Dissertation

A Practical Framework to Improve Infectious Disease Study Design based on Antibody Kinetics

submitted by

Dipl.-Ing.

Stefan EMBACHER, BSc

for the Academic Degree of

Doctor of Medical Science

(Dr. scient. med.)

at the

Medical University of Graz

Institute for Medical Informatics, Statistics and Documentation

under the Supervision of

Assoc. Prof. Priv.-Doz. Sereina HERZOG, PhD

2026

Declaration of Academic Integrity

I hereby confirm that this thesis is the result of my own independent scholarly work. I also confirm that in all cases, where material from the work of others (in books, articles, essays, dissertations, and on the internet) is acknowledged, quotations and paraphrases are clearly indicated. No material other than that cited in the reference list has been used. I have read and understood the Medical University's regulations and procedures concerning plagiarism.
Furthermore, I hereby declare that if artificial intelligence (AI) tools were used for the generation and/or correction of certain text passages in the creation of this work, such employment was conducted in compliance with ethical principles, academic integrity, and the regulations of my university. Additionally, it was ensured that this usage was transparently disclosed and appropriately attributed.

Graz, 12th February 2026 Stefan Embacher

Disclosures

Part of this thesis has been published in:

Embacher S, Berghold A, Maertens K, Herzog SA. A Practical Framework to Design Immunization Studies Based on the Beta Distribution. Statistics in Medicine. 2025;44(23-24):e70293. DOI: 10.1002/sim.70293.

The following people contributed actively to the results of the thesis and the publication resulting from the thesis project and have given their written consent for the use of the associated data in this thesis:

- Sereina Annik Herzog affiliated with the Institute of Medical Informatics, Statistics and Documentation at the Medical University of Graz, Austria.
- Andrea Berghold affiliated with the Institute of Medical Informatics, Statistics and Documentation at the Medical University of Graz, Austria.
- Martin Hönigl affiliated with the Division of Infectious Diseases, Department of Internal Medicine at the Medical University of Graz, Austria.
- Kirsten Maertens affiliated with the Centre for the Evaluation of Vaccination, Vaccine and Infectious Diseases Institute, University of Antwerp, Belgium.

All content was authored by the thesis author, with AI assistance solely used for the linguistic optimization of the text. The author edited the content as needed and takes full responsibility for the content of the dissertation. The following tool was used:
ChatGPT (versions GPT-4, GPT-4-Turbo, and GPT-5), OpenAI, L.L.C., January 2025 – November 2025, url: https://chat.openai.com.

Throughout my doctoral studies I contributed actively to the following, thesis-unrelated publications:

- **Embacher S**, Maertens K, Herzog SA. Half-life Estimation of Pertussis-Specific Maternal Antibodies in (Pre)Term Infants After In-Pregnancy Tetanus, Diphtheria, Acellular Pertussis Vaccination. The Journal of infectious diseases. 2023;228(11): 1640–1648. DOI: 10.1093/infdis/jiad212.

- Eglseer D, Traxler M, **Embacher S**, Reiter L, Schoufour JD, Weijs PJM, et al. Nutrition and Exercise Interventions to Improve Body Composition for Persons with Overweight or Obesity Near Retirement Age: A Systematic Review and Network Meta-Analysis of Randomized Controlled Trials. Advances in Nutrition. 2023;14(3) DOI: 10.1016/j.advnut.2023.04.001

- Glachs L, **Embacher S**, Berghold A, Wildner B, Michelitsch M, Tscherne A, et al. Treatment of myopic choroidal neovascularization: a network meta-analysis and review. Graefe's Archive for Clinical and Experimental Ophthalmology. 2024;262(6): 1693-1722. DOI: 10.1007/s00417-023-06271-2

- Zurl H, **Embacher S**, Schöllnast H, Talakic E, Schöpfer-Schwab S, Pohl K, et al. MRI prostate tumor volume predicts the need for systematic biopsies in patients undergoing MRI-targeted biopsy. Minerva Urol Nephrol 2026 DOI: 10.23736/S2724-6051.25.06375-X

Acknowledgements

First and foremost, I want to sincerely thank Sereina Herzog who, in countless hours of discussion, provided patient support, guidance, and mentorship. I am deeply grateful that such an exceptional person guided me through this journey. I would also like to thank Andrea Berghold for her advice, support, and trust over the past years, and for fostering an environment in which I was able to develop academically while also growing as a person. I am also grateful to Martin Hönigl for opening new perspectives, his consistent availability and constructive feedback.
I am deeply thankful for the time and care all of them invested in this dissertation, and even more so in me as a person.
I also want to thank my colleagues at the Institute for Medical Informatics, Statistics and Documentation for the support, the shared laughter, and making my workplace a place I enjoy going to.
As a doctoral student, I received funding from the Medical University of Graz through the Doctoral School Sustainable Health Research (SHR). Parts of this work were also supported by a bilateral project of the Austrian Science Fund (I 6376-B) and the Fonds Wetenschappelijk Onderzoek (FWO; G004723N).
Most importantly, I would like to thank my family and friends for supporting me in ways they may not even recognize as support, for their belief in me as a person, and for always caring for my wellbeing in ways that truly matter. Without them as an anchor, this dissertation would not have been possible to finish.

Contents

Abbreviations

AMA-1 Apical Membrane Antigen 1

AR(1) Autoregressive of order 1

AUC Area under the Curve

FIM Fisher Information Matrix

FIMs Fisher Information Matrices

HIV Human Immunodeficiency Virus

IgA Immunoglobulin A

IgD Immunoglobulin D

IgE Immunoglobulin E

IgG Immunoglobulin G

IgM Immunoglobulin M

MHC Major Histocompatibility Complex

MLE Maximum Likelihood Estimator

ODE Ordinary Differential Equation

ODEs Ordinary Differential Equations

Q1 First Quartile

Q3 Third Quartile

SIR Susceptible, Infected, Recovered

WHO World Health Organization

List of Figures

List of Tables

Zusammenfassung

Einleitung. Ein optimal geplantes Experiment liefert Informationen effizienter als ein suboptimal designtes Experiment. Obwohl die Theorie des optimalen Designs viele Vorteile bietet, hat unsere systematische Übersichtsarbeit gezeigt, dass sie noch nicht ihren Weg in die Anwendung in Immunisierungsstudien, deren Hauptziel die Modellierung der Antikörperkinetik ist, gefunden hat. Dies könnte zum einen an mangelndem Bewusstsein und Zugänglichkeit liegen und zum anderen an den inhärenten Herausforderungen bei der Nutzung der Theorie des optimalen Designs bei komplexen Modellen. In dieser Dissertation wird ein Framework vorgestellt, das es medizinischem Fachpersonal ermöglicht, mittels leicht interpretierbarer Informationen Immunisierungsstudien zu planen.

Methoden. Unter der Annahme, dass die Antikörperkinetik bis zu einem Plateau der Dichte einer Beta-Verteilung entspricht, sind Anfangswert, die Zeit und Höhe des Maximums sowie die Zeit und Höhe des Plateaus als Information ausreichend, um einen optimalen Zeitplan zur Probennahme zu finden. Unter der Verwendung der analytischen Beziehung zwischen dem Modus und den Parametern der Verteilung kann das Studiendesign auf Grundlage der funktionalen Form der Beta-Dichte optimiert werden. Um die Robustheit des Frameworks gegenüber Misspezifikationen in den Ausgangsinformationen zu überprüfen, wurden 12 Szenarien definiert und Misspezifikationen in einem beziehungsweise zwei Parametern evaluiert.

Ergebnisse. Die Analyse der Robustheit zeigte für alle einfachen Misspezifikationen eine mediane D-Effizienz über 0.95 und für alle paarweisen Misspezifikationen eine mediane D-Effizienz von mindestens 0.9, was insgesamt für robuste Ergebnisse bezüglich der Effizienz des Studiendesigns spricht. Darüber hinaus wird eine vollständig implementierte Shiny Anwendung zur Verfügung gestellt, um eine einfache Nutzung des Frameworks ohne Programmierkenntnisse zu ermöglichen.

Diskussion. Der größte Vorteil des Frameworks und der dazugehörigen Implementierung besteht darin, dass die Optimierung von Zeitplänen zur Probenentnahme mit minimalen mathematischen Kenntnissen und Programmierfähigkeiten ermöglicht wird und dadurch einen leicht zugänglichen Einstieg für die Anwendung optimaler Designs in Immunisierungsstudien bietet.

Abstract

Introduction. An optimally designed experiment provides information more efficiently compared to suboptimally designed experiments. Despite the advantages of optimal design theory, our systematic review highlighted that its application has not yet been adopted in immunization studies where modelling antibody kinetics is the main objective. This lack of application might be attributed to limited awareness and accessibility, as well as the inherent challenges of combining complex models and the theory of optimal design. In this dissertation, a framework is introduced that enables healthcare professionals to conveniently design immunization studies using clear and easily interpretable initial information.

Methods. Assuming that antibody kinetics follow the shape of the density of a beta distribution until reaching a plateau, the baseline level, the time and height of the maximum and the time and height of the plateau are sufficient to find an optimal sampling schedule. These easily understandable information are translated into parameters of the beta distribution by using the analytical relationship between the distribution's mode and its parameters. Optimal design theory is then applied to the functional form of the beta density to obtain optimal sampling times. To assess the robustness of this framework against misspecification in the initial information, we defined 12 scenarios and discussed single and double parameter misspecification.

Results. The Robustness Analysis showed median D-efficiencies above 0.95 for all single parameter misspecifications and median D-efficiencies greater equal 0.9 across all pairs of misspecification, thereby showing generally robust results in terms of design efficiency. Additionally, a fully implemented Shiny application is provided to allow for convenient use of the framework without programming knowledge.

Discussion. The major advantage of the introduced framework, together with the provided implementation, is that it enables the optimization of sampling schedules with minimal mathematical knowledge and coding skills and thereby provides an accessible entry point for applying optimal design in immunization studies.

1. Introduction

Infectious diseases such as HIV, malaria, measles, pertussis or in the recent years Covid-19 contribute substantially to public health issues worldwide.[1–4] The immune system protects the body against pathogens through highly coordinated and complex mechanisms. Fundamentally, it consists of the fast acting and non-specific innate immunity, and the specific and longer lasting adaptive immunity.[5] Adaptive immunity against many infectious diseases is driven by antibodies and understanding the dynamics of antibodies and the change of antibody concentration over time, known as antibody kinetics, provides additional benefit in tackling issues related to the global burden of infectious diseases.[6] Information on antibody kinetics can help to predict the duration of antibody-mediated immunity, might guide vaccination strategies in diseases for which vaccines exist, and improve disease control strategies in general. Consequently, many immunization studies describe antibody kinetics.[6] Rigorous studies are needed to describe the immunological response to infections and also to evaluate the effect of interventions aiming at reducing the susceptibility to infectious diseases, infectiousness or the overall burden of infection, such as vaccination. A good study design aims to answer the research question with satisfying power, while avoiding under- and over-enrolment of participants. In studies involving longitudinal data, such as immunization studies, it is additionally crucial to determine the optimal sampling frequency and schedule to plan and design studies accordingly.

When conducting research, whether experimental, clinical or observational and irrespective of the field it is conducted in, it feels natural to think about whether the design of the experiment, such as sample size or timing of observations, influences the outcome and its conclusions. Besides the consideration, if the design could influence the outcome and conclusions, we can also think about whether there are ways to improve the 'efficiency' of the experiment, i.e. an optimal number of samples, timing or location. Since resources such as time, money, materials or participants are notoriously limited in most scientific experiments,[7, 8] there are several reasons why one would aim for optimally designed studies. We therefore want to gain as much information from an experiment, while keeping the required resources at a minimum. An optimally designed experiment provides more information about parameters of interest compared to an experiment with subop-

timal design (with an equal number of observations), or correspondingly, provides the same amount of information with fewer observations. Therefore, an optimally designed experiment achieves the same results quicker, cheaper or with less observations.[9, 10] For that reason designing a study to the best of one's knowledge, especially in the medical field, is an ethical obligation. Further, an optimal designed experiment enhances the overall scientific rigour of research and promotes a methodical approach to experimentation. The utility of optimal design theory lies in its ability to provide a structured approach to data collection and analysis. By maximizing information gain and improving resource efficiency, optimal designs ensure that experiments are cost-effective and scientifically robust contemporaneously, which not only benefits research but also adheres to ethical standards and ensures compliance with regulatory requirements, making it an indispensable part of modern experimental research. Despite its clear advantages, optimal design theory is still underutilized in practice,[11] potentially due to missing links between theory and practice or simply lack of awareness. There is a need for more research on the practical application of optimal design principles across various disciplines.[11]

Model selection is an integral part of study design and models used to analyze antibody kinetics are often complex, include parameters that are typically unmeasured in clinical practice, present identifiability issues, are difficult to interpret or are only of theoretical nature.[12–14] Inevitably, they present substantial challenges when designing a study, such as the need to provide guesses of typically unmeasured parameters or simply by the complexities arising from the model needed to design the study. Therefore, a gap between what is practically applicable and theoretical possible arises. The advantages of optimal design theory are still overlooked in many practical applications.[11] This is especially true in immunization studies, where modelling the antibody kinetics is the main objective (cp. Section 1.2.2). In these studies, study design would include the optimization of the number and timing of blood samples. Arguably, this gap between theory and practice can be attributed to lack of awareness, the complexities arising from complex models in combination with optimal design theory, being of both, analytical and numerical nature, and accessibility of convenient software solutions. Since many of the more complex models already provide challenges in estimation of the parameters, it is evident that at the stage of study design the need of prior information on these parameters of interest might be especially challenging. This dissertation focuses on optimal sampling times despite the fact that determining the required sample size to enrol neither too many nor too few participants, is an equally crucial aspect in study design.

In this dissertation, a framework is introduced that enables healthcare professionals to

conveniently design immunization studies using clear and easily interpretable initial information. This is achieved by translating the easily understandable information into parameters of the beta distribution, using the property that the mode can be expressed in terms of the parameters and then by applying optimal design theory to the functional form of the beta density to obtain optimal sampling times. Additionally, a full implementation, in form of a Shiny application is provided to allow for convenient use without programming knowledge.

To assess the robustness of the framework against misspecification in the initial information, an approach referred to as 'Robustness Analysis' is employed. In contrast to simulation studies, this approach involves no randomness but instead predefined scenarios are specified, each representing different underlying truths along with ranges of misspecification in the initial information used to design the study. In statistical research, simulation studies play a crucial role in the evaluation of new methods and the comparison of existing methods, since they allow for a systematic assessment of the methods' performance, because the underlying "truth" is known.[15] The term 'Robustness Analysis' is chosen to highlight that the main interest lies in the robustness of the framework against misspecification in the initial information and to emphasize that no random sampling is involved in this approach. Nevertheless, the underlying truth is known and is utilized to assess the effect of misspecifications on the study design and its efficiency.

The main contribution of this dissertation is the introduction of a framework that allows healthcare professionals to design an immunization study with optimally timed samples with easily understandable information. Thereby, optimal design theory and the modelling of antibody kinetics are connected, while keeping the practical relevance and applicability.
Chapter 1 'Introduction' provided a general introduction, Section 1.1 shortly introduces the key aspects of the immune system relevant to this work followed by the description of mathematical models in epidemiology in Section 1.2 including between-host models and the results of our systematic review of within-host models. Further, optimal design theory is introduced in Section 1.3 and the general problem discussed in this dissertation is described in Section 1.4. The 'Introduction' ends with a summary of the most relevant parts including the justification of the research question, the aim of this dissertation and the novelty of the research in Section 1.5. In Chapter 2 'Methods', the necessary essential concepts are introduced in Section 2.1, relevant parts of optimal design theory are described in Section 2.2, the framework is presented in Section 2.3 and the set up of the Robustness Analysis is described in Section 2.4. In Chapter 3 'Results', the Robustness

Analysis results are presented in Section 3.1 and a Shiny application is introduced in Section 3.2, followed by three examples in Section 3.3. Lastly, in Chapter 4 'Discussion', the used methods are discussed in Section 4.1 and the results are discussed in Section 4.2, followed by a discussion of the limitations in Section 4.3 and extension possibilities in Section 4.4. A final conclusion is presented in Section 4.5.

1.1. Introduction to the Immune System

The immune system is a highly coordinated, complex and dynamic system operating on several, interconnected scales by which the body protects itself against potentially harmful substances. This complexity makes it difficult to view, more so to model, the immune system as a whole.[16, 17] The following sections provide a simplistic overview of the major parts of the immune system in a conceptual way that should serve as a basis for the within-host mathematical models presented in Section 1.2.2. When being interested in the modelling of antibody kinetics, most of the mechanistic models described in Section 1.2.2 conceptually model the generation of antibodies, through for example, short- and long-lived plasma cells. We therefore focus on the humoral immunity and antibodies, and some of the basic properties of the immune system in the following subsections.
The immune system can be generally be split into innate immunity (cp. Section 1.1.1) and the adaptive immunity (cp. Section 1.1.2), while adaptive immunity can be further divided into the cell-mediated immunity and the humoral immunity.[5] However, there are various links between the innate and adaptive immune system.[5] The following short overview of the immune system is primarily based on the book by Abbas, Lichtman and Pillai 'Cellular and Molecular Immunology'.[5]

1.1.1. Innate Immunity

The innate immunity, also known as native immunity, is the first line of defence of the body. It consists of chemical and physical barriers, cellular components and humoral components. It is characterized by a fast response without a memory and therefore does not rely on prior exposure. Chemical and physical barriers are, for example, the skin and its fatty acids, antimicrobial factors in saliva and tears or acid in the stomach. It has a self-cleaning property, through coughing, sneezing or vomiting among other.
When pathogens get through these barriers, cellular components such as phagocytes (macrophages, neutrophils) or natural killer cells take over. Phagocytes engulf and destroy microbes ("phagocytosis"), while natural killer cells attack and destroy infected (stressed) cells but not the pathogens themselves. The complement system is part of the humoral, innate immune system and consists of several serum proteins. They enhance the capabil-

ity to eliminate microbes and cells from an organism, attack the pathogen's cell membrane or promote inflammation. Inflammation itself is a key component of the innate immunity, characterized by redness of skin, heat, swelling and pain, and helps to eliminate the cause of cell injury and promotes tissue repair. Often the innate immune response is followed by an antigen specific response and therefore innate and adaptive immune responses are bridged by antigen presenting cells.[5]

1.1.2. Adaptive Immunity

The adaptive immunity, also known as specific or acquired immunity, can be considered the second line of defence of the body against pathogens. It is characterized by the ability to distinguish between pathogens, called specificity, and the ability to respond stronger to repeated exposure of the same pathogen, known as memory. The adaptive immune response is typically slower (days to weeks) compared to the innate immune response (seconds to hours). The adaptiveness of the acquired immunity is the main reason why vaccines work or why individuals gain at least temporary immunity after (most) infections. Vaccines work by following the principles of adaptive immune responses, namely by inducing an immune response that protects the host from infection and/or disease following future pathogen exposure using its memory function. Most of the commonly available vaccines contain one or more antigens, which stimulate immune responses and triggers the differentiation of B-cells. Lymphocytes, e.g., B- and T-cells, their differentiated cell types and especially antibodies constitute the main parts of adaptive immunity. Natural killer cells, although primarily part of the innate immunity, may serve as an evolutionary bridge between the innate and adaptive immunity, by presenting "adaptive-like" features, such as the ability to develop immunological memory, specifically in response to certain viral infections. Generally, the adaptive immune system can be distinguished into cell-mediated and humoral immunity.

Cell-mediated Immunity

Cell-mediated immunity, also known as cellular immunity, primarily relies on T lymphocytes. Broadly T-cells can be split into T-helper cells (also known as $CD4^+$-cells) and cytotoxic T-cells (also known as $CD8^+$ T-cells) and be distinguished by the major histocompatibility complex (MHC) molecules they target. T-helper cells recognize antigens presented by antigen-presenting cells on the MHC class II molecules and become thereby activated and release cytokines to aid the immune response, cellular and humoral. This involves the activation of cytotoxic T-cells, macrophages or natural killer cells or the activation of B-cells and the production of antibodies. Cytotoxic T-cells on the other hand kill infected cells by recognizing antigens presented on the MHC class I molecules

and then inducing apoptosis, programmed cell death. Cell-mediated immunity is a crucial component of the adaptive immune system against intracellular microbes, such as viruses, bacteria, fungi or cancerous and transplanted cells.

Humoral Immunity

The humoral immunity is primarily driven by B lymphocytes and the antibodies they secrete. B lymphocytes are the only cells capable of producing antigen-specific antibodies. Antibodies have a Y-shaped protein structure, while there are five different isotypes of antibodies: IgG, IgM, IgD, IgA, IgE. They are structurally and functionally distinct. In humans, IgG is present in 4 different forms, IgG1, IgG2, IgG3, IgG4 and constitutes the majority of antibody based immunity.[18] It is also the only antibody class that can can cross the placenta and therefore provide passive immunity to the fetus.[19] B-cells originate and mature in the bone marrow, upon activation they differentiate either into plasma cells or memory B-cells. Plasma cells, fully differentiated B-cells, are the cells that secret antigen-specific antibodies, while they are typically divided into short-lived plasma cells, responsible for the early immune response and long-lived plasma cells maintaining sustained antibody production relevant to long-term protection. Memory B-cells remain in circulation in the bloodstream in an inactive state, with the role to memorize the antigen-characteristic that activated their parent B-cell during primary infection and therefore allowing for a faster and stronger secondary immune response upon re-exposure to the same antigen. Memory B-cells can survive for decades. There are thymus independent antigens, which can stimulate B-cells without the help of T-cells and thymus dependent antigens, which need the help of T-cells to activate B-cells, however, going into details of these mechanisms would go beyond the scope of this dissertation and is barely needed to understand the functionality of the already complex models described in Section 1.2.2.

While antibodies play a crucial role in the adaptive immune response, other components, such as B and T-cells also contribute considerably to immunity.[5] However, only a small proportion of infectious diseases are vaccine-preventable, partly because many pathogens exhibit high variability, making broad vaccine coverage challenging. When a practical surrogate is needed to quantify the immune response to allow for subsequent analyses in immunization studies the most practical and cost-effective approach is the quantification of antibodies, as antibody assays are typically affordable, standardized and can be performed using standard blood serum or plasma.[20–23] Assays for T and B-cells, while providing more detailed information about immunity, are generally less useful for broad application due to their higher costs, work intensity and other complexities.[24–26] Further, they often only capture fractions of the total response, due to different

cell subtypes, non-standardized protocols or different functional characteristics.[25, 27, 28] Consequently, immunization studies rely on the measurement of antibodies and the models used in these studies either exclusively model the antibodies or rely on implicit assumptions on the other cell types and their interplay with antibodies without explicitly measuring them (cp. Section 1.2.2). For these reasons the focus of this dissertation lies on the design of studies aiming at modelling the antibody response.

1.2. Mathematical Modelling in Epidemiology

Mathematical models have long been used in epidemiology and became an important tool, especially regarding infectious diseases, to improve understanding and prediction of disease dynamics and thus to aid public health decision making, such as controlling and assessing intervention strategies.[29–31] For example, Daniel Bernoulli published in 1766, but discussed in the years before, a model to describe the dynamics of smallpox infections and whether universal inoculation is beneficial.[32] The famous SIR-Model (susceptible, infected, recovered), introduced by Kermack and McKendrick in 1927 might be considered the foundation of modern infectious disease modeling.[33] Generally infectious disease modelling can be separated into two major classes: between-host models and within-host models. As the names suggest both types of models consider hosts, which can be humans or animals, playing a major role in infectious diseases. Both model types have in common that they translate processes, either of biological nature or population based, to mathematical expressions and in many cases to (systems of) differential equations. While between-host models describe the transition of individuals between different (health) states, typically called compartments, on a population level, e.g., the number of individuals being susceptible or infected, within-host models focus on the individual and aim at capturing dynamics and underlying biological processes of the immune response to pathogens. In the following, ideas behind between-host models will be briefly discussed in Section 1.2.1, while the focus lies on within-host models and the results of our systematic review as described in Section 1.2.2 as the main objective of this work is the improvement of study design in immunization studies where within-host models are used.

1.2.1. Between-Host Models

Between-host models describe how infectious diseases spread among individuals in a population. Typically, the population is divided into disjoint states, commonly referred to as compartments, and the transition between these states are modelled through mathematical equations. The main objective then lies in determining the number of individuals in the respective states over time. In contrast, agent-based models represent disease

transmission at the individual level, where each agent is individually characterized by its attributes, such as age, vaccination status and disease status. In agent-based models disease transmission is on an individual level instead of a group level and can therefore more easily account for variability in the behaviour of the individuals.[34] However, they are complex, require more computational power and rely on more data.[34] From a public health perspective the major role of between-host models lies in predicting or understanding infectious disease dynamics and consequently control and reduce the number of infected and infectious individuals.[30] For example, vaccinations can reduce the number of susceptible individuals within the population, decrease the infectiousness of infected individuals or lessen the consequences of an infection.[30] As previously stated, the focus of this dissertation is on within-host model, however, in the following the SIR-model is introduced as a simple example of between-host models as the general modelling idea can be translated to within-host compartmental models.

The SIR-model consists of three compartments, namely, susceptible (S), infected (I) and recovered (R).[31] The simplistic SIR-model has several underlying assumptions: individuals are born into the susceptible compartment, individuals can only move from the susceptible to the infected compartment (with infection rate β) and from the infected to the recovered (with recovery rate γ), which means that after infection lifelong immunity is achieved. Additionally, all individuals have an equal chance of interacting with each other, known as homogeneous mixing and infection dynamics are age-homogeneous. The system of differential equations then depends on the assumptions made on birth, disease-related mortality and all-cause mortality.[31] In the following we will assume that the number of births depend on the birth rate ν and the fixed population size N and that there is no disease related mortality. The system of differential equations can then be stated as

$$\begin{cases} \dfrac{dS(t)}{dt} = \nu N - \beta S(t) - \mu S(t), \\ \dfrac{dI(t)}{dt} = \beta S(t) - \gamma I(t) - \mu I(t), \\ \dfrac{dR(t)}{dt} = \gamma I(t) - \mu R(t), \end{cases}$$

where μ denotes the death rate, which is equal for all compartments. Since the population size is fixed, the birth rate equals the death rate, i.e. $\nu = \mu$.

As it is for all modelling approaches, several assumptions that typically do not hold in real-world scenarios have been made. Relaxing these assumptions includes, among others, the incorporation of additional compartments, such as a deceased compartment,[35], age-heterogeneous models with maternal antibodies,[31] allowing for vaccination and/or breakthrough infections,[36] a latent phase,[30] incorporating spatial data [30] or hetero-

geneous mixing.[31] Other approaches also include stochastic models [37] or agent-based models.[38] However, going into details would go beyond the scope of this dissertation.

1.2.2. Within-Host Models - Systematic Review

Within-host models provide a mathematical framework to describe how a host's immune system reacts to pathogen exposure. Within-host models can have different aims, they can be modelled to capture the underlying biological processes, describe how concentrations of different cell types change over time or describe different binding mechanisms, for example, between antibodies and antigens. Our focus in this work lies on within-host models describing the change of antibody concentration over time, i.e. antibody kinetics, and thus focusing on the humoral immune response. The major advantage of within-host models incorporating the longitudinal structure of the data lies in the fact that they treat time as a continuous variable instead of discrete time-points and therefore allow for the assessment of time-effects. Within-host models generally, but especially those that focus on antibody kinetics, can be split into two major classes, mechanistic and phenomenological models.[6] Typically, it is assumed that antibody concentrations are distributed around a often nonlinear mean depending on time, which can take many functional forms, depending on the modelling assumptions.[6] Mechanistic models, typically based on a system of differential equations, describe an underlying biological process and the timely evolution of specific cell types and their interaction. Therefore, their main goal lies in improving the understanding of immunological processes including the quantification of antibodies. On the other hand, phenomenological models are statistical models that do not explicitly specify a biological system (often there is an biological system implied by the resulting function), but specify the mean function directly. Phenomenological models can be considered more of a statistical nature, while mechanistic models might be more mathematical modelling, even though the distinction is often difficult.[6]

In our finished, but yet unpublished, systematic review with the main objective of identifying which mathematical or statistical models were used to model antibody kinetics and to give an overview to what extend these models were used in the process of study design, we found an extensive list of models.[39] However, we have not identified any publications that use the functional form of the antibody kinetics in study design, i.e. to determine the optimal timing or the optimal number of blood samples.

Our search strategy consisted of three categories, the first consisting of words describing antibodies and known abbreviations, the second of words that describe the 'kinetics'-part, and the third the longitudinal modelling aspect. Details can be found in the published

protocol and in the Appendix A.1.[39] The initial search was conducted on the 8th of November 2021 and updated using the same search strategy on 9th of April 2024. Two independent reviewers (Sereina Herzog and Stefan Embacher) screened the abstracts, full texts and performed the data extraction. The publications need to fulfil the following three key points:

- The publication should deal with antibodies, specific or as theoretical concept;
- The publication should use a mathematical or statistical model to describe antibody kinetics in humans or animals;
- The used mathematical or statistical model should explicitly consider a continuous time component.

More details on inclusion and exclusion criteria can be found in the study protocol.[39] In total, we have found 1811 abstracts, out of which 857 were decided to be eligible for full text screening. Out of these 377 were eligible for data extraction, in which we became aware that further 76 publications need to be excluded. In total we have included 301 publications in our systematic review. A more detailed flow diagram can be found in Appendix A.2. Due to the extensive work load, we have deviated from the strategy described in the study protocol and omitted additional searches. For each of the publications, we have extracted a variety of information (see Appendix A.3), however, in the following we will focus on selected models, representative for the different model types and aiming at illustrating the major components of within-host models focusing on antibody kinetics.
In most publications the antibody concentrations were modelled on the original scale, log-2 transformed, ln-transformed or log-10 transformed scale.
In many publications simple linear regression models or linear mixed effects models, including random intercept and/or random slopes, were used. The advantage of random effects models is that the longitudinal structure of the data can be explicitly modelled and the advantage of linear mixed effects models is that two observations per participant are sufficient to provide meaningful estimates of the constant decay parameter. By the logarithmic change of base rule, i.e. $log_b(a) = \frac{log_x(a)}{log_x(b)}$, it does not matter which logarithmic transformation is chosen since all bases would result in a linearisation of exponential decay, i.e. $f(t) = f(0) * exp(-\lambda t)$, but with different numerical values on the transformed scale. Consider the natural-log: $log_e(f(t)) = log_e(f(0)) - \lambda * t$ and compare it with the log to base b (e.g. $b = 2$ or $b = 10$)

$$log_b(f(t)) = \frac{log_e(f(t))}{log_e(b)} = \frac{log_e(f(0)) - \lambda * t}{log_e(b)} = \frac{log_e(f(0))}{log_e(b)} - \frac{\lambda}{log_e(b))}t.$$

Through the assumption of exponential decay, i.e. $f(t) = f(0) * exp(-\lambda t)$, the half-life of antibodies can be explicitly calculated. Depending on the modelling choice different options exist to examine the influence of covariates on these half-lives.[40]
There exist several extension of these exponential decay models, many trying to relax the assumption of constant decay over time. The generalized exponential decay can be described through the function $f(t) = c_1 * exp(c_2 * t^{c_3})$; note that $c_3 = 1$ implies ordinary exponential decay.[41, 42] Another example is the Gompertz decay model, which can be expressed through $f(t) = c_1 * exp(c_2 * exp(c_3 * t))$. Several models that use alternative functional forms to characterize the antibody kinetics have been proposed, including: the power law model,[43–45] the extended power law model allowing for long-term plateau,[43, 46], the gamma decay model and its modifications [47] or the biexponential model.[44, 48, 49] Typically, these models are fit using nonlinear mixed effects models.
Since the antibody response to exposure (vaccination or infection) is characterized by an relatively sharp increase followed by a phase of faster and then a phase of slower decay, several publications used the idea to split the antibody kinetics into (two or more) phases.[50–53] The framework introduced in Section 2.3 follows this idea and splits the phases into a non-plateau and a plateau phase.

The distinction between mechanistic and phenomenological models, or mathematical and statistical models, is not sharp and a clear cut is difficult to make. For example, ordinary exponential decay and, therefore, simple linear mixed effects models on the log-transformed scale might be considered mechanistic models since the exponential decay can be written as an ODE: $\frac{df(t)}{dt} = -\lambda f(t)$, with the starting value $f(0) = f_0$. The same is true for other models that might be considered phenomenological, for example, the generalized exponential or the Gompertz decay model can be expressed as ODEs.[41, 42] Additionally, many of the ODE systems presented on the next pages posses an explicit solution and their parameters are estimated through nonlinear mixed effects models.

Most mechanistic models assume at least a compartment of antibody secreting cells and an antibody compartment.[54, 55] A reasonable adoption of this model would be to split the antibody secreting cells into short- and long-lived plasma cells,[56–58] as nicely described by Andraud et al., using the following system of differential equations.[56]

$$\begin{cases} \dfrac{dP_S(t)}{dt} = -\mu_S P_S(t), \\ \dfrac{dP_L(t)}{dt} = -\mu_L P_L(t), \\ \dfrac{dA(t)}{dt} = \psi_S P_S(t) + \psi_L P_L(t) - \mu_A A(t), \end{cases}$$

with $A(0) = A_0, P_S(0) = P_S^0$ and $P_L(0) = P_L^0$ denoting the initial population sizes of the respective cell types. In this model, antibodies (A) are assumed to be produced by short- and long-lived plasma cells (P_S and P_L) at rates of ψ_S and ψ_L, respectively, while decaying at a rate of μ_A. The plasma cells can only decay at rates μ_S and μ_L, respectively. Andraud et al. were also discussing two simplifications of the aforementioned model: an asymptotic model and the plasma cell driven kinetic model. All three models allow to express the antibodies as an explicit function with different grades of complexity, all coming with advantages and disadvantages regarding the research question of interest, accuracy of estimates and the interpretability of parameters.[56] Balelli et al. introduced an additional antigen compartment (Ag), which stimulates the differentiation of the additional memory B-cells compartment (M) into short and long-lived plasma cells.[12]

$$\begin{cases} \frac{dAg(t)}{dt} = -\mu_{Ag} Ag(t) \\ \frac{dM(t)}{dt} = \rho Ag(t) - (\phi_S + \phi_L) Ag(t) M(t) - \mu_M M(t) \\ \frac{dP_S(t)}{dt} = \phi_S Ag(t) M(t) - \mu_S P_S(t) \\ \frac{dP_L(t)}{dt} = \phi_L Ag(t) M(t) - \mu_L P_L(t) \\ \frac{dA(t)}{dt} = \psi_S P_S(t) + \psi_L P_L(t) - \mu_A A(t), \end{cases}$$

where μ_{Ag} and μ_M denote the decay rates of the antigens and the memory B-cells, respectively, ρ the generation rate of B-cells through antigen exposure and ϕ_S and ϕ_L the differentiation rate of the B-cells into short and long-lived plasma cells. Their model is feasible for describing the humoral immune response to a two-dose vaccination regimen and to predict a subsequent booster dose with a non-replicating vaccine. However, the authors point out that antibody concentrations alone are insufficient for structural identifiability of the parameters.[12]

As the immune system is a very complex system with numerous different cell types and interaction between those, the complexity of mechanistic models can easily increase to impracticability. Many of these complex models present identifiability issues, use unmeasured compartments or are inherently theoretical.[12, 14, 59, 60] Exemplary, these models incorporated additional compartments such as viral load,[59, 60] macrophages,[60] antigen presenting cells,[60, 61] different antibodies,[14, 60, 62] $CD4^+$ and $CD8^+$ T-cells,[61–63] or interferon-γ.[63] Blanco-Rodriguez et al. introduce 6 models in increasing complexity, where the simplest already represents dynamics of virus, T-cells, B-cells, IgM, and IgG and discuss different restrictions on their interaction dynamics.[62] When pharmacokinetic models were used to model the change of antibody concentration over time, the

antibodies were not a part of the immune response but some kind of treatment.[64–66]

None of the in total 301 included publications and the resulting large variety of different models described methods to optimize the study design. Some discussed optimality in term of timing of vaccination,[67–69] but neglected the possibilities which optimal design would provide at the stage of study design. Partly, this might be explained by the complexity of the models and the resulting difficulties when discussing optimal sampling times. On the other hand, the "simpler" phenomenological models could be used to optimize sampling schedules with the already available optimal design methods. Therefore, another main reason for the lack of optimal design in immunization studies using within-host models is, arguably, the lack of awareness and accessible tools.

1.3. Introduction to Optimal Design Theory

Statistical design of experiments can historically be attributed to mainly three fields: agriculture, industrial engineering and medicine.[70] Many consider Kirstine Smith with her publication in 1918 as the first contributor to this field of statistics.[10, 70, 71] She found designs that minimize the maximum variance of prediction (nowadays known as G-optimality) in the context of linear regression with polynomials.[72] Ronald Fisher is also widely considered as one of the founding fathers of experimental design due to the concepts and methodologies introduced in his influential book 'The Design of Experiments', published in 1935.[73] Wald introduced in 1943 the idea of optimizing the determinant of the information matrix, later known as D-optimality.[74] In the year 1952, Elfving introduced the idea to minimize the average variance of the parameter estimates (nowadays known as A-optimality).[75] In the 1950s and 1960s, Jack Kiefer can be considered as one of the main contributors with his work being summarized in Wynn 1984.[71] Kiefer and Wolfowitz showed the equivalence of G-optimal and D-optimal design which is commonly known as equivalence theorem.[76] A more general form was later provided in 1974, again by Jack Kiefer.[77] Box and Lucas (1959) found locally D-optimal designs for nonlinear models.[78] First attempts at constructing optimal sampling schemes date back to Sacks and Ylvisacker (1966).[79, 80] A nice overview and short introduction to several of those mentioned ideas can be found in Atkinson 1982.[81] Numerous contributions to the field of optimal design theory were made by Valerii Fedorov, non exclusively ranging from a "pioneering book" [9, 82], iterative construction of optimal designs, numerical methods of optimal design construction to a book on *"Optimal Design for Nonlinear Response Models"* together with Sergei Leonov, which builds the foundation of the methodological part of this dissertation.[10]

Many optimality criteria, and most of the prominent ones, map the Fisher Information matrix (FIM) to a real number.[9, 10] This idea is based on the fact that the inverse of the FIM can be used (under certain regularity assumptions) to approximate the variance-covariance matrix of the maximum likelihood estimator.[83, 84] For instance, D-optimal designs maximize the determinant of the FIM, while for E-optimal designs the largest eigenvalue of the inverse of the FIM is minimized. The, arguably, most popular and known optimality criterion is D-optimality, primarily due to its simple graphical interpretation (cp. Section 2.2), explainability and its generally good performance compared to other criteria.[9, 10, 85] The study design of nonlinear models, however, are less easy implemented in practical application, since the FIM depends on the unknown model parameters.[85, 86] Over time various approaches were developed to address this issue. Probably, the easiest/most practical strategy is to base the study design on guesses of the unknown parameters, commonly referred to as a locally optimal design.[10, 78] Optimal design for nonlinear models is utilized in various scientific and economic fields, including pharmacokinetics, a field closely related to our area of interest, yet different in its specific modelling goals.[10, 87, 88] As described in Section 1.2.2, there are no reported applications of optimal design theory in immunization studies, where modelling the antibody kinetics is the main objective. Optimal designs are based on models that make inherent assumptions, including controlled conditions. Clinical practice, however, involves unpredictable factors, like unexpected patient heterogeneity or adherence issues. Consequently, optimally designed experiments may be optimal theoretically but suboptimal or at least impractical in real settings. Further, with each additional sample the risk of less adherence increases.

1.4. Problem

The following section presents a motivational example illustrating the influence of sampling times and the rapid increase in the number of possible sampling combinations with increasing number of samples. An analytical formulation of the problem is subsequently introduced.

1.4.1. Illustrative Example

Let's assume we are conducting a hypothetical study, in which our budget allows for 600 blood samples. In our research group, discussion led to three possible options: either include 300 participants with 2 samples each, 200 with 3 samples each or 150 participants with 4 samples each. After further discussion, taking 2 samples was ruled out, because we want to distinguish between linear and exponential decay and we want to allow for the

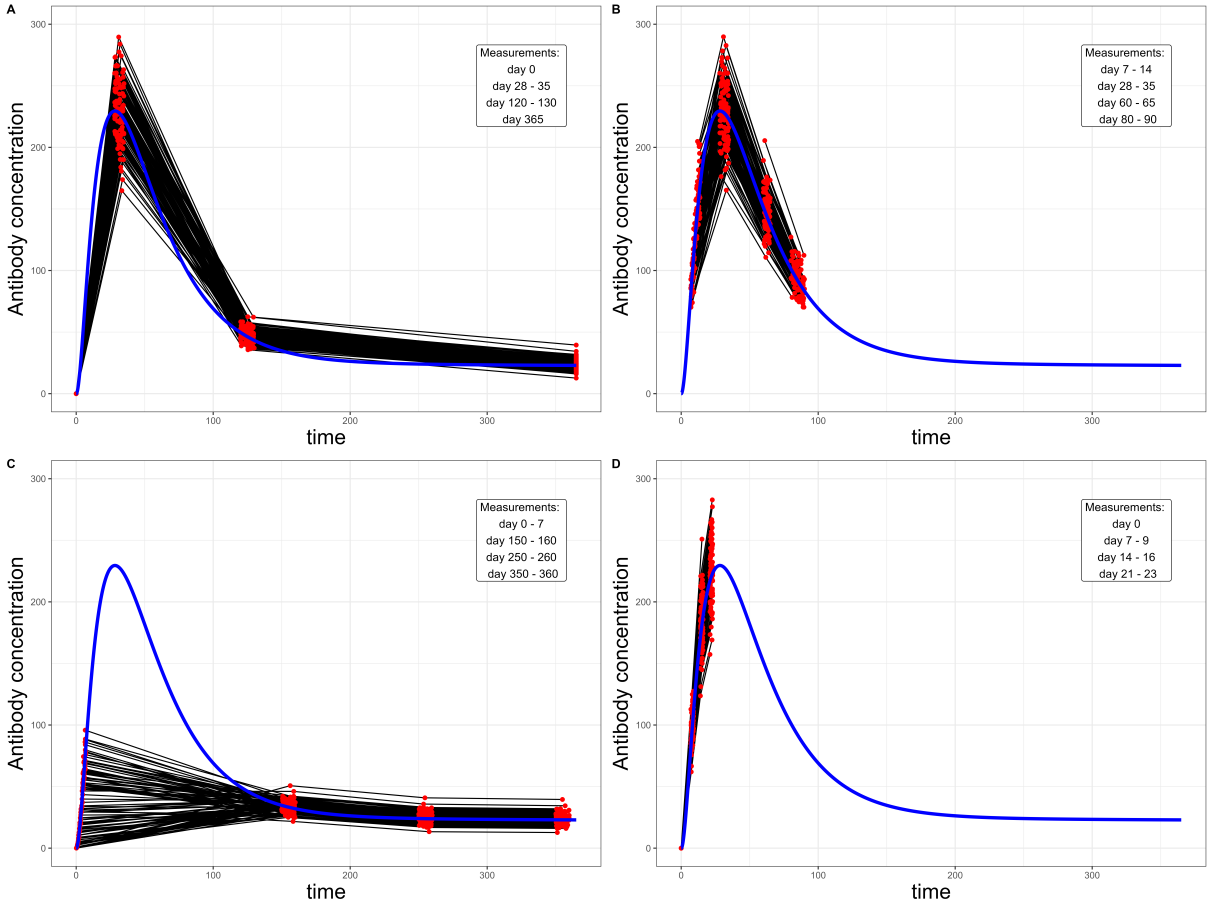

Figure 1.1.: The 4 panels illustrate how different sampling schedules might impact estimation. The blue line is the same in all 4 panels and shows the 'true' mean antibody kinetics. The red dots denote samples of individual patients in the prespecified time-windows which differ between the panels. The black lines connect the samples coming from the same individual linearly.

general exponential decay to incorporate the longitudinal structure of the data, i.e. mixed effects model. Finally, based on the available literature it is reasonable to assume that the antibodies show an initial increase followed by a decay and for logistic reasons the study should end within a year, we therefore opted for 4 samples within a year. The logical next question could be: When should the samples be taken? If we would allow for all time-point combination on a day-basis, a total of $\binom{365}{4} = 727.441.715$ different schedule would be possible. Under the assumptions that we fix one sample at day 0 and one sample at day 365 and that at least one week should be between two consecutive samples this would result in 59.340 possibilities (on a day basis). The number of possibilities increases with additional samples, for example, 5 samples within a year would result in 52.521.291.823 possible schedules. Figure 1.1 illustrates why the timing of samples might impact the estimation of the curve. In Panel A and Panel B, the samples are taken before the maximum, around the maximum and in the decaying phase. A simple linear interpolation between samples would already provide an approximation of the underlying antibody kinetics. In Panel C, through a second measurement that is too late, the peak would be missed and in Panel D, through 4 early samples, we would only observe the increasing phase. This could potentially lead to wrong conclusions and ultimately to waste of resources. Optimal design allows to optimize the timing of the samples with an analytical criterion, for example, minimizing the confidence ellipsoid of the parameter estimates, and therefore to an optimized use of resources.

