapy". In: *Blood, The Journal of the American Society of Hematology* 118.11, pp. 3163–3171.

González-Sales, Mario et al. (2021). "Population pharmacokinetic analysis of entrectinib in pediatric and adult patients with advanced/metastatic solid tumors: support of new drug application submission". In: *Cancer Chemotherapy and Pharmacology* 88, pp. 997–1007.

Goodsell, David S et al. (2020). "RCSB Protein Data Bank: Enabling biomedical research and drug discovery". In: *Protein Science* 29.1, pp. 52–65.

Haario, Heikki, Marko Laine, et al. (2006). "DRAM: efficient adaptive MCMC". In: *Statistics and Computing* 16, pp. 339–354.

Haario, Heikki, Eero Saksman, and Johanna Tamminen (2001). "An Adaptive Metropolis Algorithm". In: *Bernoulli* 7.2, pp. 223–242.

Hallgrimsson, Benedikt and Brian K Hall (2005). "Variation and variability: central concepts in biology". In: *Variation*. Elsevier, pp. 1–7.

Hamberg, A-K, M-L Dahl, et al. (2007). "A PK–PD model for predicting the impact of age, CYP2C9, and VKORC1 genotype on individualization of warfarin therapy". In: *Clinical Pharmacology & Therapeutics* 81.4, pp. 529–538.

Hamberg, A-K, Lena E Friberg, et al. (2013). "Warfarin dose prediction in children using pharmacometric bridging—comparison with published pharmacogenetic dosing algorithms". In: *European Journal of Clinical Pharmacology* 69.6, pp. 1275–1283.

Hamberg, A-K, Jacob Hellman, et al. (2015). "A Bayesian decision support tool for efficient dose individualization of warfarin in adults and children". In: *BMC Medical Informatics and Decision Making* 15.1, pp. 1–9.

Hamberg, A-K and Mia Wadelius (2014). "Pharmacogenetics-based warfarin dosing in children". In: *Pharmacogenomics* 15.3, pp. 361–374.

Hamberg, A-K, Mia Wadelius, et al. (2010). "A pharmacometric model describing the relationship between warfarin dose and INR response with respect to variations in CYP2C9, VKORC1, and age". In: *Clinical Pharmacology & Therapeutics* 87.6, pp. 727–734.

Hanahan, Douglas and Robert A Weinberg (2000). "The hallmarks of cancer". In: *Cell* 100.1, pp. 57–70.

— (2011). "Hallmarks of cancer: the next generation". In: *Cell* 144.5, pp. 646–674.

Hansen, Nikolaus, Sibylle D Müller, and Petros Koumoutsakos (2003). "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)". In: *Evolutionary Computation* 11.1, pp. 1–18.

Hartmann, S, K Biliouris, L J Lesko, et al. (2016). "Quantitative systems pharmacology model to predict the effects of commonly used anticoagulants on the human coagulation network". In: *CPT: Pharmacometrics & Systems Pharmacology* 5.10, pp. 554–564.

— (2020). "Quantitative systems pharmacology model-based predictions of clinical endpoints to optimize warfarin and rivaroxaban anti-thrombosis therapy". In: *Frontiers in Pharmacology* 11, p. 1041.

Hartmann, S, K Biliouris, Himanshu Naik, et al. (2020). "A clinical population pharmacokinetic/pharmacodynamic model for BIIB059, a monoclonal antibody for the

treatment of systemic and cutaneous lupus erythematosus". In: *Journal of Pharmacokinetics and Pharmacodynamics* 47, pp. 255–266.

Hasenauer, Jan et al. (2011). "Identification of models of heterogeneous cell populations from population snapshot data". In: *BMC Bioinformatics* 12.1, pp. 1–15.

Hastings, W Keith (1970). "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* 57.1, pp. 97–109.

Herbst, Roy S (2004). "Review of epidermal growth factor receptor biology". In: *International Journal of Radiation Oncology* Biology* Physics* 59.2, S21–S26.

Hoffman, Matthew D, Andrew Gelman, et al. (2014). "The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." In: *Journal of Machine Learning Research* 15.1, pp. 1593–1623.

Holden, Philip B et al. (2018). "ABC for climate: dealing with expensive simulators". In: *Handbook of Approximate Bayesian Computation*. Chapman and Hall/CRC, pp. 569–595.