1.4.2. Analytical Formulation

Lets assume we are conducting a study, where N participants take part. We have n different combination of predictor levels x_i, with $i \in \{1, \dots, n\}$ (for example sampling schedules, i.e. multiple time-points). Let's assume $dim(x_i) = k$, which means we are taking, for example, k samples. Suppose r_i participants follow the predictor level x_i with $\sum_{i=1}^{n} r_i = N$. However, we could further allow for different length of those combination of predictor levels, i.e. x_{ij} with $j \in \{1, \dots, k_i\}$. For example, if we deal with sampling times, different patients can have different number of samples taken, i.e. the $x_i = (x_{i1}, \dots, x_{ik_i})$ can have different length for different patients. We therefore take k_i samples from patient $i \in \{1, \dots, N\}$, where x_{ij} denotes the j-th sampling time for patient i and y_{ij} the measurement at time j for patient i, with $j \in \{1, \dots, k_i\}$. Let $y_i = (y_{i1}, \dots, y_{ik_i})$ denote the vector of samples from participant i.

Without any assumption on the structure of the function, we can say that the measurements of the antibody levels y_{ij} follow a function η, which depends on the time of sampling

x_{ij}, some parameters γ_i plus an additive sampling error with zero mean ϵ_{ij}, i.e.

$$y_{ij} = \eta(x_{ij}, \gamma_i) + \epsilon_{ij}, \text{ for } j = 1, \ldots, k_i \text{ and } i = 1, \ldots, N$$

The ϵ_{ij} are uncorrelated measurement errors with mean zero. For example, they can follow a normal distribution with variance σ^2. Further, we can assume that the individual parameters are independent between patients and that the γ_i follow a distribution with mean γ_0 and variance-covariance matrix Ω. Then it follows that

$$\mathbb{E}[y_{ij}|\gamma_i] = \eta(x_{ij}, \gamma_i). \tag{1.1}$$

In an immunization study, one aim could be to model the expected individual antibody kinetics over time, depending on some specific individual parameters, which is given by equation 1.1. However, most studies focus on population means, i.e. $\mathbb{E}[y_{ij}]$. Now if we denote by $\theta = (\gamma_0, \Omega, \sigma^2)$ all parameters and consider that $\eta(x_{ij}, \gamma_0)$ does not depend on Ω and σ^2 it follows that [10]

$$\mathbb{E}[y_{ij}] \approx \eta(x_{ij}, \gamma_0) = \eta(x_{ij}, \theta).$$

The aim lies now in determining sampling schedules x_i that improve (optimize) the estimation of θ. Usually, also some distributional assumptions on the measurements are made, for example, a multivariate normal distribution with a nonlinear mean and correlated measurements, i.e.

$$y_i|x_i \sim N(\eta(x_i, \theta), S(x_i, \theta)),$$

with $\eta(x_i, \theta) = (\eta_1(x_i, \theta), \ldots, \eta_{k_i}(x, \theta))^T$, $S(x, \theta)$ is a $k \times k$ matrix, and θ are unknown parameters.
The function $\eta(.)$ can take several forms depending on the underlying modelling assumptions, for which we discussed multiple options in Section 1.2.2. However, when we take this general problem formulation into application we have to make some practical considerations:

- Which η do we choose?
- Which distribution should the measurements follow and how are different measurements correlated?
- Restrictions on the number of different predictor levels n and the length of those k_i enter the application via the definition of eligible designs and restrictions on the design space (cp. Section 2.2). Therefore: What are analytical and/or practical restrictions on the design space?

1.5. Aim of the Dissertation

Advantages of optimally designed studies are indisputable, more so from an ethical perspective and in terms of sustainability. In the previous sections, we saw that the use of optimal design, a non-trivial mathematical problem, is heavily underutilized in immunization studies where modelling the antibody kinetics is the main objective (cp. Section 1.2.2). The aim of this dissertation is to bridge the gap between theory and practice by introducing a framework that allows to design immunization studies without sophisticated mathematical knowledge or advanced programming abilities while also reducing the needed prior information to easily understandable and interpretable information. Thereby, the aim is to increase the accessibility of optimal design theory to a field in which no endeavour is made up to now to optimize study design analytically.

2. Methods

The following sections introduce the necessary essential concepts, including the Fisher information matrix, in Section 2.1, which serves as a basis and reference for subsequent developments. Key principles of optimal design theory are then presented in Section 2.2. The proposed framework is described in Section 2.3, starting with its basic idea, extending it to a more flexible setting, and describing its integration within the context of optimal design theory. Finally, Section 2.4 outlines the Robustness Analysis of the framework, detailing the methodology used to assess its sensitivity to misspecification in the initial information.

2.1. Essential Concepts

Section 2.1.1 presents matrix properties relevant to this dissertation, followed by Section 2.1.2 which introduces the Fisher Information matrix as the foundation of optimal design theory, while Section 2.1.3 provides a brief introduction to the beta and digamma functions as they are part of the Fisher Information matrix in our framework. Throughout this dissertation, the natural logarithm is denoted by *log* unless a different base is explicitly stated.

2.1.1. Matrix Properties

Definition 1. *A quadratic real matrix $A \in \mathbb{R}^{n \times n}$ is called:*

- *symmetric if $A = A^\intercal$*
- *invertible or non-singular if there exists A^{-1} such that $AA^{-1} = A^{-1}A = I$, where I denotes the identity matrix*
- *the trace of A is defined by $tr(A) = \sum_i a_{ii}$*

Definition 2. *A quadratic real matrix $A \in \mathbb{R}^{n \times n}$ is called*

- *positive-definite if $x^\intercal Ax > 0 \quad \forall x \in \mathbb{R}^n \setminus \{0\}$*
- *positive-semidefinite (or non-negative definite) if $x^\intercal Ax \geq 0 \quad \forall x \in \mathbb{R}^n$*

- *negative-definite if* $x^\intercal Ax < 0 \quad \forall x \in \mathbb{R}^n \setminus \{0\}$
- *negative-semidefinite (or non-positive definite) if* $x^\intercal Ax \leq 0 \quad \forall x \in \mathbb{R}^n$

Definition 3. *The determinant of a quadratic matrix A is a scalar-valued function of it's entries. It is typically denoted by* $\det(A)$ *or* $|A|$*. There are different ways of defining the determinant, out of which the Leibniz formula is one of the most prominent ones.*

$$\det(A) = \sum_{\tau \in S_n} sgn(\tau) \prod_{i=1}^{n} a_{i\tau(i)},$$

where S_n denotes the group of permutations, $\tau(i)$ the permutation of element i and sgn denotes the sign function of the respective permutation. It holds that A is invertible $\iff \det(A) \neq 0$.

Definition 4. *Let $A \in \mathbb{R}^{n \times n}$ be a quadratic matrix and $x \in \mathbb{R}^n \setminus \{0\}$. If $Ax = \lambda x$, where $\lambda \in \mathbb{R}$, then x is called an eigenvector and λ an eigenvalue of A.*

The definiteness of a real symmetric matrix A are directly connected to it's eigenvalues via their sign.

Theorem 1. *For a symmetric real matrix $A \in \mathbb{R}^{n \times n}$ the following equivalences hold:*

- *A is positive definite $\iff$ all eigenvalues are positive*
- *A is positive semi-definite $\iff$ all eigenvalues are non-negative*
- *A is negative definite $\iff$ all eigenvalues are negative*
- *A is negative semi-definite $\iff$ all eigenvalues are non-positive*
- *A is indefinite $\iff$ A has positive and negative eigenvalues*

Definition 5. *Let $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times n}$ be symmetric matrices, we then say that $A \geq B$ if $A - B$ is positive semi-definite and $A > B$ if $A - B$ is positive definite. This partial order is called Loewner ordering.[89]*

2.1.2. Fisher Information Matrix

The definition and derivation of the Fisher information matrix (FIM) is very similar to the scalar case of the Fisher information.[90] Let X be distributed with probability density function $f(X;\theta)$, where $\theta = (\theta_1, \ldots, \theta_p)^\intercal$ is a p-dimensional vector, and with the following three assumptions (i) the parameter space Θ is open, (ii) the set $A = \{x : f(x;\theta) > 0\}$ is independent of θ (corresponding to common support), (iii) for all $x \in A$ and all $\theta \in \Theta$ the

derivative $\frac{\partial}{\partial\theta_i} log f(x;\theta)$ exists and is finite $\forall i \in 1,\ldots,p$, then the FIM may be introduced as in [91]

$$FIM(\theta)_{ij} = \mathbb{E}\left[\left(\frac{\partial}{\partial\theta_i} log f(X;\theta)\right)\left(\frac{\partial}{\partial\theta_j} log f(X;\theta)\right)|\theta\right]$$

or equivalently in matrix notation:

$$FIM = \mathbb{E}[D_\theta \log f(X;\theta)^\intercal D_\theta \log f(X;\theta)|\theta].$$

Under the additional assumption that (iv) the integral of $f(X;\theta)$ can be differentiated under the integral sign with respect to θ, then [91]

$$\mathbb{E}\left[\frac{\partial}{\partial\theta_i} log f(X;\theta)|\theta\right] = 0.$$

If further (v) the second derivatives $\frac{\partial^2}{\partial\theta_i\partial\theta_j} log f(X;\theta)$ exist, an alternative and often more convenient way to write the FIM is given as in [91]

$$FIM(\theta)_{ij} = -\mathbb{E}\left[\frac{\partial^2}{\partial\theta_i\partial\theta_j} log f(X;\theta)|\theta\right]$$

One pivotal result used in Section 2.2 is that the FIMs of independently distributed random variables X and Y are additive, [10, 91] i.e.,

$$FIM_{(X,Y)}(\theta) = FIM_X(\theta) + FIM_Y(\theta)$$

Under technical and relatively mild conditions (see Appendix B assumptions 1-5) the maximum likelihood estimator (MLE) $\hat{\theta}$ of θ is a consistent estimator, i.e. $\hat{\theta} \xrightarrow{\mathrm{p}} \theta$.[83, 84] If additional technical conditions (see Appendix B assumptions 6-11) hold, it can be shown that the MLE is asymptotically normal with $\sqrt{n}\left(\hat{\theta} - \theta\right) \xrightarrow{\mathrm{d}} N(0, FIM(\theta)^{-1})$.[83, 84] With the following result, described underneath, the MLE is therefore an asymptotically efficient estimator.

Let θ be a p-dimensional parameter vector, with a given probability density function, and let $FIM(\theta)$ denote the corresponding Fisher information matrix. Further let $T(X) = (T_1(X),\ldots,T_p(X))^\intercal$ be an unbiased estimator of θ, then the variance-covariance matrix fulfils the following inequality:[92–94]

$$cov(T(X)) \geq FIM(\theta)^{-1},$$

where the matrix inequality is understood as Loewner ordering.[95] The right hand side of

the inequality is called Cramer-Rao bound and could be extended to several other cases, including a biased estimator.[94, 95]

Theorem 2. *Let $g(X) \sim N(\mu(\theta), \Sigma)$, where g is continuous and the partial derivatives $\frac{\partial \mu(\theta)}{\partial \theta_i}$ exist. Further, let Σ be a symmetric, positive definite $p \times p$ matrix not depending on the parameters, i.e. $\Sigma^\intercal = \Sigma$, $\Sigma(\theta) = \Sigma$ and $det(\Sigma) > 0$. Then the Fisher information matrix is defined by*

$$FIM_{ij} = \frac{\partial \mu(\theta)}{\partial \theta_i}^\intercal \Sigma^{-1} \frac{\partial \mu(\theta)}{\partial \theta_j}$$

Proof. Let $g(X) \sim N(\mu(\theta), \Sigma)$ and let Σ fulfil the regularity assumptions. Then the probability density function of $g(X)$ exists and the log-likelihood has the following form

$$\begin{aligned}\log f(g(X), \mu(\theta), \Sigma) = &-\frac{p}{2}\log(2\pi) - \frac{1}{2}\log(\det(\Sigma)) \\ &-\frac{1}{2}(g(X) - \mu(\theta))^\intercal \Sigma^{-1}(g(X) - \mu(\theta))\end{aligned}$$

Using that the derivative $D_x(x^\intercal A x) = x^\intercal(A + A^\intercal)$ if A does not depend on x, the chain-rule $D[f(g(x))] = Df(g(x)) * Dg(x)$ and the symmetry of Σ and therefore of Σ^{-1} then the derivative of the log-likelihood function has the following form

$$\begin{aligned}D_\theta \log f(g(X), \mu(\theta), \Sigma) = & \\ &= D_\theta(-\frac{1}{2}(g(X) - \mu(\theta))^\intercal \Sigma^{-1}(g(X) - \mu(\theta))) \\ &= -\frac{1}{2} D_\theta((g(X) - \mu(\theta))^\intercal \Sigma^{-1}(g(X) - \mu(\theta))) \\ &= -\frac{1}{2}((g(X) - \mu(\theta))^\intercal(\Sigma^{-1} + \Sigma^{-1T})(-D_\theta \mu(\theta)) \\ &= (g(X) - \mu(\theta))^\intercal \Sigma^{-1} D_\theta \mu(\theta)\end{aligned}$$

Set $H := D_\theta \mu(\theta)$, then the Fisher information matrix can be calculated using the linearity of the expectation

$$\begin{aligned}FIM &= \mathbb{E}[D_\theta \log f(g(X), \mu(\theta), \Sigma)^\intercal D_\theta \log f(g(X), \mu(\theta), \Sigma) | \theta] \\ &= \mathbb{E}[H^\intercal \Sigma^{-1\intercal}(g(X) - \mu(\theta))(g(X) - \mu(\theta))^\intercal \Sigma^{-1} H | \theta] \\ &= H^\intercal \Sigma^{-1\intercal} \mathbb{E}[(g(X) - \mu(\theta))(g(X) - \mu(\theta))^\intercal | \theta] \Sigma^{-1} H \\ &= H^\intercal \Sigma^{-1} H\end{aligned}$$

and therefore component-wise

$$FIM_{ij} = \frac{\partial \mu(\theta)}{\partial \theta_i}^{\mathsf{T}} \Sigma^{-1} \frac{\partial \mu(\theta)}{\partial \theta_j}$$

□

This result is of particular interest, since it relates to two special cases typically seen when working with antibodies or laboratory values in general. In practice the following two distributions are used:

- If our data is normally distributed, g is the identity function and therefore $X \sim N(\mu(\theta), \Sigma)$
- If our data is log-normally distributed, g is the logarithm and $\log(X) \sim N(\mu(\theta), \Sigma)$

If the variance-covariance matrix depends on the parameters, i.e. $\Sigma(\theta)$, it holds generally that if X is multivariate normal $N(\mu(\theta), \Sigma(\theta))$, that there exists a closed-form solution for the FIM.[96, 97] It is given by

$$FIM(\theta)_{ij} = \frac{\partial \mu(\theta)}{\partial \theta_i}^{\mathsf{T}} \Sigma^{-1}(\theta) \frac{\partial \mu(\theta)}{\partial \theta_j} + \frac{1}{2} tr\left(\Sigma^{-1}(\theta) * \frac{\partial \Sigma(\theta)}{\partial \theta_i} * \Sigma^{-1}(\theta) * \frac{\partial \Sigma(\theta)}{\partial \theta_j} \right)$$

Note that the FIM does not generally have a closed-form solution when working with nonlinear mixed effects models due to the fact that there is no analytical expression of the log-likelihood function.[96, 98–100] Therefore, approximations are used, where linear approximations, especially Taylor expansion, seem the natural choice.[99, 100]

2.1.3. Beta Function and Digamma Function

The beta function is defined by the integral

$$B(z_1, z_2) = \int_0^1 t^{z_1 - 1}(1-t)^{z_2 - 1} dt,$$

where $z_1, z_2 \in \mathbb{C}$ with $Re(z_1), Re(z_2) > 0$. It is characterized through its close relationship to the gamma function[101]

$$B(z_1, z_2) = \frac{\Gamma(z_1)\Gamma(z_2)}{\Gamma(z_1 + z_2)}.$$

It follows that $B(z_1, z_2) = B(z_2, z_1)$.
The derivative of the beta function is given by

$$\frac{\partial}{\partial z_1} B(z_1, z_2) = B(z_1, z_2)(\psi(z_1) - \psi(z_1 + z_2)),$$

where $\psi(z)$ denotes the digamma function which is defined as

$$\psi(z) = \frac{d}{dz} log\Gamma(z) = \frac{\Gamma'(z)}{\Gamma(z)}.$$

2.2. Optimal Design Theory

The following sections formally introduce discrete and continuous designs, optimality criteria, and ways to deal with unknown parameters, followed by a theorem that sets out conditions for the existence of optimal designs. Additionally, a list of practical considerations is given in Section 2.2.1. From now onwards, different Fisher information matrices (FIMs) are used and therefore the notation is changed: $\underline{M}$ denotes the FIM of a design, M the normalized FIM of a design and $m(x)$ the FIM of a single observational unit x. While in the previous section the emphasize was on the definition and properties of the FIM, the focus lies now on the definition of designs and their properties.

Definition 6. *A discrete design is a combination of n support points $x_i \in \mathbb{X}$, where $\mathbb{X}$ denotes the design space and r_i the number of replications at x_i, such that $\sum_{i=1}^{n} r_i = N$ and $n \leq N$. A design is denoted by*

$$\xi_N = \{x_i, p_i\}_{i=1}^{n}, \textit{where } p_i = r_i/N.$$

For example, a design space could consist of all possible sampling schedules with two, three or four measurements within a year, i.e.

$$\mathbb{X} = \{x = (t_1, \ldots, t_k), t_j \in [0, 365], j = 1, \ldots, k \text{ and } 2 \leq k \leq 4\},$$

and a design could then for example be the combination of two schedules, where 20% follow the first one and 80% follow the second one, i.e.

$$\xi_N = \{(x_1 = (0, 28, 128, 256), p_1 = 0.2), (x_2 = (0, 60, 120, 180), p_2 = 0.8)\}.$$

It appears conceptually intuitive to set the optimality of a design into context to the variance-covariance matrix $\underline{D}(\xi_N)$ of estimated parameters, which depends on the design ξ_N. Further, under mild regularity assumptions (cp. Section 2.1.2 and Appendix B), the variance-covariance matrix of the maximum likelihood estimator can be approximated by the inverse of the FIM $\underline{M}(\xi_N)^{-1}$. The Fisher information matrix of the design is additive, i.e.

$$\underline{M}(\xi_N) = \sum_{i=1}^{n} r_i m(x_i),$$

where $m(x_i)$ denotes the FIM of a single observational unit. It is assumed that $\underline{M}(\xi_N)$ is non-singular for any reasonable ξ_N, which is a common and reasonable assumption.[86] Irrespectively, Fedorov and Leonov give a very nice example of why it is necessary to introduce optimality criteria.[10] They show that even for simple models the solution of

the optimization problem

$$\xi_N^* = argmax_{\xi_N} \underline{M}(\xi_N) = argmin_{\xi_N} \underline{D}(\xi_N)$$

does in general not exist. Therefore, optimality criteria are introduced.

Definition 7. *The solution of the optimization problem*

$$\xi_N^* = argmin_{\xi_N} \Psi[\underline{M}(\xi_N)],$$

where Ψ is a scalar function, is called an optimal design with respect to the optimality criterion Ψ.

There are numerous optimality criteria available and providing an extended list goes beyond the scope of this dissertation.[9, 10] In the following, some of the most prominent ones are defined.

Definition 8.

- *D-optimality:*
  The function Ψ is defined as the determinant of the corresponding matrix, i.e.

  $$\Psi = \det(\underline{D}) = \det(\underline{M})^{-1}$$

- *E-optimality:*
  The function Ψ is defined as the minimal/maximal eigenvalue of the corresponding matrix, i.e.

  $$\Psi = \lambda_{max}(\underline{D}) = \lambda_{min}^{-1}(\underline{M})$$

- *A-optimality:*
  The function Ψ is defined as the trace of the product of a non-negative matrix A with $\underline{D}$, i.e.

  $$\Psi = tr(A\underline{D})$$

  A convenient choice of A would be $A = m^{-1}I_m$, where I_m is the identity matrix. This results in

  $$\Psi = m^{-1}tr(\underline{D}) = m^{-1}tr(\underline{M}^{-1}) = m^{-1}\sum_{i=1}^{m} Var(\hat{\theta}_i)$$

As shown in Figure 2.1, D- and E-optimality can be interpreted graphically. D-optimality corresponds to the 'size' of the confidence ellipsoid (proportional to the volume), while E-optimality corresponds to the length of the largest axis of the confidence

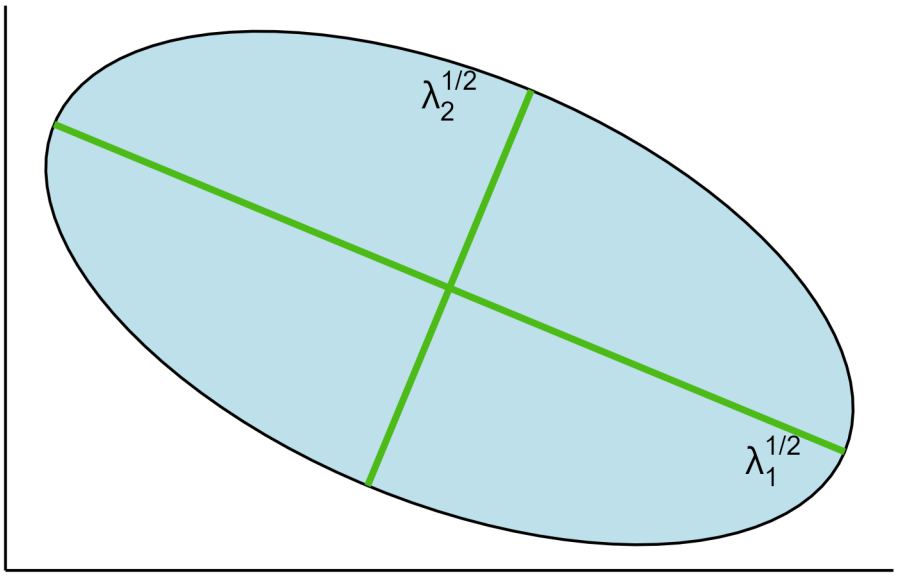

Figure 2.1.: Optimality criteria represented on the 2-dimensional confidence ellipsoid. The blue area represents the D-criterion and the green axis (eigenvalues) represent E-criterion.

ellipsoid.[10] The A-criterion, based on the choice presented in Definition 8, is the average variance of the parameter estimates. Arguably, D-optimality is the most popular criterion among researchers.[10, 85] Fedorov and Leonov reason this by the ease of explanation, the invariance with respect to nondegenerate transformations of parameters and because D-optimality often performs well compared to other criteria.[9, 10, 85]

Definition 9. *In general, the optimality criteria should fulfil three properties. [10]*

- *Monotonicity*

  $$\Psi(\underline{M}) \leq \Psi(\underline{M}') \text{ if } \underline{M} \geq \underline{M}',$$

  where both are nonnegative definite matrices and Ψ *therefore a monotonically non-increasing function, where the matrix inequality is understood as Loewner ordering.*

- *Homogeneity*

  $$\Psi(\underline{M}) = \Psi(NM) = \gamma(N)\Psi(M),$$

  where γ *is a nonincreasing function,* N *the number of participant and* M *the normalized fisher information matrix,*

  $$M = \frac{1}{N}\underline{M} = \sum_{i=1}^{n} p_i m(x_i).$$

- *Convexity*

The optimality criterion is or can be transformed into a convex function,i.e.

$$\Psi(M) \leq (1-\alpha)\Psi(M_1)+\alpha\Psi(M_2),$$

for $\alpha \in [0,1]$, $M_1, M_2 \in \mathbb{M}$, *where* $\mathbb{M}$ *is a convex set and* $M = (1-\alpha)M_1+\alpha M_2$.

The concept of monotonicity corresponds to the idea, that extra observations can only be beneficial and never worsen the optimality criteria. Homogeneity means that the dependence of the criterion on N and $\underline{M}$ can be split. However, M still depends on N through $p_i = r_i/N$. Especially when working with continuous designs, it is useful to work with the normalized information matrix. Note, that D-optimality as defined in Definition 8 is not convex, however, it can be easily transformed in such, by using the logarithm.[10, 85] As stated by Fedorov and Leonov it is possible to combine criteria, when pursuing multiple objectives.[10]

To find a solution of the discrete optimization problem defined by Definition 6 and Definition 7 might be numerically and/or analytically costly.[10] It is therefore useful to relax the assumption of the discreteness of the weights and introduce continuous designs.[10, 85]

Definition 10. *A continuous design is an arbitrary combination of n support points* x_i *and weights* p_i, *such that*

$$\xi = \{x_i, p_i\}, \textit{where } x_i \in \mathbb{X}, 0 \leq p_i \leq 1 \textit{ and } \sum_{i=1}^{n} p_i = 1$$

In contrast to discrete designs, the weights p_i in continuous designs can vary continuously in [0,1]. Consequently, the weights p_i do not need to fulfil $p_i * N = r_i \in \mathbb{N}$. Therefore, the solution of the optimization problem using continuous designs is only an approximation of the problem using discrete designs.[10] However, an advantage of continuous designs is that there is no need to study variations in the optimal designs as a function of N.[102] Nevertheless, in practice it is not possible, that $r_i \notin \mathbb{N}$. An easy approximate solution would then be to round $r_i = N * p_i$, while fixing $\sum_{i=1}^{n} r_i = N$.[103]

Lemma 1. *If* $\xi_1 = \{x_{1i}, p_{1i}\}_{i=1}^{n}$ *and* $\xi_2 = \{x_{2i}, p_{2i}\}_{i=1}^{n}$ *are two continuous designs and* $0 \leq \alpha \leq 1$, *then a new continuous design is defined by*

$$\xi = (1-\alpha)\xi_1 + \alpha\xi_2,$$

understood in the sense that the weights are combined

$$p_i = (1-\alpha)p_{1i} + \alpha p_{2i}.$$

To show that the combination is actually a design, we consider the assignment of weights to design points. If a design point x_i belongs to ξ_1 only, its weight in the new design is $(1-\alpha)p_{1i}$, if a design point belongs to ξ_2 only, its weight in the new design is αp_{2i} and if it belongs to both its weight is the convex combination. Consequently, $\sum p_i = 1$, $0 \leq p_i \leq 1$. If the design space for both designs is identical or corresponds to their union, then all $x_i \in \mathbb{X}$.
In practice, it may be relevant to quantify or compare the efficiency of designs. Therefore, the notion of D-efficiency is introduced.

Definition 11. *To compare two designs, denoted by $\xi_{N_1,1}$ and $\xi_{N_2,2}$, the relative D-efficiency is defined by*

$$\mathit{Eff}_D(\xi_{N_1,1}, \xi_{N_2,2}) = \left(\frac{\det(\underline{M}(\xi_{N_1,1}))}{\det(\underline{M}(\xi_{N_2,2}))}\right)^{1/m},$$

where m denotes the number of parameters.

Having established the ability to compare different designs, the next difficulty arises. In the case of nonlinear models, the information matrix and the covariance matrix depend on the unknown parameters, i.e. $\underline{D}(\xi_N, \theta)$ and $\underline{M}(\xi_N, \theta)$.[85, 86, 104] There are different ways to deal with this issue, formally defined in Definition 12 below. The first, and probably simplest one, is to find locally optimal designs. This means that a preliminary guess of the parameters θ_G is specified and the optimization problem is addressed with respect to these parameters. Others are the so called minimax criterion or the average criterion. In the minimax approach, one finds the optimal design for the worst-case value of θ. This approach has the disadvantage that it may give rise to such an extreme design that the strategy is unduly defensive.[105] The average criterion uses the idea to integrate the unknown parameters out of the optimization problem. The average criterion is often called Bayesian approach, because the ideas are closely connected.[106] Additionally, it is possible to work with adaptive designs, wherein acquired information is iteratively integrated into the process. The procedure typically begins with a locally optimal design, and after a predetermined number of samples one uses the estimates of the unknown parameters to refine the design for subsequent data collection. These steps are then repeated several times. Hill (1980) has stressed that in models containing both linear and nonlinear parameters, it is only the values of the latter that affect the D-optimal design.[107] Further, when working with locally optimal designs, it is always necessary to

check the properties of a particular optimal design performing a sensitivity analysis with regards to the assumed parameter values.[10]

Definition 12.

- *Let θ_G be an guess of the unknown parameters θ. Then*

$$\xi^* = argmin_\xi \Psi(M(\xi, \theta_G))$$

  is called an locally optimal design.

- *Let Θ be the parameters space, the minimax design is then*

$$\xi^* = argmin_\xi max_{\theta \in \Theta} \Psi(M(\xi, \theta))$$

- *Let $\mathbb{A}(d\theta)$ be interpreted as a measure of our trust/importance in particular values of θ (in the bayesian setting: a priori distribution). The average criterion is then defined as*

$$\xi^* = argmin_\xi \int_\Theta \Psi(M(\xi, \theta)) \mathbb{A}(d\theta).$$

Lets now assume that (i) $\mathbb{X}$ is compact, (ii) the information matrix $m(x)$ is continuous in x (for nonlinear models, that corresponds to the continuity of the partial derivatives of $\eta(x, \theta)$ $\forall \theta$),[10] (iii) $\Psi(M)$ is convex and monotonically nonincreasing, and that (iv) $\exists q < \infty$ such that $\{\xi : \Psi(M(\xi)) \leq q\}$ is a non empty set.
(v) Further let Ψ be differentiable in the following sense:
Let $\overline{\xi}$ put unit mass at the point x and let $\xi' = (1-\alpha)\xi + \alpha\overline{\xi}$ then it follows that $M(\xi') = (1-\alpha)M(\xi) + \alpha M(\overline{\xi})$ and the derivative of ψ in the direction $\overline{\xi}$ is given by

$$\phi(x, \xi) = lim_{\alpha -> 0^+} \frac{1}{\alpha} [\Psi\{(1-\alpha)M(\xi) + \alpha M(\overline{\xi})\} - \Psi(M(\xi)].$$

Theorem 3. *If the mentioned assumptions (i)-(v) hold, then*

1. *There exists an optimal design with no more than $p(p+1)/2$ support points,*
2. *The set of optimal designs is convex.*

For a proof of (1) in Theorem 3, see Fedorov and Leonov 2019.[10] To show (2) in Theorem 3, i.e. the convexity of the set of optimal solutions, let's assume ξ_1^* and ξ_2^* are optimal designs, i.e. $\Psi(M(\xi_1^*)) = \Psi(M(\xi_2^*)) = min_\xi \Psi(M(\xi))$ and let $\xi^* = (1-\alpha)\xi_1^* + \alpha\xi_2^*$, then it follows from the convexity of Ψ that $\Psi(M(\xi^*)) \leq (1-\alpha)\Psi(M(\xi_1^*)) + \alpha\Psi(M(\xi_2^*)) = min_\xi \Psi(M(\xi))$. With Theorem 3, conditions for the existence of optimal designs have been established. However, an optimal design is not necessarily unique.[85]

The equivalence theorem(s) plays a crucial role in optimal design theory, especially in showing the equivalence between different optimality criteria and verifying the optimality of a given design. The original equivalence theorem, which shows equivalence between D- and G-optimality, is published by Kiefer and Wolfowitz 1960.[76] Which means, that the design which maximizes the determinant of the information matrix (D-optimal) also minimizes the maximum prediction variance (G-optimal). For the formulation of the general equivalence theorem, we follow Atkinson.[102] The general equivalence theorem basically uses the fact that derivatives of a function are zero at a minimum of the respective function. Under the given (mostly mild) assumptions, out of which the compactness of $\mathbb{X}$ and the convexity and differentiability of Ψ are the most important once,[9] the general equivalence theorem holds:

Theorem 4. *Under the given assumptions the following three conditions on ξ^* are equivalent:*

1. *ξ^* minimizes $\Psi(M(\xi))$*
2. *ξ^* maximizes the minimum over $\mathbb{X}$ of $\phi(x,\xi)$*
3. *$min_{x\in\mathbb{X}}\phi(x,\xi^*) = 0$ and this minimum is achieved at the at the support points of the design.*

The general equivalence theorem provides a relatively simple way to check the optimality of a given design. In the case of D-optimality, the necessary and sufficient condition for a design ξ to be locally optimal is given by [82, 96, 102, 104]

$$d(x,\xi,\theta) = tr[m(x,\theta)M^{-1}(\xi,\theta)] \leq p,$$

where $p = dim(\theta)$, where equality is attained in the support points of the optimal design. This function is often called sensitivity function.[85, 96] Note, that for the usually unknown parameters θ, the guess θ_G is used.

2.2.1. Practical Considerations

Summarizing, to take the methodological framework of optimal design into practice, several practical considerations must be made. Specifically, the following decisions need to be addressed:

- Which optimality criterion is chosen?
- Because antibody kinetics are typically nonlinear: How to deal with the fact that the objective function is a function in unknown parameters?

- Are there restrictions on the design space? For example, a fixed visit after 2 month.
- Are there restrictions on the weights of the design points? For example, all participants should follow the same schedule or a 1:1 split.

2.3. Framework

The various shapes of the beta distribution, determined by different choices of the shape parameters and their relationship to the mode, are described in Section 2.3.1. This serves as a foundation for our proposed framework to enhance infectious disease study design, which is introduced in Section 2.3.2. A potential extension is explored in Section 2.3.3, and its application to optimal design theory is discussed in Section 2.3.4. The implementation is then described in Section2.3.5. The major parts are already published in Embacher et al.[108]

2.3.1. Beta Distribution

The beta distribution is a continuous probability distribution defined on the interval $[0,1]$ characterized by two positive parameters α and β. The probability density function is given by

$$f(x,\alpha,\beta) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1},$$

where $B(\alpha,\beta)$ denotes the beta function.[109] The parameters α and β are called shape parameters, because they shape the density function and also determine its mode. Notably, the values $\alpha = 1$ and $\beta = 1$ are crucial cases. Each parameter can either be less than 1, equal to 1 or greater than 1, resulting in a total of 9 cases as presented in Table 2.1. Figure 2.2 illustrates the different variations of the beta density for the mentioned

$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	**Mode**	**shape**
$\alpha < 1$	$\beta < 1$	No mode	u-shaped
$\alpha < 1$	$\beta = 1$	0	decreasing
$\alpha < 1$	$\beta > 1$	0	decreasing
$\alpha = 1$	$\beta < 1$	1	increasing
$\alpha = 1$	$\beta = 1$	Any value	constant
$\alpha = 1$	$\beta > 1$	0	decreasing
$\alpha > 1$	$\beta < 1$	1	increasing
$\alpha > 1$	$\beta = 1$	1	increasing
$\alpha > 1$	$\beta > 1$	$\frac{\alpha-1}{\alpha+\beta-2}$	peaking

Table 2.1.: Modes of the beta distribution for different α and β values

cases. There are five basic shapes: The density is either constant, monotonically increasing, monotonically decreasing, u-shaped while diverging at the borders or peaking within the interval $(0,1)$. Since the objective is to model the antibody kinetics after exposure, which typically show an increase followed by a decrease, the parameters in this work are restricted to the case $\alpha > 1$ and $\beta > 1$. For the beta distribution and the respective choice of parameters, it is then known that the mode x_{max} is given by [109]

$$x_{max} = \frac{\alpha - 1}{\alpha + \beta - 2}. \tag{2.1}$$

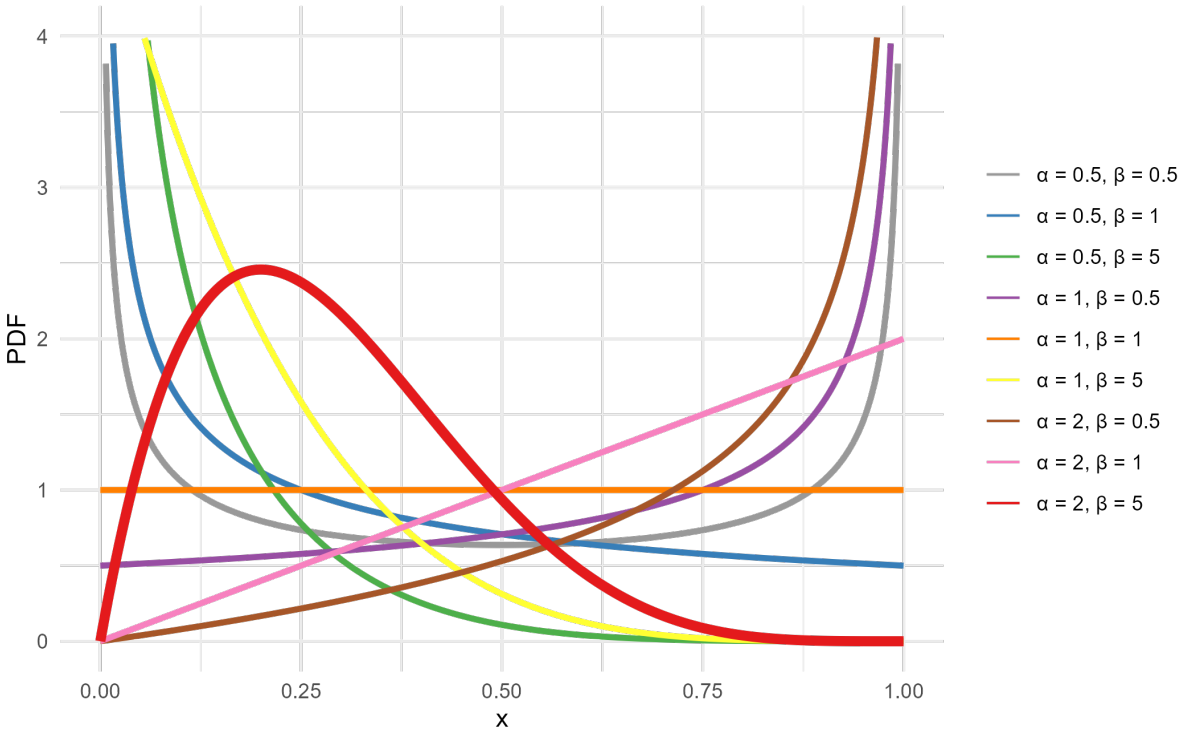

Figure 2.2.: The density of the beta distribution for different choices of the shape parameters α and β. The red line highlights the case, where $\alpha > 1$ and $\beta > 1$. For better visibility the y-axis is limited to the interval $[0, 4]$.

The ability to describe the mode of the beta density through its parameters is one of the core properties utilized in the framework.

2.3.2. Framework - Basic Idea

One of the fundamental assumptions in the framework is that the antibody concentration $A(t)$ over time t follows a mean antibody concentration $\mu(t, \theta)$. It is assumed that $\mu(t, \theta)$ reaches a plateau after following the structural form of a beta density. Reaching a plateau is equivalent to assuming that antibody levels either fall to and remain at undetectable levels or stabilize at a particular threshold. Based on the fact that the antibody levels should first increase and then decrease, the parameters of the beta distribution are restricted to be $\alpha > 1$ and $\beta > 1$ (cp. Section 2.3.1). With the restrictions on the parameters, the mode x_{max} can therefore be described by equation (2.1). For now, it is assumed that participants do not have antibodies at time $t = 0$, i.e. $A(0) = 0$. Under these assumptions, the mean antibody concentration $\mu(t, \theta)$ can then be described and determined using clear and intuitive information provided by medical professionals. Specifically, $\mu(t, \theta)$ can be uniquely determined by the time of the maximum antibody concentration t_{max}, the value of the maximum antibody concentration $A_{max} = \mathbb{E}[A(t_{max})] = \mu(t_{max}, \theta)$, the time un-

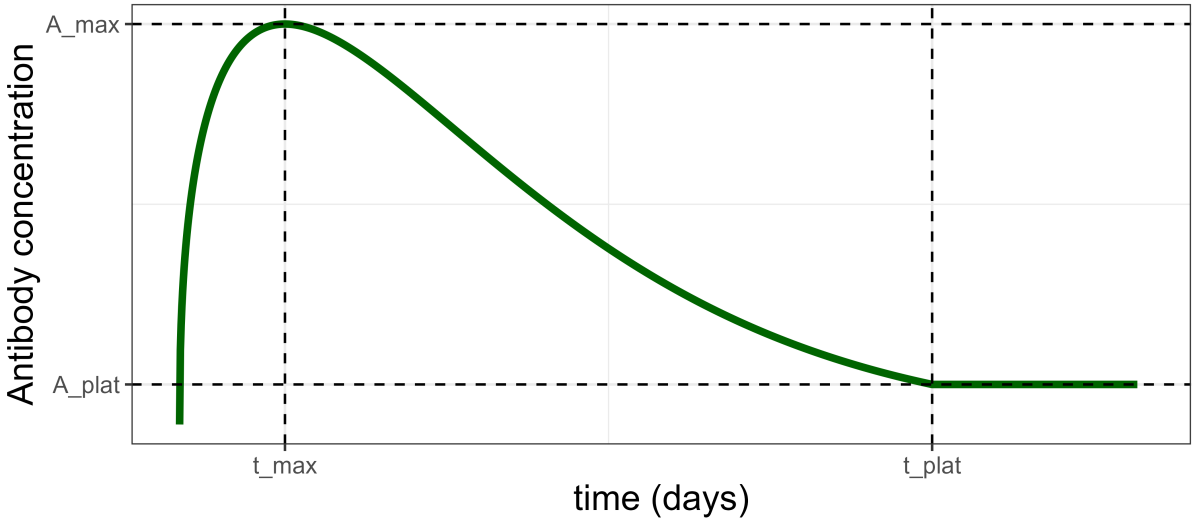

Figure 2.3.: Graphical display of the basic idea of using the time of maximum t_{max}, the height of the maximum A_{max}, the time of the plateau t_{plat} and the height of the plateau A_{plat} to describe the change in antibody concentration over time. Reproduced from Embacher et al. [108]

til the plateau is reached t_{plat} and the value of the plateau $A_{plat} = \mathbb{E}[A(t_{plat})] = \mu(t,\theta)$. Figure 2.3 gives a graphical illustration of this idea. Since the beta distribution has support $[0,1]$, an additional scaling parameter t_{scale} is introduced to ensure that time can be expressed on a suitable scale, e.g. day scale. Via $x_{max} = \frac{t_{max}}{t_{scale}}$ and $x_{plat} = \frac{t_{plat}}{t_{scale}}$ this scaling parameter assures that $x_{max} \in [0,1]$ and $x_{plat} \in [0,1]$, given the restriction that $t_{scale} \geq t_{plat} > t_{max}$.