Holford, N. H. G., H. C. Kimko, et al. (2000). "Simulation of clinical trials". In: *Annual Review of Pharmacology and Toxicology* 40.1, pp. 209–234.

Holford, N. H. G., S. C. Ma, and B. A. Ploeger (2010). "Clinical trial simulation: a review". In: *Clinical Pharmacology & Therapeutics* 88.2, pp. 166–182.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White (1989). "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5, pp. 359–366.

Hucka, M. et al. (2003). "The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models". In: *Bioinformatics* 19.4, pp. 524–531.

Hughes, Alex J et al. (2014). "Single-cell western blotting". In: *Nature Methods* 11.7, pp. 749–755.

Hughes, Jasmine H, Janel Long-Boyle, and Ron J Keizer (2024). "Maximum a posteriori Bayesian methods out-perform non-compartmental analysis for busulfan precision dosing". In: *Journal of Pharmacokinetics and Pharmacodynamics*, pp. 1–10.

IWPC (2009). "Estimation of the warfarin dose with clinical and pharmacogenetic data". In: *New England Journal of Medicine* 360.8, pp. 753–764.

Johnson, Julie A, Kelly E Caudle, et al. (2017). "Clinical Pharmacogenetics Implementation Consortium (CPIC) guideline for pharmacogenetics-guided warfarin dosing: 2017 update". In: *Clinical Pharmacology & Therapeutics* 102.3, pp. 397–404.

Johnson, Julie A, Li Gong, et al. (2011). "Clinical Pharmacogenetics Implementation Consortium Guidelines for CYP2C9 and VKORC1 genotypes and warfarin dosing". In: *Clinical Pharmacology & Therapeutics* 90.4, pp. 625–629.

Jones, Galin L and Qian Qin (2022). "Markov chain Monte Carlo in practice". In: *Annual Review of Statistics and Its Application* 9, pp. 557–578.

Joyce, Paul and Paul Marjoram (2008). "Approximately Sufficient Statistics and Bayesian Computation". In: *Statistical Applications in Genetics and Molecular Biology* 7.1.

Kaitin, Kenneth I (2010). "Deconstructing the drug development process: the new face of innovation". In: *Clinical Pharmacology & Therapeutics* 87.3, pp. 356–361.

Karlsson, MO and Lewis B Sheiner (1993). "The importance of modeling interoccasion variability in population pharmacokinetic analyses". In: *Journal of Pharmacokinetics and Biopharmaceutics* 21.6, pp. 735–750.

Keizer, Ron J et al. (2018). "Model-informed precision dosing at the bedside: scientific challenges and opportunities". In: *CPT: Pharmacometrics & Systems Pharmacology* 7.12, pp. 785–787.

Kennedy, James and Russell Eberhart (1995). "Particle swarm optimization". In: *Proceedings of ICNN'95-International Conference on Neural Networks*. Vol. 4. IEEE, pp. 1942–1948.

Keutzer, Lina and Ulrika SH Simonsson (2020). "Individualized dosing with high inter-occasion variability is correctly handled with model-informed precision dosing—using rifampicin as an example". In: *Frontiers in Pharmacology* 11, p. 794.

Khorikov, Vladimir (2020). *Unit Testing Principles, Practices, and Patterns*. Simon and Schuster.

Khozin, Sean et al. (2014). "US Food and Drug Administration approval summary: erlotinib for the first-line treatment of metastatic non-small cell lung cancer with epidermal growth factor receptor exon 19 deletions or exon 21 (L858R) substitution mutations". In: *The Oncologist* 19.7, pp. 774–779.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *ArXiv preprint arXiv:1412.6980*.

Klein, Sanford and Gregory Nellis (2011). *Thermodynamics*. Cambridge University Press.

Kobuchi, Shinji et al. (2024). "Translational PK-PD/TD modeling of antitumor effects and peripheral neuropathy in gemcitabine and nab-paclitaxel chemotherapy from xenograft mice to patients for optimal dose and schedule". In: *Cancer Chemotherapy and Pharmacology* 93.4, pp. 365–379.

Koole, Simone N et al. (2024). "Lessons learned from postmarketing withdrawals of expedited approvals for oncology drug indications". In: *The Lancet Oncology* 25.3, e126–e135.