Under the given assumptions, the antibody kinetics can be split into two phases, namely the phase until the plateau is reached and the plateau-phase. Therefore, it is assumed that the mean antibody kinetics follow the structure

$$\mathbb{E}[A(t)] = \mu(t,\theta) = \begin{cases} f\left(\frac{t}{t_{scale}}, \alpha, \beta\right) & \text{for } 0 < t < t_{plat} \\ A_{plat} & \text{for } t \geq t_{plat}. \end{cases}$$

By providing the initial information $A_{max}, t_{max}, A_{plat}$, and t_{plat}, the function $\mu(t,\theta)$ can be uniquely determined and leads to the following system of equations

$$\begin{cases} 0 = \dfrac{\alpha - 1}{\alpha + \beta - 2} - \dfrac{t_{max}}{t_{scale}} \\ f(t_{max}, \alpha, \beta, t_{scale}) = A_{max} = \dfrac{1}{B(\alpha,\beta)} \left(\dfrac{t_{max}}{t_{scale}}\right)^{\alpha-1} \left(1 - \dfrac{t_{max}}{t_{scale}}\right)^{\beta-1} \\ f(t_{plat}, \alpha, \beta, t_{scale}) = A_{plat} = \dfrac{1}{B(\alpha,\beta)} \left(\dfrac{t_{plat}}{t_{scale}}\right)^{\alpha-1} \left(1 - \dfrac{t_{plat}}{t_{scale}}\right)^{\beta-1} \end{cases} \tag{2.2}$$

which needs to be solved for α, β and t_{scale}. Since the FIM is given in terms of the partial derivatives of $\mu(t, \theta)$ (cp. Section 2.1.2) and the FIM is used to optimize study design, the respective partial derivatives must be derived. In the case of patients without pre-existing antibodies, the partial derivatives of the beta density are given by

$$\begin{cases} \frac{\partial \mu}{\partial \alpha} = \frac{1}{B(\alpha,\beta)} \Big[log\left(\frac{t}{t_{scale}}\right) \left(1 - \frac{t}{t_{scale}}\right)^{\beta-1} \left(\frac{t}{t_{scale}}\right)^{\alpha-1} \\ \qquad - (\psi(\alpha) - \psi(\alpha+\beta)) \left(1 - \frac{t}{t_{scale}}\right)^{\beta-1} \left(\frac{t}{t_{scale}}\right)^{\alpha-1} \Big] \\ \frac{\partial \mu}{\partial \beta} = \frac{1}{B(\alpha,\beta)} \Big[log\left(1 - \frac{t}{t_{scale}}\right) \left(1 - \frac{t}{t_{scale}}\right)^{\beta-1} \left(\frac{t}{t_{scale}}\right)^{\alpha-1} \\ \qquad - (\psi(\beta) - \psi(\alpha+\beta)) \left(1 - \frac{t}{t_{scale}}\right)^{\beta-1} \left(\frac{t}{t_{scale}}\right)^{\alpha-1} \Big] \\ \frac{\partial \mu}{\partial t_{scale}} = \frac{1}{B(\alpha,\beta) t_{scale}^2} \Big[(\beta-1)t \left(1 - \frac{t}{t_{scale}}\right)^{\beta-2} \left(\frac{t}{t_{scale}}\right)^{\alpha-1} \\ \qquad - (\alpha-1)t \left(1 - \frac{t}{t_{scale}}\right)^{\beta-1} \left(\frac{t}{t_{scale}}\right)^{\alpha-2} \Big] \end{cases}$$

2.3.3. Framework - Extension

In the system of equations (2.2), one of the main underlying assumptions is $A(0) = 0$. In practice this assumption is often violated. Exemplary, one could deal with patients with prior exposure to antigens or vaccination or with newborns in whom transplacental transfer provided maternal antibodies.[5, 40, 110] To extend the flexibility of our framework, a shift c is introduced to incorporate the possibility of modelling $A(0) > 0$ in the antibody levels. Regardless of the shift, the model still needs to exhibit an initial increase, followed by a decay in the antibody concentration. The restrictions on the parameters, i.e. $\alpha > 1$ and $\beta > 1$, directly imply that $f(x, \alpha, \beta) = 0$ for $x = 0$. Consequently, $c > 0$ if $A(0) > 0$ and $c = 0$ if $A(0) = 0$ must be fulfilled. Similarly to x_{max} and x_{plat}, $x_c = \frac{c}{t_{scale}}$ is defined. As a result, incorporating the additional information about the expected starting value

A_0, the following system of equations is obtained

$$\begin{cases} 0 = \dfrac{\alpha-1}{\alpha+\beta-2} - \dfrac{t_{max}+c}{t_{scale}} \\ f(\alpha,\beta,c,t_{scale}) = A_0 = \dfrac{1}{B(\alpha,\beta)}\left(\dfrac{c}{t_{scale}}\right)^{\alpha-1}\left(1-\dfrac{c}{t_{scale}}\right)^{\beta-1} \\ f(t_{max},\alpha,\beta,c,t_{scale}) = A_{max} = \dfrac{1}{B(\alpha,\beta)}\left(\dfrac{t_{max}+c}{t_{scale}}\right)^{\alpha-1}\left(1-\dfrac{t_{max}+c}{t_{scale}}\right)^{\beta-1} \\ f(t_{plat},\alpha,\beta,c,t_{scale}) = A_{plat} = \dfrac{1}{B(\alpha,\beta)}\left(\dfrac{t_{plat}+c}{t_{scale}}\right)^{\alpha-1}\left(1-\dfrac{t_{plat}+c}{t_{scale}}\right)^{\beta-1} \end{cases} \tag{2.3}$$

which needs to be solved for α, β, t_{scale} and c. With this adaptation, the partial derivatives of the beta density are given by

$$\begin{cases} \dfrac{\partial\mu}{\partial\alpha} = \dfrac{1}{B(\alpha,\beta)}\Big[log\left(\dfrac{t+c}{t_{scale}}\right)\left(1-\dfrac{t+c}{t_{scale}}\right)^{\beta-1}\left(\dfrac{t+c}{t_{scale}}\right)^{\alpha-1} \\ \qquad -(\psi(\alpha)-\psi(\alpha+\beta))\left(1-\dfrac{t+c}{t_{scale}}\right)^{\beta-1}\left(\dfrac{t+c}{t_{scale}}\right)^{\alpha-1}\Big] \\ \dfrac{\partial\mu}{\partial\beta} = \dfrac{1}{B(\alpha,\beta)}\Big[log\left(1-\dfrac{t+c}{t_{scale}}\right)\left(1-\dfrac{t+c}{t_{scale}}\right)^{\beta-1}\left(\dfrac{t+c}{t_{scale}}\right)^{\alpha-1} \\ \qquad -(\psi(\beta)-\psi(\alpha+\beta))\left(1-\dfrac{t+c}{t_{scale}}\right)^{\beta-1}\left(\dfrac{t+c}{t_{scale}}\right)^{\alpha-1}\Big] \\ \dfrac{\partial\mu}{\partial t_{scale}} = \dfrac{1}{B(\alpha,\beta)t_{scale}^2}\Big[(\beta-1)(t+c)\left(1-\dfrac{t+c}{t_{scale}}\right)^{\beta-2}\left(\dfrac{t+c}{t_{scale}}\right)^{\alpha-1} \\ \qquad -(\alpha-1)(t+c)\left(1-\dfrac{t+c}{t_{scale}}\right)^{\beta-1}\left(\dfrac{t+c}{t_{scale}}\right)^{\alpha-2}\Big] \\ \dfrac{\partial\mu}{\partial c} = \dfrac{1}{B(\alpha,\beta)t_{scale}}\Big[(\alpha-1)\left(1-\dfrac{t+c}{t_{scale}}\right)^{\beta-1}\left(\dfrac{t+c}{t_{scale}}\right)^{\alpha-2} \\ \qquad -(\beta-1)\left(1-\dfrac{t+c}{t_{scale}}\right)^{\beta-2}\left(\dfrac{t+c}{t_{scale}}\right)^{\alpha-1}\Big] \end{cases}$$

where $\psi(z)$ denotes the digamma function and $B(\alpha,\beta)$ the beta function.

2.3.4. Applied Optimal Design

With the introduced framework the primary objective is to facilitate the improvement of infectious disease study design for healthcare professionals. To optimize the properties of a design, an optimality criterion needs to be introduced.[10] D-optimality was chosen, as it

is one of the most popular criteria and is generally recognized for its superior performance compared to other criteria.[9, 10, 85] For more details, see Section 2.2 and Definition 8. To fulfil the desired convexity property of optimality criteria (cp. Definition 9), we will use $-\ln(\det(FIM)+1)$ instead of the determinant. The constant 1 is added to the determinant for mainly two reasons. Firstly, by adding one, designs with a determinant of zero still correspond to a value of zero in the objective function. Secondly, numerical stability is improved for designs, where the determinant is very small. Let's assume a time dependent mean, where n measurements are taken $\boldsymbol{\mu}(\boldsymbol{t},\boldsymbol{\theta}) = (\mu(t_1,\boldsymbol{\theta}),\ldots,\mu(t_n,\boldsymbol{\theta}))^\intercal$, with $\boldsymbol{\theta} = (\theta_1,\ldots,\theta_p)^\intercal$ being a p-dimensional parameter vector. Further, let's assume that the n measurements are multivariate normally distributed $Y \sim N_n(\boldsymbol{\mu}(\boldsymbol{t},\boldsymbol{\theta}),\Sigma(\boldsymbol{\theta}))$, with the simplification that the covariance matrix does not depend on $\boldsymbol{\theta}$, i.e. $\Sigma(\boldsymbol{\theta}) = \Sigma$. As described in Section 2.1.2, the (i,j)-entry of the FIM is then given by

$$FIM_{ij} = \frac{\partial \boldsymbol{\mu}^\intercal}{\partial \theta_i}\Sigma^{-1}\frac{\partial \boldsymbol{\mu}}{\partial \theta_j}.$$

The corresponding partial derivatives for the framework are given in Section 2.3.2 and Section 2.3.3. When working with locally optimal designs, the existence of one true parameter vector is assumed. This, however, implies that the number of samples per participant equals the number of system parameters and that additional samples would be replicates of one of the original samples.[111] Consequently, the FIM is a 3x3 matrix in the basic case (starting in zero) and a 4x4 matrix in the extended framework (not starting in zero), due to inclusion of the additional parameter c. Since the decision was made to work with D-optimal designs, D-efficiency will be used to compare different designs, where higher values are more favourable (cp. Definition 11).

2.3.5. Implementation

Beta parameters

The parameters of the system of equations (2.2) or (2.3) are determined using the R-Package 'nleqslv', Version 3.3.5.[112] Both, the implemented Broyden and Newton methods, are employed with the default global strategy of 'double dogleg', as per the package's author's arguments.[112, 113] The default settings are used for all tolerances, with automatic scaling and a maximum of 1000 iterations. Solutions are only used if the algorithm converges and meets the parameters' restrictions (e.g., $\alpha > 1$). As shown in the Appendix D, solutions can be considered global optima.

Optimal design

The Hooke and Jeeves method, a pattern search procedure used to minimize nonlinear functions, while not relying on gradients, is used to find the optimal solution, i.e. the optimal time-points.[114] The method is implemented as the function 'hjn' in the R-package 'optimx' version 2023-10.21.[115, 116] The following values were chosen as starting values for the optimization algorithm: day 1, the provided time of maximum t_{max} and time of the plateau t_{plat}. In the extended framework, the fourth starting time is added as $t_{max} + (t_{plat} - t_{max})/2$, corresponding to the middle between t_{max} and t_{plat}. When designing a study, the Hooke and Jeeves method offers the advantage of adding constraints on lower and upper boundaries of the time-points. All time-points should fall between the start (day 0) and the end of the study, which is assumed to occur no later than when the plateau is reached. Clinicians can also prespecify time-periods for samples with this option, for example, if participants have a fixed visit in the 3rd week. For numerical considerations, the step size was increased by a factor of 1.5 compared to the default and set the number of function evaluations to 10^5.

Scenario Nr.	Description	A_0	t_{max}	A_{max}	t_{plat}	A_{plat}
1	Start in zero, faster increase, lower plateau	0	30	10	365	1
2	Start in zero, slower increase, lower plateau	0	70	10	365	1
3	Start in zero, faster increase, higher plateau	0	30	10	365	2.5
4	Start in zero, slower increase, higher plateau	0	70	10	365	2.5
5	Start at lower level, faster increase, lower plateau	1	30	10	365	1
6	Start at lower level, slower increase, lower plateau	1	70	10	365	1
7	Start at lower level, faster increase, higher plateau	1	30	10	365	2.5
8	Start at lower level, slower increase, higher plateau	1	70	10	365	2.5
9	Start at higher level, faster increase, lower plateau	5	30	10	365	1
10	Start at higher level, slower increase, lower plateau	5	70	10	365	1
11	Start at higher level, faster increase, higher plateau	5	30	10	365	2.5
12	Start at higher level, slower increase, higher plateau	5	70	10	365	2.5

Table 2.2.: Definition of the 12 scenarios used for the Robustness Analysis, where time variables are given in days and antibody concentration in arbitrary units. A_0: Provided starting value; t_{max}: Provided time of maximum antibody concentration; A_{max}: Provided maximum antibody concentration; t_{plat}: Provided time of plateau; A_{plat}: Provided height of the plateau. Reproduced from Embacher et al. [108]

2.4. Robustness Analysis

To evaluate the robustness and other key properties of our proposed framework, a Robustness Analysis is conducted. Specifically, different scenarios were defined, ranges of misspecification were introduced, and assumptions regarding the variance-covariance matrix were made in Section 2.4.1. The framework's sensitivity to misspecification in the initial information was then examined, which commonly arise during the planning phase of any (immunization) study. Details on the single parameter misspecification are given in Section 2.4.2, while details on the double parameter misspecification are given in Section 2.4.3. All analyses were performed using R, version 4.4.1,[117] while the implementation of the methods to solve for the beta parameters and the optimal time-points are described in Section 2.3.5.

2.4.1. Scenarios

Twelve scenarios were defined, as shown in Table 2.2, to capture a broad range of possible antibody kinetics following exposure. The scenarios account for differences in the starting levels, the speed of increase and the height of the plateau. Specifically, a distinction is made between cases where antibody levels start at zero, at a lower level or a higher level. The speed of increase is varied through the timing of the maximum antibody concentration and the height of the plateau is differentiated between lower and higher values. The primary objective of the Robustness Analysis is to evaluate how sensitive the framework reacts to misspecification in the initial information. In real-world applications, where the framework is used to design studies, there is a high chance that the initial information,

such as the timing of maximum antibody concentration, may not be correctly specified. To systematically assess this sensitivity, specific assumptions regarding the extent of potential misspecifications are defined, with the correct value of the respective scenario denoted by the additional subscript $\_true$:

- $A_0 \in [max(0, A_{0\_true} - 2), A_{0\_true} + 2]$
- $A_{max} \in [A_{max\_true} - 2, A_{max\_true} + 2]$
- $A_{plat} \in [max(0, A_{plat\_true} - 2), A_{plat\_true} + 2]$
- $t_{max} \in [t_{max\_true} - 14, t_{max\_true} + 14]$
- $t_{plat} \in [t_{plat\_true} - 50, t_{plat\_true} + 50]$

In scenarios 1-4, A_0 is not varied, as it is assumed to be known at the planning stage whether the population has pre-existing antibodies or not. According to the definitions in Table 2.2 and the specified ranges of misspecification, A_0 is fixed at 0 for scenarios 1-4, while it varies within the interval $[0, 3]$ for scenarios 5-8 and $[3, 7]$ for scenarios 9-12. The initial information A_{max} is varied within the range $[8, 12]$ for all scenarios and can be interpreted as a standardization for the other antibody concentrations. For scenarios 1, 3, 5, 7, 9, 11, t_{max} is varied in the interval $[16, 44]$ and for scenarios 2, 4, 6, 8, 10, 12 in the interval $[56, 84]$. Furthermore, t_{plat} varies between $[315, 415]$ for all scenarios, while A_{plat} varies in the interval $[0, 3]$ for scenarios 1, 2, 5, 6, 9, 10 and in the interval $[0.5, 4.5]$ for scenarios 3, 4, 7, 8, 11, 12. The parameters of the beta distribution and the resulting optimal time-points for the true scenarios are given in Section 3.1.

Covariance Matrix

To calculate the FIM, it is assumed, that each measurement shows the same, known, variability σ^2 and that the variance-covariance matrix has the structure of an AR(1) model, [118] where the parameter ρ is also assumed to be known. This implies that the variance-covariance matrix does not depend on the unknown parameters, is therefore constant, and that the formula described in Section 2.1.2 can be used. The AR(1)-matrix is also known as Kac-Murdock-Szegö matrix and is a symmetric Toeplitz matrix defined by [119–121]

$$AR(1)_{ij} = \rho^{|i-j|}, \quad i, j = 1, \ldots, n,$$

where $0 < \rho < 1$ and $n > 1$. The matrix is therefore given by

$$AR(1) = \begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{n-1} \\ \rho & 1 & \rho & \cdots & \rho^{n-2} \\ \rho^2 & \rho & 1 & \cdots & \rho^{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{n-1} & \rho^{n-2} & \rho^{n-3} & \cdots & 1 \end{bmatrix}.$$

The inverse of this matrix is given by[120, 121]

$$AR(1)^{-1} = \frac{1}{1-\rho^2} \begin{bmatrix} 1 & -\rho & 0 & 0 & \cdots & 0 & 0 \\ -\rho & 1+\rho^2 & -\rho & 0 & \cdots & 0 & 0 \\ 0 & -\rho & 1+\rho^2 & -\rho & \cdots & 0 & 0 \\ 0 & 0 & -\rho & 1+\rho^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1+\rho^2 & -\rho \\ 0 & 0 & 0 & 0 & \cdots & -\rho & 1 \end{bmatrix}.$$

Suppose that the variance-covariance matrix $\Sigma = S * AR(1)$, where S is a diagonal matrix. In this case $S = diag(\sigma^2, \ldots, \sigma^2) \in \mathbb{R}^{n \times n}$, which corresponds to multiplying the matrix $AR(1)$ with the scalar σ^2, however, formulating S as a diagonal matrix allows to extend the framework in future work as discussed in Section 4.4. The inverse of the variance-covariance matrix, which is needed to get the FIM, is given by

$$\Sigma^{-1} = (S * AR(1))^{-1} = AR(1)^{-1} * S^{-1}.$$

Under the assumptions mentioned before this again corresponds to multiplying the matrix $AR(1)^{-1}$ with the scalar $\frac{1}{\sigma^2}$ or equivalently the diagonal matrix $S^{-1} = diag(\frac{1}{\sigma^2}, \ldots, \frac{1}{\sigma^2})$. For the Robustness Analysis the standard deviation of the measurements is fixed to 0.25. Based on the data used in one of our publications, we assume $\rho = 0.73$, which represents a strong correlation between measurements.[40, 122] However, the choice of σ and ρ do not affect the resulting optimal sampling times analytically and only to a very limited extent numerically as shown in Appendix C.

2.4.2. Single Parameter Misspecification

To investigate the influence of a single information on the determination of optimal sampling times, a controlled approach is employed, in which every initial information is fixed to their true values except for one, which is varied across the predefined range of mis-

specification. Specifically, this variation is implemented on an equidistant grid consisting of $N = 1001$ points. To assess the impact of misspecification, the D-efficiency (cp. Definition 11) of the resulting design is computed by using the 'true' parameter values and evaluating the 'misspecified' time-points compared with the 'true' time-points. This allows for the quantification of how deviations from the optimal time-points affect the efficiency of the design. To analyse the impact of misspecification in the initial information on deviations from the optimal time-points, the differences between the estimated optimal time-points under misspecification and the true optimal time-points is computed. This provides a direct measure of how inaccuracies in initial information propagate into the study design, enabling a more comprehensive understanding of the framework's sensitivity to parameter uncertainty and its practical effects.

2.4.3. Double Parameter Misspecification

To investigate the influence of pairwise misspecification, which means that three of the five initial information are fixed and the other two are varied in the specified ranges of misspecification, a comparable analysis as described in Section 2.4.2 is conducted. In total, there are 10 pairs of initial information and the same 12 scenarios (see Table 2.2) and the same ranges of misspecification are used. For runtime reasons the grid size of the single information is reduced to $N = 101$, resulting in 10201 runs per pair and scenario. D-efficiency was again used to evaluate the effect of misspecification in the initial information. Following the same principle as before, the 'true' parameter values were used to compare the 'misspecified' time-points with the 'true' time-points. Heat maps, coloured tables, and boxplots were employed to graphically display the results. Additionally, for each pair, the proportion of runs with a D-efficiency surpassing a given level was assessed.

3. Results

In the following the results of the Robustness Analysis regarding misspecification in the initial information in the proposed framework will be presented in Section 3.1, starting with the single parameter misspecification followed by the results of double parameter misspecification. Then the implementation of a Shiny application is described in Section 3.2, which is intended to bring the framework to practice, while keeping the Shiny application user-friendly to lower the hurdle of usage for the main target group of non-statistician. The input is described thoroughly, and the output is explained in detail. A corresponding version of the manual is provided in Appendix F. Last, in Section 3.3, the framework is applied to three examples.

3.1. Robustness Analysis - Results

To analyse the robustness of the framework and to evaluate how misspecification in the initial information affect the resulting study design a Robustness Analysis is performed. The methods for the Robustness Analysis are described in Section 2.4. Section 3.1.1 provides the results for single parameter misspecifications, where the corresponding results have already been published in Embacher et al.[108] Additionally, Section 3.1.2 provides the results of how pairwise misspecifications affect the robustness of the framework. Table 3.1 shows the parameters of the beta distribution and the resulting optimal time-points given the initial information are correctly specified. When examining the values in Table 3.1, we can observe patterns across different scenarios. In scenarios, where antibody levels do not start in zero, a sample is consistently taken at time 0. There is also a connection observable between the speed of increase and the height of the plateau. Scenarios 1 and 5, both characterized by a faster increase and a higher plateau, yield nearly identical parameters $(\alpha, \beta, t_{scale}, c)$ and sampling times, with an extra sample at time 0. Scenario 9, similarly showing a faster increase with a lower plateau but starting at a higher level, again results in comparable parameters, with differences in the shifting term c, but different optimal sampling times. A comparable pattern emerges in scenarios characterized by a faster increase with a higher plateau. Scenarios 3 and 7, starting in zero or at a low level, yield very similar parameters and sampling times. In contrast, scenario 11, starting at a higher

Table 3.1.: Parameters of the beta distribution and optimal time-points by scenario. The scenarios are defined in Table 2.2 (page 40). Note that in scenarios 1-4 $A_0 = 0$ and therefore three time-points are optimized. Reproduced from Embacher et al. [108]

Scenario	alpha	beta	t_{scale}	c	time1	time2	time3	time4
1	1.24	15.81	1919.88	0.00	0.30	29.67	183.83	-
2	1.79	23.46	2056.98	0.00	11.44	69.91	211.03	-
3	1.15	14.12	2706.91	0.00	0.03	29.54	260.40	-
4	1.49	19.80	2741.19	0.00	5.54	69.76	269.56	-
5	1.24	15.81	1919.89	0.00	0.00	0.35	30.17	184.44
6	1.82	23.72	2067.73	1.63	0.00	15.02	77.01	219.17
7	1.15	14.12	2706.91	0.00	0.00	0.07	31.63	262.93
8	1.49	19.83	2742.92	0.25	0.00	7.07	73.83	274.33
9	1.24	15.92	1926.89	0.66	0.00	3.28	44.92	202.59
10	2.23	27.84	2218.27	27.31	0.00	30.80	108.87	254.92
11	1.15	14.13	2707.91	0.10	0.00	0.91	43.30	277.31
12	1.59	21.13	2812.25	10.68	0.00	21.67	108.27	313.48

level, leads to a different optimal sampling schedule. In scenarios with a slower increase, a higher plateau and starting in zero or at a lower level, specifically scenarios 4 and 8, the parameter values are quite similar, though the sampling times can vary by up to 5 days. In contrast, scenario 12, starting at a higher level, shows notable differences in both, parameter estimates and sampling times. A similar pattern appears in scenarios 2 and 6, characterized by a slower increase and a lower plateau, the parameters are comparable, but sampling times differ up to 8 days. Scenario 10, however, deviates in parameters and optimal sampling times. To analyse the effect of misspecification in the initial information on the study design, we vary them on ranges as defined in Section 2.4.1. In total 12 scenarios are distinguished between, as described in Table 2.2 on page 40.

3.1.1. Single Parameter Misspecification

Figure 3.1 and Table 3.2 show that when pooling all scenarios, we observe that misspecifications in the height of the maximum A_{max} influence the D-efficiency of the respective design only marginally (median 1, range: 0.98-1.0). A comparable outcome is observed for the time of the plateau t_{plat}, where only marginal effects on the D-efficiency can be observed (median 0.99 range: 0.97-1.00). On the other hand, misspecification in the height of the plateau A_{plat} (median 0.96, range: 0.55-1.0) and the time of the maximum t_{max} (median 0.98, range: 0.43-1.0) show a stronger effect on the D-efficiency. For the starting value A_0, we assume that it is known, whether $A_0 = 0$ or not and therefore it is only varied in scenarios 5-12, with a median D-efficiency of 0.99 across the scenarios (range:

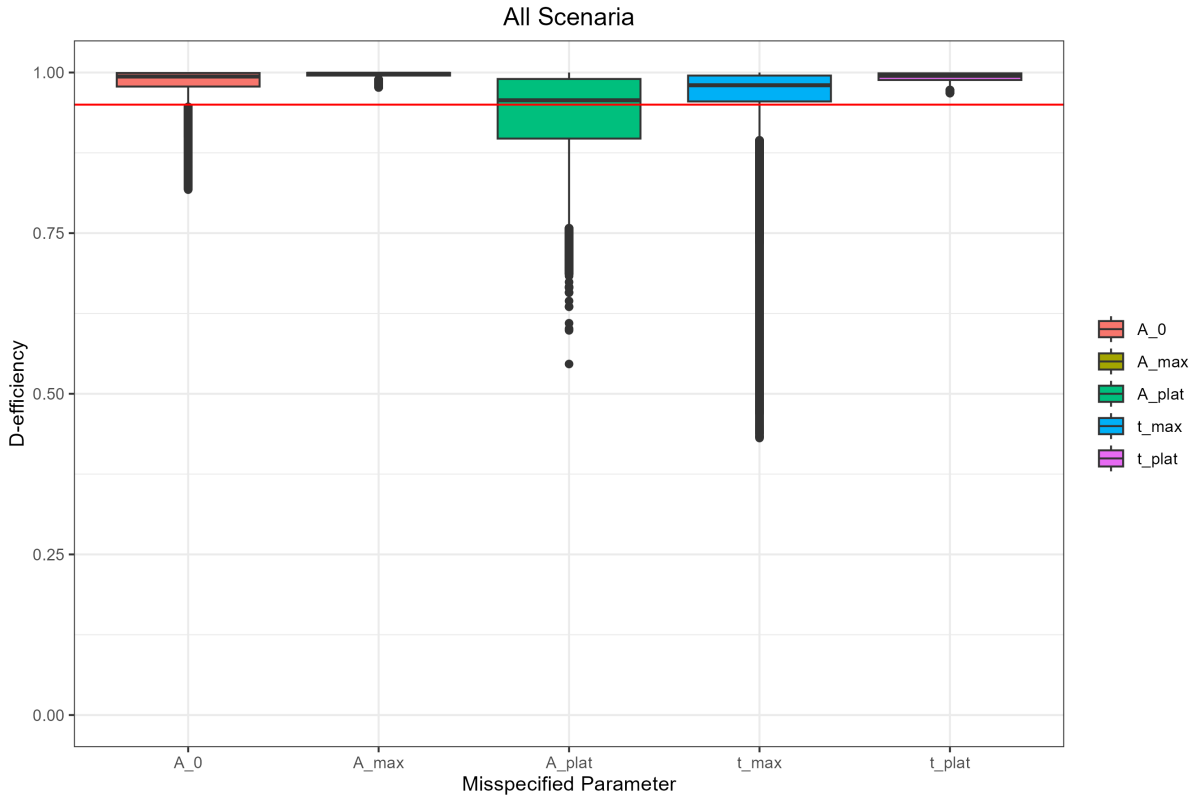

Figure 3.1.: Boxplot displaying the D-efficiency across all pooled scenarios, with the red line marking a D-efficiency of 0.95. The misspecified parameters represent the initial information used to determine the beta distribution parameters, with misspecifications varying within specified ranges on a grid of size N=1001 for each scenario. Reproduced from Embacher et al. [108]

0.82-1.0).

Examining the effect of misspecification on the D-efficiency scenariowise, as displayed in Figure 3.2, shows that for all scenarios the D-efficiency of the design is affected by misspecification in A_{plat}, where we observe minimal D-efficiencies ranging from 0.55 (scenario 6; cp. Table 3.3) to 0.86 (scenario 12; cp. Table 3.3). Scenario 9 and scenario 11 show the strongest effect for t_{max} and A_0, while for t_{max} we observe minimal D-efficiencies of 0.47 (scenario 9) and 0.43 (scenario 11), for A_0 the minimal D-efficiencies are 0.90 (scenario 9) and 0.82 (scenario 11). Both scenarios have in common that they are starting at a higher level and show a faster increase in antibody levels (cp. Table 2.2 on page 40).

Since the time of the maximum and the height of the plateau were identified as the most sensitive initial information, a detailed analysis of how the misspecifications influence deviations from the optimal time-points is conducted. Figure 3.3 shows the deviation of the sampling times and the D-efficiency of the respective design for t_{max}, while Figure 3.4 shows it analogously for A_{plat}. All measurements are influenced by misspecifications in t_{max} (cp. Figure 3.3), where for scenarios 1-4 the second measurement and for scenarios

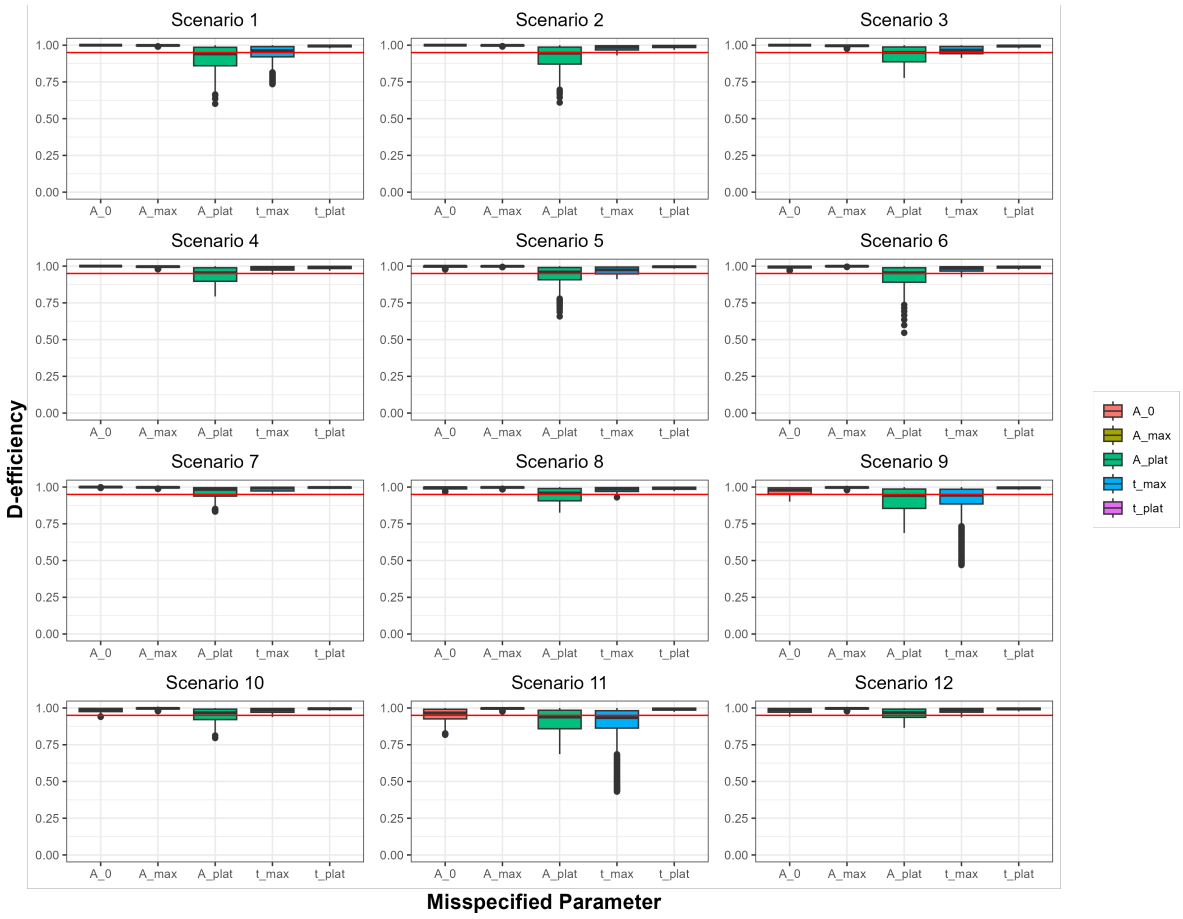

Figure 3.2.: Boxplot displaying the D-efficiency across all scenarios separated, with the red line marking a D-efficiency of 0.95. The misspecified parameters represent the initial information used to determine the beta distribution parameters, with misspecifications varying within specified ranges on a grid of size N=1001 for each scenario. Reproduced from Embacher et al. [108]

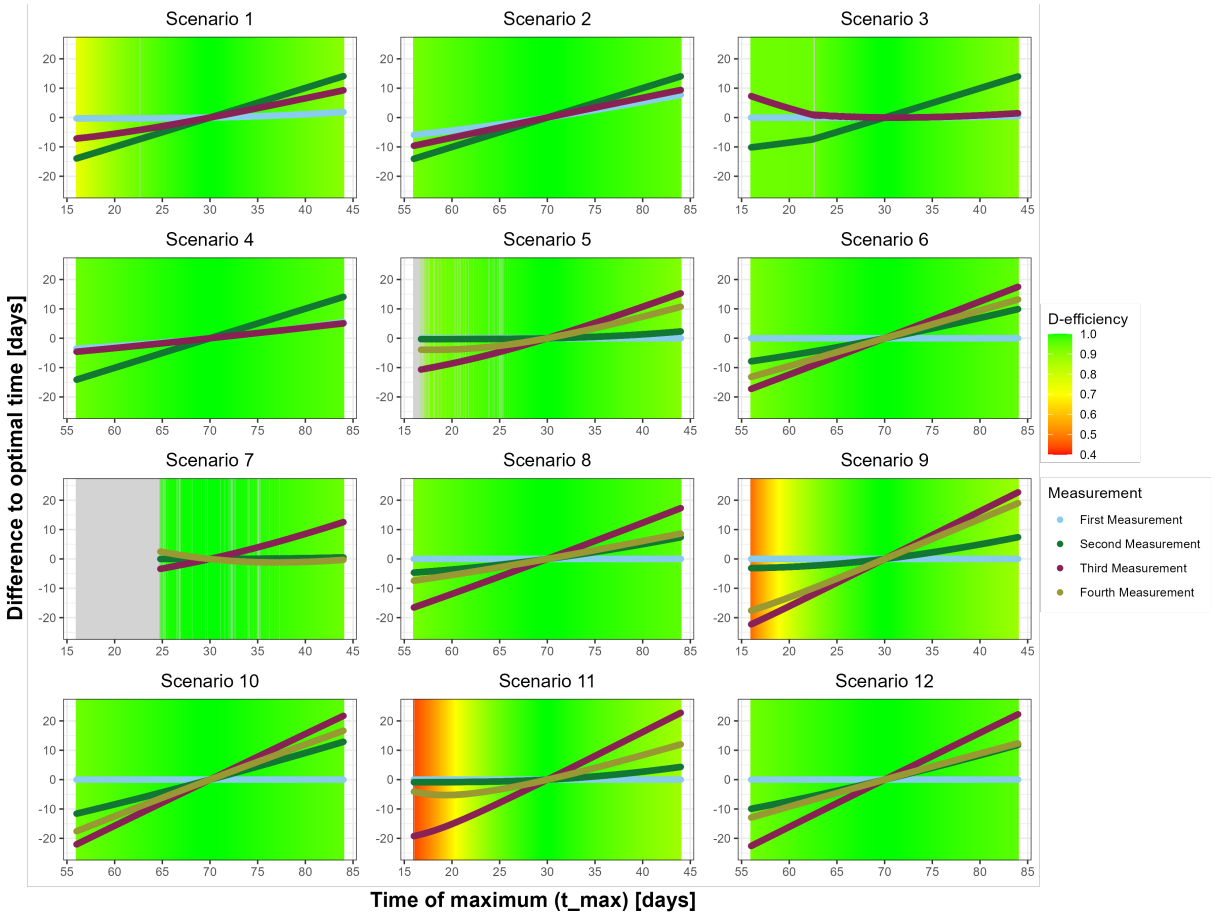

Figure 3.3.: Plots showing the deviation of sampling times in relation to misspecification in the time of maximum t_{max}. Each plot corresponds to a different scenario. The lines show the deviation of optimal sampling times based on misspecified initial information compared to those derived from the 'true' initial information. For some scenarios the deviation for the first measurement are not visible, because it is overlaid by the others. The color gradient represents the D-efficiency of each design, illustrating the impact of sampling time deviations on D-efficiency. In scenarios 1–4, three time-points are optimized, whereas in scenarios 5–12, four time-points are optimized. The grey area denotes settings where no solution could be determined. Reproduced from Embacher et al. [108]

Table 3.2.: D-efficiency for all scenarios pooled together, reported as minimum, first quartile (Q1), median, third quartile (Q3), and maximum. Additionally, the absolute and relative frequencies of non-converging runs are reported. For A_0, a total of 8008 runs were conducted, while all others had 12,012 runs. Reproduced from Embacher et al. [108]

Characteristic	Minimum	Q1	Median	Q3	Maximum	Not converging
A_0	0.82	0.98	0.99	1	1	269 (3.4%)
A_{max}	0.98	1	1	1	1	151 (1.3%)
A_{plat}	0.55	0.90	0.96	0.99	1	437 (3.6%)
t_{max}	0.43	0.95	0.98	1	1	442 (3.7%)
t_{plat}	0.97	0.99	0.99	1	1	133 (1.1%)

Table 3.3.: The D-efficiency for misspecification in the respective initial information and scenario. Values are reported as median (range). Reproduced from Embacher et al. [108]

	A_0	A_{max}	A_{plat}	t_{max}	t_{plat}
Scenario 1	-	1.00 (0.99-1.00)	0.94 (0.60-1.00)	0.96 (0.73-1.00)	1.00 (0.98-1.00)
Scenario 2	-	1.00 (0.99-1.00)	0.95 (0.61-1.00)	0.99 (0.93-1.00)	0.99 (0.97-1.00)
Scenario 3	-	1.00 (0.98-1.00)	0.95 (0.78-1.00)	0.97 (0.91-1.00)	0.99 (0.98-1.00)
Scenario 4	-	1.00 (0.98-1.00)	0.96 (0.79-1.00)	0.99 (0.94-1.00)	0.99 (0.97-1.00)
Scenario 5	1.00 (0.98-1.00)	1.00 (0.99-1.00)	0.96 (0.66-1.00)	0.97 (0.91-1.00)	1.00 (0.98-1.00)
Scenario 6	0.99 (0.97-1.00)	1.00 (1.00-1.00)	0.96 (0.55-1.00)	0.98 (0.92-1.00)	0.99 (0.98-1.00)
Scenario 7	1.00 (0.99-1.00)	1.00 (0.99-1.00)	0.98 (0.83-1.00)	0.99 (0.94-1.00)	1.00 (0.99-1.00)
Scenario 8	1.00 (0.97-1.00)	1.00 (0.98-1.00)	0.96 (0.83-1.00)	0.99 (0.93-1.00)	0.99 (0.97-1.00)
Scenario 9	0.98 (0.90-1.00)	1.00 (0.98-1.00)	0.94 (0.69-1.00)	0.94 (0.47-1.00)	1.00 (0.98-1.00)
Scenario 10	0.99 (0.94-1.00)	1.00 (0.98-1.00)	0.97 (0.80-1.00)	0.99 (0.94-1.00)	1.00 (0.98-1.00)
Scenario 11	0.97 (0.82-1.00)	1.00 (0.98-1.00)	0.94 (0.69-1.00)	0.93 (0.43-1.00)	0.99 (0.97-1.00)
Scenario 12	0.99 (0.94-1.00)	1.00 (0.98-1.00)	0.97 (0.86-1.00)	0.99 (0.94-1.00)	0.99 (0.98-1.00)

5-12 the third measurement are showing the strongest deviation from the optimal sampling times. As previously mentioned and displayed in Figure 3.2 and Figure 3.3, the strongest effect on the D-efficiency with regards of misspecification in t_{max} is observed for scenario 9 and scenario 11. For all other scenarios the effect of misspecification in t_{max} on the D-efficiency are practically negligible. Misspecification in A_{plat} primarily influence the last measurement (cp. Figure 3.4), which is the third measurement for scenarios 1-4 and the fourth measurement for scenarios 5-12. If A_{plat} is specified as too low, the last measurement is planned too early and vice versa, while depending on the degree of misspecification, deviations of up to 100 days are observed for the given scenarios and ranges of misspecification. Nevertheless, this strong deviation only slightly worsens the D-efficiency of the respective design. The figures for A_{max}, t_{plat} and A_0 are given in the Appendix as Figures E.1, E.2 and E.3 (from page 106 onwards). For A_{max} and t_{plat} the minimally observed D-efficiencies are 0.977 and 0.968, respectively. For both, mainly the last measurement is influenced by the misspecification, while for A_{max} scenarios 9-

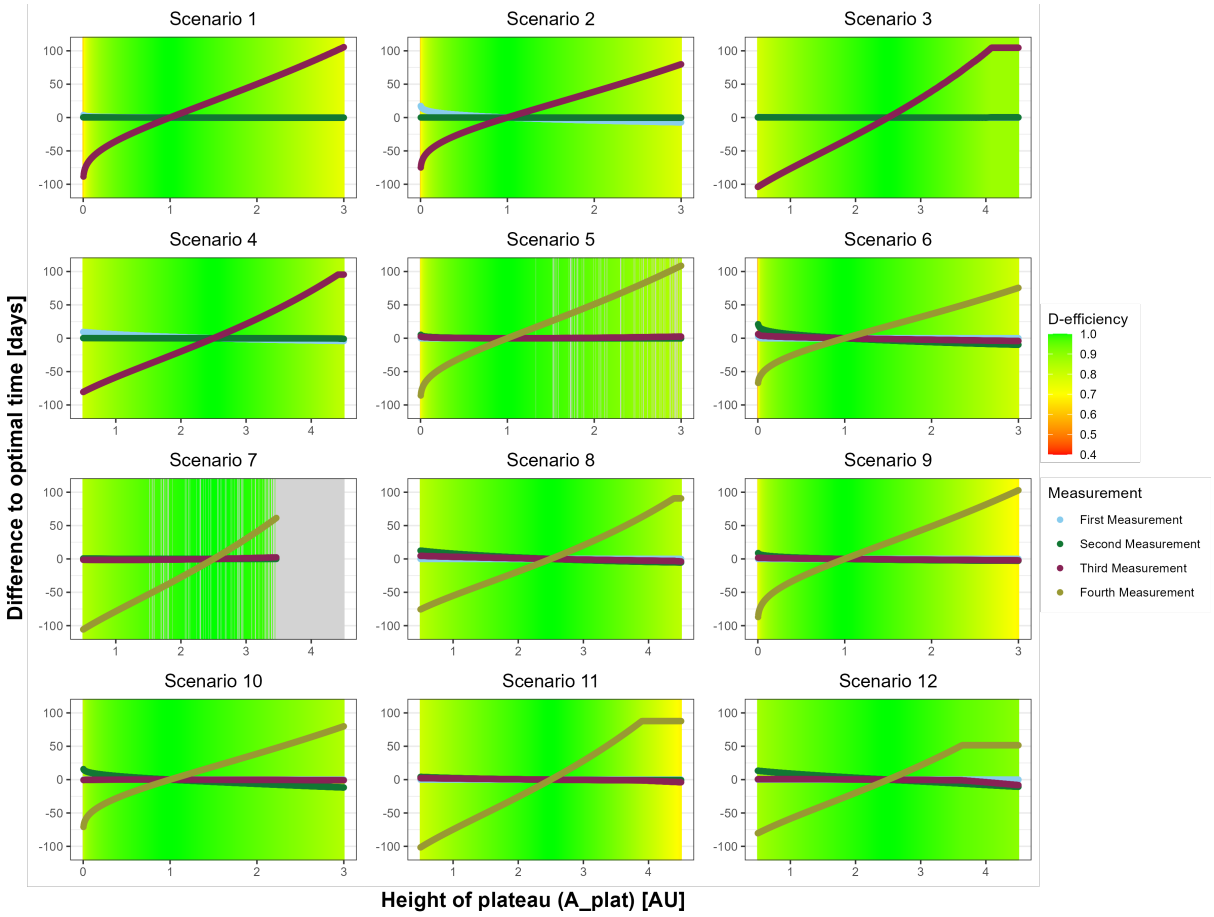

Figure 3.4.: Plots showing the deviation of sampling times in relation to misspecification in the height of the plateau A_{plat}. Each plot corresponds to a different scenario. The lines show the deviation of optimal sampling times based on misspecified initial information compared to those derived from the 'true' initial information. For some scenarios the deviation for the first measurement are not visible, because it is overlaid by the others. The color gradient represents the D-efficiency of each design, illustrating the impact of sampling time deviations on D-efficiency. In scenarios 1–4, three time-points are optimized, whereas in scenarios 5–12, four time-points are optimized. The grey area denotes settings where no solution could be determined. Reproduced from Embacher et al. [108]

12 show observable deviations in all measurements (cp. Figure E.1). Figure E.3 shows that for misspecification in A_0 all measurements are influenced, while scenarios 1-4 do not show any results by design. Across all 5 initial information, certain scenarios lead to numerical issues. Specifically, there are issues in scenario 7 for all initial information, while scenario 5 shows numerical issues for all except A_{max}. These numerical problems arise from both, solving for the parameters of the beta distribution and solving for the optimal time-points. Further, for t_{max} Figure 3.3 shows issues around the value of 22.7 in scenario 3. All three scenarios have in common, that they start low (or in zero) and show a faster increase. A discussion of the results, potential reasons for the numerical issues and practical implications are given in Section 4.

3.1.2. Double Parameter Misspecification

To analyse the robustness of our framework regarding simultaneous misspecification in two initial information, we discuss the D-efficiency, specifically the minimum, first quartile (Q1), and the median, as displayed in Figures 3.5, 3.6 and 3.7. In these figures, each row represent one scenario (see Table 2.2 on page 40) and each column represent one pair of misspecification. The corresponding figures for the third quartile (Q3; Figure E.4), the number of runs not converging (Figure E.5) and the number of runs showing a D-efficiency of 0.95 or above (Figure E.6) can be found in the Appendix (from page 109 onwards). Given that there are 5 initial information, ten possible pairs of double misspecification are discussed.

- The pair (t_{plat}, A_{max}) shows robust values with regard of misspecification (minimum: 0.90 (scenario 4), Q1: 0.98 (several scenarios)). See also Figure E.7 in the Appendix for a graphical representation by scenario.

- The pairs (A_0, A_{max}) and (A_0, t_{plat}) show robust results for scenarios 1-8, where all minimal values are greater equal to 0.92 and Q1-values are greater equal to 0.98. However, scenarios 9-12 show less robust results with minimum values ranging between 0.65 (scenario 10 for (A_0, A_{max})) and 0.92 (scenario 10 for (A_0, t_{plat})), with several values being below 0.8. See also Figures E.8 and Figure E.9 in the Appendix for a graphical representation by scenario.

- The pair (A_0, t_{max}) shows robust results for scenarios 1-8, where all Q1-values are greater equal 0.92 and all medians exceed 0.95. The minimal values are comparably good, while not being optimal (ranging from 0.73 (scenario 1) to 0.94 (scenario 4)). In scenarios 9-12, the minimal values are ranging from 0.22 (scenario 9) to 0.81 (scenario 10), while the median values for scenario 9 and scenario 11 are below 0.95.

See also Figure E.10 in the Appendix for a graphical representation by scenario. In this figure, we can also observe that numerical issues arise due to a combination of a starting value that is too low and a maximum value that is specified too early.

- The pair (t_{max}, A_{plat}) shows the lowest or close to lowest values for Q1 (ranging from 0.75 (scenario 9) to 0.92 (scenarios 7 and 12)) and the median (ranging from 0.90 (scenarios 1, 9 and 11) to 0.96 (scenarios 7 and 12)) across all scenarios, while also showing low minimal values, ranging from 0.02 (scenario 5) to 0.69 (scenario 7, 10 and 12; cp. Figure 3.5). See also Figure E.11 in the Appendix for a graphical representation by scenario. We can also observe that numerical issues primarily arise from specifying the maximum too early, while setting the plateau height too high amplifies the impact of this misspecification.

- When visually examining Figures 3.5, 3.6 and 3.7 the pairs (A_0, A_{plat}), (A_{max}, A_{plat}), (t_{plat}, A_{plat}) can be considered suboptimal compared to the others. All three show a minimal D-efficiency of 0.02 in scenario 5, while all minimal D-efficiencies are below 0.8 (except scenarios 7 and 12 for (A_{max}, A_{plat})). All Q1-values are around 0.9, with scenarios 1, 9 and 11 showing the lowest values. The same is true for the median, where all values are around 0.95. See also Figures E.12, E.13 and E.14 in the Appendix for a graphical representation by scenario.

- The pairs (t_{max}, t_{plat}) and (t_{max}, A_{max}) show robust values for scenarios 2-8, 10 and 12, with minimal values ranging from 0.84 (scenarios 3 and 8 for (t_{max}, t_{plat})) to 0.91 (scenario 7 and 12 for (t_{max}, A_{max})) and Q1-values being greater equal 0.94. In scenario 1, both show a minimal D-efficiency of 0.71 and a Q1-value of 0.92. In scenarios 9 and 11 the minimal D-efficienies are less than 0.45 and all four do not exceed a median D-efficiency of 0.95. See also Figure E.15 and Figure E.16 in the Appendix for a graphical representation by scenario. In these figures we can also see, that numerical issues mainly arise in scenarios with an earlier peak and if we specify the maximum as too early.

In the Figures 3.5, 3.6 and 3.7, we can see a distinction between scenarios 1-8 and 9-12. In scenarios 1-4 A_0 is not varied, because we assume that it is known whether we work with patients with pre-existing antibodies. Scenarios 1-8 assume that the initial antibody concentration is non-existing or low, while scenarios 9-12 assume higher initial concentration. Figures for each scenario showing all pairwise misspecifications can be found in the Appendix as Figures E.17- E.28 (from page 122 onwards). In general, scenarios 1-8 show good results, where the Q1-values are above 0.92 for all pairs not involving A_{plat}. For all pairs, scenario 1 has the tendency to show slightly lower Q1-values and median

values compared to scenarios 2-8 (cp. Figures 3.6 and 3.7). Scenario 5 shows the smallest minimum with a value of 0.02 (for pairs including A_{plat}), where Figure E.21 shows that those small values can be found on the boarder to non-convergence. When analyzing convergence, as shown in Figure E.5, which displays the number of non-converging runs, we observe that all scenarios except scenario 12 exhibit some degree of non-convergence. However, scenario 5 and especially scenario 7 have high numbers of runs not converging (up to 3770 for scenario 7 and pair (t_{max}, A_{plat})). Scenarios 10 and 12 show robust Q1-values, where all values are above 0.9 for scenario 10 and above 0.92 for scenario 12. Scenario 9 and 11, however, show less robust results, where Q1-values are ranging between 0.75 and 0.98, with several values being below 0.9. All 4 scenarios (9-12) show small minimal values, which are observed on the boarder to convergence (cp. Figures E.25, E.26, E.27 and E.28). Despite the fact, that some scenarios show better results than others, robustness can be mainly attributed to the pair of misspecification. The worst values are, in general, observed for pairs where A_{plat} is misspecified. Further, Figures E.11, E.12, E.13 and E.14 show that convergence issues arise when $A_{plat} = 0$. Most of the convergence issues arise however, in scenarios with an earlier maximum misspecified as too early. Since we consider a D-efficiency of 0.95 as practically optimal, the number of runs that show a D-efficiency > 0.95 are provided in Figure E.6. These numbers confirm what we already observed in the other figures. Namely, that pairs including A_{plat} show the least robust results, followed by pairs that include t_{max} in scenarios with an earlier maximum. As a result, the pair (t_{max}, A_{plat}) show the least robust results of all pairs. A discussion of the results and potential practical implications are given in Section 4.

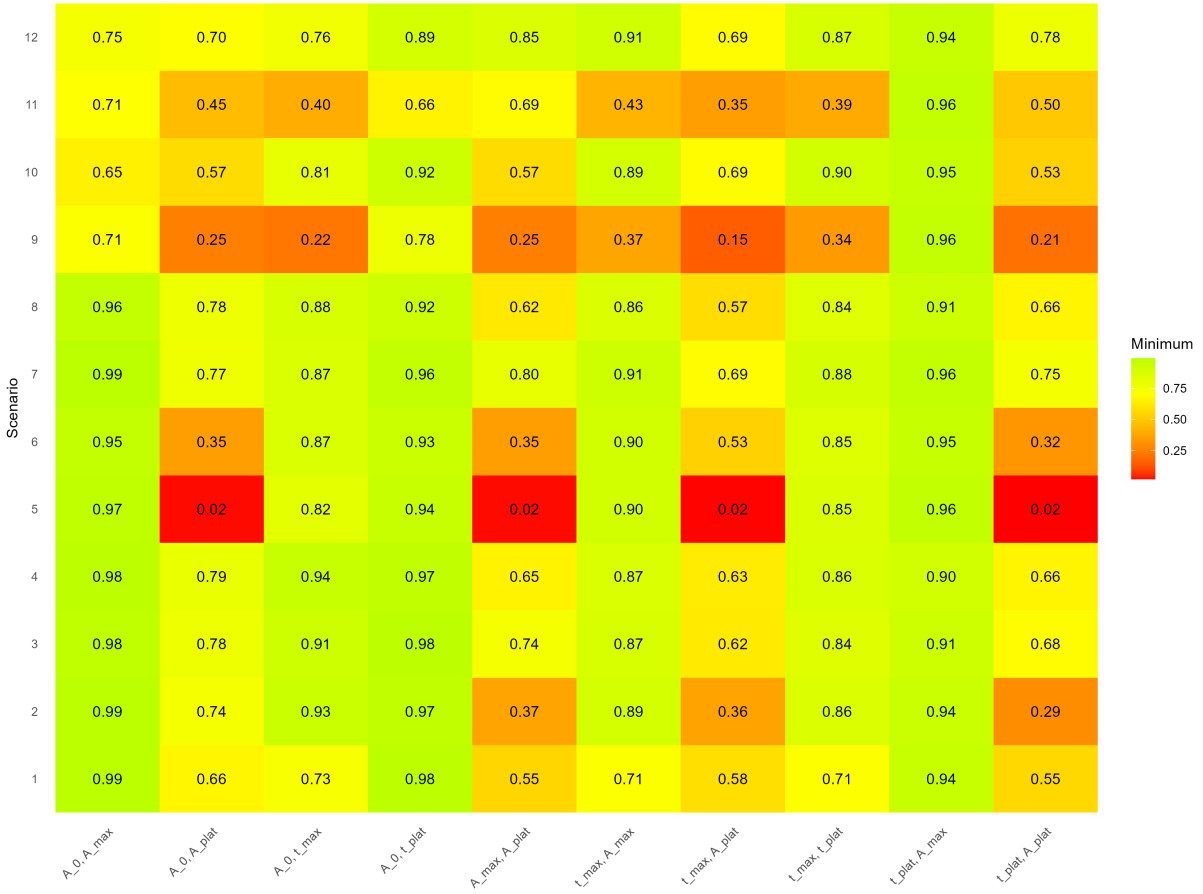

Figure 3.5.: Heatmap of the minimum D-efficiency values across different scenarios and pairs of misspecification in the initial information. The color gradient represents the D-efficiency, with green indicating higher values and red representing lower values. The x-axis corresponds to different pairs, while the y-axis represents the scenario index.

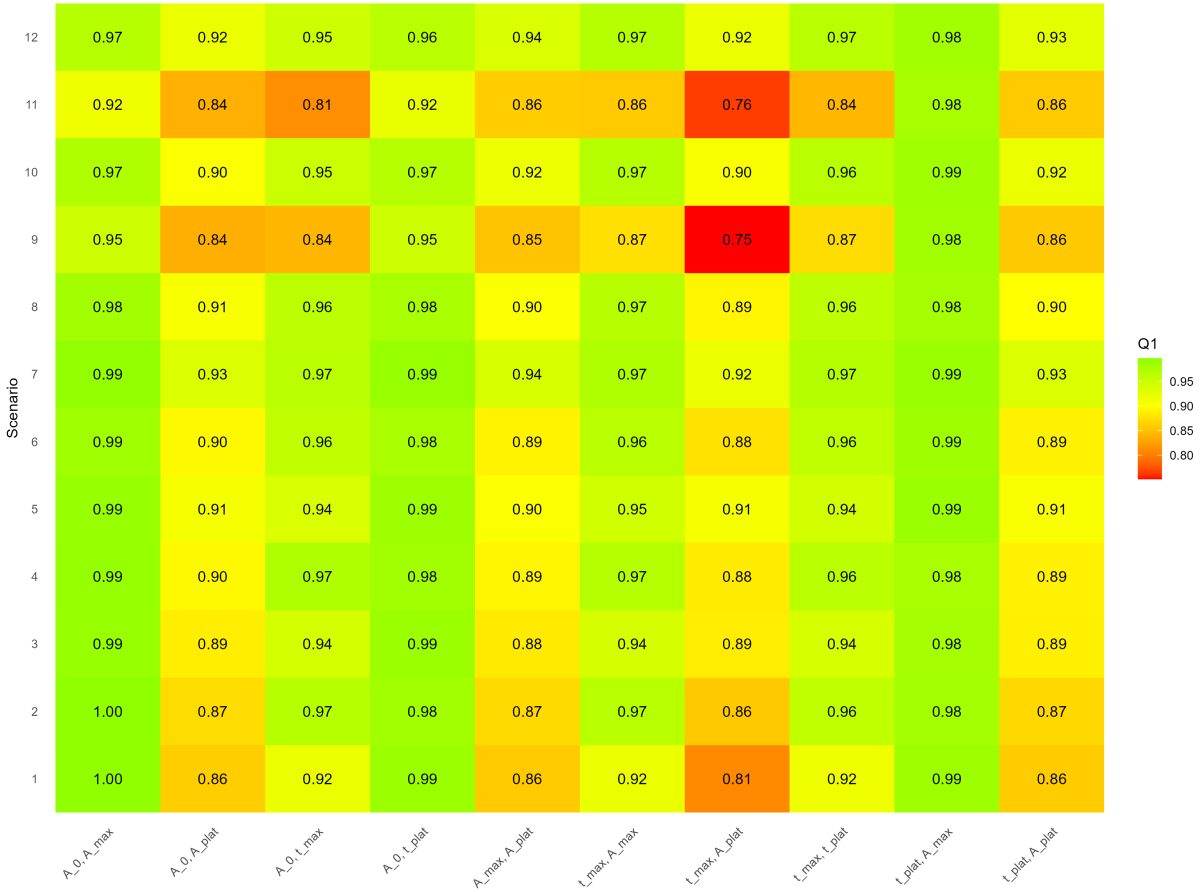

Figure 3.6.: Heatmap of the first quartile D-efficiency values across different scenarios and pairs of misspecification in the initial information. The color gradient represents the D-efficiency, with green indicating higher values and red representing lower values. The x-axis corresponds to different pairs, while the y-axis represents the scenario index.

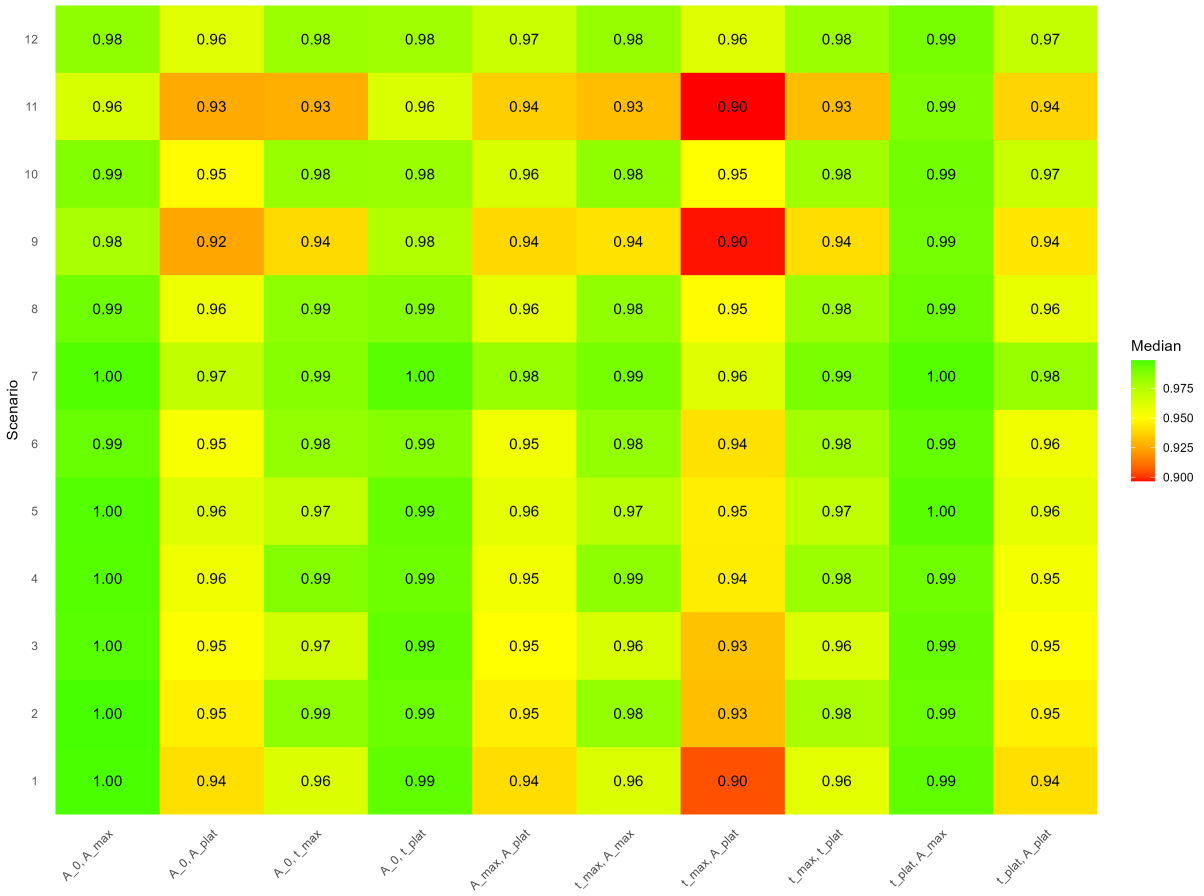

Figure 3.7.: Heatmap of the median D-efficiency values across different scenarios and pairs of misspecification in the initial information. The color gradient represents the D-efficiency, with green indicating higher values and red representing lower values. The x-axis corresponds to different pairs, while the y-axis represents the scenario index.

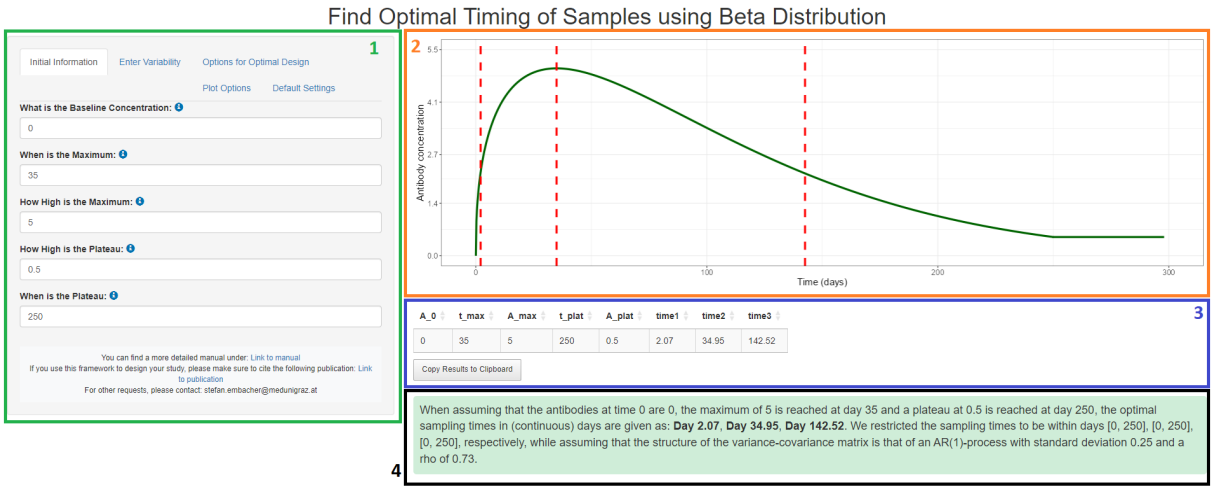

Figure 3.8.: The graphical user interface, when opening the Shiny application. The interface consists of four main panels. Green Panel 1: In total 5 different tabs allowing the user to specify input. Orange Panel 2: The resulting curve and optimal sampling times graphically displayed. Blue Panel 3: The most important input and the numerical values of the optimal sampling times. Black Panel 4: Everything summarized to share with colleagues or in a publication, assuring reproducibility.

3.2. Shiny Application

The main objective is to provide an easily accessible tool to apply the developed framework (cp. Section 2.3). The described version of our deployed Shiny application is Version 1.0 and accessible on Zenodo.[123] This description and the implementation is based on R Version 4.4.1 and Shiny version 1.10.0.[117, 124] In Section 3.2.1 the different panels of the user interface are described, while Section 3.2.2 describes the input and Section 3.2.3 the output in more detail.

3.2.1. Graphical User Interface

The Shiny application is structured in four panels, as displayed in Figure 3.8, where values for all possible variables are set to default values. Panel 1 (green) asks for the input (initial information and variability) and allows to set options or restrictions on the time-windows. Panel 2 (orange) displays the resulting curve based on the assumptions made and the resulting parameters of the beta distribution (cp. Section 2.3.2) indicating the optimal sampling times as dashed red lines (cp. Section 2.3.4). Additionally, panel 3 (blue) displays the used initial information and the resulting sampling times while providing the option to copy all variables, including variability and restrictions on time-windows, to the clipboard. Further, panel 4 (black) shows a text-phrase including all relevant information which can be copied and used with collaborators or in a publication.

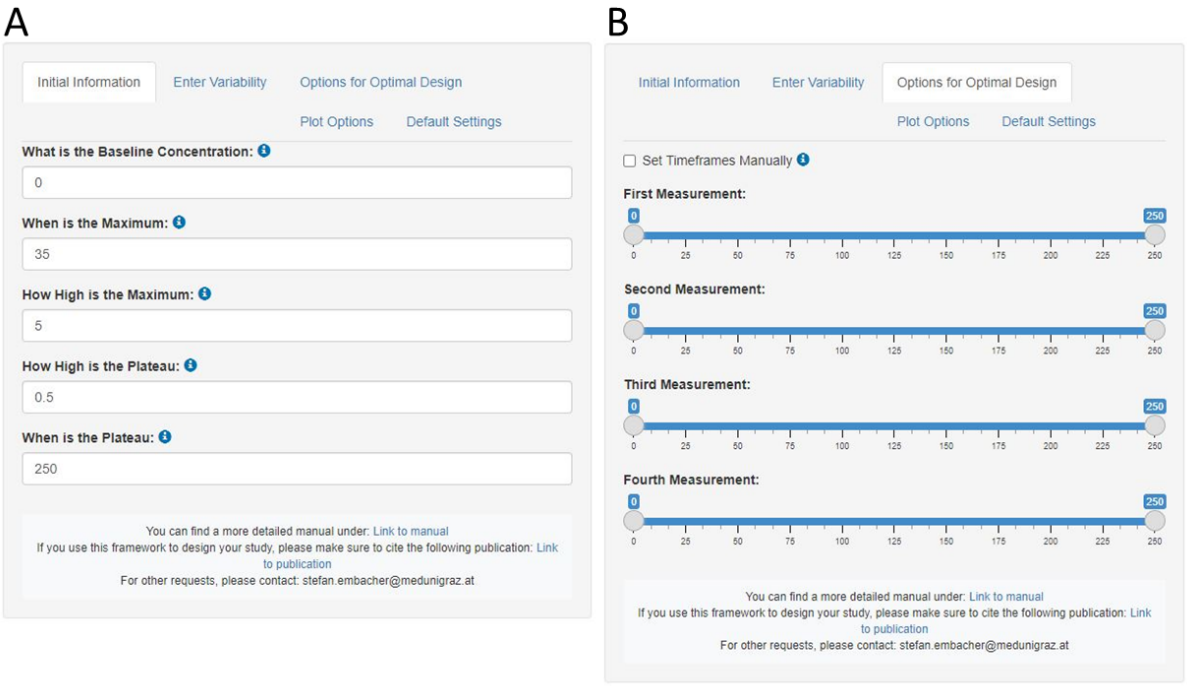

Figure 3.9.: Two input tabs of the Shiny application. Panel A asks for the initial information $A_0, t_{max}, A_{max}, A_{plat}, t_{plat}$. Panel B allows to set restrictions on the time-windows for the measurements.

3.2.2. Input

Initial Information

As displayed in Figure 3.9, panel A, the tab 'Initial Information' asks for the five information as specified in Section 2.3.2 ($A_0, t_{max}, A_{max}, A_{plat}, t_{plat}$). The info-icon has the purpose to clarify the input and draw the users attention to specific issues. The value 'What is the Baseline Concentration' specifies A_0 and therefore directly determines whether we optimize three or four measurements. The text in the info-icon *"The starting value A(0) denoted by A_0 in the framework. Note that A_0 = 0 implies 3 measurements and A_0 > 0 implies 4 measurements"* should draw the users attention to that fact. 'When is the Maximum' and 'How High is the Maximum' determine the values of t_{max} and A_{max}, respectively. The user is provided this information to clarify what they specify. A_{plat} is specified in 'How High is the Plateau' and through the text in the info-icon *"The (mean) antibody concentration when reaching a plateau, denoted by A_plat in the framework. Choose zero, the LLOD or a assumed stabilization value. Note: This is a modelling simplification."* offers practical choices and draws the users attention to the fact that the assumption of a plateau is a modelling simplification. 'When is the Plateau' specifies

t_{plat} and warns the user to take care when restricting the time-windows, through the text in the info-icon *"The time until the plateau is reached. Note: If you change t_plat, the limits on the time-windows adapt in Tab 'Options for Optimal Design'"*. More details on restrictions of the time-windows can be found under 'Options for Optimal Design'.

Enter Variability

As described in Section 2.4.1 we assume that the variance-covariance matrix is of an AR(1) structure, where the standard deviation σ and the correlation ρ are assumed to be known. Under the assumption $\Sigma(\theta) = \Sigma$, the choices of these values only influence a constant and therefore do not have an effect on the resulting optimal sampling times. However, they have an effect on the numerical properties, such as scaling, and might influence convergence. Figure 3.10, panel A, shows the corresponding input tab. In anticipation of the functionality in future versions, to allow for different variance-covariance structures and/or unknown variability, a single tab is already dedicated to the variability input within the Shiny application.

Options for Optimal Design

As displayed in Figure 3.9, panel B, the user can check a box, whether they want to set restrictions on the lower and upper limit of the respective measurement time-points. As described in Section 2.3.5, day 0 is the default value for the lower limit and t_{plat} is the maximum for the upper limit. Therefore ranges adapt to the value chosen as t_{plat}. Since the Hooke and Jeeves optimization algorithm interprets the time-points as parameters in an optimization problem, it does not enforce an ordered sequence for the sampling times. Therefore, users should be aware that if the restrictions are not properly ordered, the results may cause time-points to switch. For example, if the user specifies that the first sample should be taken between day 14 and 28, while not giving restrictions on the second time-point, we might observe something like $t_1 = 28$ and $t_2 = 0$. Furthermore, the Hooke and Jeeves method requires starting values that fall within the specified limits. If the predefined starting values (cp. Section 2.3.5) exceed these limits, simply the closest valid value is selected as the new starting point. To draw the user's attention to these points, the following text is provided in the info-icon: *"Select this option if you want to set restrictions on the time-windows. Be aware that measurements do not need to be ordered for the algorithm. So make sure to select the correct timeframes. Limits are max(0,selected value) and min(t_plat,selected_value). Please be aware, that if you have set A_0 = 0, that restrictions on the fourth time-window do not have any effect. Note: Restrictions on the time-windows might change the starting values of the optimization algorithm"*.

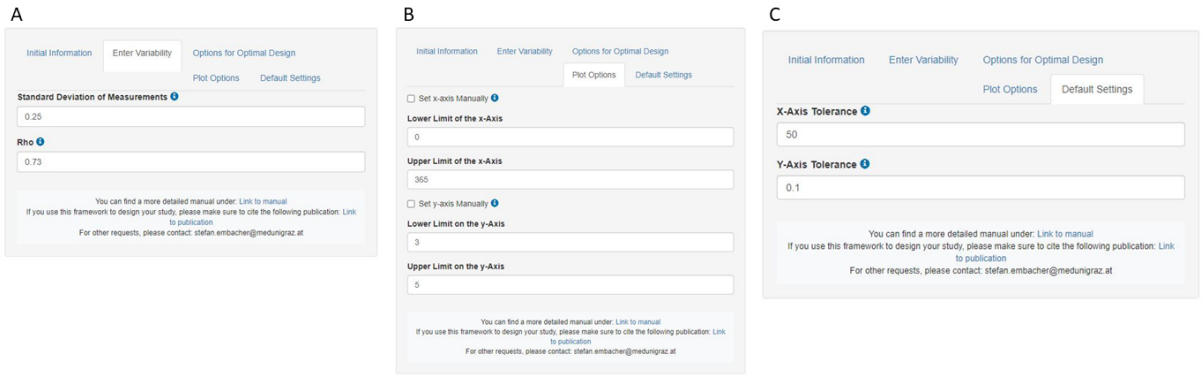

Figure 3.10.: Three input tabs of the Shiny application. Panel A asks for the standard deviation and the the correlation used in the AR(1) matrix. Panel B allows the user to set lower and upper limits on the respective axis in the plot and Panel C gives the option to change default values.

Plot Options

Under plot options (see Figure 3.10, panel B) the user can specify limits for the x-axis and the y-axis.

Default Settings

Under default settings, for now, there are only two values (see Figure 3.10, panel C). The first one is the display length of the plateau, which is called 'x-axis tolerance'. If the x-axis limits are set manually this value is ignored. The second one specifies the extra space shown above the y-axis maximum, unless the limits are specified by the user. The y-axis tolerance is the proportion of the maximum additionally display, i.e. the upper limit is $A_{max} + y\_axis\_tolerance * A_{max}$. In future versions, we plan to enable users to configure options for relevant functions throughout the framework's implementation.

3.2.3. Output

The output section of the Shiny application consists of three parts (Panel 2, 3, 4 in Figure 3.8. The orange panel (panel 2 in Figure 3.8) provides a visualisation of the curve derived from the assumptions made through the input and the resulting beta distribution, while the resulting optimal sampling times are displayed as dashed red lines. The visual representation allows the user to directly assess the effect of changes in the initial information or when applying restrictions on the time-windows. The blue panel (panel 3 in Figure 3.8) highlights the key input parameters and displays the resulting optimal sampling times numerically. Additionally, the user can copy the input and the results to the clipboard for further use. In addition to the displayed values, all others, includ-

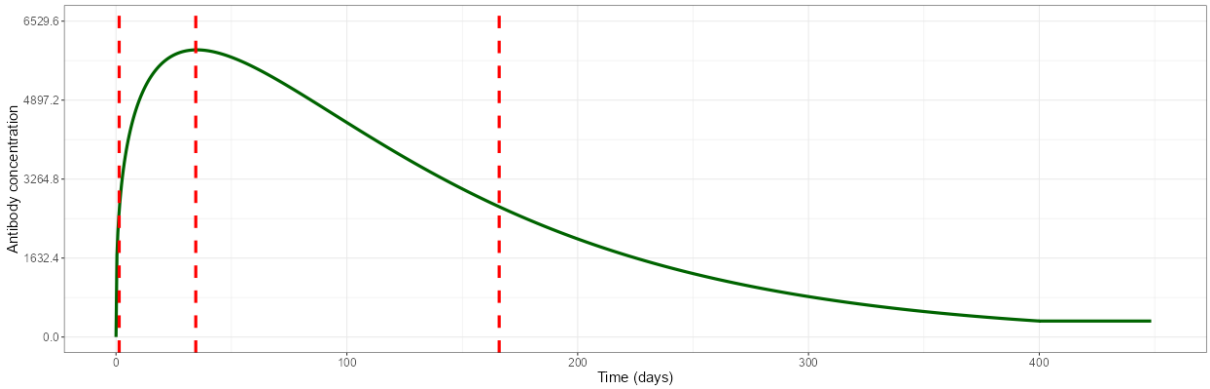

Figure 3.11.: The resulting curve and optimal sampling times based on the values shown in Example 1.

ing input on variability, time restrictions and beta parameters, are also copied to ensure reproducibility. The black panel (panel 4 in Figure 3.8) contains a comprehensive text summary of all relevant information, formatted for convenient sharing. Users can copy this summary for collaboration with colleagues or for direct inclusion in publications, ensuring transparency and reproducibility of the results.

3.3. Examples

The following section presents three examples illustrating how the framework works in hypothetical cases, while aiming to keep a strong practical connection. The primary challenge is the difficulty in obtaining reliable estimates for the timing or value of the plateau. A practical approach would be to interpolate results or rely on expert knowledge; however, for the following examples, numerically derived values are used to minimize subjectivity.

3.3.1. Example 1

In the first example we assume that we want to design a (hypothetical) study, where we aim to assess the antibody response in naive individuals following a SARS-CoV-2 vaccination. We base our study design on results published by Srivastava et al., [125], which have the great advantage, that they explicitly model a stable component corresponding to a phase where there is practically no decay. Because our hypothetical study is designed for naive individuals, the baseline concentration A_0 is assumed to be zero. Results in Srivastava et al. provide a peak value at day 35 with a value of 5936 AUC for naive individuals. Therefore we set $t_{max} = 35$ and $A_{max} = 5936$. Further, Srivastava et al. state that the *"... stable component had a magnitude of 330 AUC"* and they had an *"... approximately 400-day-long follow-up period ..."*.[125, pp. 589, 591] This is the reason why

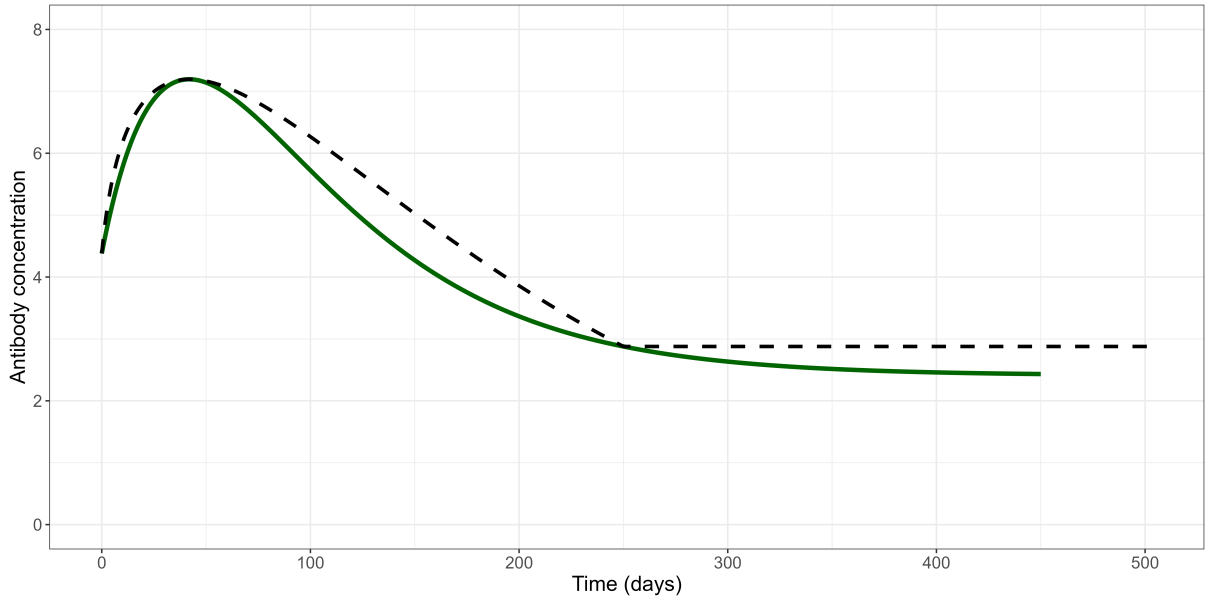

Figure 3.12.: The darkgreen line represents the fitted model by Violan et al.,[126] while the black dashed line shows the resulting curve from the framework under the given assumptions.

we chose $t_{plat} = 400$ and $A_{plat} = 330$. We are not restricted on any time-windows and assume a standard deviation of 200 and a strong correlation of 0.8; even though changes in the variability input influence the results only slightly due to numerical reasons. The resulting curve with the optimal sampling times are given in Figure 3.11. The time-points are day 1.38, day 34.71 and day 166, which practically means we are taking a sample on the 2nd day, the 35th day and the 166th day. The provided text states *"When assuming that the antibodies at time 0 are 0, the maximum of 5936 is reached at day 35 and a plateau at 330 is reached at day 400, the optimal sampling times in (continuous) days are given as: Day 1.38, Day 34.71, Day 166.15. We restricted the sampling times to be within days [0, 400], [0, 400], [0, 400], respectively, while assuming that the structure of the variance-covariance matrix is that of an AR(1)-process with standard deviation 200 and a rho of 0.8."*

3.3.2. Example 2

In the second hypothetical example, also described in the supplementary material of Embacher et al., [108] we design a study in which we aim to model the antibody kinetics after SARS-CoV-2 infection. We base the study design on results published by Violán et al.,[126] where we plan for the IgG against the nucleocapsid protein (IgG(N)). In their supplementary material they provide the modelling results, which allows us to directly derive the numerical values. We use the provided general equation, because at the plan-

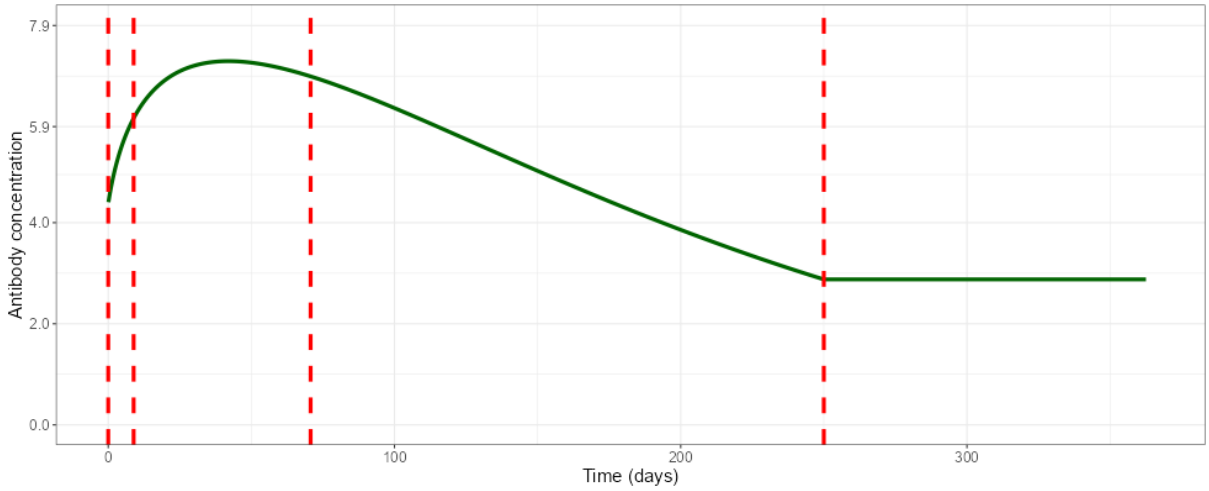

Figure 3.13.: The resulting curve and optimal sampling times based on the values shown in Example 2.

ning stage of the study we do not know, whether we face asymptomatic, mild or severe disease. The equation is given as

$$IgG(N)(t) = 2.41 + 1.97\exp(-10.85t) - 26.15(\exp(-10.85t) - \exp(-7.03t)),$$

where t is normalized for the maximum length of follow-up, which is 450 for IgG(N). Note, that this scaling corresponds to the opposite direction as used in the framework, instead of bringing it on an interpretable scale they normalize by the time of follow up. The resulting numerical values are $A_0 = 4.38$ and a maximum at day 42 with a value of 7.195, i.e. $t_{max} = 42$ and $A_{max} = 7.195$. When graphically inspecting the resulting curve described by the equation, as displayed in Figure 3.12, we assume the plateau at day 250, i.e. $t_{plat} = 250$, resulting in $A_{plat} = 2.878$. Assuming a standard deviation of 1 and a moderate correlation of 0.5, this results in samples taken at day 0, day 8.88, day 70.71, day 250 as displayed in Figure 3.13. However, in our hypothetical scenario, we want to restrict the time-windows for sample collection. Given that we are dealing with infections, we want to make sure, that patients are well monitored and therefore ask them to visit the hospital regularly. Consequently, we restrict the first sample to be taken within the first week, the second sample within the second week, and the third sample approximately one month after infection. Further, we want our study to end within half a year after infection and restrict the fourth time-window accordingly. Given these restrictions, we get the following output: *"When assuming that the antibodies at time 0 are 4.38, the maximum of 7.195 is reached at day 42 and a plateau at 2.878 is reached at day 250, the optimal sampling times in (continuous) days are given as: Day 0, Day 8, Day 38, Day 180. We restricted the sampling times to be within days [0, 7], [8, 14], [25, 38], [39, 180],*

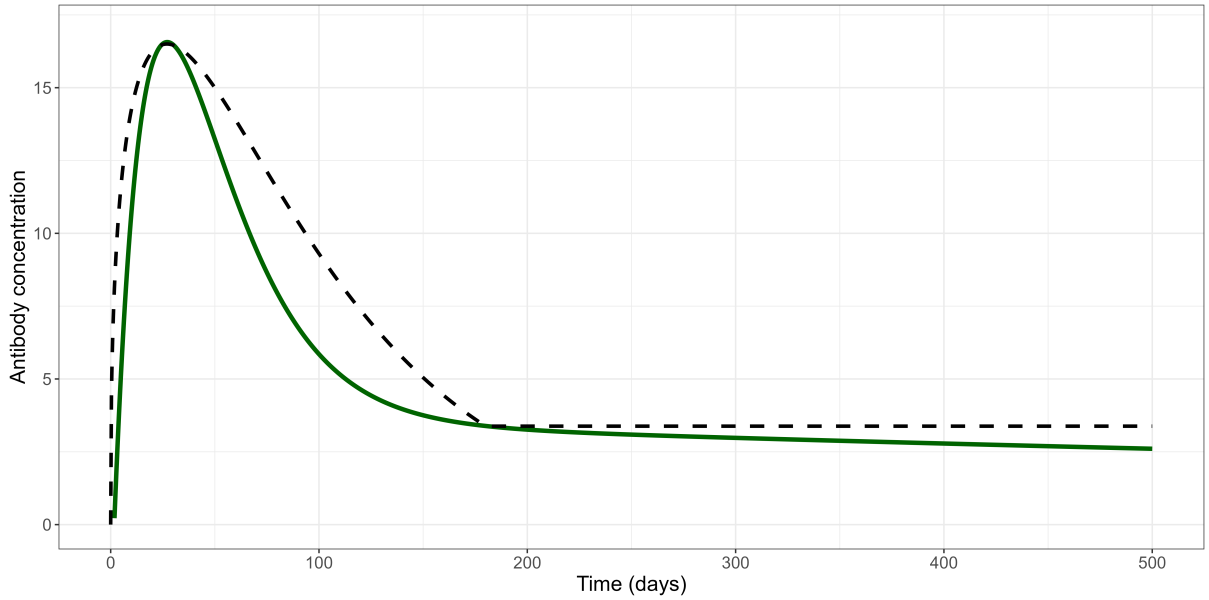

Figure 3.14.: The resulting curve and optimal sampling times based on the values shown in Example 3.

respectively, while assuming that the structure of the variance-covariance matrix is that of an AR(1)-process with standard deviation 1 and a rho of 0.5."

3.3.3. Example 3

In the third example, we plan a study to investigate the antibody response to natural *Plasmodium falciparum* infection in previously naïve individuals based on the results by Yman et al.[127] They present the functional structure used in their analysis, which is based on the model described by White et al.,[57] and provide a detailed list of parameter estimates in their supplementary material, including IgG-subclasses for different antigens. We base our study design on the total IgG for the apical membrane antigen 1 (AMA-1). Since we are designing the study for previously naïve individuals, the antibody concentration at time zero is assumed to be zero, i.e. $A_0 = 0$. Furthermore, since the functional structure and the parameter estimates are thoroughly described, we are able to determine the time and height of the maximum as $t_{max} = 27$ and $A_{max} = 16.5$. Again by visual inspection of the resulting curve (cp. Figure 3.14), we assume the plateau being reached at day 180, i.e. $t_{plat} = 180$ with an corresponding value of $A_{plat} = 3.38$. Since the variability parameters do not affect the solutions analytically, we stay with the default values. Additionally, we do not impose any constraints on the sampling windows. Overall, this provides us with the following result: *"When assuming that the antibodies at time 0 are 0, the maximum of 16.5 is reached at day 27 and a plateau at 3.38 is reached at day 180, the optimal sampling times in (continuous) days are given as: Day 1.34, Day 26.82, Day*

119.11. We restricted the sampling times to be within days [0, 180], [0, 180], [0, 180], respectively, while assuming that the structure of the variance-covariance matrix is that of an AR(1)-process with standard deviation 0.25 and a rho of 0.73."

4. Discussion

In this dissertation, a novel framework is introduced that utilizes easily understandable information to facilitate the design of immunization studies. Specifically, a practical tool for clinicians to design an optimal sampling schedule in studies, where modelling the antibody kinetics is the main goal, is provided. Assuming that antibody kinetics follow the shape of the density of a beta distribution until reaching a plateau, the baseline antibody concentration (A_0), the time and level of the maximum (t_{max}, A_{max}) and the time and value of the plateau (t_{plat}, A_{plat}) are sufficient to find an optimal sampling schedule. To examine how sensitive the framework reacts to misspecification in the initial information, a Robustness Analysis with single and double parameter misspecification was conducted. Generally, the framework showed robust results in most scenarios for a reasonable range of misspecification in terms of design efficiency. As described in Chapter 1, potential reasons for the lack of optimal design in immunization studies are the complexities involved in modelling antibody kinetics, paired with complexities arising from optimal design theory, and the absence of easily applicable and intuitively usable and understandable tools. In study design, guesses of the parameters of interest are needed to find optimal designs and obtaining these guesses might be very tricky, especially if models involve differential equations containing compartments with typically unmeasured information or complex functional structures. To allow clinicians to easily access and apply the developed framework, while lowering the programmatic burden, a Shiny application is made publicly available.[123] It allows to use easily understandable initial information, translates them to parameters used in the chosen functional structure and provides a closed implementation, where no coding and minimal mathematical understanding are needed to obtain optimal sampling times. Parts of the methods, limitations and results of the Robustness Analysis are already discussed in Embacher et al.[108]

This chapter starts with a detailed discussion about the selected methods in Section 4.1 and their implications, followed by a discussion of the results in Section 4.2. We then discuss limitations of the framework and limitations of the Robustness Analysis in Section 4.3, while providing potential solutions to some of these limitations and other extension possibilities in Section 4.4. The dissertation closes with a conclusion in Section 4.5.

4.1. Discussion of Methods

There is a large variety of optimality criteria, most of which follow a specific optimization purpose. In Section 2.2, for example, we saw a graphical representation of D-optimality compared with E-optimality. Others might allow for a compound criteria pursuing multiple optimization purposes or allow for the incorporation of costs.[10] However, since the purpose of this work is to introduce a framework, D-optimality was chosen as it is a very prominent and generally robust optimality criterion.[10] Besides the choice of the optimality criterion, optimal designs depend on the unknown values of the parameters. The simplest and arguably most intuitive way is to work with the best available guesses of those unknown parameters (e.g. coming from the literature), an approach typically referred to as locally optimal designs. Following the previous argumentation, locally optimal designs were used in the described framework. However, several potential approaches exist to reduce the dependence on the unknown parameters. The most pragmatic solution is to perform sensitivity analysis across the spectrum of parameter uncertainty to assess variation in the resulting optimal designs. Analytical approaches typically involve distributional assumptions on the parameter uncertainty (cp. Definition 12). For example, in ED-optimal designs, one considers the unknown parameters to come from an known prior distribution. The ED-optimal design will then be found by maximizing the expected value of the determinant of the FIMs. In contrast DE-optimal designs maximize the determinant of the expected value of the FIM.[128, 129] Adaptive designs might also be a feasible solution to account for the uncertainty in the parameters, however, since the purpose of the described framework is to introduce an easily accessible tool without the need of a strong mathematical background, adaptive designs would undermine this idea. Additional to the uncertainty in the unknown parameters, there is also model uncertainty. Strategies like model selection or model averaging could be used to tackle this problem.[130]

The FIM obviously depends on the distributional assumptions of the antibody measurements. Laboratory values are usually transformed to continuous values which typically follow either a normal or a log-normal distribution. We showed in Section 2.1.2 that the FIM is the same for both, commonly used distributional assumptions. If the variance-covariance matrix includes unknown parameters, complexities increase and the previous statement is, without further prove, not true anymore. Further work would be needed to proof the more general statement.

In many applications of optimal design sensitivity functions are used to find D-optimal designs (cp. Section 2.2 on page 31).[10] However, since our design space is strongly restricted (only one schedule with measurements in an interval) and the sensitivity function involves the evaluation and the inversion of multiple FIMs, an implementation that relies

on the Hooke and Jeeves method maximizing the FIM directly, is chosen. With an extension to more complex settings it might be beneficial to adapt the numerical procedures (cp. extensions in Section 4.4).