Kumar, Anil, Arti Singh, et al. (2015). "A review on Alzheimer's disease pathophysiology and its management: an update". In: *Pharmacological Reports* 67.2, pp. 195–203.

Kumar, Ravin et al. (2019). "ArviZ a unified library for exploratory analysis of Bayesian models in Python". In: *Journal of Open Source Software* 4.33, p. 1143.

Lambert, Ben, David J Gavaghan, and Simon J Tavener (2021). "A Monte Carlo method to estimate cell population heterogeneity from cell snapshot data". In: *Journal of Theoretical Biology* 511, p. 110541.

Lambert, Ben and Aki Vehtari (2022). "R: A robust MCMC convergence diagnostic with uncertainty using decision tree classifiers". In: *Bayesian Analysis* 17.2, pp. 353–379.

Laverty, HG et al. (2011). "How can we improve our understanding of cardiovascular safety liabilities to develop safer medicines?" In: *British Journal of Pharmacology* 163.4, pp. 675–693.

Lavielle, M (2014). *Mixed Effects Models for the Population Approach: Models, Tasks, Methods and Tools*. 1st ed. Chapman and Hall/CRC.

Lin, Lan-Hsi et al. (2023). "Population pharmacokinetics and pharmacodynamics of carfilzomib in combination with rituximab, ifosfamide, carboplatin, and etoposide in adult patients with relapsed/refractory diffuse large B cell lymphoma". In: *Targeted Oncology* 18.5, pp. 685–695.

Lin, Long-Ji (1992). "Self-improving reactive agents based on reinforcement learning, planning and teaching". In: *Machine Learning* 8, pp. 293–321.

Lin, Yvonne S et al. (2021). "Sources of interindividual variability". In: *Enzyme Kinetics in Drug Metabolism: Fundamentals and Applications*, pp. 481–550.

Luecken, Malte D and Fabian J Theis (2019). "Current best practices in single-cell RNA-seq analysis: a tutorial". In: *Molecular Systems Biology* 15.6, e8746.

Lunn, David J et al. (2002). "Bayesian analysis of population PK/PD models: general concepts and software". In: *Journal of Pharmacokinetics and Pharmacodynamics* 29, pp. 271–307.

Ma, Yuxiang et al. (2021). "Therapeutic drug monitoring of docetaxel by pharmacokinetics and pharmacogenetics: A randomized clinical trial of AUC-guided dosing in nonsmall cell lung cancer". In: *Clinical and Translational Medicine* 11.4.

Mager, Donald E (2006). "Target-mediated drug disposition and dynamics". In: *Biochemical Pharmacology* 72.1, pp. 1–10.

Mager, Donald E and William J Jusko (2001). "General Pharmacokinetic Model for Drugs Exhibiting Target-Mediated Drug Disposition". In: *Journal of Pharmacokinetics and Pharmacodynamics* 28.6, pp. 507–532.

Mager, Donald E and Wojciech Krzyzanski (2005). "Quasi-equilibrium pharmacokinetic model for drugs exhibiting target-mediated drug disposition". In: *Pharmaceutical Research* 22, pp. 1589–1596.

Maier, Corinna, Niklas Hartung, Jana de Wiljes, et al. (2020). "Bayesian data assimilation to support informed decision making in individualized chemotherapy". In: *CPT: Pharmacometrics & Systems Pharmacology* 9.3, pp. 153–164.

Maier, Corinna, Niklas Hartung, Charlotte Kloft, et al. (2021). "Reinforcement learning and Bayesian data assimilation for model-informed precision dosing in oncology". In: *CPT: Pharmacometrics & Systems Pharmacology* 10.3, pp. 241–254.

Mandema, Jaap W, Davide Verotta, and Lewis B Sheiner (1992). "Building population pharmacokineticpharmacodynamic models. I. Models for covariate effects". In: *Journal of Pharmacokinetics and Biopharmaceutics* 20.5, pp. 511–528.

Matsumoto, Kazuaki et al. (2022). "Clinical practice guidelines for therapeutic drug monitoring of vancomycin in the framework of model-informed precision dosing: a consensus review by the Japanese society of chemotherapy and the Japanese society of therapeutic drug monitoring". In: *Pharmaceutics* 14.3, p. 489.