An optimal design does not necessarily need to be unique.[131, 132] This means that different starting values in the used optimization algorithm to determine the sampling times might affect the resulting optimal design. It is common, when optimizing study design, to only provide a single optimal solution, since if multiple designs are optimal, they show the same efficiency. Being able to describe the set of optimal solutions, however, might provide additional benefit and allows for a greater freedom in selecting an appropriate design.[132] In the case of D-optimal designs, it is easy to see that two designs can be equally as effective, since different FIMs can have the same determinant. On the other hand, we saw that the solution to the parameters of the beta distribution, which correspond to the guesses of the unknown parameters in our locally optimal design problem, are unique. We restricted ourselves to perform a small simulation study, as shown in Appendix D, which let us to conclude that these solutions are rather global than local.

Throughout the framework we distinguished between two different systems of equations, with the main difference of an additionally shifting term c. Theoretically nothing speaks against simply using the larger system, however, numerically issues arise if c becomes too small, it therefore is sensible to choose the corresponding system based on the expected value at time zero. Further, the assumption that the antibody kinetics follow the functional structure of the density of the beta distribution is used to simplify the needed initial information due to its flexibility compared to the low number of parameters. Generally, many different functional structures could be used to model the antibody kinetics and to optimize study design (cp. Section 1.2.2 and Section 4.4.1).

In immunization studies, where modelling antibody kinetics is the main objective, typically multiple measurements are made for each participant. Simplified it reduces to the questions of how often and when should we measure how many participants? In this work, the focus lies on optimal sampling times, while the chosen number of unknown parameters fixes the number of time-points.

For the Robustness Analysis, in both the single and double misspecification, the standard deviation σ and the correlation ρ are set to specific values. Through the made modelling assumptions, the choice of these values only affect the optimal solutions numerically, as shown in Appendix C. If we would assume another covariance structure or assume the parameters to be unknown, they would become part of the optimization problem and therefore increase the complexity (cp. Section 4.4.2).

Examples of the framework's application are presented in Section 3.3. We outline different approaches of gathering the required initial information, through the visual examination

of existing results to using modelling outcomes of prior studies. As with study design in general, the quality of the study design is directly proportional to the effort invested in determining the initial information. Ideally, all available resources, expert knowledge, modelling outcomes and graphical inspections should be combined.

4.2. Discussion of Results

The results of the Robustness Analysis for single misspecification are provided in Section 3.1.1, while the results for double parameter misspecification are given in Section 3.1.2. The Shiny application is introduced in Section 3.2 .

4.2.1. Single Parameter Misspecification

To assess the framework's sensitivity to misspecification in the initial information, a Robustness Analysis was performed. The results show median D-efficiencies above 0.95 for all single parameter misspecification and first quartiles greater than 0.9 for all parameters (cp. Table 3.2). The proposed framework shows generally robust results against single misspecification in the initial information. The Robustness Analysis indicates that for single parameter misspecification of the initial information, the height of the plateau and the time of the maximum are most sensitive. This can be, potentially, attributed to the impact of the information on the structure of the curve. In the flat and stabilizing phase, the height of the plateau is more important than its timing, while the timing of the maximum has more impact on the steepness and overall shape of the curve than the height of the maximum. The highest degree of inefficiency can be observed in scenario 9 and scenario 11. They differ in the height of the plateau, while both scenarios start at a higher level with a faster increase. A higher starting antibody concentration combined with a faster increase, corresponding to an earlier maximum, results in potentially less efficient designs, especially if the maximum is misspecified as too early. This observation is intuitively plausible, as starting at a higher level in combination with a very early peak strongly influences the overall structure of the curve. These results further suggest that the design's D-efficiency is at least indirectly influenced by the initial antibody concentration A_0. Particularly, we observed that scenarios with zero ($A_0 = 0$) and lower values show more similarities than scenarios with lower and higher A_0 levels. Furthermore, the deviation of optimal sampling times are differently affected by misspecification in different initial information. While for misspecification in the time of the maximum, every measurement is influenced (cp. Figure 3.3), misspecification in the height of the plateau mostly affects the last measurement (cp. Figure 3.4). This can again be attributed to the corresponding effect of the misspecification in the respective initial information on the

overall structure of the curve. Based on the previously discussed results and the fact that in reality a misspecification of a single initial information, while correctly specifying all others, is an unrealistic assumption, the subsequent step of simultaneous misspecification of two initial information is well justified.

4.2.2. Double Parameter Misspecification

For double misspecification in the initial information the median D-efficiencies across all pairs of misspecification and scenarios are greater equal 0.90 and the Q1-value above 0.8 for all expect for one pair in two scenarios (scenario 9 and scenario 11 for (t_{max}, A_{plat})). We therefore conclude that the framework is robust against double misspecification as well, however, some scenarios and pairs are more robust than others. Generally, we observe two types of grouping. Firstly, scenarios 1-8 compared to scenarios 9-12 in which scenarios 1-8 (starting in zero or at a lower level) show more robust results than scenarios 9-12 (starting at a higher level), irrespectively if A_0 is misspecified or not. Secondly, pairs where A_{plat} is involved compared to pairs without A_{plat}. Pairs in which the height of the plateau is misspecified are more sensitive to misspecification across all scenarios. Not surprisingly, the two initial information which showed the most sensitive behaviour to misspecification in the single misspecification, namely the time of the maximum t_{max} and the height of the plateau A_{plat}, also show the most sensitive results when misspecifying them simultaneously. They show the lowest or close to the lowest Q1 (0.75 (scenario 9) to 0.92 (scenarios 7 and 12)) and median values (0.9 (scenario 1) to 0.96 (scenarios 7 and 12)) across all scenarios. Nevertheless, these values remain within an acceptable range and thus the framework can be considered robust against double misspecification. The pair (t_{plat}, A_{max}) shows the most robust results, with an overall minimal D-efficiency of 0.90, while the pair (t_{max}, t_{plat}) also shows generally robust results, indicating that the time of plateau has less impact on the designs efficiency than the height of the plateau. Somewhat surprisingly, the simultaneous misspecification of the timing and height of the maximum shows robust results for many scenarios. Here scenarios 9 and 11 show the most sensitive results, both of which start at higher level with a faster increase, showing convergence issues and inefficient designs. The inefficiency in the study design arises from the maximum being misspecified as too early (cp. Figure E.16), because the combination of an higher starting value combined with an earlier maximum has the most impact on the overall structure of the curve, therefore the guesses of the parameters and therefore the optimal sampling times. Overall, scenario 7 followed by scenario 5 show the largest numbers of non-converging runs (cp. Figure E.5). Both scenarios start at a lower level and show a faster increase, while differing in the height of the plateau. The lower the starting value is chosen, the smaller the shifting term c becomes, causing numerical issues

in determining the parameters of the beta parameter and subsequently in determining the optimal sampling times. The overall minimal D-efficiency of 0.02 (cp. Figure 3.5) are all observed for scenario 5 and pairs involving A_{plat}, specifically, if converging at initial values of $A_{plat} = 0$. If choosing A_{plat} to be non-zero, the minimal D-efficiencies immediately improve to 0.62 to 0.71, depending on the respective pair of misspecification.
Unsurprisingly, Figures E.7 to E.28 in the Appendix show that the smaller the magnitude of misspecification, the more efficient the resulting designs are. However, they also show that the area of misspecification in which the framework provides efficient designs (the green area in these figures) is quite large, highlighting that the framework is generally robust against double misspecification. Based on the results of the simultaneous misspecification of two initial information in combination with the results of the single misspecification, practical recommendations can be made. As for all scenarios with pre-existing antibodies the first optimal measurement is time 0, we advice for taking a sample, if possible, at (or shortly before) time of exposure. Since the framework shows numerical issues, if A_0 is small but non-zero, due to its effects on the shifting term c, it might be beneficial to choose A_0 accordingly. Furthermore, throughout the framework numerical issues can be observed if $A_{plat} = 0$ is chosen. Therefore, choosing the lower limit of detection/quantification might be more robust and could improve the numerical stability of the framework. However, the results of the Robustness Analysis show a tendency towards non-convergence or inefficiency if the height of the plateau is specified as too high compared to the true value. Therefore, choosing A_{plat} as non-zero but conservatively low might be the best option. Further, most of the convergence issues appeared when an earlier maximum is misspecified as too early, it might therefore be beneficial to set the time of the maximum conservatively late.

4.2.3. Shiny Application

To make the framework easily accessible, it is publicly available at Zenodo.[123] Throughout the implementation of the Shiny application, the focus was on providing an easily applicable tool, where no mathematical background is needed to optimize the optimal sampling schedule of an immunization study. Therefore, a manual is provided (cp. Appendix F). The user interface is structured in 4 panels, while only one of them is used as input. This panel focuses on the easily understandable initial information, while allowing in different tabs to set different options (eg. restrictions on the time-windows). The other three provide a graphical overview of the results of the input, a display of the resulting numerical optimal sampling times, with the option to copy a complete set of used values to the clipboard and a text focusing on reproducibility.

4.3. Limitations

The proposed framework is considered phase I in methodological research, as described by Heinze et al.,[133] and therefore introduces a new idea and provides reasoning why it is valid from a theoretical point of view and provides general proof of work. Consequently, it comes with several limitations which will be discussed briefly. Directions of future research are presented in Section 4.4, however, as suggested by the Robustness Analysis the framework could be used in practical settings to improve the current status quo of study design in immunization studies. Generally, it is common practice to simplify complexities at the stage of study design. These simplifications, on the other hand, allow for potential extensions of the framework.

The framework assumes that the antibody kinetics follow the density of a beta distribution, while other functional forms could also be considered (cp. Section 4.4.1). By the restrictions on the parameters of the beta distribution to be $\alpha > 1$ and $\beta > 1$, it can only be used if the antibody kinetics show an initial increase followed by a decay. If the assumptions on the parameters do not hold, we can not utilize the property of the mode x_{max}. Further, the variability and correlation between measurements are assumed to be known. The variability parameters could be assumed to be unknown or other covariance matrices could be used (cp. Section 4.4.2). This, however, would result in additional analytical and numerical complexity, while requiring additional initial information on these parameters. For now, we saw that through the assumption of known variability parameters, they do not affect the optimal sampling times analytically.

This work focuses on a single schedule for all patients. Optimal designs allowing for multiple schedules, with patients being allocated according to (predefined) ratios, could potentially further improve design efficiency (cp. Section 4.4.3). Nonetheless, given that clinical studies regularly use a single schedule all patients should follow, optimizing this schedule would already represent an improvement of the status quo. Furthermore, the framework is based on mean values of the initial information, thereby neglecting inter-individual variability (cp. Section 4.4.6). While theory on optimal design of nonlinear mixed effects models is available, the practical application remains challenging.[96, 98–100]

The Robustness Analysis also has limitations. The most significant of these is that varying one or two parameters at a time, while fixing the others to be known is a strong simplification of reality. In practice, all parameters are likely to be misspecified simultaneously. Furthermore, the values chosen for the initial information in the different scenarios, might not reflect practical conditions perfectly. It might also be confusing that not all initial information have the same number of potential values. For t_{max} and A_{plat} two values are

chosen to distinguish between slower and faster increase and lower and higher plateaus, respectively. To distinguish between zero and non-zero initial antibody concentrations and to distinguish between different magnitudes, three values for A_0 were selected. The maximum antibody concentration A_{max} selected for the Robustness Analysis is somehow arbitrary, but it can be interpreted as a standardization, allowing the other antibody concentrations being interpreted as relative to the maximum. Nonetheless, typical observations, specifically on the log-scale, align with the selected ranges. The framework also allows studies to be designed on different scales, which can be of a larger magnitude in practice, especially when working on the original scale. However, scaling might have an effect on the numerical behaviour, the magnitude of the FIM and therefore convergence. The selected ranges for misspecification in the initial information were informed by clinical experience, although a greater variability may occur in real-world scenarios. Estimating the mean maximum antibody concentration within a range of $\pm 20\%$ is considered reasonable. Similarly, when designing a study, it is a practically appropriate assumption to specify the timing of the maximum within ± 2 weeks of the true maximum, as well as to assume knowledge of whether patients have pre-existing antibodies. It is also reasonable to specify the time of the plateau within a time-window of 100 days as a model simplification. For the initial antibody concentration and the height of the plateau the same ranges of misspecification as for the maximum concentration were chosen, representing relatively wider ranges of misspecification.

Consequently, the Shiny application comes with limitations, which to a great extent reflect the limitations of the framework or result by limitations in the implementation. Throughout the Robustness Analysis and therefore during the use of the Shiny application, numerical issues were observed resulting either from determining the beta parameters or the optimal sampling times. This can mainly be attributed to two causes: The beta density is infeasible to model the change in antibody concentration over time (e.g. a very early maximum or an almost flat curve) or convergence issues resulting from the optimization algorithms. In the former case, the framework should not be used, while for the latter case, a pragmatic solution is to include a sensitivity analysis. A sensitivity analysis is advised at the stage of study design in any case in order to check the sensitivity of the optimal design and to assess numerical and analytical stability.[10] In a future version of the Shiny application it is planned to include a sensitivity analysis, i.e. varying the initial information slightly, automatically. Future work should also aim to improve numerical convergence by investigating alternative optimization algorithms and working on tolerances and scaling strategies to address issues related to potentially small parameter values. In many cases non-convergence can be observed, when the shifting term c reaches the or-

der 10^{-8}, corresponding to the numerical tolerances in the R-package 'nleqslv'.[112] The current version of the Shiny application might not be flexible enough to cover the broad range of different antibody kinetics scenarios, or the error messages appearing when encountering non-convergent input might discourage the use of the framework for scientists without programming experience.

4.4. Extension Possibilities

The introduced framework allows for further development, increasing flexibility and stability. The following discusses potential extension possibilities, with applicability always marking the main focus to ensure practical relevance.

4.4.1. Other Functional Structures

One of the key properties we are using throughout the framework is that the mode of the beta density can be described by the parameters θ. For each function (if regular enough) describing an initial increase followed by a decay, it should be possible to describe the time of the maximum through the parameters, by building the first derivative with respect to t and setting it to zero, i.e. finding the maximum of the function. Additionally it should also be possible to use A_0 as the value at time $t = 0$ and A_{max} as the value at t_{max}. In the described framework, we additionally use the height of the plateau A_{plat} at time t_{plat} to provide enough information to uniquely determine the parameters of the beta distribution. This provides the necessary estimates to work with locally optimal designs. Specifically, we have chosen the beta density due to it's desired but flexible functional form relying only on few parameters. If we increase the model complexity, i.e. the number of parameters, we would automatically increase the complexity of study design. Furthermore, we might need to provide additional initial information, for which careful consideration of selection and interpretation is essential in order to avoid losing the direct practical connection and ending up with parameters that are difficult to interpret in complex models.

4.4.2. Other Covariance Matrix or Unknown Parameters

Throughout the framework, strong assumptions on the variance-covariance matrix were made. Primarily, assuming its structure and the components to be known. While it is common practice to simplify complexities at the stage of study design, some adaptation could be made. The current form of the variance-covariance matrix is described in Section 2.4.1. Firstly, the structure of the matrix S could allow for different variability for each measurement, i.e. $S = diag(\sigma_1, \ldots, \sigma_n)$. Secondly, the overall structure of the

variance-covariance matrix could be a fully unstructured one or be assumed to show compound symmetry, i.e. equal correlation between measurements. The choice of an AR(1) matrix is reasonable as it mimics the fact that two closer measurements are stronger correlated than two more distant ones. However, the described AR(1)-matrix assumes that the distance between two measurements is equidistant and a potential extension would be to include the time difference between two measurements in the matrix, as for example given by Atkinson et al. the correlation between two measurements taken at time t_1 and t_2 could be described by [134]:

$$c(t_1, t_2) = exp(-\tau|t_2 - t_1|).$$

The closer the two measurements are taken, the stronger their correlation, while it is exactly 1, if two measurements are taken at the same time and therefore provide no additional information. Further, two measurements that are far apart show a lower correlation, while τ determines the "strength" of correlation. However, this would immediately imply that the covariance matrix is part of the optimization process and can not be interpreted as a constant any more. Additionally, as stated by Atkinson et al., the difficulty is to build an efficient algorithm allowing to find optimal designs to account for the non trivial effect of correlation between measurements.[134]
Furthermore, if the variability parameters are part of the unknown parameters or the variance-covariance matrix depends on those, the formula for the FIM changes as shown in Section 2.1.2. As a consequence, the number of necessary measurement changes and the implemented optimization algorithm needs adaptations.

4.4.3. Multiple Different Schedules

Throughout the framework the weights in the design are fixed to $p_1 = 1$ and $p_i = 0 \quad \forall i > 1$, which corresponds to optimizing one single design point, i.e. one single schedule. It might be beneficial and of practical interest to allow for more than one schedule, while in principle there are three possible extension options. Firstly, the number of schedules and their weights can be preset (n schedules and the weights p_i fixed for $i = 1, \ldots, n$), for example $p_1 = p_2 = 0.5$ if the goal is a 1:1 split between two schedules. Secondly, a specific number of possible schedules could be allowed while the weights are not specified (n fixed, while the p_i are not fixed for $i = 1, \ldots, n$) or as a third option the number of samples and their weights are not specified at all (n and p_i not fixed). These three extension possibilities are based on the additivity of the FIM, namely $M(\xi) = \sum_{i=1}^{n} p_i m(x_i)$, where $m(x_i)$ is the FIM of a single observational unit and $\xi = \{x_i, p_i\}_{i=1}^{n}$ the corresponding design. These changes, however, would directly affect the complexity of the optimization problem, the implemented method

(see Section 2.3.5) might not be efficient enough for this extension. Further work is needed to explore properties of the implemented method and compare it with other algorithms, while the open question remains of how the algorithm(s) can be implemented in such a flexible way that they can handle fixed and unfixed number of schedules n and their fixed or unfixed weights p_i. Additionally, some practical considerations require more careful analysis, exemplary if the user wants to fix the number of schedules n to be larger than the optimal solution would yield in the unrestricted case.

4.4.4. Include Covariates

In some studies, the primary interest might not lie in modelling antibody kinetics, but rather in assessing the effect of covariates on antibody levels over time. These covariates may include demographic factors such as sex (binary) or age (continuous), as well as clinical factors that affect immune function, such as underlying immunosuppression due to chronic disease or receipt of immunosuppressive treatment (e.g. systemic corticosteroids), either prior to or during course of infection. Optimal design theory, in principle, optimizes the allocation of covariates in a regression model and it is therefore theoretically relatively easy to include the two levels of the covariate in the design space, simply by adding an extra dimension in addition to the sampling times. However, the initial information (t_{max}, A_{max}, etc.) might differ between the groups, which results in different parameters of the beta distribution and therefore in different sampling schedules, not focusing on detecting the difference between the two curves. Additionally, more rigorous considerations of how to incorporate the additional covariate(s) into modelling, estimation and design optimization are needed.

4.4.5. Include Uncertainty in Initial Information

Typically in study design, guesses of the unknown parameters come with uncertainty. The same is true for our simplified initial information. If it would be perfectly clear, where the time and height of the maximum and others are, we would not need to conduct a study. Despite working with locally optimal designs, i.e. best available guesses, improvement in the study design could be reached by allowing for uncertainty in the provided initial information. Typically one would not specify a precise value but a plausible range, e.g., the maximum is somewhere between day 28 and 42. Reasonable solutions could be to either performing sensitivity analysis, i.e. using all values and comparing the resulting optimal sampling times or to use the average criterion/Bayesian criterion to integrate over the "prior" belief of the initial information. The difficulties arising from this approach is that we can not directly integrate over the parameters as we translate the initial infor-

mation to the parameters that are actually used in study design. Therefore one needs to specify a distributional assumption of the initial information, while the distribution of the parameters in the optimality criteria might be different and a priori not clear.

4.4.6. Include Inter-Individual Variability

Antibody concentrations over time typically present substantial inter-individual variability and nonlinear mixed effects models are a feasible approach for their analysis. However, several challenges arise when designing a study based on nonlinear mixed effects models. Firstly, the FIM generally lacks a closed-form solution, since there is no analytical expression of the log-likelihood, while typically approximations such as linearization are used.[96, 98–100] Secondly, additional information would be required such as assumptions on the variability of the random effect or if there are multiple random effects, additionally their structure.[98] Major difficulties arise in incorporating the inter-individual variability into the initial information, since the random effects are (potentially) part of the resulting parameters used in the modelling function. Despite the relevance of nonlinear mixed effects models in the analysis of longitudinal nonlinear data, for study design priority should lie on the incorporation of uncertainty in the initial information (cp. Section 4.4.5).

4.4.7. Restrict Number of Samples per Patient

If the number of samples per participant is restricted to be smaller than the number of parameters, the optimization problem can not be solved in its current form, since the corresponding information matrix is rank-deficient. However, allowing for multiple schedules (cp. Section 4.4.3) can overcome this limitation. This is based on the fact that the sum of matrices with determinant zero is not necessarily zero and the optimization problem can therefore be solved. Specifically, while

$$\det(A) = 0 \implies \det(n * A) = 0,$$

it does not hold in general that the determinant of the sum of zero determinant matrices is zero, i.e.

$$\det(A_j) = 0 \ \forall j \not\implies \det(\sum_{j=1}^{S} A_j) = 0.$$

4.4.8. Add a Sample

Including additional measurements in the study design might be desired for several reasons. It can provide the flexibility to fit a more complex model at the analysis stage (cp. Section 4.4.10) or might account for practical issues, such as missed measurements or

drop-outs. Moreover, additional samples could compensate for uncertainty in the initial information and the functional form, allowing for a more robust estimation in cases of model misspecification. Furthermore, adding additional measurements might simply have pragmatic reasons, such as multiple routine visits. In the current framework, an additional sample would be a replicate of an already existing sampling time.[111] One could increase the number of measurements by introducing additional parameter to be estimated such as σ and ρ and therefore increase the dimensionality of the FIM (cp. Section 2.1.2). From an analytical perspective, the addition of a sample to a D-optimal design without increasing the number of parameters is not straightforward. A very pragmatic solution would be to optimize the design based on the initial information and simply add samples in addition to the optimized ones. Further, one could optimize two different schedules, e.g. by setting different restrictions on the time-windows, and combine those.

4.4.9. Double (Multiple) Exposure

In many settings patients face multiple exposure, for example, SARS-CoV-2 vaccination consisting of multiple doses or pertussis immunization starting with maternal vaccination and continuing through early life with multiple exposures. Therefore it might be reasonable to incorporate this possibility into the framework. To illustrate a potential solution, consider the case of double exposure, while the idea should be extendable to multiple exposure. In the case of double exposure, the second exposure (E_2) occurs either before the first maximum ($t_{E2} \leq t_{max1}$), in which case it is possible to simplify to a single exposure since the antibody kinetics are still in the increasing phase, or after the first maximum ($t_{E2} > t_{max1}$), which implies that there was an increase followed by some decay. Under the assumption, that $t_{max1} < t_{E2} < t_{plat}$ the following information would be necessary to design a study: A_0, t_{max1}, A_{max1}, the time of the second exposure t_{E2}, the antibody concentration A_{E2}, the time of the second maximum t_{max2}, the height of the second maximum A_{max2} and the plateau values t_{plat} and A_{plat}.

This would result in the following system of equations

$$
\begin{aligned}
f\left(\frac{c_1}{t_{scale1}}, \alpha_1, \beta_1\right) &= A_0 \\
f\left(\frac{t_{max1} + c_1}{t_{scale1}}, \alpha_1, \beta_1\right) &= A_{max1} \\
f\left(\frac{t_{E2} + c_1}{t_{scale1}}, \alpha_1, \beta_1\right) &= A_{E2} \\
\frac{\alpha_1 - 1}{\alpha_1 + \beta_1 - 2} &= \frac{t_{max1} + c_1}{t_{scale1}} \\
f\left(\frac{c_2}{t_{scale2}}, \alpha_2, \beta_2\right) &= A_{E2} \\
f\left(\frac{t_{max2} - t_{E2} + c_2}{t_{scale2}}, \alpha_2, \beta_2\right) &= A_{max2} \\
f\left(\frac{t_{plat} - t_{E2} + c_2}{t_{scale2}}, \alpha_2, \beta_2\right) &= A_{plat2} \\
\frac{\alpha_2 - 1}{\alpha_2 + \beta_2 - 2} &= \frac{t_{max2} + c_2 - t_{E2}}{t_{scale2}},
\end{aligned}
$$

where $\alpha_1, \beta_1, t_{scale1}$ and c_1 denote the parameters of the beta distribution modelling the first maximum, while $\alpha_2, \beta_2, t_{scale2}$ and c_2 denote the parameters of the beta distribution modelling the second maximum.
Initial implementation attempts resulted in numerical issues, such as non-convergence or mismatched curves. Future work should carefully investigate these aspects and also pay attention to questions of parameter identifiability. However, based on the described idea, two closely related extension options are available. First, instead of relying on the plateau one could simply specify a second "known" value within the framework and use these initial information to obtain the parameters of the beta distribution. Second, studies with double exposure could be designed such that, rather than a simultaneous design, each peak is modelled separately with the starting value of the second curve being equal to the endpoint of the first. A potential drawback of this approach is the potential occurrence of a duplicated sample at the cut point, which is impractical.

4.4.10. What if the Analysis Model differs from the Design Model?

Optimal designs based on the FIM are inherently model dependent, which means that the optimal design is based on the assumed probability function, especially the mean function and its parameters. If a different model is used to analyse the data, involving a different functional form and a different set of parameters, the corresponding optimal design might differ as well. An illustrative example is the case where the study design is using a

nonlinear model, but the analysis is performed using a nonlinear mixed effects model. Furthermore, this issue might become more relevant if the underlying functional form of the model also changes. Since the number of measurements per participant depends on the number of parameters used in the model, discrepancies may arise between the design and analysis models, resulting in an imbalance between the number of measurements and parameters. A more complex analysis model might require additional parameters and the number of available measurements are then not sufficient to ensure a reliable, yet optimal, estimation.

Further methodological considerations and theoretical work are needed to determine the practical relevance of these issues for study design. Simulation studies could be employed to evaluate design efficiency and compare scenarios in which: (i) the study is planned using a model whose functional form aligns with the analysis model, (ii) a completely different model is used, (iii) time-points are allocated completely random or (iv) commonly applied standard designs, e.g. taking a sample at the immunization event, after 2 weeks, after 2 month and after 6 month. Additionally, clinical trial simulations could provide insight into the benefits of variability and bias in parameter estimates between an optimal design, a misspecified optimal design, and a non-optimized design.

4.4.11. Numerical Stability and Sensitivity Analysis

Throughout the Robustness Analysis, we repeatedly encountered numerical issues. One common source of numerical issues is that the parameters take very small values relative to the numerical tolerances, typically the shifting term c, leading to instability in the optimization. Typically, these convergence issues occur only in small regions and are highly dependent on the specific choice of the initial information. Here even minor adjustments to the initial information resolve the numerical issues. In other cases, the non-convergence regions are larger and even a modification of the initial information would result in numerical issues. In both cases sensitivity analysis, i.e. optimizing the design for a range of initial information, would provide additional insight. In the first case, it solves the problem of non-convergence and provides a solution, while in the second case it highlights systematic issues. More generally, sensitivity analysis should be performed in the context of optimal design anyway and is also a pragmatic solution to account for the uncertainty in the initial information (cp. Section 4.4.5).[10]

Most optimization algorithms in optimal design theory are based on sensitivity functions (cp. Section 2.2; not to be confused with sensitivity analysis), while our current implementation does rely on other optimization algorithms. Exploring different optimization algorithms or adapting sensitivity function based methods may improve the numerical stability of our framework. Particularly, the use of sensitivity function based methods might

become useful when incorporating some of the described extension, especially multiple schedule with unfixed weights and numbers of samples, where our current implementation might become too slow, unreliable and inefficient. In its current form, however, sensitivity analysis and rethinking parameter scaling, might be the easiest way of improvement.
An alternative strategy is to rely on simulation-based approaches and to disregard the analytical solution altogether. While this is a feasible approach it somehow undermines the ideas behind optimal design theory.

4.5. Conclusion

The main goal of this dissertation was to introduce a framework that allows the incorporation of optimal design theory in immunization studies, where modelling the antibody kinetics is the main objective, in an easily understandable and accessible way. Throughout our systematic review, we saw that optimal design theory is not used in immunization studies, which we attributed to mathematical complexities arising from the combination of modelling antibody kinetics and optimal design theory and lack of awareness/accessibility (cp. Section 1.2.2).[39] To reduce the mathematical complexities (from the user's perspective) the framework translates easily understandable initial information to mathematical parameters and utilizes these parameters to optimize a sampling schedule. By that the framework allows studies to be designed by users with a limited mathematical background, keeping the mathematical complexities at a deeper, hidden level. To increase the accessibility of the framework, a publicly available Shiny application can be used, thereby reducing the programmatic burden.[123] The Shiny application is easily accessible, allows the users to vary their input and explore different input parameters, while keeping a high standard of reproducibility. Additionally, the Shiny application is implemented in a way that allows for potential extensions. The major advantage of the framework is that it does not require mathematical or coding skills, allowing users to tackle the complex problem of optimizing sampling schedules from an easily accessible, less complex, entry point.
The Robustness Analysis showed that the framework itself is robust against misspecification, highlighting that the translation of easily understandable input information to mathematical parameters is a valid approach. Specifically, the framework showed very robust results, when the misspecification in the initial information was of smaller extent. Through the Robustness Analysis we saw that a baseline sample is typically recommended to quantify the value of the initial antibody concentration and that in most cases an additional sample is recommended close to the maximum.
Throughout the methodological development of the framework the focus was on using sim-

pler concepts (e.g., D-optimality, single exposure, not using mixed models, etc.) to bridge the gap between theory and practice. Although a substantial amount of theory on optimal design exists, it is not used in the practical design of immunization studies. One can use these developed methodological concepts to further refine the introduced framework; however, practical applicability should remain the central interest if the goal is real-world adoption. This dissertation has demonstrated this trade-off at several instances. Future (research) efforts should aim at improving the flexibility of the framework (e.g., multiple exposure, adding or removing samples, include uncertainty, etc.), improving numerical stability and raising awareness for the benefits and efficiency gains in study design when considering optimal sampling schedules in immunization studies.

In summary, the proposed framework shows robustness against misspecifications in the initial information, particularly when these misspecifications are minor. While certain limitations remain, it offers a good starting point for the application of optimal design theory in immunization studies. A key strength of the framework is the reliance on interpretable input information, enhancing the accessibility for healthcare professionals. Accordingly, it marks a valuable step in the study design of immunization studies, specifically in optimizing sampling schedules, a field that has received little attention to date and in which systematic approaches remain underutilized.

Bibliography

[1] Vos T, Barber RM, Bell B, Bertozzi-Villa A, Biryukov S, Bolliger I, et al. Global, regional, and national incidence, prevalence, and years lived with disability for 301 acute and chronic diseases and injuries in 188 countries, 1990–2013: a systematic analysis for the Global Burden of Disease Study 2013. The Lancet. 2015;386(9995): 743–800. DOI: 10.1016/S0140-6736(15)60692-4.

[2] Cassini A, Colzani E, Pini A, Mangen M.-JJ, Plass D, McDonald SA, et al. Impact of infectious diseases on population health using incidence-based disability-adjusted life years (DALYs): results from the Burden of Communicable Diseases in Europe study, European Union and European Economic Area countries, 2009 to 2013. Euro Surveillance. 2018;23(16). DOI: 10.2807/1560-7917.ES.2018.23.16.17-00454.

[3] Naghavi M, Mestrovic T, Gray A, Gershberg Hayoon A, Swetschinski LR, Robles Aguilar G, et al. Global burden associated with 85 pathogens in 2019: a systematic analysis for the Global Burden of Disease Study 2019. The Lancet Infectious Diseases. 2024;24(8): 868–895. DOI: 10.1016/S1473-3099(24)00158-0.

[4] World Health Organization. World health statistics 2025: monitoring health for the SDGs, Sustainable Development Goals. Geneva: World Health Organization, 2025. ISBN: 9789240110496.

[5] Abbas AK, Lichtman AH, Pillai S. Cellular and Molecular Immunology. 8th ed. Philadelphia, PA: Elsevier Saunders, 2014. ISBN: 978-0-323-22275-4.

[6] Garcia-Fogeda I, Besbassi H, Larivière Y, Ogunjimi B, Abrams S, Hens N. Within-host modeling to measure dynamics of antibody responses after natural infection or vaccination: A systematic review. Vaccine. 2023;41(25): 3701–3709. DOI: 10.1016/j.vaccine.2023.04.030.

[7] Theofanidis D, Fountouki A. Limitations and delimitations in research process. Perioperative nursing. 2018;7: 155–162. DOI: 10.5281/zenodo.2552022.

[8] Puhan MA, Akl EA, Bryant D, Xie F, Apolone G, Riet G. Discussing study limitations in reports of biomedical studies- the need for more transparency. Health and quality of life outcomes. 2012;10. DOI: 10.1186/1477-7525-10-23.

[9] Atkinson AC, Donev AN. Optimum Experimental Designs. Vol. 8. Oxford science publications. Oxford: Clarendon, 1992. ISBN: 0198522541.

[10] Fedorov VV, Leonov SL. Optimal Design for Nonlinear Response Models. 1st ed. Boca Raton: CRC Press, 2013. ISBN: 9780429103988. DOI: 10.1201/b15054.

[11] Duan N, Bhaumik DK, Palinkas LA, Hoagwood K. Optimal Design and Purposeful Sampling: Complementary Methodologies for Implementation Research. Administration and policy in mental health. 2015;42(5): 524–532. DOI: 10.1007/s10488-014-0596-7.

[12] Balelli I, Pasin C, Prague M, Crauste F, van Effelterre T, Bockstal V, et al. A model for establishment, maintenance and reactivation of the immune response after vaccination against Ebola virus. Journal of theoretical biology. 2020;495. DOI: 10.1016/j.jtbi.2020.110254.

[13] Davis CL, Wahid R, Toapanta FR, Simon JK, Sztein MB. A clinically parameterized mathematical model of Shigella immunity to inform vaccine design. PLOS ONE. 2018;13(1): e0189571. DOI: 10.1371/journal.pone.0189571.

[14] Rundell A, DeCarlo R, HogenEsch H, Doerschuk P. The Humoral Immune Response to Haemophilus influenzae Type b: a Mathematical Model Based on T-zone and Germinal Center B-cell Dynamics. Journal of theoretical biology. 1998;194(3): 341–381. DOI: 10.1006/jtbi.1998.0751.

[15] Morris TP, White IR, Crowther MJ. Using simulation studies to evaluate statistical methods. Statistics in medicine. 2019;38(11): 2074–2102. DOI: 10.1002/sim.8086.

[16] Subramanian N, Torabi-Parizi P, Gottschalk RA, Germain RN, Dutta B. Network representations of immune system complexity. Wiley interdisciplinary reviews. Systems biology and medicine. 2015;7(1): 13–38. DOI: 10.1002/wsbm.1288.

[17] Varadé J, Magadán S, González-Fernández Á. Human immunology and immunotherapy: main achievements and challenges. Cellular & molecular immunology. 2021;18(4): 805–828. DOI: 10.1038/s41423-020-00530-6.

[18] Olafsdottir TA, Thorleifsson G, Lopez de Lapuente Portilla A, Jonsson S, Stefansdottir L, Niroula A, et al. Sequence variants influencing the regulation of serum IgG subclass levels. Nature Communications. 2024;15(1): 8054. DOI: 10.1038/s41467-024-52470-8.

[19] Ciobanu AM, Dumitru AE, Gica N, Botezatu R, Peltecu G, Panaitescu AM. Benefits and Risks of IgG Transplacental Transfer. Diagnostics. 2020;10. DOI: 10.3390/diagnostics10080583.

[20] Tian L, Elsheikh EB, Patrone PN, Kearsley AJ, Gaigalas AK, Inwood S, et al. Towards Quantitative and Standardized Serological and Neutralization Assays for COVID-19. International Journal of Molecular Sciences. 2021;22(5). DOI: 10.3390/ijms22052723.

[21] Liu K.-T, Han Y.-J, Wu G.-H, Huang K.-YA, Huang P.-N. Overview of Neutralization Assays and International Standard for Detecting SARS-CoV-2 Neutralizing Antibody. Viruses. 2022;14(7). DOI: 10.3390/v14071560.

[22] Dutt TS, Spencer JS, Karger BR, Fox A, Obregon-Henao A, Podell BK, et al. ELISA-R: an R-based method for robust ELISA data analysis. Frontiers in Immunology. 2024;15.

[23] Chentoufi HA, Galipeau Y, Arnold C, Dewar-Darch D, Dyks A, Cooper C, et al. Comparative performance of serum and plasma samples in SARS-CoV-2 serology and neutralization assays. Journal of Virological Methods. 2025;337: 115186. DOI: 10.1016/j.jviromet.2025.115186.

[24] Sette A, Crotty S. Adaptive immunity to SARS-CoV-2 and COVID-19. Cell. 2021;184(4): 861–880. DOI: 10.1016/j.cell.2021.01.007.

[25] Ameratunga R, Woon S.-T, Jordan A, Longhurst H, Leung E, Steele R, et al. Perspective: diagnostic laboratories should urgently develop T cell assays for SARS-CoV-2 infection. Expert Review of Clinical Immunology. 2021;17(5): 421–430. DOI: 10.1080/1744666X.2021.1905525.

[26] Kirchenbaum GA, Pawelec G, Lehmann PV. The Importance of Monitoring Antigen-Specific Memory B Cells, and How ImmunoSpot Assays Are Suitable for This Task. Cells. 2025;14(3). DOI: 10.3390/cells14030223.

[27] Bowyer G, Rampling T, Powlson J, Morter R, Wright D, Hill AVS, et al. Activation-induced Markers Detect Vaccine-Specific $CD4^+$ T Cell Responses Not Measured by Assays Conventionally Used in Clinical Trials. Vaccines. 2018;6(3). DOI: 10.3390/vaccines6030050.

[28] Binayke A, Zaheer A, Vishwakarma S, Singh S, Sharma P, Chandwaskar R, et al. A quest for universal anti-SARS-CoV-2 T cell assay: systematic review, meta-analysis, and experimental validation. NPJ vaccines. 2024;9(1): 3. DOI: 10.1038/s41541-023-00794-9.

[29] Anderson RM, May RM. Infectious Diseases of Humans: Dynamics and Control. Oxford: Oxford University Press, 1991. ISBN: 9780198545996. DOI: 10.1093/oso/9780198545996.001.0001.

[30] Keeling MJ, Rohani P. Modeling Infectious Diseases in Humans and Animals. Princeton University Press, 2008. ISBN: 9780691116174.

[31] Hens N, Shkedy Z, Aerts M, Faes C, van Damme P, Beutels P. Modeling Infectious Disease Parameters Based on Serological and Social Contact Data. New York, NY: Springer, 2012. ISBN: 978-1-4614-4071-0. DOI: 10.1007/978-1-4614-4072-7.

[32] Bernoulli D, Blower S. An attempt at a new analysis of the mortality caused by smallpox and of the advantages of inoculation to prevent it. 1766. Reviews in medical virology. 2004;14(5): 275–288. DOI: 10.1002/rmv.443.

[33] Kermack WO, McKendrick A. G. A contribution to the mathematical theory of epidemics. Proceedings of the Royal Society of London. Series A. 1927;115(772): 700–721. DOI: 10.1098/rspa.1927.0118.

[34] CDC: Centers for Disease Control and Prevention. Technical Explainer: Infectious Disease Transmission Models [Internet]. [cited 11th December 2025]. 2025. URL: https://www.cdc.gov/cfa-modeling-and-forecasting/about/explainer-transmission-models.html.

[35] Schlickeiser R, Kröger M. Mathematics of Epidemics: On the General Solution of SIRVD, SIRV, SIRD, and SIR Compartment Models. Mathematics. 2024;12(7). DOI: 10.3390/math12070941.

[36] Ariyaratne P, Ramasinghe LP, Ayyash JS, Kelley TM, Plant-Collins TA, Shinkle LW, et al. Application and significance of SIRVB model in analyzing COVID-19 dynamics. Scientific Reports. 2025;15(1): 8526. DOI: 10.1038/s41598-025-90260-4.

[37] Andersson H, Britton T. Stochastic Epidemic Models and Their Statistical Analysis. New York, NY: Springer, 2000. ISBN: 978-0-387-95050-1. DOI: 10.1007/978-1-4612-1158-7.

[38] Tracy M, Cerdá M, Keyes KM. Agent-Based Modeling in Public Health: Current Applications and Future Directions. Annual Review of Public Health. 2018;39: 77–94. DOI: 10.1146/annurev-publhealth-040617-014317.

[39] Embacher S, Herzog SA. Mathematical and statistical models describing antibody kinetics: a systematic review: CRD42022378665 [Internet]. [cited 11th December 2025]. 2022. URL: https://www.crd.york.ac.uk/prospero/display_record.php?ID=CRD42022378665.

[40] Embacher S, Maertens K, Herzog SA. Half-life Estimation of Pertussis-Specific Maternal Antibodies in (Pre)Term Infants After In-Pregnancy Tetanus, Diphtheria, Acellular Pertussis Vaccination. The Journal of Infectious Diseases. 2023;228(11): 1640–1648. DOI: 10.1093/infdis/jiad212.

[41] Leuridan E, Hens N, Hutse V, Aerts M, van Damme P. Kinetics of maternal antibodies against rubella and varicella in infants. Vaccine. 2011;29(11): 2222–2226. DOI: 10.1016/j.vaccine.2010.06.004.

[42] Goeyvaerts N, Leuridan E, Faes C, van Damme P, Hens N. Multi-disease analysis of maternal antibody decay using non-linear mixed models accounting for censoring. Statistics in Medicine. 2015;34(20): 2858–2871. DOI: 10.1002/sim.6518.

[43] Fraser C, Tomassini JE, Xi L, Golm G, Watson M, Giuliano AR, et al. Modeling the long-term antibody response of a human papillomavirus (HPV) virus-like particle (VLP) type 16 prophylactic vaccine. Vaccine. 2007;25(21): 4324–4333. DOI: 10.1016/j.vaccine.2007.02.069.

[44] Jõgi P, Soeorg H, Oona M, Kaart T, Toompere K, Maskina T, et al. Dynamics of pertussis toxin IgG after symptomatic pertussis in children and adults. Vaccine. 2020;38(16): 3196–3200. DOI: 10.1016/j.vaccine.2020.02.080.

[45] Martin JC, Petrecz ML, Stek JE, Simon JK, Goveia MG, Klopfer SO. Using the power law model to predict the long-term persistence and duration of detectable hepatitis A antibody after receipt of hepatitis A vaccine (VAQTA™). Vaccine. 2021;39(20): 2764–2771. DOI: 10.1016/j.vaccine.2021.03.052.

[46] David M.-P, van Herck K, Hardt K, Tibaldi F, Dubin G, Descamps D, et al. Long-term persistence of anti-HPV-16 and -18 antibodies induced by vaccination with the AS04-adjuvanted cervical cancer vaccine: Modeling of sustained antibody responses. Gynecologic Oncology. 2009;115(3). DOI: 10.1016/j.ygyno.2009.01.011.

[47] Grandjean L, Saso A, Torres Ortiz A, Lam T, Hatcher J, Thistlethwayte R, et al. Long-Term Persistence of Spike Protein Antibody and Predictive Modeling of Antibody Dynamics After Infection With Severe Acute Respiratory Syndrome Coronavirus 2. Clinical Infectious Diseases. 2022;74(7): 1220–1229. DOI: 10.1093/cid/ciab607.

[48] van Ravenhorst MB, Marinovic AB, van der Klis FR, van Rooijen DM, van Maurik M, Stoof SP, et al. Long-term persistence of protective antibodies in Dutch adolescents following a meningococcal serogroup C tetanus booster vaccination. Vaccine. 2016;34(50): 6309–6315. DOI: 10.1016/j.vaccine.2016.10.049.

[49] van der Lee S, van Rooijen DM, Zeeuw-Brouwer M.-L, Bogaard MJM, van Gageldonk PGM, Marinovic AB, et al. Robust Humoral and Cellular Immune Responses to Pertussis in Adults After a First Acellular Booster Vaccination. Frontiers in Immunology. 2018.

[50] Graaf WF, Kretzschmar M, Teunis P, Diekmann O. A two-phase within-host model for immune response and its application to serological profiles of pertussis. Epidemics. 2014;9: 1–7. DOI: 10.1016/j.epidem.2014.08.002.

[51] Scheltema NM, Kavelaars XM, Thorburn K, Hennus MP, van Woensel JB, van der Ent CK, et al. Potential impact of maternal vaccination on life-threatening respiratory syncytial virus infection during infancy. Vaccine. 2018;36(31): 4693–4700. DOI: 10.1016/j.vaccine.2018.06.021.

[52] Tran TMP, Maertens K, Hoang HTT, van Damme P, Leuridan E, Hens N. Elucidating the difference in the kinetics of antibody titres of infants in Belgium and Vietnam. Vaccine. 2020;38(45): 7079–7086. DOI: 10.1016/j.vaccine.2020.09.003.

[53] Aiemjoy K, Seidman JC, Saha S, Munira SJ, Islam Sajib MS, Sium SMA, et al. Estimating typhoid incidence from community-based serosurveys: a multicohort study. The Lancet Microbe. 2022;3(8): e578–e587. DOI: 10.1016/S2666-5247(22)00114-8.

[54] Le D, Miller JD, Ganusov VV. Mathematical modeling provides kinetic details of the human immune response to vaccination. Frontiers in Cellular and Infection Microbiology. 2015;4.