McKinley, Trevelyan J et al. (2018). "Approximate Bayesian computation and simulation-based inference for complex stochastic epidemic models". In: *Statistical Science* 33.1, pp. 4–18.

Merlé, Yann, Azzedine Aouimer, and Michel Tod (2004). "Impact of model misspecification at design (and/or) estimation step in population pharmacokinetic studies". In: *Journal of Biopharmaceutical Statistics* 14.1, pp. 213–227.

Metropolis, Nicholas et al. (1953). "Equation of state calculations by fast computing machines". In: *The Journal of Chemical Physics* 21.6, pp. 1087–1092.

Mnih, Volodymyr et al. (2013). "Playing atari with deep reinforcement learning". In: *ArXiv preprint arXiv:1312.5602*.

Moini, Jahangir, Anthony Logalbo, and Jennifer G Schnellmann (2023). "Chapter 4 - Pharmacodynamics". In: *Neuropsychopharmacology*. Ed. by Jahangir Moini, Anthony Logalbo, and Jennifer G Schnellmann. Academic Press, pp. 55–74.

Moon, Todd K (1996). "The expectation-maximization algorithm". In: *IEEE Signal Processing Magazine* 13.6, pp. 47–60.

Moore, Brett L et al. (2004). "Intelligent Control of Closed-Loop Sedation in Simulated ICU Patients." In: *Flairs Conference*, pp. 109–114.

Mourya, Alkesh S and Rajesh Kumar Pandey (2024). "A study on factors attributed to failure of pharmaceutical products". In: *Applied Science and Biotechnology Journal for Advanced Research* 3.2, pp. 12–21.

Mueller, Laurence (2019). *Conceptual breakthroughs in evolutionary ecology*. Academic Press.

Muller, Patrick Y and Mark N Milton (2012). "The determination and interpretation of the therapeutic index in drug development". In: *Nature Reviews Drug Discovery* 11.10, pp. 751–761.

Musteata, Florin Marcel (2011). "Monitoring free drug concentrations: challenges". In: *Bioanalysis* 3.15, pp. 1753–1768.

Neal, Radford M. (2010). "MCMC Using Hamiltonian Dynamics". In: *Handbook of Markov Chain Monte Carlo* 54, pp. 113–162.

Neath, Andrew A and Joseph E Cavanaugh (2012). "The Bayesian information criterion: background, derivation, and applications". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 4.2, pp. 199–203.

Nebendahl, Klaus (2000). "Routes of administration". In: *The Laboratory Rat*. Elsevier, pp. 463–483.

Norris, James R (1998). *Markov chains*. 2. Cambridge University Press.

Oliver, Meredith B et al. (2023). "Considerations for implementation of vancomycin Bayesian software monitoring in a level IV NICU population within a multisite health system". In: *American Journal of Health-System Pharmacy* 80.11, pp. 670–677.

Paalzow, Lennart K (1995). "Torsten Teorell, the father of pharmacokinetics". In: *Upsala Journal of Medical Sciences* 100.1, pp. 41–46.

Paszke, Adam et al. (2019). "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in Neural Information Processing Systems* 32.

Peck, Richard W (2021). "Precision dosing: an industry perspective". In: *Clinical Pharmacology and Therapeutics* 109.1, p. 47.

Peters, Sheila Annie (2021). *Physiologically based pharmacokinetic (PBPK) modeling and simulations: principles, methods, and applications in the pharmaceutical industry*. John Wiley & Sons.

Polasek, Thomas M, Craig R Rayner, et al. (2019). "Toward dynamic prescribing information: codevelopment of companion model-informed precision dosing tools in drug development". In: *Clinical Pharmacology in Drug Development* 8.4, pp. 418–425.

Polasek, Thomas M and Amin Rostami-Hodjegan (2020). "Virtual twins: understanding the data required for model-informed precision dosing". In: *Clinical Pharmacology and Therapeutics* 107.4, pp. 742–745.

Polasek, Thomas M, Sepehr Shakib, and Amin Rostami-Hodjegan (2018). "Precision dosing in clinical medicine: present and future". In: *Expert Review of Clinical Pharmacology* 11.8, pp. 743–746.

Prangle, Dennis (2017). "Adapting the ABC distance function". In: *Bayesian Analysis* 12.1, pp. 289–309.