[55] Besbassi H, Garcia-Fogeda I, Quinlivan M, Breuer J, Abrams S, Hens N, et al. Modeling antibody dynamics following herpes zoster indicates that higher varicella-zoster virus viremia generates more VZV-specific antibodies. Frontiers in Immunology. 2023;14.

[56] Andraud M, Lejeune O, Musoro JZ, Ogunjimi B, Beutels P, Hens N. Living on Three Time Scales: The Dynamics of Plasma Cell and Antibody Populations Illustrated for Hepatitis A Virus. PLOS Computational Biology. 2012;8(3): e1002418. DOI: 10.1371/journal.pcbi.1002418.

[57] White MT, Griffin JT, Akpogheneta O, Conway DJ, Koram KA, Riley EM, et al. Dynamics of the antibody response to Plasmodium falciparum infection in African children. The Journal of Infectious Diseases. 2014;210(7): 1115–1122. DOI: 10.1093/infdis/jiu219.

[58] Pasin C, Balelli I, van Effelterre T, Bockstal V, Solforosi L, Prague M, et al. Dynamics of the Humoral Immune Response to a Prime-Boost Ebola Vaccine: Quantification and Sources of Variation. Journal of Virology. 2019;93(18): 10.1128/jvi.00579–19. DOI: 10.1128/jvi.00579-19.

[59] Poehler E, Gibson L, Lustig A, Moreland NJ, McGregor R, James A. Estimating decay curves of neutralizing antibodies to SARS-CoV-2 infection. Mathematical Medicine and Biology. 2022;39(4): 368–381. DOI: 10.1093/imammb/dqac008.

[60] Xu Z, Zhang H, Yang D, Wei D, Demongeot J, Zeng Q. The Mathematical Modeling of the Host–Virus Interaction in Dengue Virus Infection: A Quantitative Study. Viruses. 2024;16(2). DOI: 10.3390/v16020216.

[61] Bonin CRB, Fernandes GC, Menezes Martins R, Camacho LAB, Teixeira-Carvalho A, Da Mota LMH, et al. Validation of a yellow fever vaccine model using data from primary vaccination in children and adults, re-vaccination and dose-response in adults and studies with immunocompromised individuals. BMC Bioinformatics. 2020;21(17). DOI: 10.1186/s12859-020-03845-3.

[62] Blanco-Rodríguez R, Ordoñez-Jiménez F, Almocera AES, Chinney-Herrera G, Hernandez-Vargas E. Topological data analysis of antibody dynamics of severe and non-severe patients with COVID-19. Mathematical Biosciences. 2023;361: 109011. DOI: 10.1016/j.mbs.2023.109011.

[63] Moyles IR, Korosec CS, Heffernan JM. Determination of significant immunological timescales from mRNA-LNP-based vaccines in humans. Journal of Mathematical Biology. 2023;86. DOI: 10.1007/s00285-023-01919-3.

[64] Schenk DB, Koller M, Ness DK, Griffith SG, Grundman M, Zago W, et al. First-in-human assessment of PRX002, an anti-alpha-synuclein monoclonal antibody, in healthy volunteers. Movement Disorders. 2017;32(2): 211–218. DOI: 10.1002/mds.26878.

[65] Almholt K, Wang J, Pass J, Røder G, Padkjær SB, Hebsgaard JB, et al. Identification and preclinical development of an anti-proteolytic uPA antibody for rheumatoid arthritis. Journal of Molecular Medicine. 2020;98(4): 585–593. DOI: 10.1007/s00109-020-01889-9.

[66] Gordon O, Brosnan MK, Yoon S, Jung D, Littlefield K, Ganesan A, et al. Pharmacokinetics of high-titer anti–SARS-CoV-2 human convalescent plasma in high-risk children. JCI Insight. 2022;7(2). DOI: 10.1172/jci.insight.151518.

[67] Stolfi P, Castiglione F, Mastrostefano E, Di Biase I, Di Biase S, Palmieri G, et al. In-silico evaluation of adenoviral COVID-19 vaccination protocols: Assessment of immunological memory up to 6 months after the third dose. Frontiers in Immunology. 2022;13. DOI: 10.3389/fimmu.2022.998262.

[68] Dogra P, Schiavone C, Wang Z, Ruiz-Ramírez J, Caserta S, Staquicini DI, et al. A modeling-based approach to optimize COVID-19 vaccine dosing schedules for improved protection. JCI Insight. 2023;8(13). DOI: 10.1172/jci.insight.169860.

[69] Wessel RE, Dolatshahi S. Quantitative mechanistic model reveals key determinants of placental IgG transfer and informs prenatal immunization strategies. PLOS Computational Biology. 2023;19(11): e1011109. DOI: 10.1371/journal.pcbi.1011109.

[70] Walsh S. "Overview of Optimal Experimental Design and a Survey of its Expanse in Application to Agricultural Studies". *Conference on Applied Statistics in Agriculture and Natural Resources*. 2022. DOI: 10.26077/fc50-a735.

[71] Wynn HP. Jack Kiefer's Contributions to Experimental Design. The Annals of Statistics. 1984;12(2). DOI: 10.1214/aos/1176346496.

[72] Smith K. On the Standard Deviations of Adjusted and Interpolated Values of an Observed Polynomial Function and its Constants and the Guidance they give Towards a Proper Choice of the Distribution of Observations. Biometrika. 1918;12. DOI: 10.2307/2331929.

[73] Fisher R. The Design of Experiments. 9th ed. New York: Hafner Press, 1974. ISBN: 0-02-844690-9.

[74] Wald A. On the Efficient Design of Statistical Investigations. The Annals of Mathematical Statistics. 1943;14(2): 134–140. DOI: 10.1214/aoms/1177731454.

[75] Elfving G. Optimum Allocation in Linear Regression Theory. The Annals of Mathematical Statistics. 1952;23(2): 255–262. DOI: 10.1214/aoms/1177729442.

[76] Kiefer J, Wolfowitz J. The Equivalence of Two Extremum Problems. Canadian Journal of Mathematics. 1960;12: 363–366. DOI: 10.4153/CJM-1960-030-4.

[77] Kiefer J. General Equivalence Theory for Optimum Designs (Approximate Theory). The Annals of Statistics. 1974;2(5): 849–879. DOI: 10.1214/aos/1176342810.

[78] Box G, Lucas HL. Design of Experiments in Non-Linear Situations. Biometrika. 1959;46: 77–90. DOI: 10.1093/biomet/46.1-2.77.

[79] Sacks J, Ylvisaker D. Designs for Regression Problems with Correlated Errors. The Annals of Mathematical Statistics. 1966;37(1): 66–89. DOI: 10.1214/aoms/1177699599.

[80] Sacks J, Ylvisaker D. Designs for Regression Problems With Correlated Errors: Many Parameters. The Annals of Mathematical Statistics. 1968;39(1): 49–69. DOI: 10.1214/aoms/1177698504.

[81] Atkinson AC. Developments in the Design of Experiments. International Statistical Review. 1982;50(2): 161–177.

[82] Fedorov V. Theory of Optimal Experiments Designs. New York and London: Academic Press, 1972.

[83] Newey WK, McFadden D. "Large sample estimation and hypothesis testing". *Handbook of Econometrics*. Elsevier, 1994, 2111–2245. ISBN: 1573-4412. DOI: 10.1016/S1573-4412(05)80005-4.

[84] Casella G, Berger RL. Statistical Inference. 2nd ed. Pacific Grove: Duxbury Press, 2002. ISBN: 978-8131503942.

[85] Fedorov V, Hackl P. Model-Oriented Design of Experiments. 1st ed. New York: Springer, 1997. ISBN: 978-1-4612-0703-0.

[86] Yang M, Biedermann S, Tang E. On Optimal Designs for Nonlinear Models: A General and Efficient Algorithm. Journal of the American Statistical Association. 2013;108(504): 1411–1420. DOI: 10.1080/01621459.2013.806268.

[87] Aarons L, Ogungbenro K. Optimal Design of Pharmacokinetic Studies. Basic & clinical pharmacology & toxicology. 2010;106: 250–255. DOI: 10.1111/j.1742-7843.2009.00533.x.

[88] Ryeznik Y, Sverdlov O, Svensson EM, Montepiedra G, Hooker AC, Wong WK. Pharmacometrics meets statistics-A synergy for modern drug development. CPT Pharmacometrics & Systems Pharmacology. 2021;10(10): 1134–1149. DOI: 10.1002/psp4.12696.

[89] Baksalary JK, Pukelsheim F, Styan GP. Some properties of matrix partial orderings. Linear Algebra and its Applications. 1989;119: 57–85. DOI: 10.1016/0024-3795(89)90069-4.

[90] Fisher RA. On the Mathematical Foundations of Theoretical Statistics. Philosophical Transactions of the Royal Society of London. Series A. 1922(;222): 309–368. DOI: 10.1098/rsta.1922.0009.

[91] Lehmann EL, Casella G. Theory of Point Estimation. 2nd ed. New York: Springer, 1998. ISBN: 978-0-387-98502-2.

[92] Cramér H. Mathematical Methods of Statistics. Uppsala: Almqvist & Wiksells, 1945.

[93] Rao R. Information and the Accuracy Attainable in the Estimation of Statistical Parameters. Bulletin of Calcutta Mathematical Society. 1945;37: 81–91.

[94] Söderström T, Stoica P. System Identification. Prentice Hall International series in systems and control engineering. Hertfordshire: Prentice Hall, 1989. ISBN: 978-0-13-881236-5.

[95] Eldar YC. Rethinking Biased Estimation: Improving Maximum Likelihood and the Cramér–Rao Bound. Foundations and Trends in Signal Processing. 2007;1(4): 305–449. DOI: 10.1561/2000000008.

[96] Gagnon R, Leonov S. Optimal population designs for PK models with serial sampling. Journal of biopharmaceutical statistics. 2005;15(1): 143–163. DOI: 10.1081/bip-200040853.

[97] Malagò L, Pistone G. "Information Geometry of the Gaussian Distribution in View of Stochastic Optimization". *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*. Aberystwyth, United Kingdom: Association for Computing Machinery, 2015, 150–162. ISBN: 9781450334341. DOI: 10.1145/2725494.2725510.

[98] Seurat J, Tang Y, Mentré F, Nguyen TT. Finding optimal design in nonlinear mixed effect models using multiplicative algorithms. Computer Methods and Programs in Biomedicine. 2021;207: 106126. DOI: 10.1016/j.cmpb.2021.106126.

[99] Davidian M, Giltinan DM. Nonlinear models for repeated measurement data: An overview and update. Journal of Agricultural, Biological, and Environmental Statistics. 2003;8(4): 387–419. DOI: 10.1198/1085711032697.

[100] Pinheiro JC, Bates DM. Approximations to the Log-Likelihood Function in the Nonlinear Mixed-Effects Model. Journal of Computational and Graphical Statistics. 1995;4(1): 12–35. DOI: 10.1080/10618600.1995.10474663.

[101] Artin E. The Gamma Function. Holt, Rinehart and Winston, 1964.

[102] Atkinson AC. Examples of the use of an equivalence theorem in constructing optimum experimental designs for random-effects nonlinear regression models. Journal of Statistical Planning and Inference. 2008;138(9): 2595–2606. DOI: 10.1016/j.jspi.2008.03.002.

[103] Pukelsheim F. Optimal Design of Experiments. New York: John Wiley & Sons, 1993. ISBN: 978-0-89871-604-7.

[104] Fedorov VV, Gagnon RC, Leonov SL. Design of experiments with unknown parameters in variance. Applied Stochastic Models in Business and Industry. 2002;18(3): 207–218. DOI: 10.1002/asmb.474.

[105] Silvey SD. Optimal Design: An Introduction to the Theory for Parameter Estimation. New York: Chapmann and Hall, 1980. ISBN: 978-94-009-5914-9.

[106] Chaloner K, Verdinelli I. Bayesian Experimental Design: A Review. Statistical Science. 1995;10(3): 273–304. DOI: 10.1214/ss/1177009939.

[107] Hill PDH. D-Optimal Designs for Partially Nonlinear Regression Models. Technometrics. 1980;22(2): 275–276. DOI: 10.1080/00401706.1980.10486145.

[108] Embacher S, Berghold A, Maertens K, Herzog SA. A Practical Framework to Design Immunization Studies Based on the Beta Distribution. Statistics in Medicine. 2025;44(23-24): e70293. DOI: 10.1002/sim.70293.

[109] Kruschke J. Doing Bayesian Data Analysis. 2nd ed. Boston: Academic Press, 2015. ISBN: 978-0-12-405888-0.

[110] Matsumoto N, Sasaki A, Kadowaki T, Mitsuhashi T, Takao S, Yorifuji T. Longitudinal antibody dynamics after COVID-19 vaccine boosters based on prior infection status and booster doses. Scientific Reports. 2024;14(1): 4564. DOI: 10.1038/s41598-024-55245-9.

[111] Tam VH, Preston SL, Drusano GL. Optimal sampling schedule design for populations of patients. Antimicrobial Agents and Chemotherapy. 2003;47(9): 2888–2891. DOI: 10.1128/AAC.47.9.2888-2891.2003.

[112] Hasselman B. nleqslv: Solve Systems of Nonlinear Equations [Internet]. [cited 11th December 2025]. 2023. URL: https://CRAN.R-project.org/package=nleqslv.

[113] Dennis JE, Schnabel RB. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Philadelphia: Society for Industrial and Applied Mathematics, 1996. DOI: 10.1137/1.9781611971200.

[114] Alkhamis TM, Ahmed MA. A modified Hooke and Jeeves algorithm with likelihood ratio performance extrapolation for simulation optimization. European Journal of Operational Research. 2006;174(3): 1802–1815. DOI: 10.1016/j.ejor.2005.04.032.

[115] Nash JC, Varadhan R. Unifying Optimization Algorithms to Aid Software System Users: optimx for R. Journal of Statistical Software. 2011;43(9): 1–14. DOI: 10.18637/jss.v043.i09.

[116] Nash JC. On Best Practice Optimization Methods in R. Journal of Statistical Software. 2014;60(2): 1–14. DOI: 10.18637/jss.v060.i02.

[117] R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria, 2024. URL: https://www.R-project.org/.

[118] Kendall MG, Ord JK. Time series. 3rd ed. New York: Oxford University Press, 1990. ISBN: 0195207068.

[119] Kac M, Murdock WL, Szegö G. On the Eigen-values of Certain Hermitian Forms. Journal of Rational Mechanics and Analysis. 1953;2: 767–800.

[120] Dow M. Explicit inverses of Toeplitz and associated matrices. ANZIAM J. 2003;44(E): E185–E215.

[121] Fikioris G. Spectral properties of Kac–Murdock–Szegö matrices with a complex parameter. Linear Algebra and its Applications. 2018;553: 182–210. DOI: 10.1016/j.laa.2018.05.004.

[122] Asuero AG, Sayago A, González AG. The Correlation Coefficient: An Overview. Critical Reviews in Analytical Chemistry. 2006;36(1): 41–59. DOI: 10.1080/10408340500526766.

[123] Embacher S, Herzog SA. Shiny Application: A practical framework to design immunization studies based on the beta distribution [Internet]. [cited 11th December 2025]. Version 1.0. 2025. URL: https://doi.org/10.5281/zenodo.15826920.

[124] Chang W, Cheng J, Allaire J, Sievert C, Schloerke B, Xie Y, et al. shiny: Web Application Framework for R. 2024. URL: https://CRAN.R-project.org/package=shiny.

[125] Srivastava K, Carreño JM, Gleason C, Monahan B, Singh G, Abbad A, et al. SARS-CoV-2-infection- and vaccine-induced antibody responses are long lasting with an initial waning phase followed by a stabilization phase. Immunity. 2024;57(3): 587–599.e4. DOI: 10.1016/j.immuni.2024.01.017.

[126] Violán C, Torán-Monserrat P, Quirant B, Lamonja-Vicente N, Carrasco-Ribelles LA, Chacón C, et al. Kinetics of humoral immune response over 17 months of COVID-19 pandemic in a large cohort of healthcare workers in Spain: the ProHEpiC-19 study. BMC Infectious Diseases. 2022;22(1): 721. DOI: 10.1186/s12879-022-07696-6.

[127] Yman V, White MT, Asghar M, Sundling C, Sondén K, Draper SJ, et al. Antibody responses to merozoite antigens after natural Plasmodium falciparum infection: kinetics and longevity in absence of re-exposure. BMC Medicine. 2019;17(1). DOI: 10.1186/s12916-019-1255-3.

[128] Walter E, Pronzato L. Optimal experiment design for nonlinear models subject to large prior uncertainties. American Journal of Physiology. 1987;253. DOI: 10.1152/ajpregu.1987.253.3.R530.

[129] Foo L.-K, and Duffull S. Methods of Robust Design of Nonlinear Models with an Application to Pharmacokinetics. Journal of Biopharmaceutical Statistics. 2010;20(4): 886–902. DOI: 10.1080/10543401003618918.

[130] Fayette L, Leroux R, Mentré F, Seurat J. Robust and Adaptive Two-stage Designs in Nonlinear Mixed Effect Models. The AAPS journal. 2023;25(4): 71. DOI: 10.1208/s12248-023-00810-9.

[131] Myung JI, Pitt MA. Optimal experimental design for model discrimination. Psychological Review. 2009;116(3): 499–518. DOI: 10.1037/a0016104.

[132] Harman R, Filová L, Rosa S. The polytope of optimal approximate designs: extending the selection of informative experiments. Statistics and Computing. 2024;34(6): 211. DOI: 10.1007/s11222-024-10527-0.

[133] Heinze G, Boulesteix A.-L, Kammer M, Morris TP, White IR, the Simulation Panel of the STRATOS initiative. Phases of methodological research in biostatistics—Building the evidence base for new methods. Biometrical Journal. 2024;66(1): 2200222. DOI: 10.1002/bimj.202200222.

[134] Atkinson AC, Donev AN, Tobias R. Optimum Experimental Designs, with SAS. Oxford: Oxford University Press, 2007. ISBN: 978-0-19-929659-0. DOI: 10.1093/oso/9780199296590.001.0001.

[135] Wolters Kluwer. Ovid: Search form [Internet]. [cited 11th December 2025]. 2021. URL: https://ovidsp.ovid.com/.

Appendix A.

Systematic Review

A.1. Search Strategy

Antibodies

#	Searches	Results
1	antibod$.mp.	2944474
2	humoral immune response.mp.	20800
3	igg.mp.	385860
4	iga.mp.	147013
5	igm.mp.	175239
6	immunoglobulin.mp.	1233509
7	ab.mp.	191363
8	1 or 2 or 3 or 4 or 5 or 6 or 7	3815520

Development of antibody levels

#	Searches	Results
9	kinetics.mp.	1239906
10	dynamic?.mp.	1959271
11	waning.mp.	10405
12	decay.mp.	160422
13	decline.mp.	599105
14	9 or 10 or 11 or 12 or 13	3791110

Model

#	Searches	Results
15	longitudinal.mp.	876753
16	model$.mp.	10645200
17	equation?.mp.	435068
18	statistical analys#s.mp.	645688
19	pharmacokinetics?.mp.	1138165
20	half-life.mp.	296566
21	15 or 16 or 17 or 18 or 19 or 20	12857809

Combination

#	Searches	Results
22	8 adj5 14	19856
23	21 and 22	6394
24	limit 23 to english language	6292
25	limit 24 to (medline or publisher or "pubmed not medline")	3443
26	remove duplicates from 25	2841
27	ab initio	37564
28	26 not 27	1811

Table A.1.: Search Strategy as published in the protocol [39], using the databases Ovid MEDLINE(R) <1946 to March Week 5 2024> and Embase <1974 to 2024 April 08> simultaneously within the search engine Ovid[135]. The Search was performed on the 9th of April 2024. '$' represents truncation, '?' replaces zero or one character, '#' replaces exactly one character and 'adj5' assures that the two searches are within a maximum range of 5 words.

A.2. Flowchart

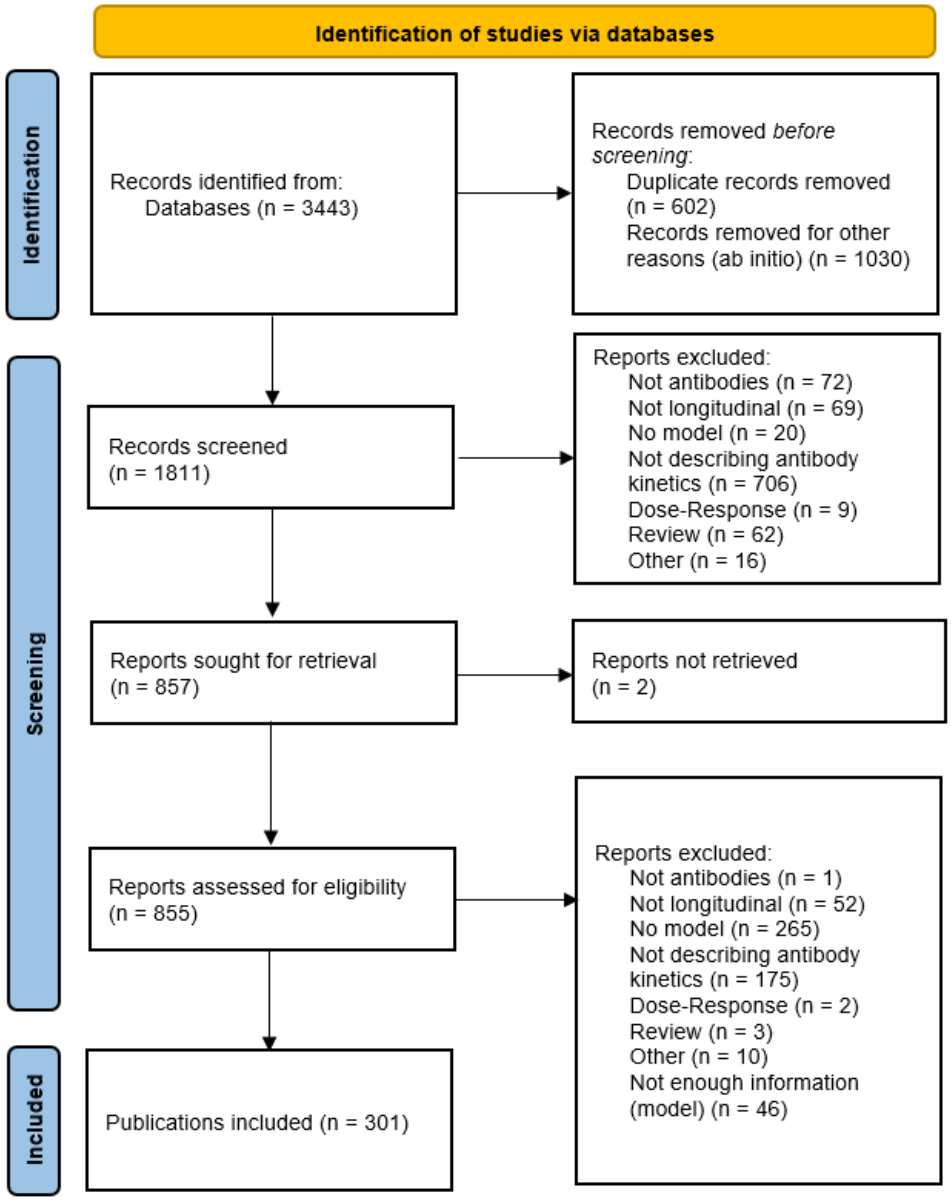

Figure A.1.: PRISMA flow diagram describing the selection process, incldluing identification, screening, eligibility assessment, exclusion criteria applied and the final number of included publications.

A.3. Extracted Information

We extract the following information for all included publications using Limesurvey:

1. Publication
    - Author(s), Title, Year, Source
    - Type: Theoretical Work, Methodological paper, Implemented Study
2. Model
    - Statistical or mathematical model?
    - Which model(s) is/are used?
    - What are the main model assumptions?
    - Are the equations stated?
    - Is there a theoretical concept described?
    - How was the model performance evaluated?
    - Is the model used for study design?
    - If yes: To answer what questions?
3. Antibodies
    - Are there specific antibodies described?
    - If yes: Which and for what disease?
    - How are the antibodies obtained?
    - How is the topic of lower limit of detection addressed?
4. Application
    - What is the population?
    - What is the sample size?
    - Number of time-points?
    - What is/are the modelling aim(s)?
    - What is/are the main outcome(s)?
5. Simulations
    - Where any simulations performed? If yes,
        - Was the data-generating model described?

  - Were the simulations based on a real data set?
  - Were simulations performed to compare the method to another? If yes, what was the outcome?

6. Data/Code available
   - Is the software/programming language stated?
   - Is the code available?
   - Is the data available?
7. Limitations
   - Were any limitations regarding the model stated? If yes, which?
8. Reference list
   - Is any interesting literature cited?
9. Importance
   - Classify the importance of this publication.

Appendix B.

Regularity Assumptions

Regularity assumptions for consistency (i.e. converges in probability to the true value of the parameter) of the maximum likelihood estimator:[83]

1. $X_i \overset{\text{iid}}{\sim} f(x|\theta)$
2. Identifiability: $\theta \neq \theta' \Rightarrow f(x|\theta) \neq f(x|\theta')$
3. $\mathbb{E}[|log f(x|\theta)|] < \infty \quad \forall \theta \in \Theta$ and $\mathbb{E}[sup_{\theta \in \Theta} |log f(x|\theta)|] < \infty]$
4. The parameter space Θ is compact
5. $log f(x_i|\theta)$ is continuous $\forall \theta \in \Theta$ a.s.

The conditions for asymptotic normality of the maximum likelihood estimator are the following:[83]

6. θ_0 is an interior point of Θ
7. $f(x|\theta)$ is twice continuously differentiable
8. $f(x|\theta) > 0$ in a neighbourhood $U \subset \Theta$ of θ_0
9. $\int sup_{\theta \in U} \|D_\theta f(x|\theta)\| dx < \infty$ and $\int sup_{\theta \in U} \|D^2_\theta f(x|\theta)\| dx < \infty$
10. The matrix $\mathbb{E}[D_\theta \log f(x|\theta)^\intercal D_\theta \log f(x|\theta)|\theta]$ exists and is nonsingular
11. $\mathbb{E}[sup_{\theta \in U} \|D^2_\theta log f(x|\theta)\|] < \infty$

Appendix C.

Analytical and Numerical Sensitivity of σ and ρ

Throughout the framework and therefore the Robustness Analysis we assumed that the number of measurements n is equal to the number of parameters p, such that the Jacobian matrix of μ with respect to θ is a square matrix of full rank. The covariance matrix Σ is assumed to be independent of θ, specifically that it is of the form $\Sigma = S * AR(1)$ (see Section 2.4.1 for more details). The following results, however, holds for any (positive definite) covariance matrix, independent of the unknown parameters θ. Given that the observations are assumed to be multivariate normal, Σ can be assumed to be a regular, symmetric, positive definite matrix of dimensionality $n \times n$ and that the matrix form of the FIM can be expressed by

$$FIM = H^\intercal \Sigma^{-1} H,$$

where $H = D_\theta \boldsymbol{\mu}(\boldsymbol{t}, \boldsymbol{\theta})$ is a square matrix of dimension $n \times n$. Then, because H and Σ are square matrices of the same dimensionality

$$\det(FIM) = \det(H^\intercal \Sigma^{-1} H) = \det(H^\intercal)\det(\Sigma^{-1})\det(H) = \frac{1}{\det(\Sigma)}\det(H^\intercal H)$$

holds. Which then results in the optimization of the following expression with respect to $\boldsymbol{t} = (t_1, \cdots, t_n)$

$$-\ln\left(\frac{1}{det(\Sigma)}\det(D_\theta \boldsymbol{\mu}(\boldsymbol{t}, \boldsymbol{\theta})^\intercal D_\theta \boldsymbol{\mu}(\boldsymbol{t}, \boldsymbol{\theta})) + 1\right).$$

The determinant of the covariance matrix is a constant, non-zero scaling factor within the convex optimization problem and thus does not influence the optimal solution analytically. To evaluate the potential numerical sensitivity, a small Robustness Analysis regarding the choice of σ and ρ is conducted. A total of 50 possible combination is resulting by varying $\sigma \in \{0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2\}$ and $\rho \in \{0, 0.2, 0.4, 0.6, 0.8\}$. For each scenario and each pair, the sampling times are optimized and the difference between

the maximal and minimal value (for each scenario), in order to quantify the numerical variability resulting by the choice σ and ρ, are calculated. The differences are described in Table C.1, where it is observable that the overall largest difference is 0.0256635 days, equivalent to 36 minutes. Consequently, we can conclude that the numerical variability introduced by the choices of σ and ρ can be practically ignored.

Table C.1.: The maximal difference [days] of optimal sampling times in 12 scenarios, across 50 different pairs of standard deviation and correlation.

Scenario	Time 1 difference	Time 2 difference	Time 3 difference	Time 4 difference
1	3.03e-05	0.0009585	0.0041970	NA
2	9.15e-05	0.0006000	0.0018000	NA
3	6.30e-06	0.0022200	0.0138315	NA
4	1.80e-04	0.0016500	0.0031652	NA
5	0.00e+00	0.0000750	0.0014534	0.0085500
6	0.00e+00	0.0004500	0.0016650	0.0030000
7	0.00e+00	0.0004500	0.0016650	0.0030000
8	0.00e+00	0.0004230	0.0026717	0.0105150
9	0.00e+00	0.0001801	0.0018000	0.0076515
10	0.00e+00	0.0045000	0.0165298	0.0150000
11	0.00e+00	0.0000600	0.0029835	0.0256635
12	0.00e+00	0.0021000	0.0030000	0.0150000

Appendix D.

Sensitivity Analysis of Starting Values for Beta Distribution

---

To evaluate the presence of parameter uncertainty in the solutions of the beta distribution, starting values were sampled from a uniform distribution within the following specified bounds:

$$\alpha_{start} \sim U(1.01, 100)$$
$$\beta_{start} \sim U(1.01, 100)$$
$$t_{scale\_start} \sim U(t_{plat}, 50000)$$
$$c \sim U(0.01, 100)$$

here, the parameter c is only used if $A_0 \neq 0$. A total of 10000 simulations were conducted for each of the 12 scenarios. The numerical variability was assessed by computing the difference between the maximum and minimum values of each parameters within each scenario. The largest observed difference was of the order 10^{-5}, suggesting that the choice of the starting value has minimal influence on the numerical solutions for α, β, t_{scale} and c. This indicates no evidence of local optima within the optimization problem. However, the starting values did influence the convergence behaviour.

Appendix E.

Robustness Analysis - Additional Results

E.1. Single Parameter Misspecification

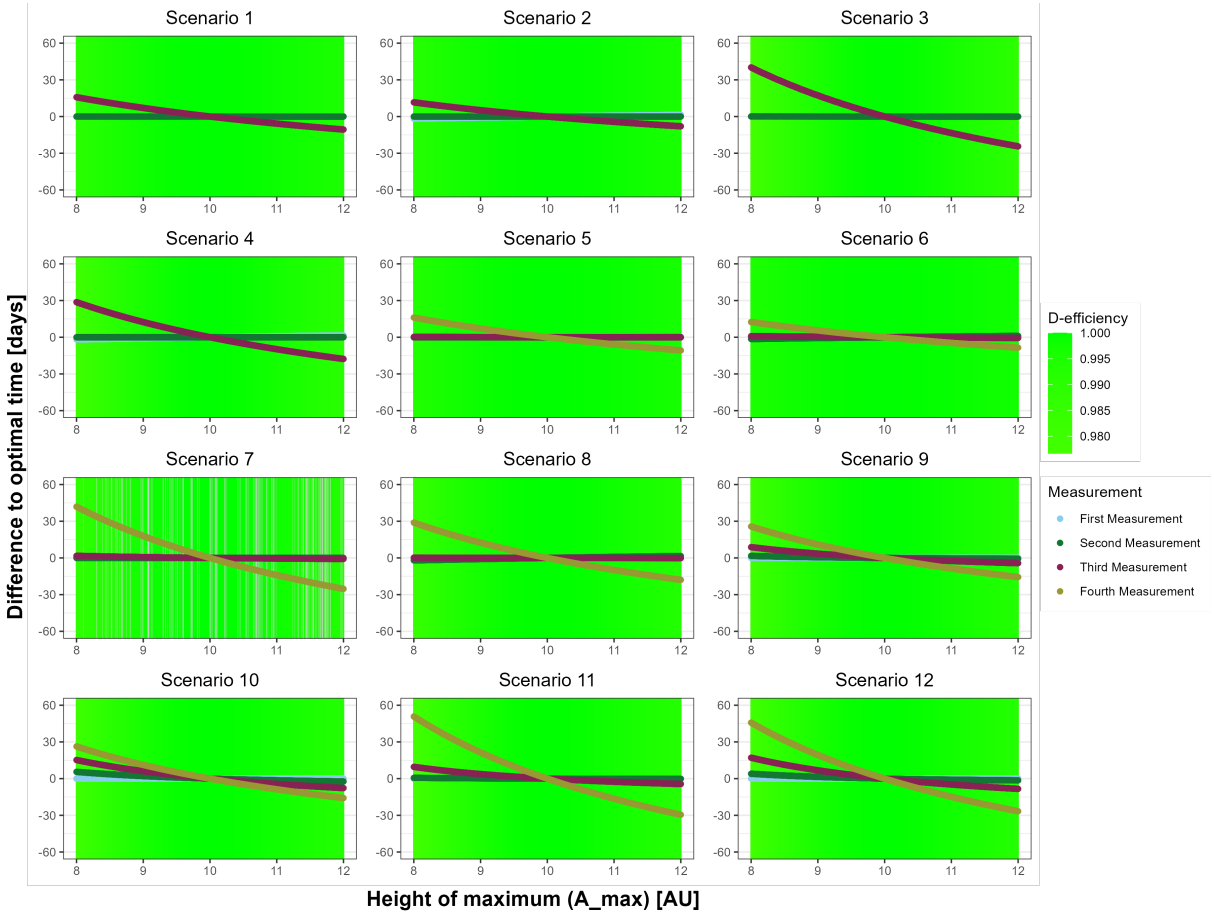

Figure E.1.: Plots showing the deviation of sampling times in relation to misspecification in the height of the maximum A_{max}. Each plot corresponds to a different scenario. The lines show the deviation of optimal sampling times based on misspecified initial information compared to those derived from the 'true' initial information. For some scenarios the deviation for the first measurement are not visible, because it is overlaid by the others. The color gradient represents the D-efficiency of each design, illustrating the impact of sampling time deviations on D-efficiency. In scenarios 1–4, three time points are optimized, whereas in scenarios 5–12, four time-points are optimized. The grey area denotes settings where no solution could be determined.

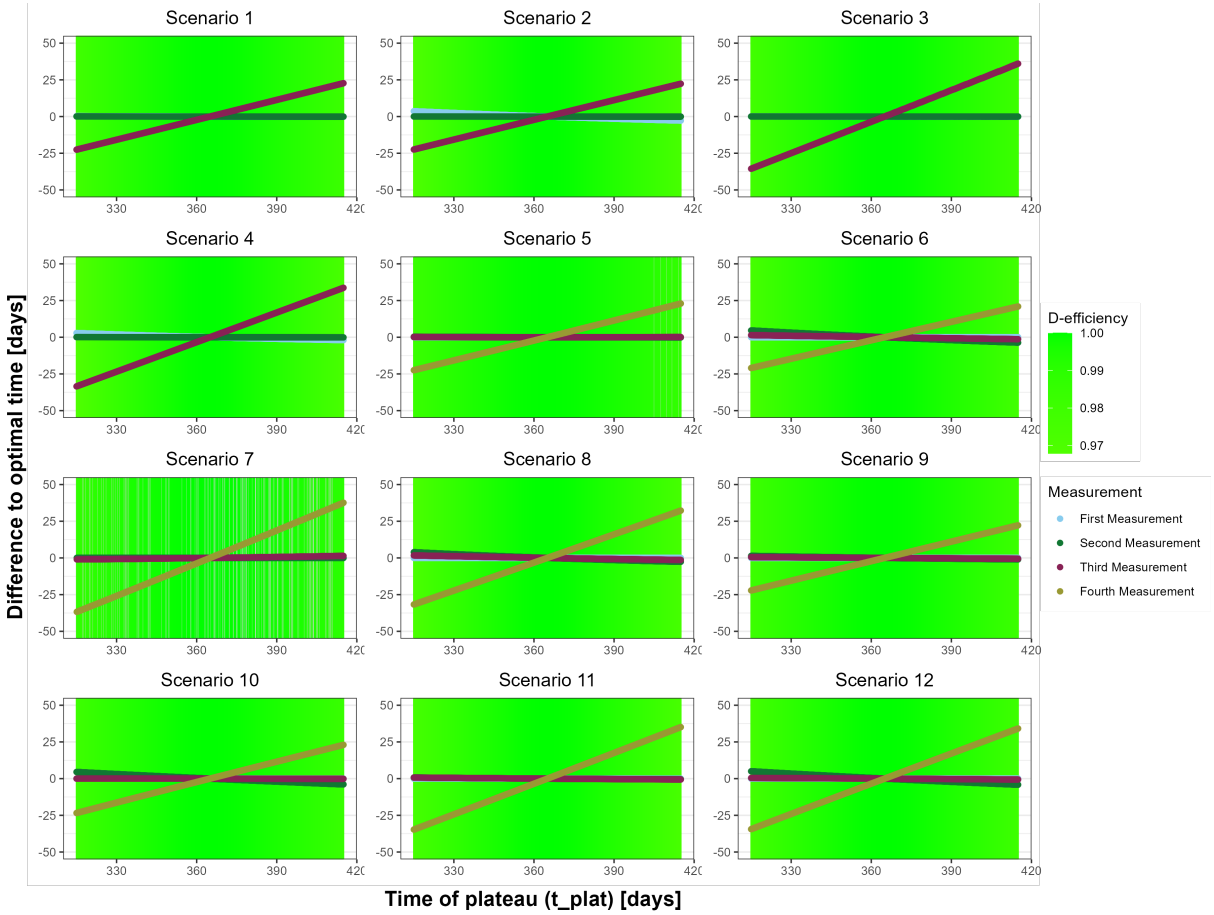

Figure E.2.: Plots showing the deviation of sampling times in relation to misspecification in the time of the plateau t_{plat}. Each plot corresponds to a different scenario. The lines show the deviation of optimal sampling times based on misspecified initial information compared to those derived from the 'true' initial information. For some scenarios the deviation for the first measurement are not visible, because it is overlaid by the others. The color gradient represents the D-efficiency of each design, illustrating the impact of sampling time deviations on D-efficiency. In scenarios 1–4, three time-points are optimized, whereas in scenarios 5–12, four time-points are optimized. The grey area denotes settings where no solution could be determined.

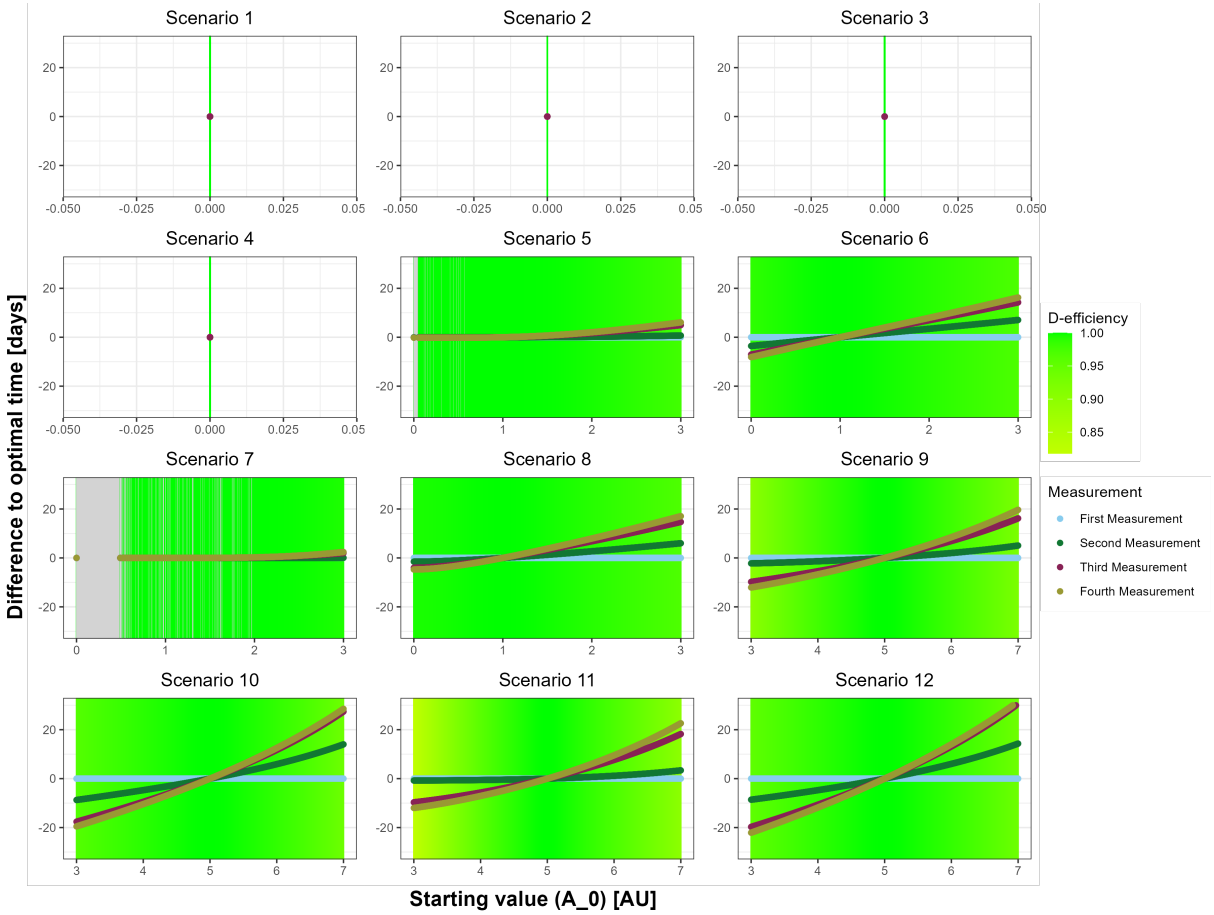

Figure E.3.: Plots showing the deviation of sampling times in relation to misspecification in the starting value A_0. Each plot corresponds to a different scenario. The lines show the deviation of optimal sampling times based on misspecified initial information compared to those derived from the 'true' initial information. For some scenarios the deviation for the first measurement are not visible, because it is overlaid by the others. The color gradient represents the D-efficiency of each design, illustrating the impact of sampling time deviations on D-efficiency. In scenarios 1–4, three time-points are optimized, whereas in scenarios 5–12, four time-points are optimized. The grey area denotes settings where no solution could be determined.

E.2. Double Parameter Misspecification

E.2.1. Heatmaps

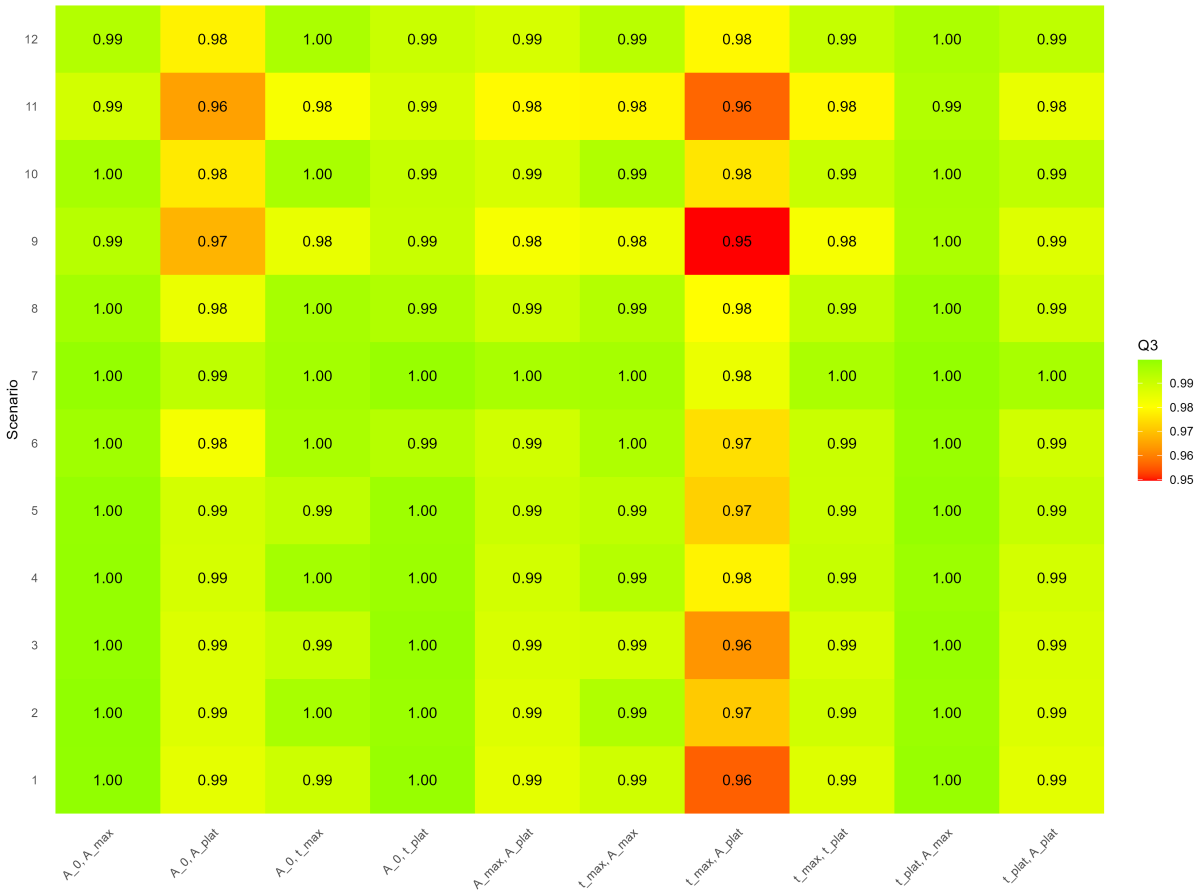

Figure E.4.: Heatmap of the third quartile D-efficiency values across different scenarios and pairs of misspecification in the initial information. The color gradient represents the D-efficiency, with green indicating higher values and red representing lower values. The x-axis corresponds to different pairs, while the y-axis represents the scenario index.