Price, Leah F et al. (2018). "Bayesian synthetic likelihood". In: *Journal of Computational and Graphical Statistics* 27.1, pp. 1–11.

Rackauckas, Chris et al. (2020). "Accelerated predictive healthcare analytics with pumas, a high performance pharmaceutical modeling and simulation platform". In: *BioRxiv*.

Raue, A. et al. (2009). "Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood". In: *Bioinformatics* 25.15, pp. 1923–1929.

Ribba, Benjamin, Dominic Stefan Bräm, et al. (2022). "Model enhanced reinforcement learning to enable precision dosing: A theoretical case study with dosing of propofol". In: *CPT: Pharmacometrics & Systems Pharmacology* 11.11, pp. 1497–1510.

Ribba, Benjamin, Sherri Dudal, et al. (2020). "Model-informed artificial intelligence: reinforcement learning for precision dosing". In: *Clinical Pharmacology & Therapeutics* 107.4, pp. 853–857.

Ribba, Benjamin, N. H. G. Holford, et al. (2014). "A Review of Mixed-Effects Models of Tumor Growth and Effects of Anticancer Drug Treatment Used in Population Analysis". In: *CPT: Pharmacometrics & Systems Pharmacology* 3.5, p. 113.

Robert, Christian P and Darren Wraith (2009). "Computational methods for Bayesian model choice". In: *ArXiv preprint arXiv:0907.5123*.

Roberts, Gareth O and Jeffrey S Rosenthal (2004). "General state space Markov chains and MCMC algorithms". In.

Roda, Giulia et al. (2020). "Crohn's disease". In: *Nature Reviews Disease Primers* 6.1, p. 22.

Rost, Simone et al. (2004). "Mutations in VKORC1 cause warfarin resistance and multiple coagulation factor deficiency type 2". In: *Nature* 427.6974, pp. 537–541.

Rowland, Malcolm, Leslie Z Benet, and Garry G Graham (1973). "Clearance concepts in pharmacokinetics". In: *Journal of Pharmacokinetics and Biopharmaceutics* 1.2, pp. 123–136.

Ruberg, Stephen J et al. (2023). "Application of Bayesian approaches in drug development: starting a virtuous cycle". In: *Nature Reviews Drug Discovery* 22.3, pp. 235–250.

Rudin, Charles M et al. (2008). "Pharmacogenomic and pharmacokinetic determinants of erlotinib toxicity". In: *Journal of Clinical Oncology: Official Journal of the American Society of Clinical Oncology* 26.7, p. 1119.

Salahudeen, Mohammed Saji and Prasad S Nishtala (2017). "An overview of pharmacodynamic modelling, ligand-binding approach and its application in clinical practice". In: *Saudi Pharmaceutical Journal* 25.2, pp. 165–175.

Sánchez-Hernández, José Germán et al. (2020). "Biomarkers of disease activity and other factors as predictors of adalimumab pharmacokinetics in inflammatory bowel disease". In: *European Journal of Pharmaceutical Sciences* 150, p. 105369.

Sandborn, William J and Stephen B Hanauer (2002). "Infliximab in the treatment of Crohn's disease: a user's guide for clinicians". In: *Official Journal of the American College of Gastroenterology* 97.12, pp. 2962–2972.

Schorning, Kirsten et al. (2016). "Model selection versus model averaging in dose finding studies". In: *Statistics in Medicine* 35.22, pp. 4021–4040.

Schräpel, Christina et al. (2021). "External model performance evaluation of twelve infliximab population pharmacokinetic models in patients with inflammatory bowel disease". In: *Pharmaceutics* 13.9, p. 1368.

Schwarz, Gideon (1978). "Estimating the dimension of a model". In: *The Annals of Statistics*, pp. 461–464.

Sébastien, Bernard et al. (2016). "Model averaging inconcentration–QT analyses". In: *Pharmaceutical Statistics* 15.6, pp. 450–458.

Seinfeld, John H, Leon Lapidus, and Myungkyu Hwang (1970). "Review of numerical integration techniques for stiff ordinary differential equations". In: *Industrial & Engineering Chemistry Fundamentals* 9.2, pp. 266–275.

Sheiner, Lewis B (1969). "Computer-aided long-term anticoagulation therapy". In: *Computers and Biomedical Research* 2.6, pp. 507–518.

Sheiner, Lewis B and Stuart Beal (1982). "Bayesian individualization of pharmacokinetics: simple implementation and comparison with non-Bayesian methods". In: *Journal of Pharmaceutical Sciences* 71.12, pp. 1344–1348.

Sheiner, Lewis B, Stuart Beal, et al. (1979). "Forecasting individual pharmacokinetics". In: *Clinical Pharmacology & Therapeutics* 26.3, pp. 294–305.

Silk, Daniel, Sarah Filippi, and Michael PH Stumpf (2013). "Optimizing threshold-schedules for sequential approximate Bayesian computation: applications to molecular systems". In: *Statistical Applications in Genetics and Molecular Biology* 12.5, pp. 603–618.

Singh, Surjit et al. (2018). "Monoclonal antibodies: a review". In: *Current Clinical Pharmacology* 13.2, pp. 85–99.

Sjödin, Per et al. (2012). "Resequencing data provide no evidence for a human bottleneck in Africa during the penultimate glacial period". In: *Molecular Biology and Evolution* 29.7, pp. 1851–1860.

Stegmann, Gabriela et al. (2018). "Nonlinear mixed-effects modeling programs in R". In: *Structural Equation Modeling: A Multidisciplinary Journal* 25.1, pp. 160–165.

Stone, Mervyn (1977). "An Asymptotic Equivalence of Choice of Model by Cross-Validation and Akaike's Criterion". In: *Journal of the Royal Statistical Society Series B-Methodological* 39, pp. 44–47.

Sutton, Richard S and Andrew G Barto (2018). *Reinforcement learning: An introduction*. MIT press.

Tamimi, Nihad AM and Peter Ellis (2009). "Drug development: from concept to marketing!" In: *Nephron Clinical Practice* 113.3, pp. c125–c131.

Tanaka, Yusuke et al. (2024). "Postmarket safety communications on drugs approved in Japan: A 25-year analysis". In: *Clinical and Translational Science* 17.4, e13803.

Teorell, Torsten (1937a). "Kinetics of distribution of substances administered to the body, I – The extravascular modes of administration". In: *Archives Internationales de Pharmacodynamie et de Therapie* 57, pp. 205–225.

— (1937b). "Kinetics of distribution of substances administered to the body, II – The intravascular modes of administration". In: *Archives Internationales de Pharmacodynamie et de Therapie* 57, pp. 226–240.

Teorell, Torsten, Robert L Dedrick, and Peter G Condliffe (1974). *Pharmacology and pharmacokinetics*. Springer.

Ternant, David et al. (2022). "Infliximab Treatment Does Not Lead to Full TNF-α Inhibition: A Target-Mediated Drug Disposition Model". In: *Clinical Pharmacokinetics*, pp. 1–12.

Thummel, Kenneth E and Yvonne S Lin (2014). "Sources of interindividual variability". In: *Enzyme Kinetics in Drug Metabolism: Fundamentals and Applications*, pp. 363–415.

Uster, David W et al. (2021). "A model averaging/selection approach improves the predictive performance of model-informed precision dosing: vancomycin as a case study". In: *Clinical Pharmacology & Therapeutics* 109.1, pp. 175–183.

Van Hasselt, Hado, Arthur Guez, and David Silver (2016). "Deep reinforcement learning with double q-learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 30. 1.

Várkonyi, Tamás and Peter Kempler (2008). "Diabetic neuropathy: new strategies for treatment". In: *Diabetes, Obesity and Metabolism* 10.2, pp. 99–108.

Vehtari, Aki, Andrew Gelman, and Jonah Gabry (2017). "Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC". In: *Statistics and Computing* 27.5, pp. 1413–1432.

Vehtari, Aki, Andrew Gelman, Daniel Simpson, et al. (2021). "Rank-Normalization, Folding, and Localization: An Improved $\widehat{R}$ for Assessing Convergence of MCMC (with Discussion)". In: *Bayesian Analysis* 16.2, pp. 667–718.

Verhoef, Talitha I et al. (2013). "A randomized trial of genotype-guided dosing of acenocoumarol and phenprocoumon". In: *New England Journal of Medicine* 369.24, pp. 2304–2312.