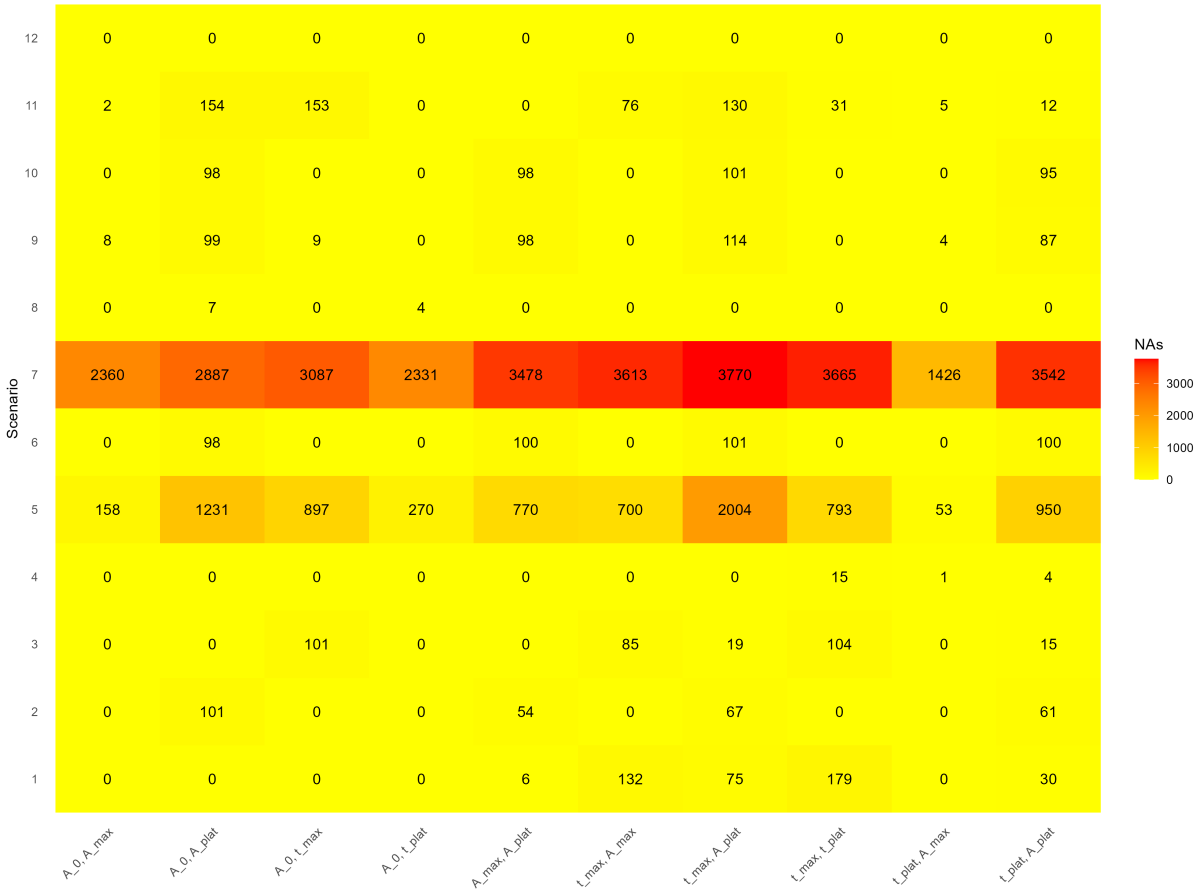

Figure E.5.: Heatmap of the number of runs not converging across different scenarios and pairs of misspecification in the initial information. The color gradient represents the number of runs not converging, with yellow indicating low or no convergence issues and red representing higher numbers. The x-axis corresponds to different pairs, while the y-axis represents the scenario index.

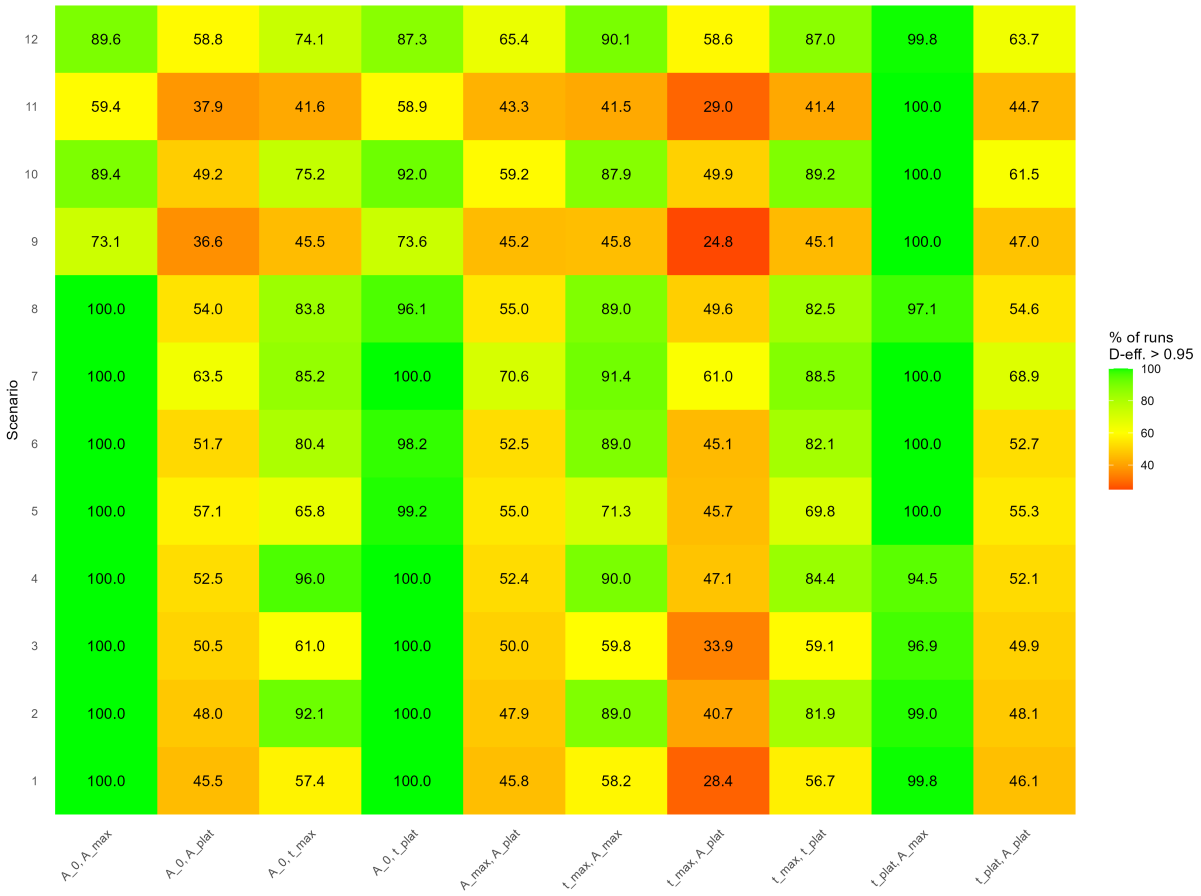

Figure E.6.: Heatmap of the percentage of runs showing a D-efficiency above 0.95, across different scenarios and pairs of misspecification in the initial information. The color gradient represents the relative number of runs, with green indicating higher values and red representing lower values. The x-axis corresponds to different pairs, while the y-axis represents the scenario index.

E.2.2. Pairwise

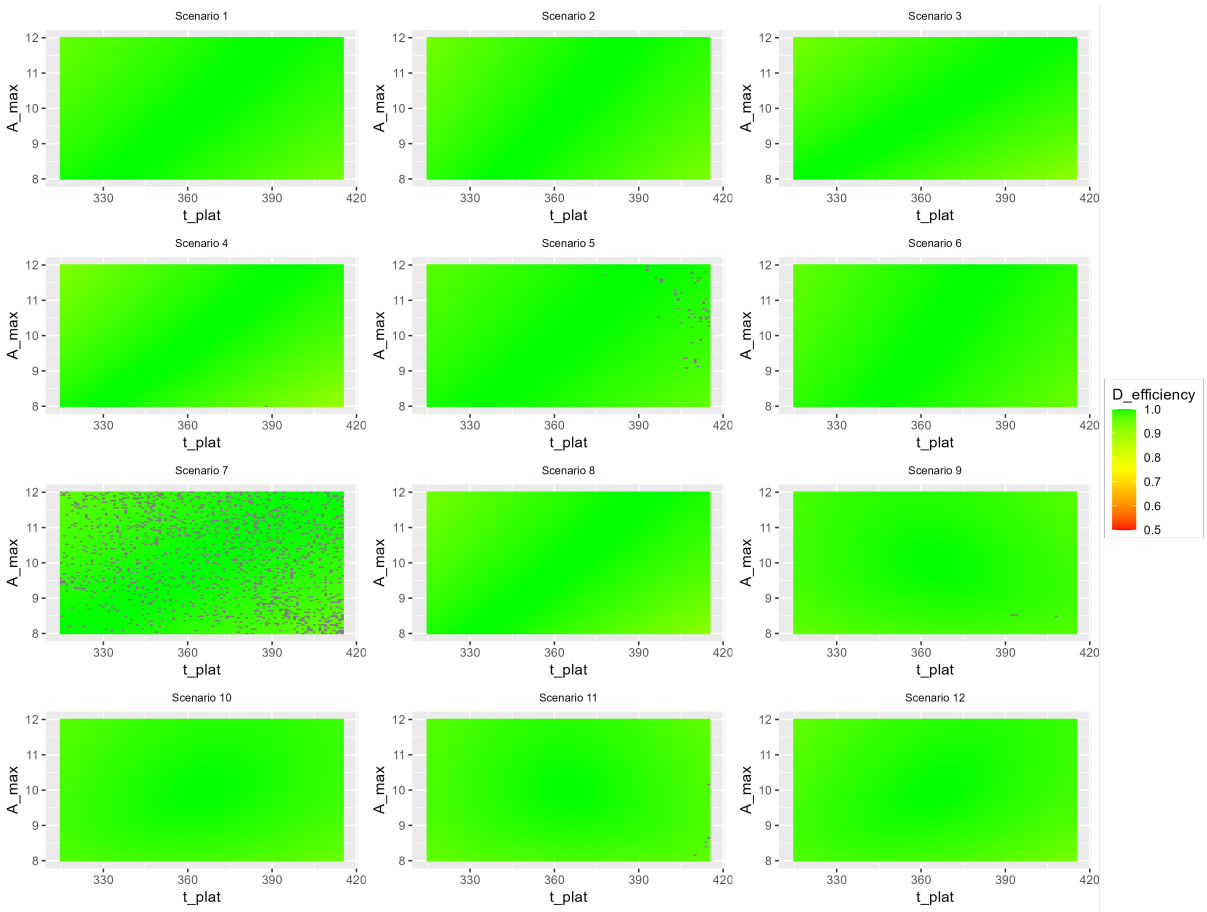

Figure E.7.: Graphical representation of misspecification in two parameters (the time of the plateau t_{plat} and the height of the maximum A_{max}) on the D-efficiency of the respective design. The color gradient represents the D-efficiency of each design, illustrating the impact of sampling time deviations on D-efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey area denotes settings where no solution could be determined. Note that in scenarios 1–4, three time-points are optimized, whereas in scenarios 5–12, four time-points are optimized.

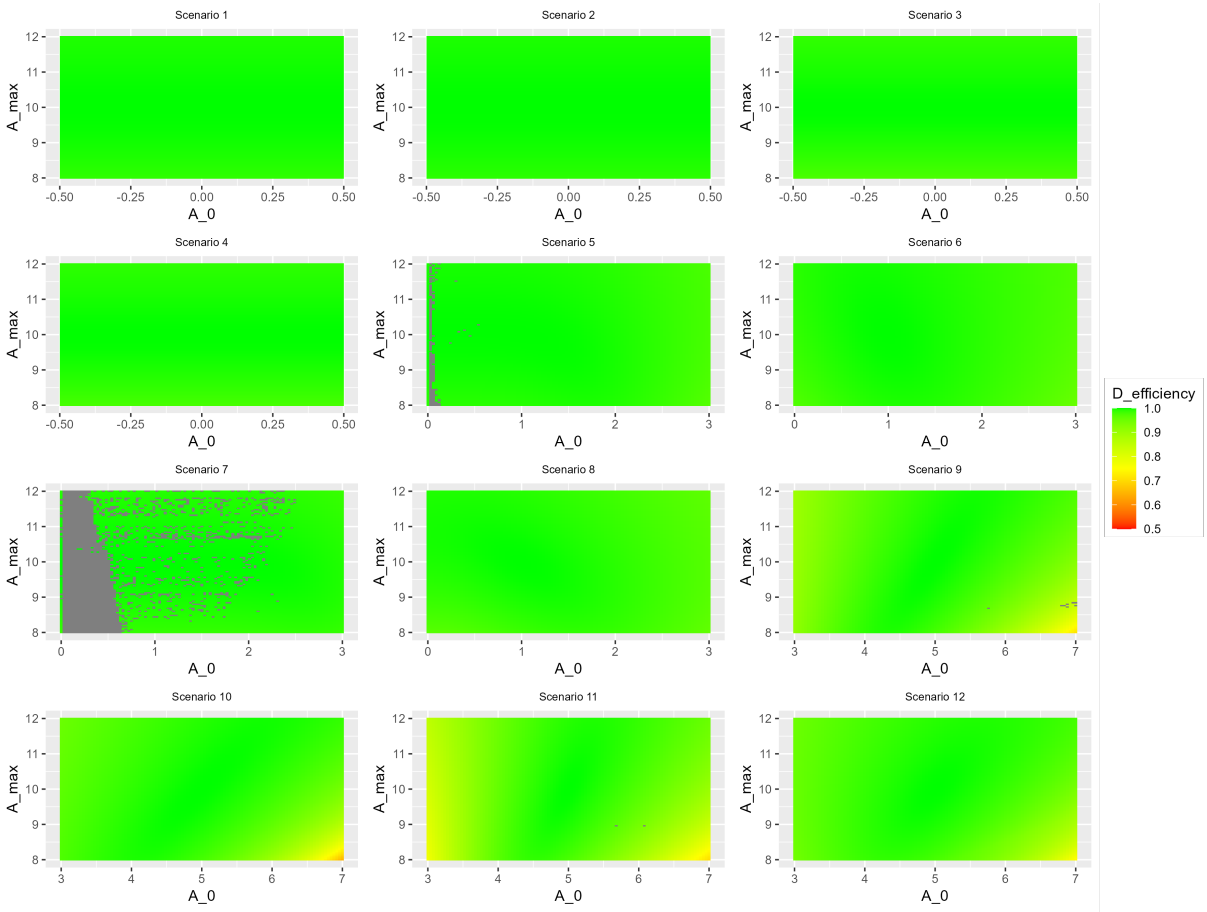

Figure E.8.: Graphical representation of misspecification in two parameters (the starting value A_0 and the height of the maximum A_{max}) on the D-efficiency of the respective design. The color gradient represents the D-efficiency of each design, illustrating the impact of sampling time deviations on D-efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey area denotes settings where no solution could be determined. Note that in scenarios 1–4, three time-points are optimized, whereas in scenarios 5–12, four time-points are optimized.

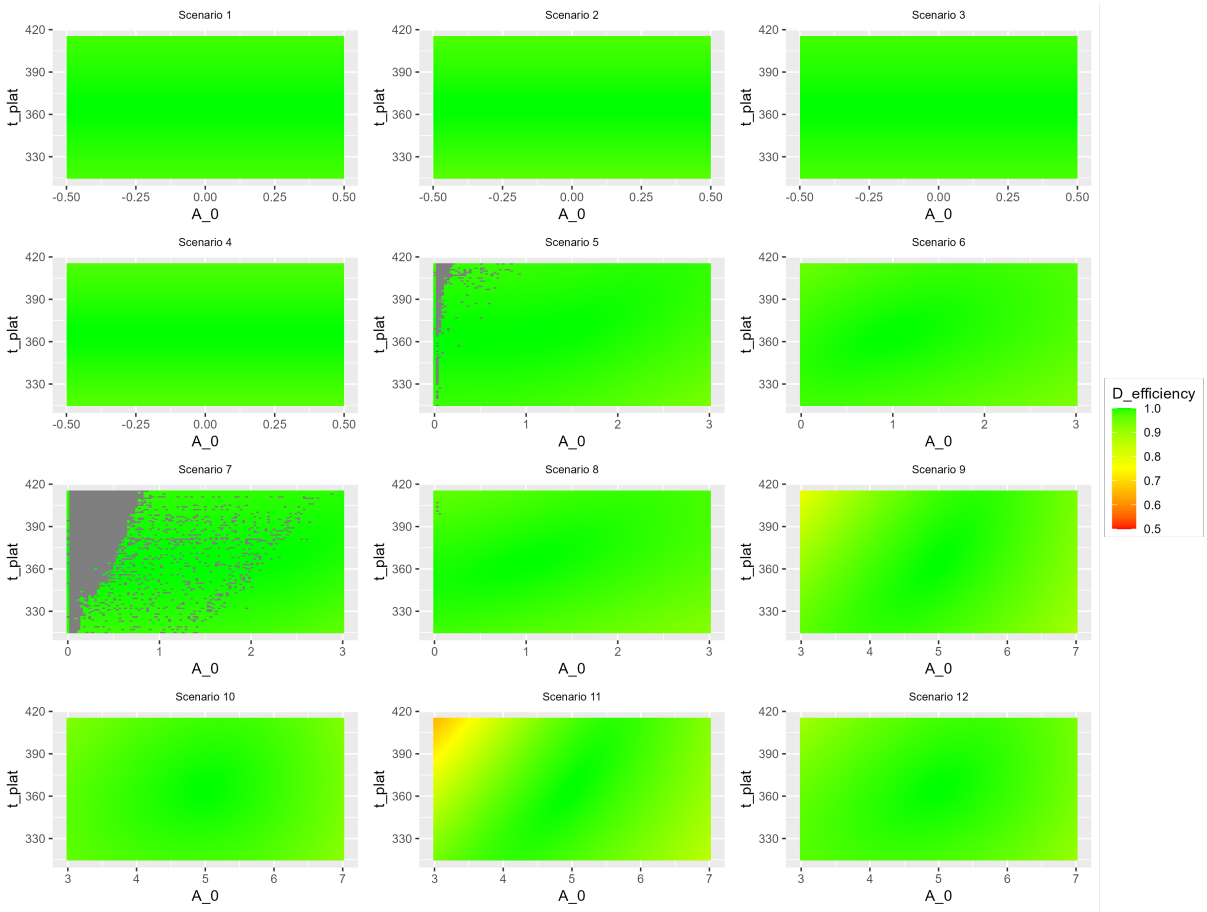

Figure E.9.: Graphical representation of misspecification in two parameters (the starting value A_0 and the time of the plateau t_{plat}) on the D-efficiency of the respective design. The color gradient represents the D-efficiency of each design, illustrating the impact of sampling time deviations on D-efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey area denotes settings where no solution could be determined. Note that in scenarios 1–4, three time-points are optimized, whereas in scenarios 5–12, four time-points are optimized.

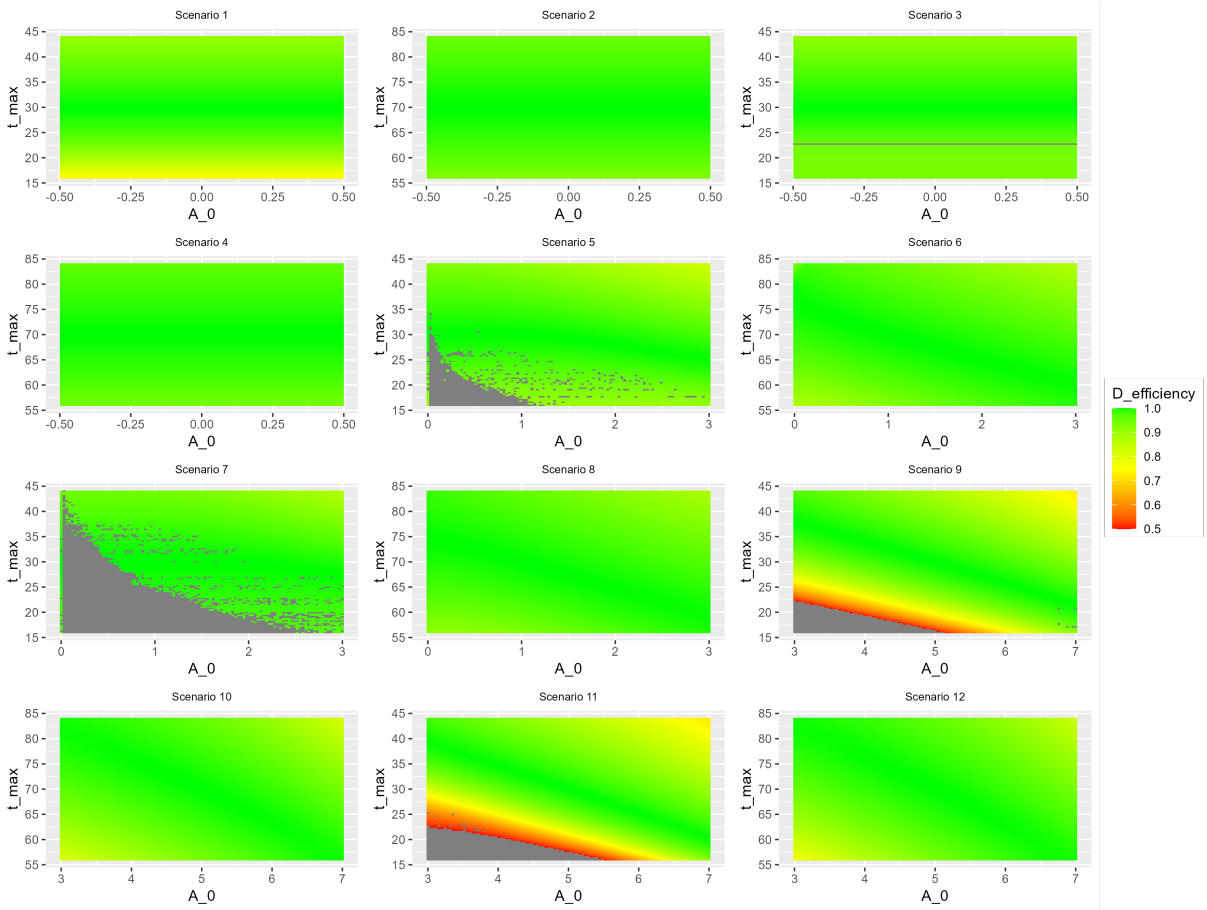

Figure E.10.: Graphical representation of misspecification in two parameters (the starting value A_0 and the time of the maximum t_{max}) on the D-efficiency of the respective design. The color gradient represents the D-efficiency of each design, illustrating the impact of sampling time deviations on D-efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey area denotes settings where no solution could be determined. Note that in scenarios 1–4, three time-points are optimized, whereas in scenarios 5–12, four time-points are optimized.

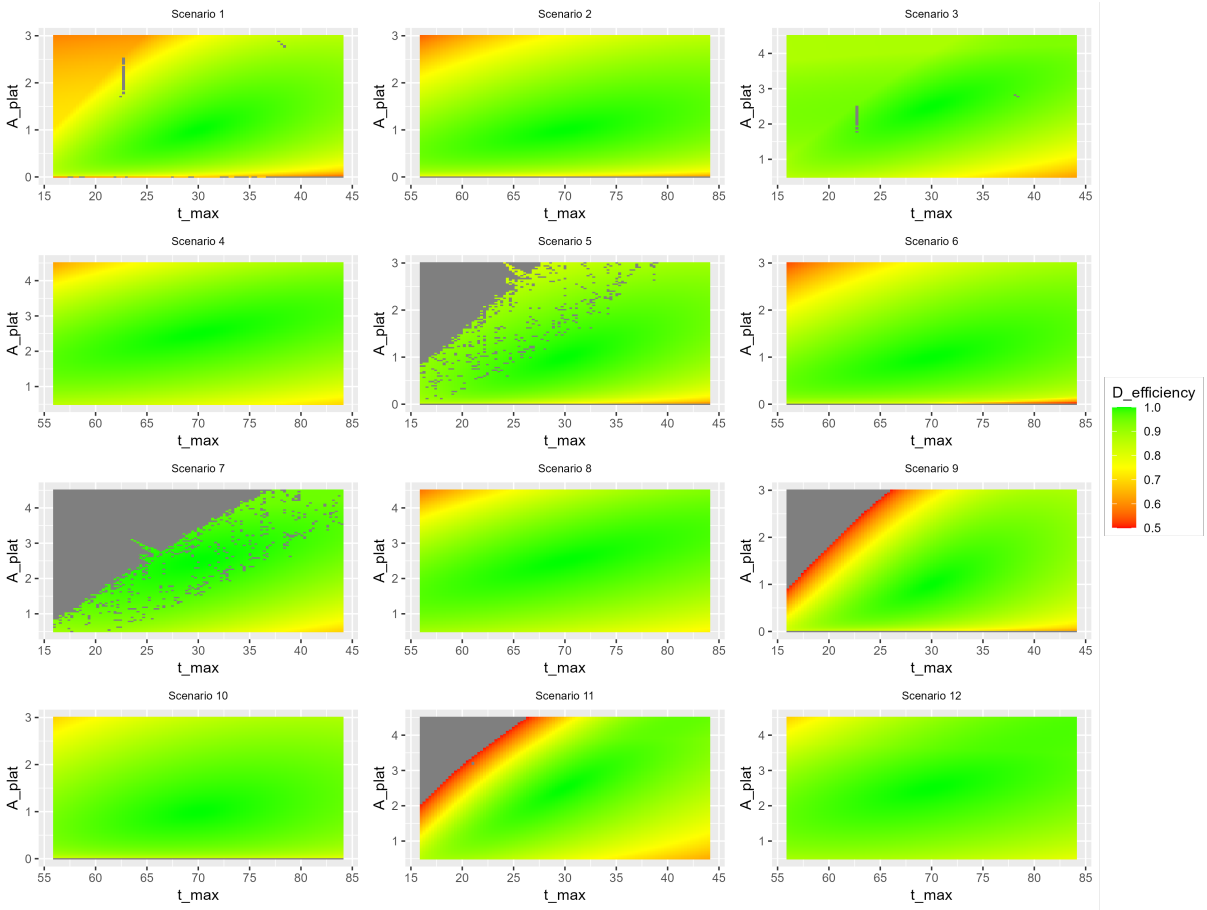

Figure E.11.: Graphical representation of misspecification in two parameters (the time of the maximum t_{max} and the height of the plateau A_{plat}) on the D-efficiency of the respective design. The color gradient represents the D-efficiency of each design, illustrating the impact of sampling time deviations on D-efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey area denotes settings where no solution could be determined. Note that in scenarios 1–4, three time-points are optimized, whereas in scenarios 5–12, four time-points are optimized. Reproduced from Embacher et al. [108]

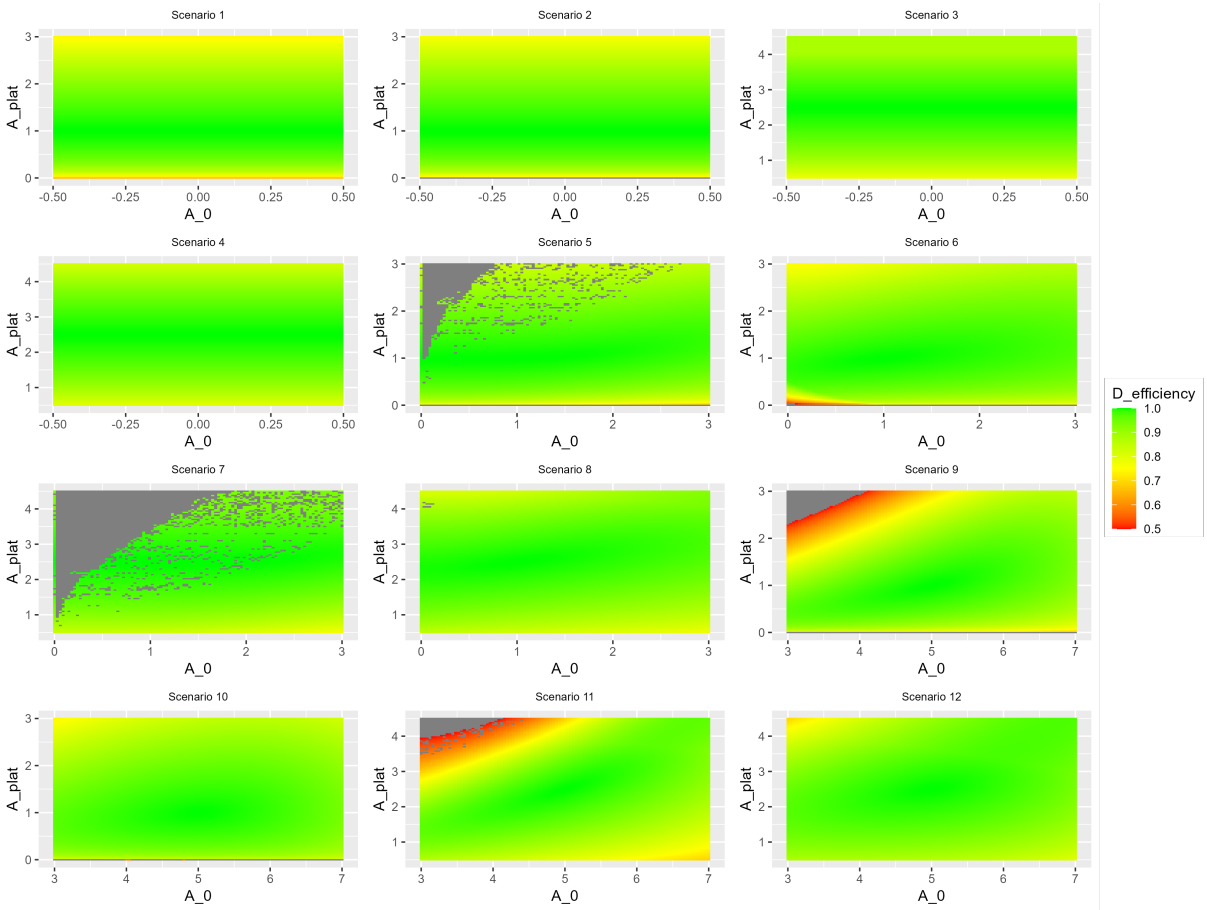

Figure E.12.: Graphical representation of misspecification in two parameters (the starting value A_0 and the height of the plateau A_{plat}) on the D-efficiency of the respective design. The color gradient represents the D-efficiency of each design, illustrating the impact of sampling time deviations on D-efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey area denotes settings where no solution could be determined. Note that in scenarios 1–4, three time-points are optimized, whereas in scenarios 5–12, four time-points are optimized.

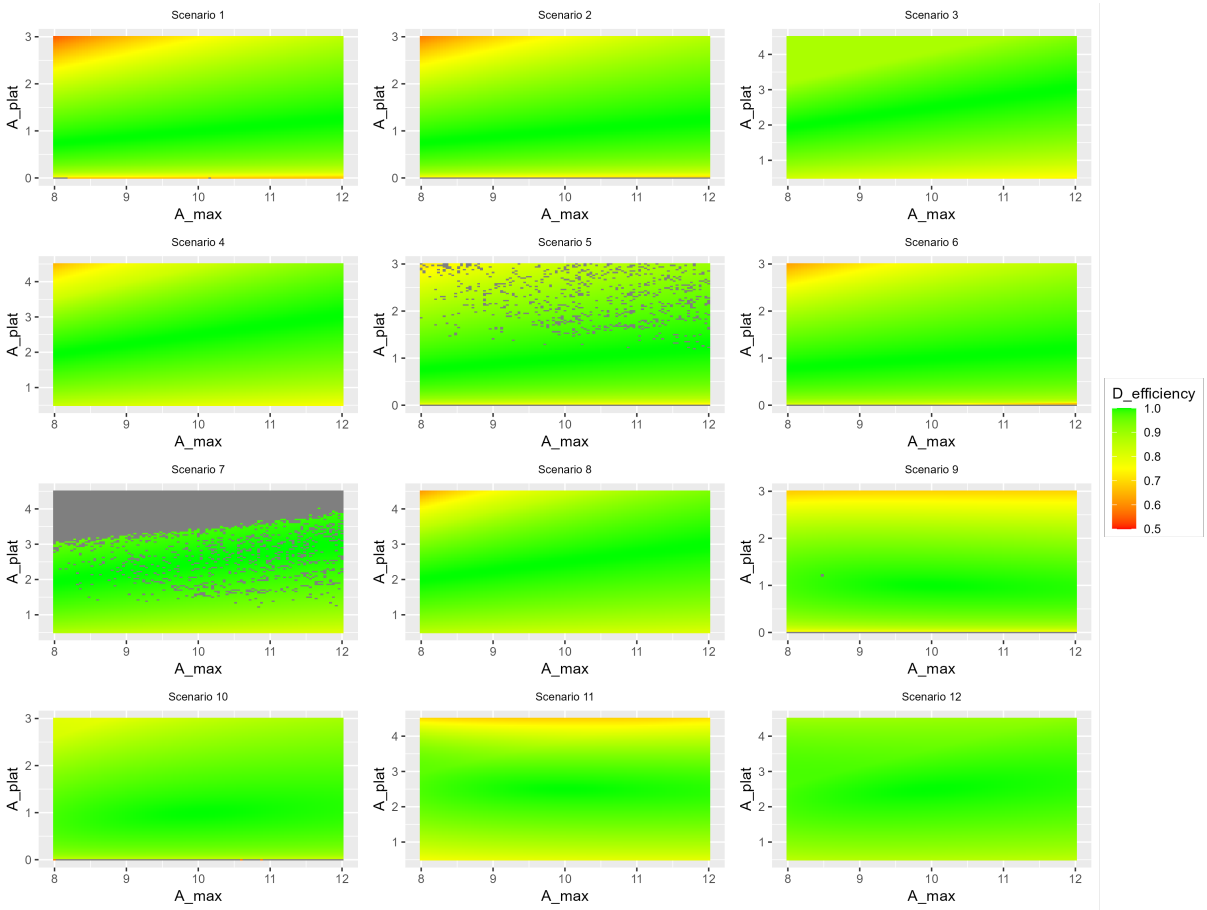

Figure E.13.: Graphical representation of misspecification in two parameters (the height of the maximum A_{max} and the height of the plateau A_{plat}) on the D-efficiency of the respective design. The color gradient represents the D-efficiency of each design, illustrating the impact of sampling time deviations on D-efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey area denotes settings where no solution could be determined. Note that in scenarios 1–4, three time-points are optimized, whereas in scenarios 5–12, four time-points are optimized.

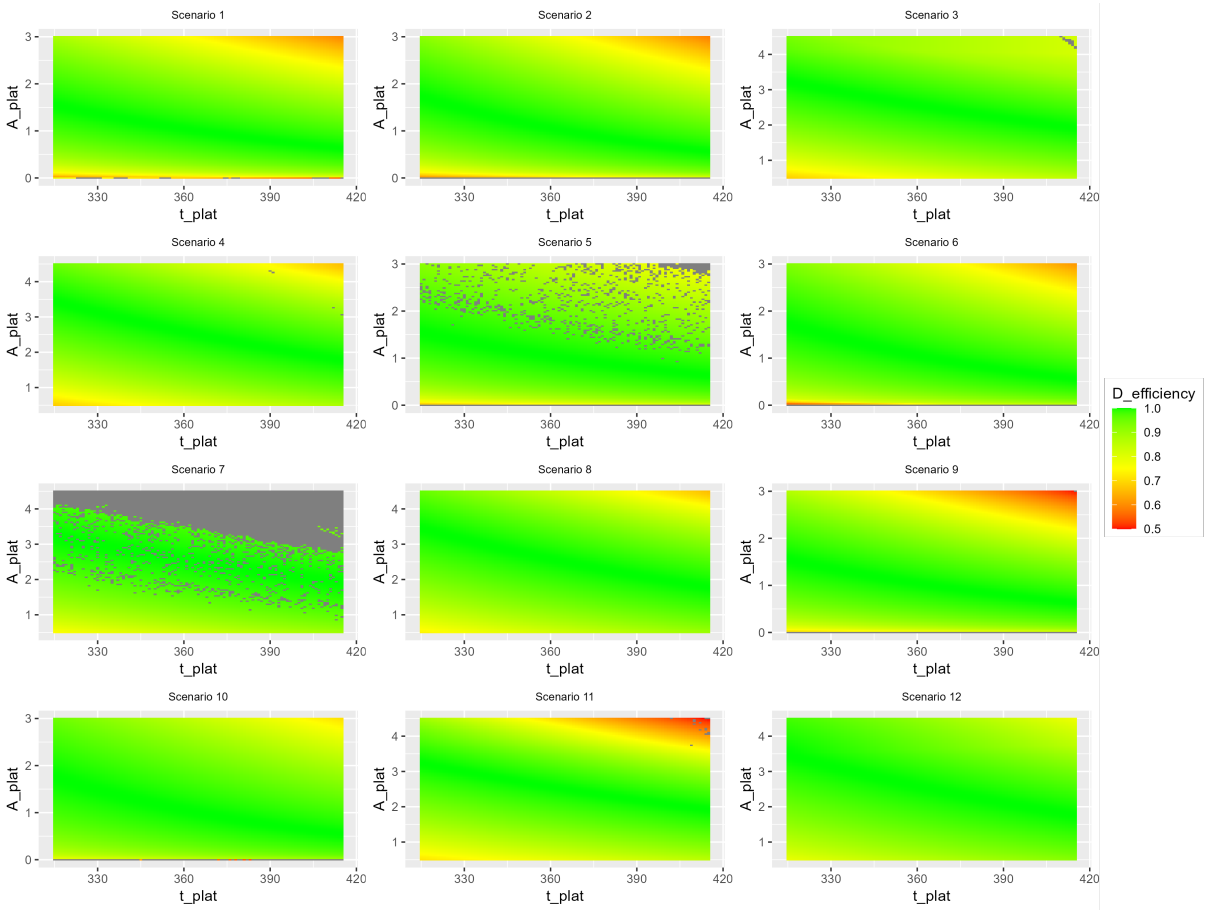

Figure E.14.: Graphical representation of misspecification in two parameters (the time of the plateau t_{plat} and the height of the plateau A_{plat}) on the D-efficiency of the respective design. The color gradient represents the D-efficiency of each design, illustrating the impact of sampling time deviations on D-efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey area denotes settings where no solution could be determined. Note that in scenarios 1–4, three time-points are optimized, whereas in scenarios 5–12, four time-points are optimized.

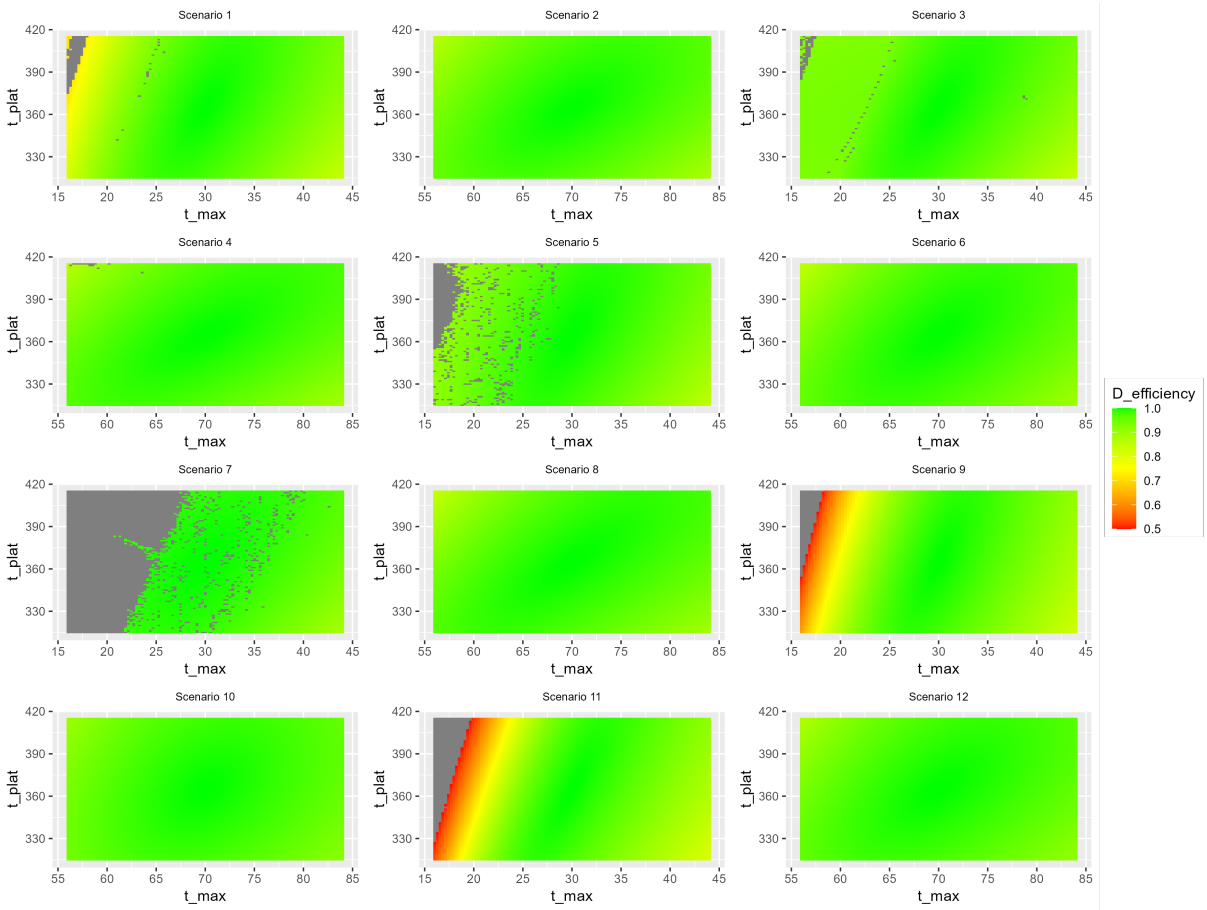

Figure E.15.: Graphical representation of misspecification in two parameters (the time of the maximum t_{max} and the time of the plateau t_{plat}) on the D-efficiency of the respective design. The color gradient represents the D-efficiency of each design, illustrating the impact of sampling time deviations on D-efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey area denotes settings where no solution could be determined. Note that in scenarios 1–4, three time-points are optimized, whereas in scenarios 5–12, four time-points are optimized.

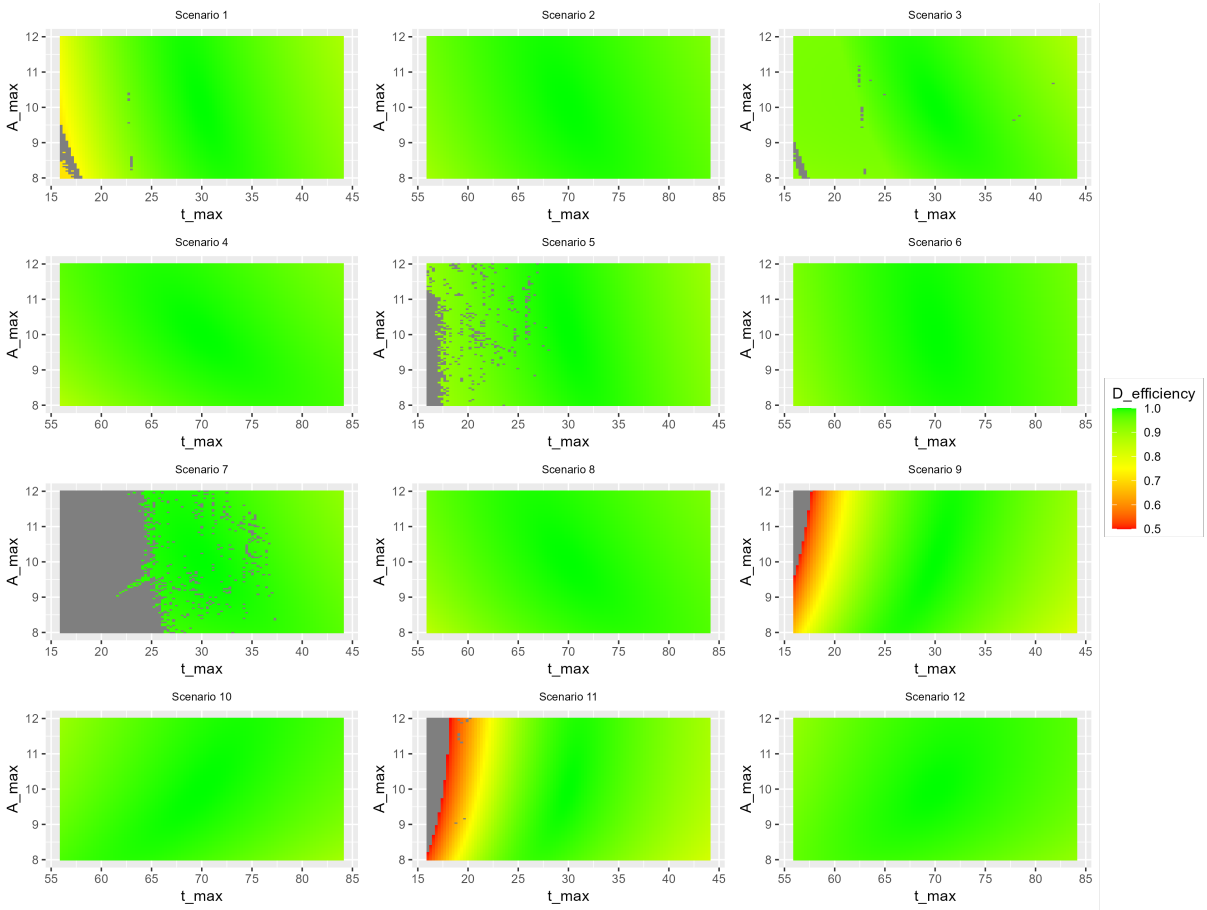

Figure E.16.: Graphical representation of misspecification in two parameters (the time of the maximum t_{max} and the height of the maximum A_{max}) on the D-efficiency of the respective design. The color gradient represents the D-efficiency of each design, illustrating the impact of sampling time deviations on D-efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey area denotes settings where no solution could be determined. Note that in scenarios 1–4, three time-points are optimized, whereas in scenarios 5–12, four time-points are optimized.