Véronneau-Veilleux, Florence et al. (2020). "Nonlinear pharmacodynamics of levodopa through Parkinson's disease progression". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 30.9.

Vrugt, Jasper A et al. (2009). "Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling". In: *International Journal of Nonlinear Sciences and Numerical Simulation* 10.3, pp. 273–290.

Wadelius, Mia, Leslie Y Chen, et al. (2009). "The largest prospective warfarin-treated cohort supports genetic forecasting". In: *Blood, The Journal of the American Society of Hematology* 113.4, pp. 784–792.

Wadelius, Mia and Munir Pirmohamed (2007). "Pharmacogenetics of warfarin: current status and future challenges". In: *The Pharmacogenomics Journal* 7.2, pp. 99–111.

Wagner, John G (1981). "History of pharmacokinetics". In: *Pharmacology & therapeutics* 12.3, pp. 537–562.

Wajima, T, GK Isbister, and SB Duffull (2009). "A comprehensive model for the humoral coagulation network in humans". In: *Clinical Pharmacology & Therapeutics* 86.3, pp. 290–298.

Wang, Yaning et al. (2019). "Model-informed drug development: current US regulatory practice and future considerations". In: *Clinical Pharmacology & Therapeutics* 105.4, pp. 899–911.

Watanabe, Sumio and Manfred Opper (2010). "Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory." In: *Journal of Machine Learning Research* 11.12.

Watkins, Christopher JCH and Peter Dayan (1992). "Q-learning". In: *Machine learning* 8, pp. 279–292.

Wicha, Sebastian G et al. (2021). "From therapeutic drug monitoring to model-informed precision dosing for antibiotics". In: *Clinical Pharmacology & Therapeutics* 109.4, pp. 928–941.

Wilbaux, Mélanie et al. (2022). "Integration of pharmacokinetics, pharmacodynamics, safety, and efficacy into model-informed dose selection in oncology first-in-human study: a case of Roblitinib (FGF401)". In: *Clinical Pharmacology & Therapeutics* 112.6, pp. 1329–1339.

Wilkinson, Richard David (2013). "Approximate Bayesian computation (ABC) gives exact results under the assumption of model error". In: *Statistical Applications in Genetics and Molecular Biology* 12.2, pp. 129–141.

Wojciechowski, Jessica et al. (2022). "Population PK and PD analysis of domagrozumab in pediatric patients with Duchenne muscular dystrophy". In: *Clinical Pharmacology & Therapeutics* 112.6, pp. 1291–1302.

Wood, Simon N (2010). "Statistical inference for noisy nonlinear ecological dynamic systems". In: *Nature* 466.7310, pp. 1102–1104.

Xue, Ling, N. H. G. Holford, and Liyan Miao (2016). "Warfarin PKPD: theory, body composition and genotype". In: *Annual Meeting of the Population Approach Group in Europe. Lisbon*.

Yano, Seiichi et al. (2003). "Distribution and function of EGFR in human tissue and the effect of EGFR tyrosine kinase inhibition." In: *Anticancer research* 23.5A, pp. 3639–3650.

Yao, Yuling et al. (2018). "Using Stacking to Average Bayesian Predictive Distributions (with Discussion)". In: *Bayesian Analysis* 13.3, pp. 917–1007.

Yu, Yichao et al. (2020). "PK/PD Approaches". In: *Drug Discovery and Evaluation: Methods in Clinical Pharmacology*, pp. 1047–1069.

Yuan, Min et al. (2019). "A quick and accurate method for the estimation of covariate effects based on empirical Bayes estimates in mixed-effects modeling: correction of bias due to shrinkage". In: *Statistical Methods in Medical Research* 28.12, pp. 3568–3578.

Zadeh, Sadjad Anzabi, W Nick Street, and Barrett W Thomas (2023). "Optimizing warfarin dosing using deep reinforcement learning". In: *Journal of Biomedical Informatics* 137, p. 104267.

Zhu, Ciyou et al. (1997). "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization". In: *ACM Transactions on Mathematical Software* 23.4, pp. 550–560.

Zhu, Taiyu et al. (2020). "An insulin bolus advisor for type 1 diabetes using deep reinforcement learning". In: *Sensors* 20.18, p. 5058.