E.2.3. Scenariowise

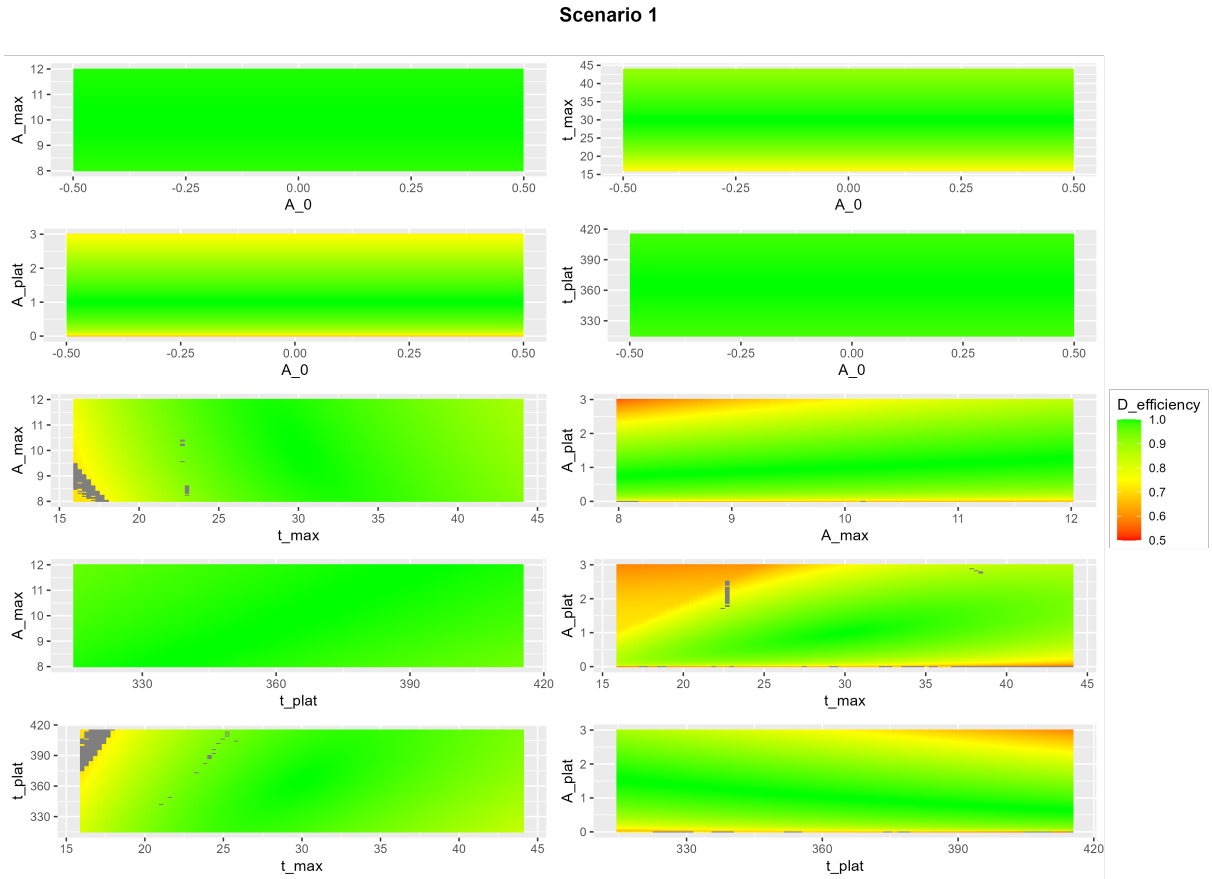

Figure E.17.: Graphical representation of the impact of simultaneous misspecification in two initial information on the D-efficiency of the respective design for Scenario 1. The color gradient represents the D-efficiency, with green indicating higher efficiency and red representing lower efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey areas denote settings where no solution could be determined.

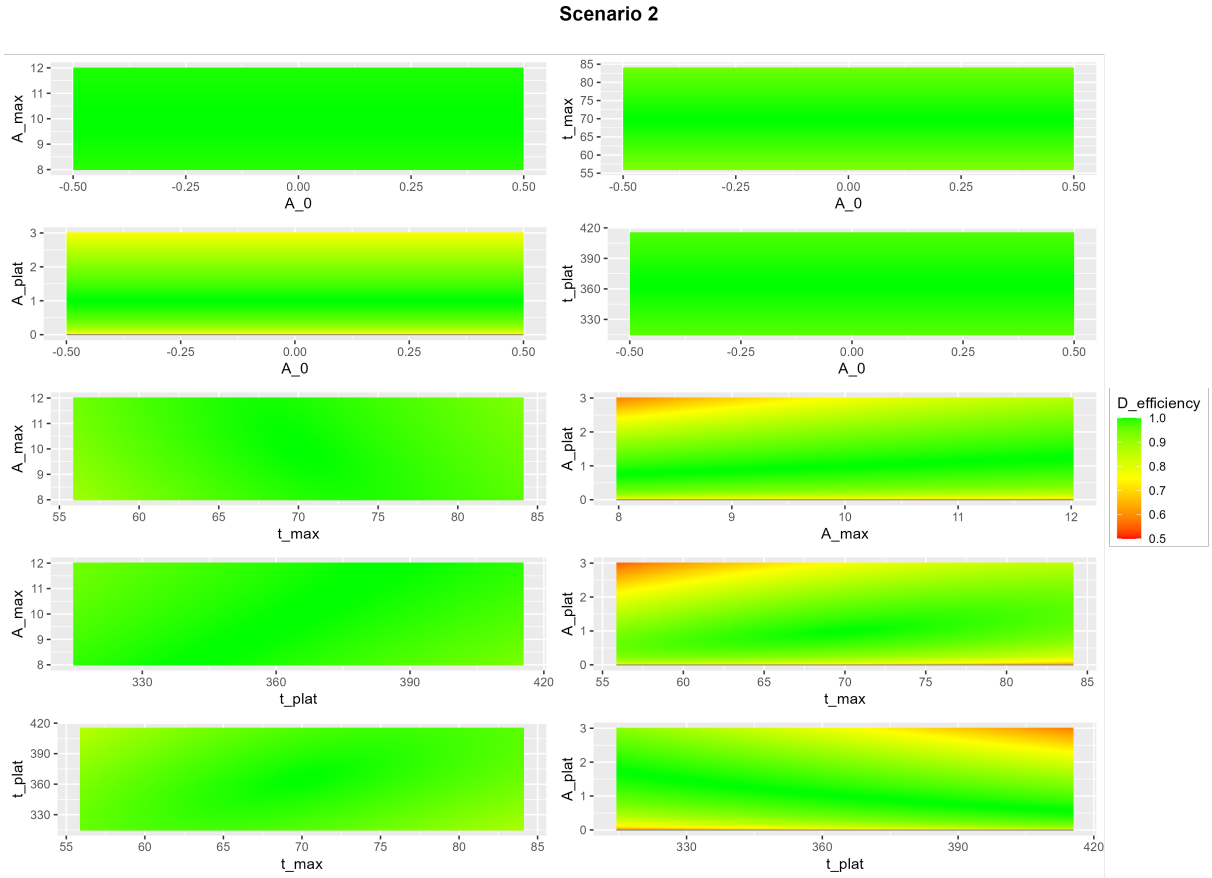

Figure E.18.: Graphical representation of the impact of simultaneous misspecification in two initial information on the D-efficiency of the respective design for Scenario 2. The color gradient represents the D-efficiency, with green indicating higher efficiency and red representing lower efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey areas denote settings where no solution could be determined.

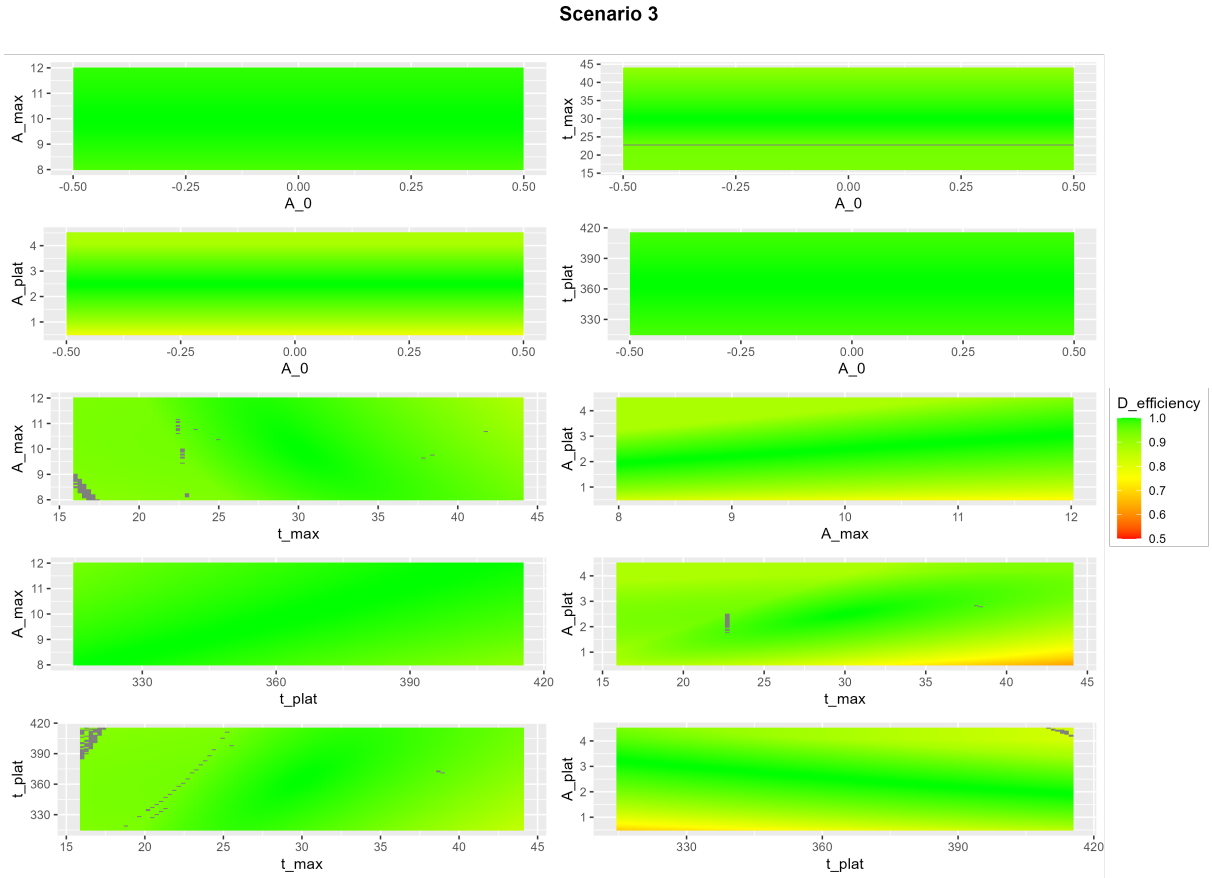

Figure E.19.: Graphical representation of the impact of simultaneous misspecification in two initial information on the D-efficiency of the respective design for Scenario 3. The color gradient represents the D-efficiency, with green indicating higher efficiency and red representing lower efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey areas denote settings where no solution could be determined.

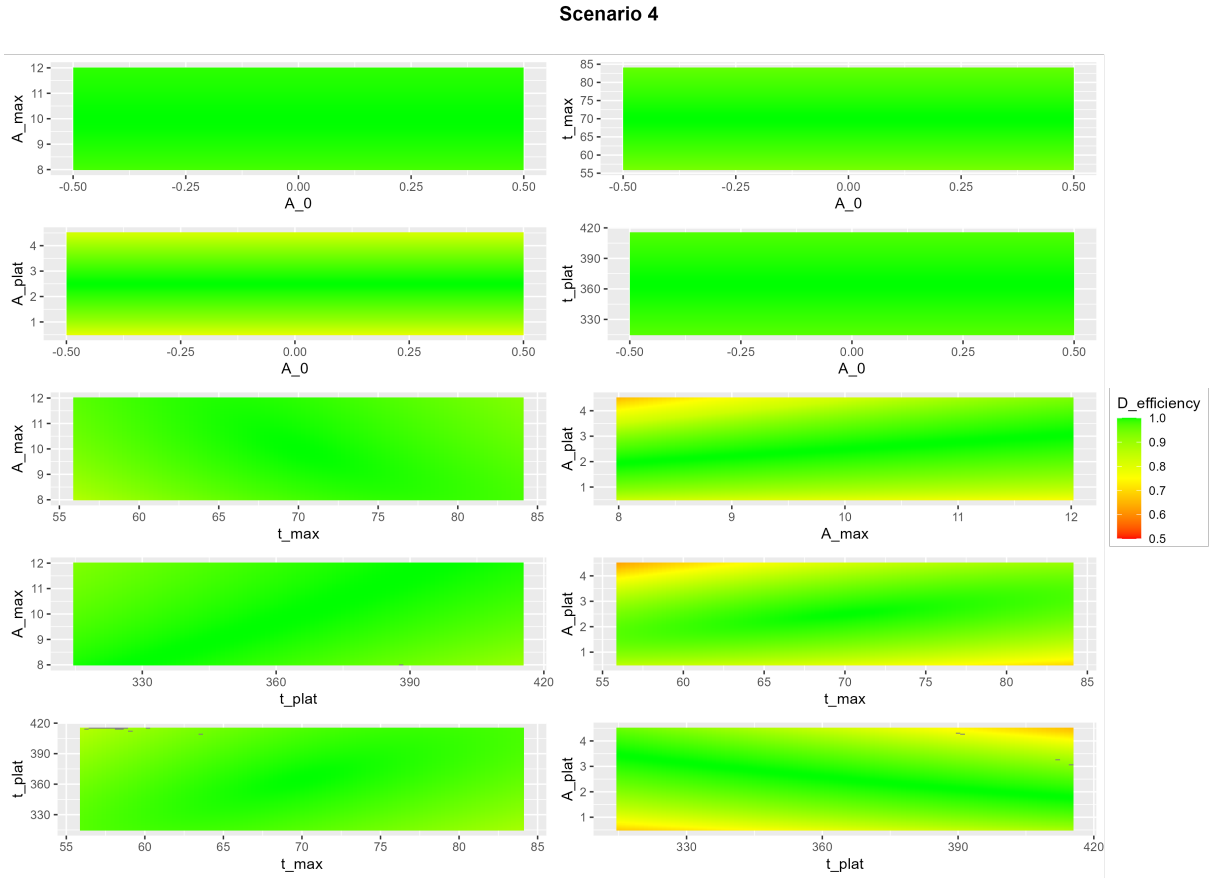

Figure E.20.: Graphical representation of the impact of simultaneous misspecification in two initial information on the D-efficiency of the respective design for Scenario 4. The color gradient represents the D-efficiency, with green indicating higher efficiency and red representing lower efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey areas denote settings where no solution could be determined.

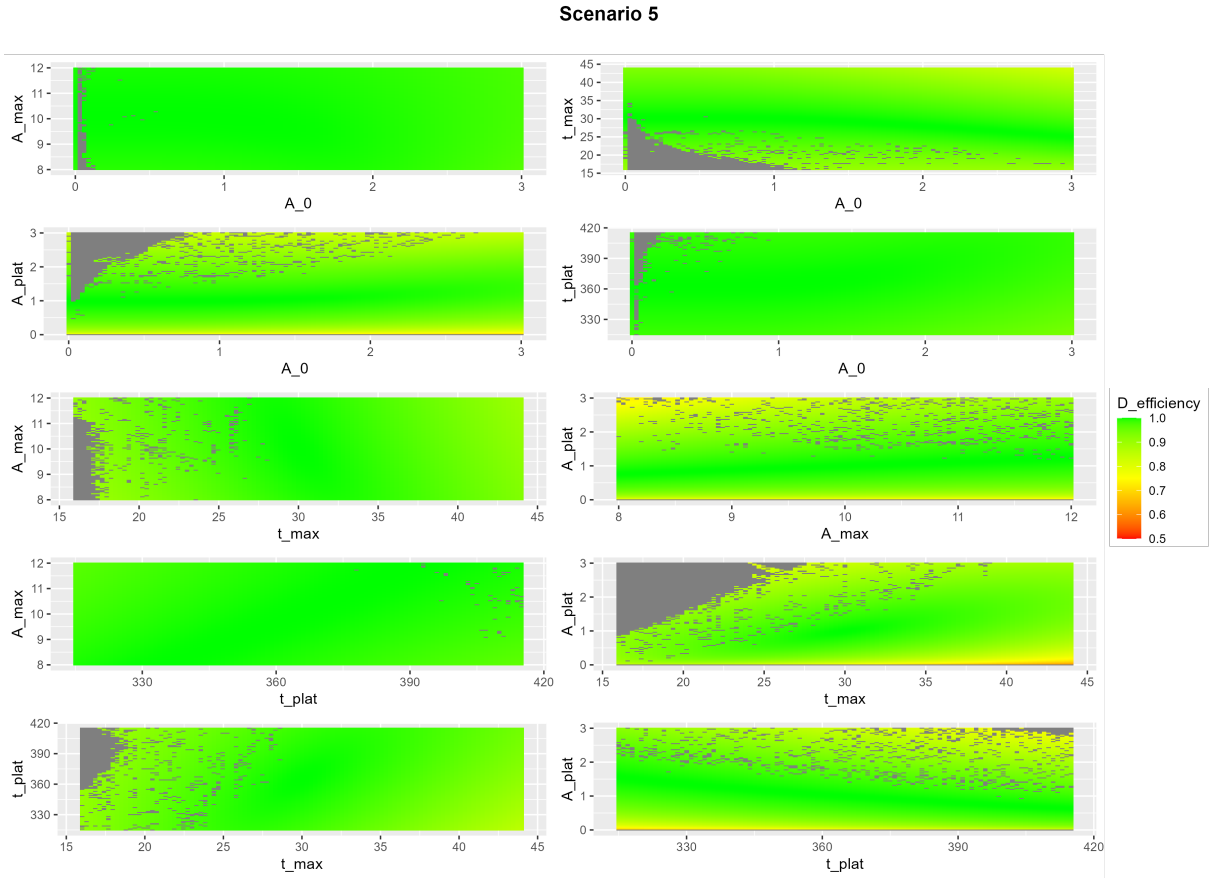

Figure E.21.: Graphical representation of the impact of simultaneous misspecification in two initial information on the D-efficiency of the respective design for Scenario 5. The color gradient represents the D-efficiency, with green indicating higher efficiency and red representing lower efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey areas denote settings where no solution could be determined.

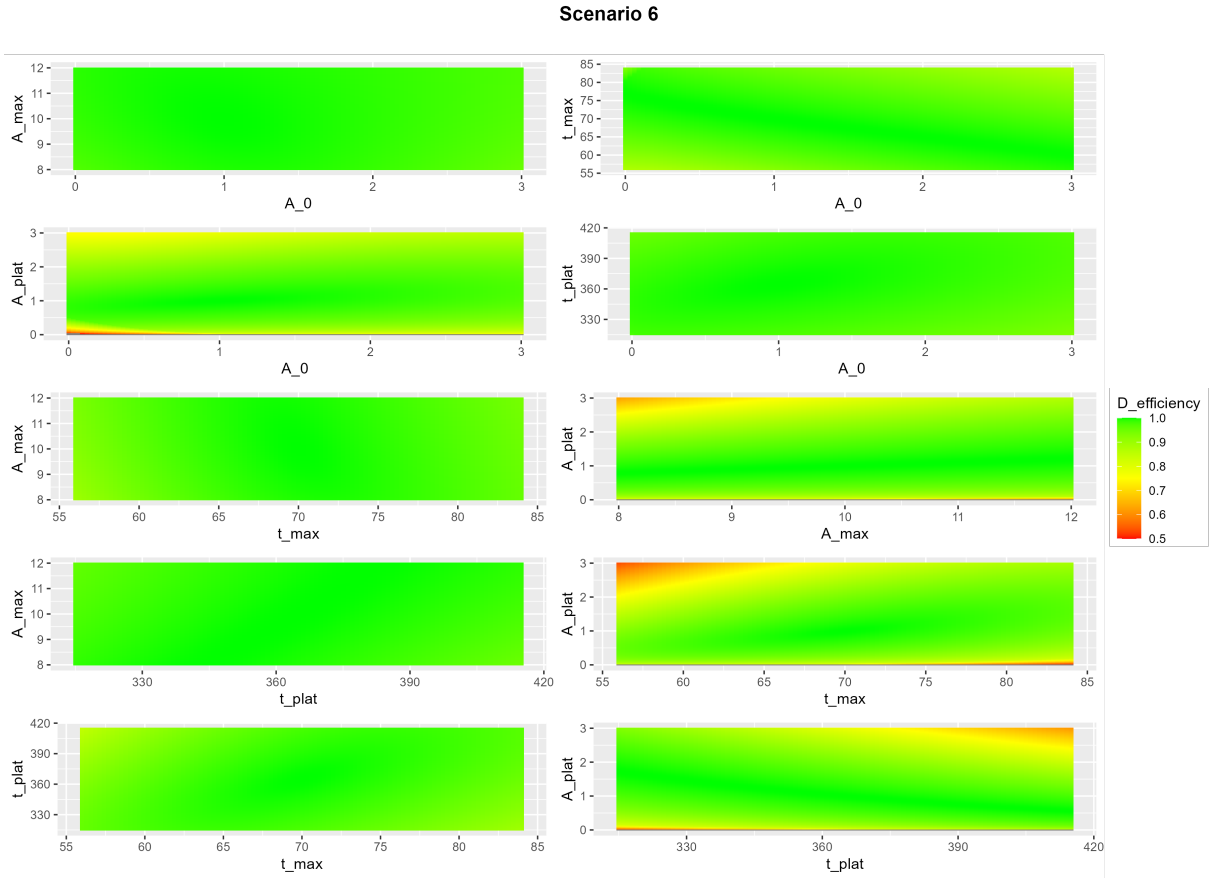

Figure E.22.: Graphical representation of the impact of simultaneous misspecification in two initial information on the D-efficiency of the respective design for Scenario 6. The color gradient represents the D-efficiency, with green indicating higher efficiency and red representing lower efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey areas denote settings where no solution could be determined.

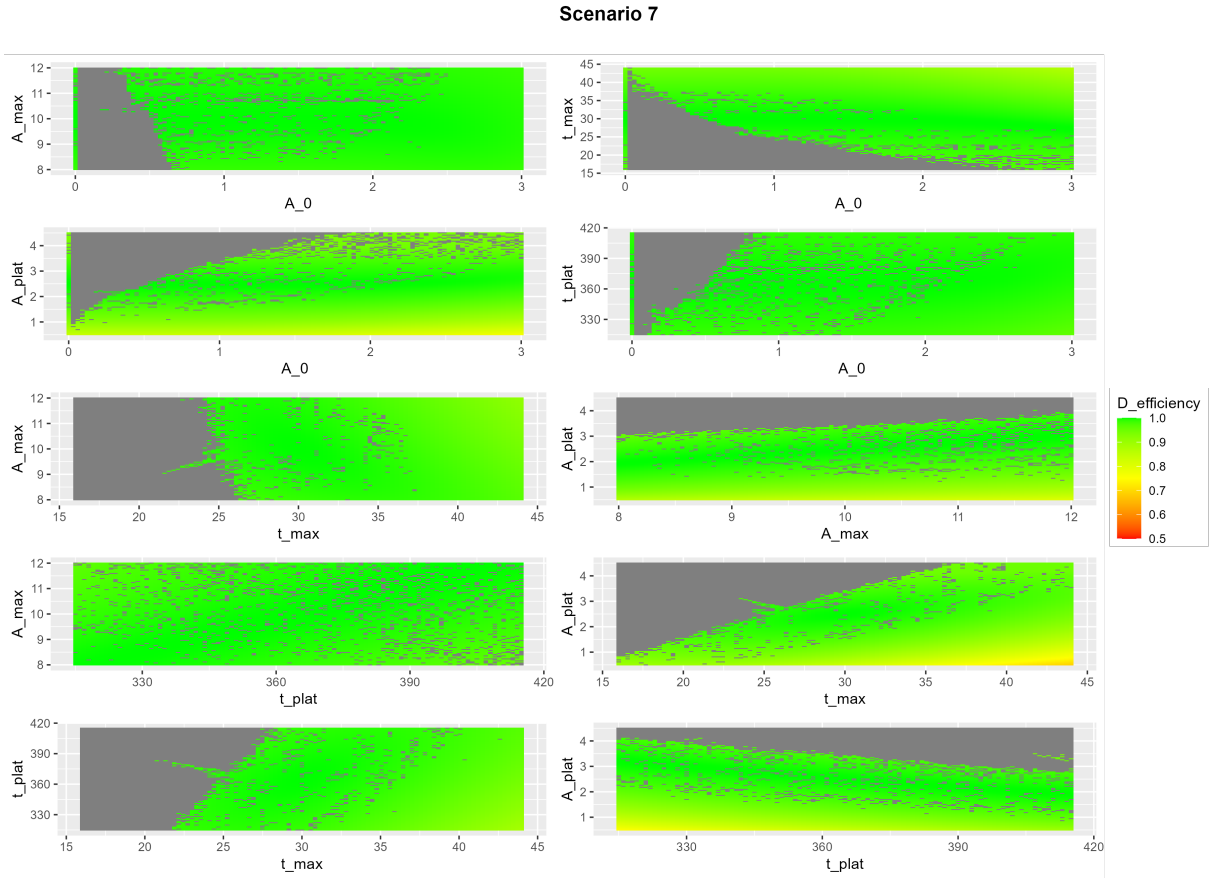

Figure E.23.: Graphical representation of the impact of simultaneous misspecification in two initial information on the D-efficiency of the respective design for Scenario 7. The color gradient represents the D-efficiency, with green indicating higher efficiency and red representing lower efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey areas denote settings where no solution could be determined.

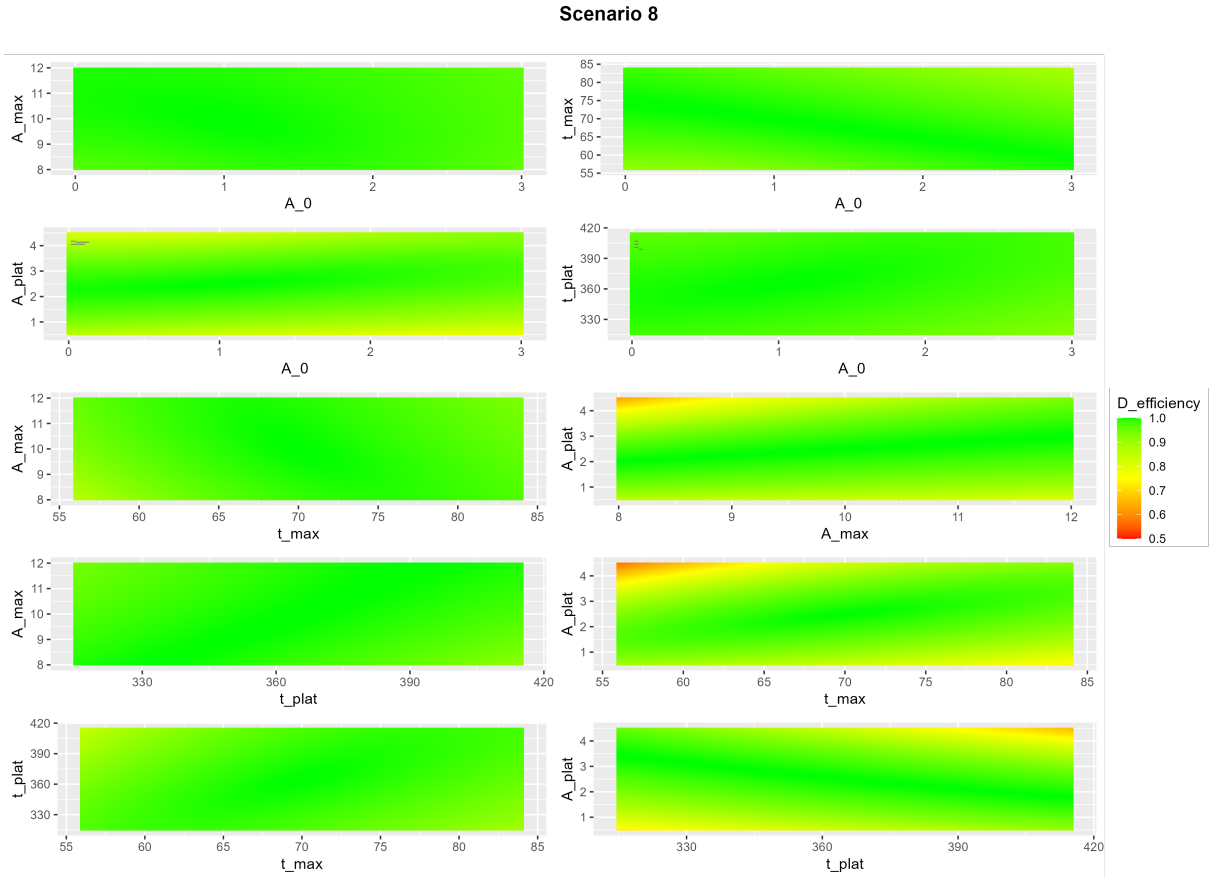

Figure E.24.: Graphical representation of the impact of simultaneous misspecification in two initial information on the D-efficiency of the respective design for Scenario 8. The color gradient represents the D-efficiency, with green indicating higher efficiency and red representing lower efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey areas denote settings where no solution could be determined.

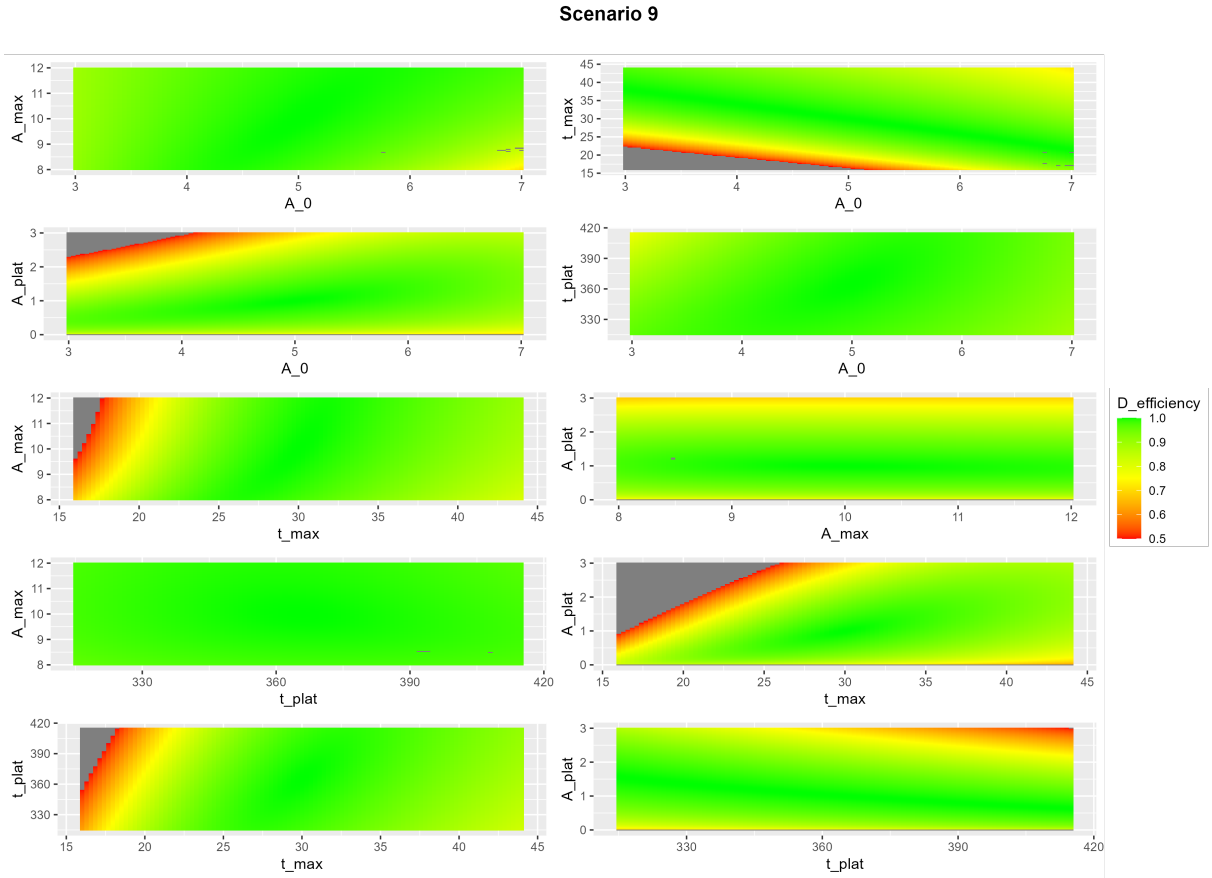

Figure E.25.: Graphical representation of the impact of simultaneous misspecification in two initial information on the D-efficiency of the respective design for Scenario 9. The color gradient represents the D-efficiency, with green indicating higher efficiency and red representing lower efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey areas denote settings where no solution could be determined.

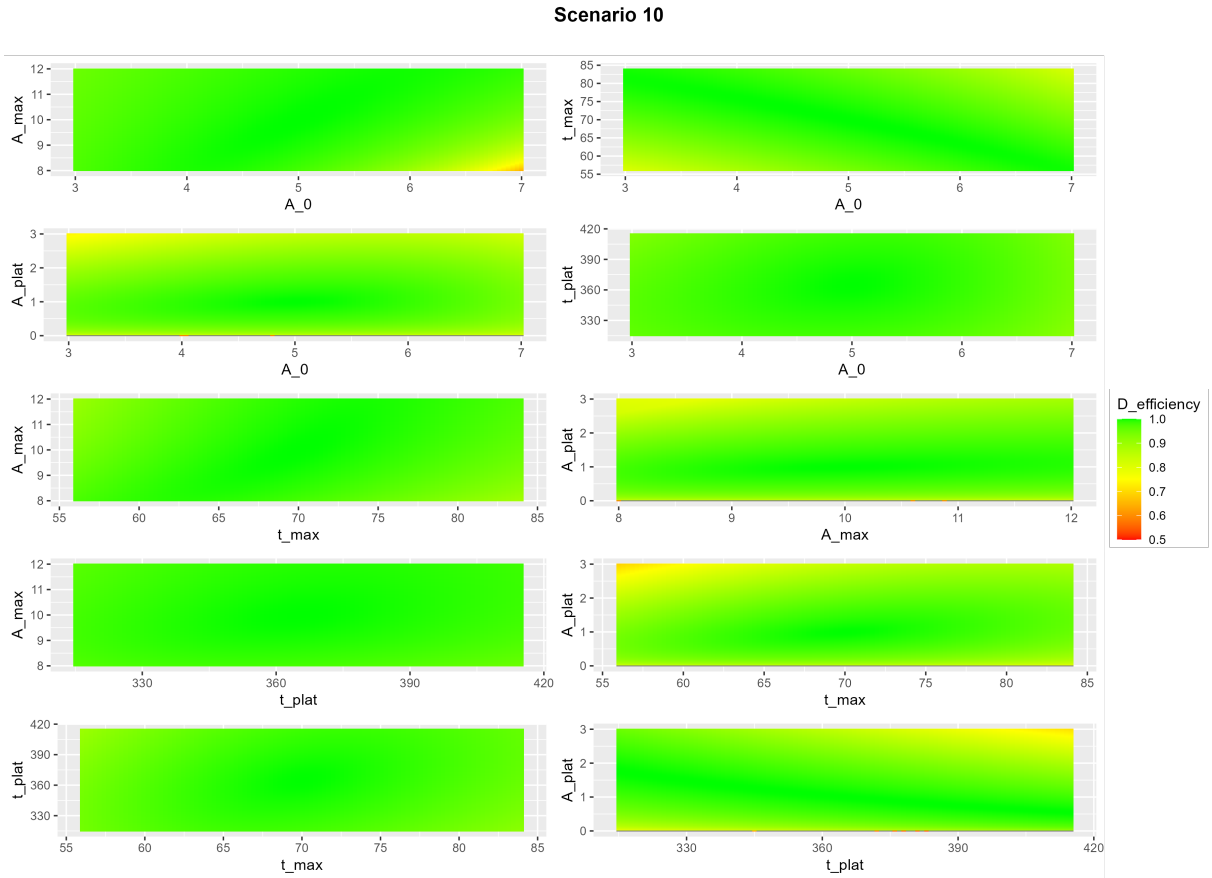

Figure E.26.: Graphical representation of the impact of simultaneous misspecification in two initial information on the D-efficiency of the respective design for Scenario 10. The color gradient represents the D-efficiency, with green indicating higher efficiency and red representing lower efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey areas denote settings where no solution could be determined.

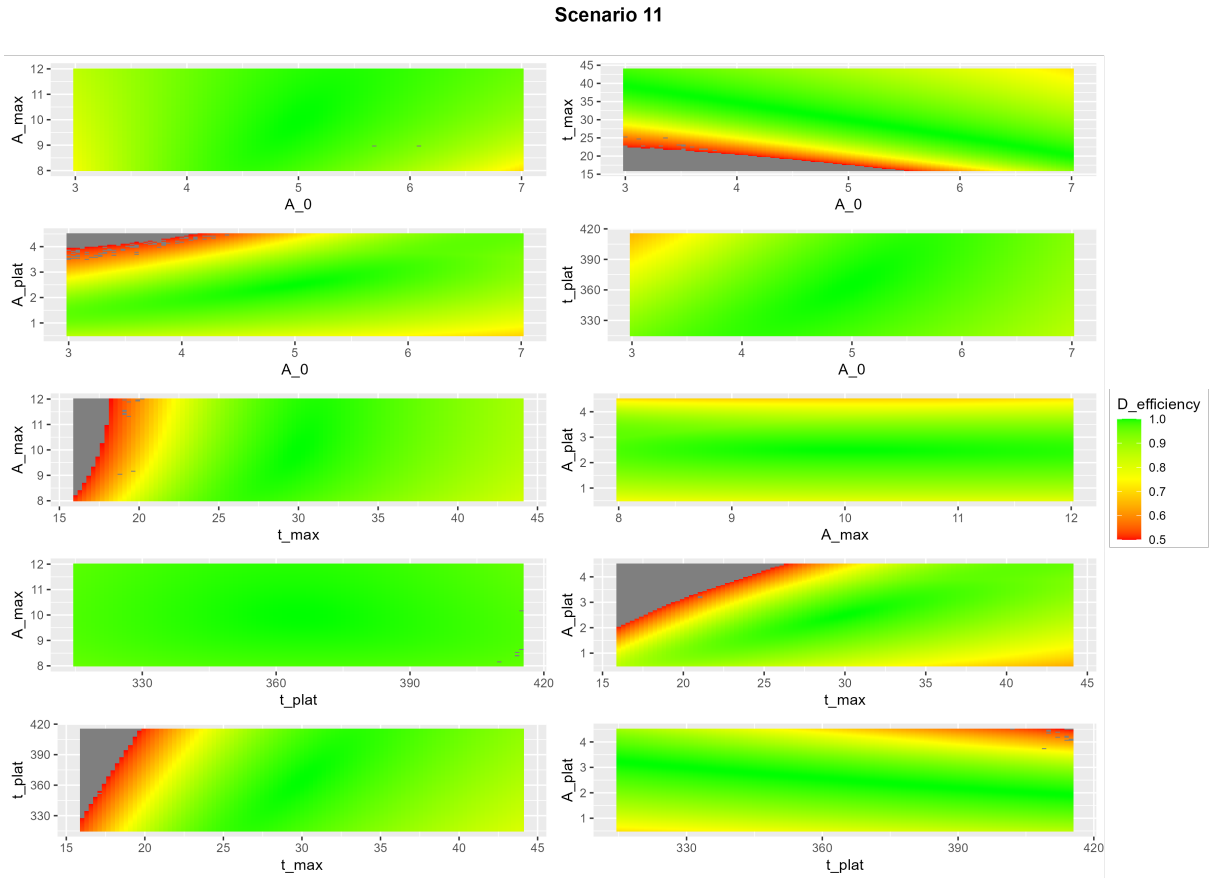

Figure E.27.: Graphical representation of the impact of simultaneous misspecification in two initial information on the D-efficiency of the respective design for Scenario 11. The color gradient represents the D-efficiency, with green indicating higher efficiency and red representing lower efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey areas denote settings where no solution could be determined.

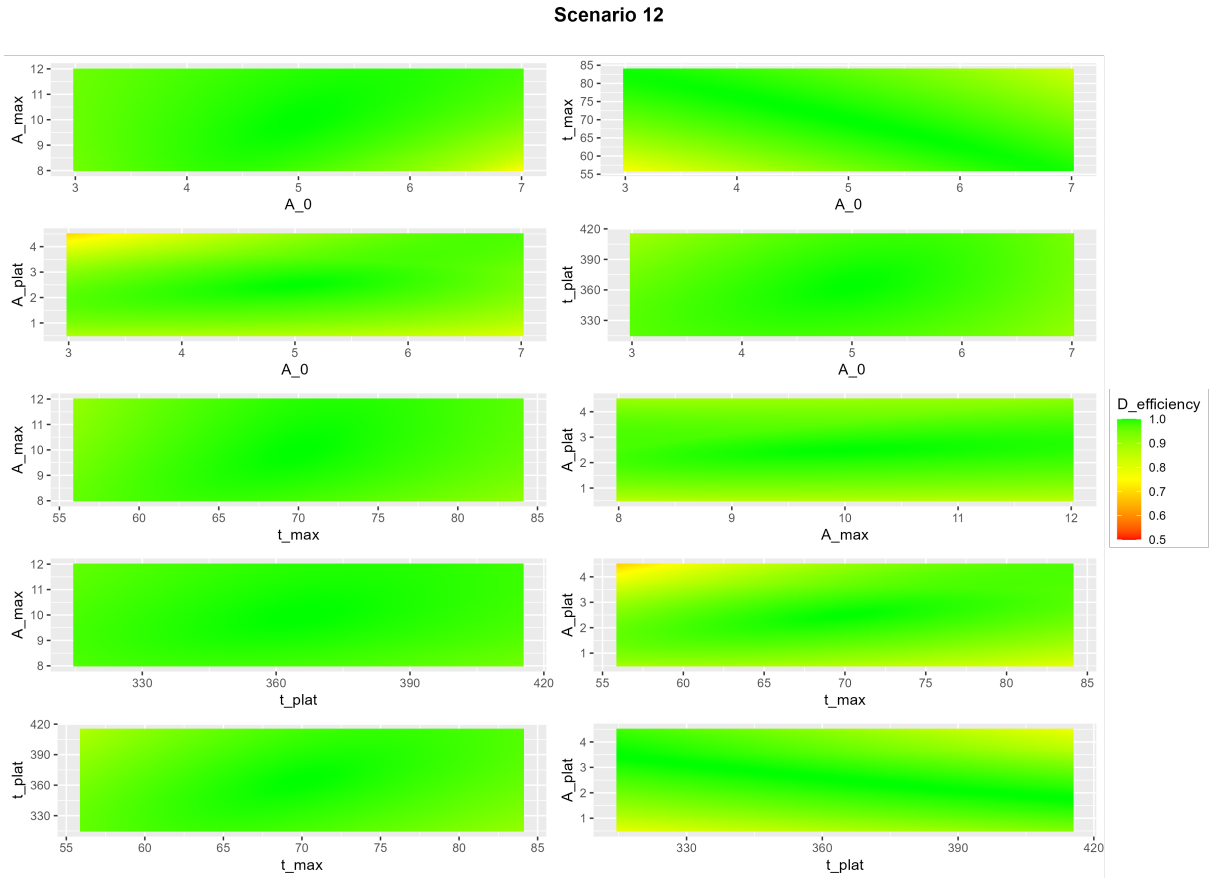

Figure E.28.: Graphical representation of the impact of simultaneous misspecification in two initial information on the D-efficiency of the respective design for Scenario 12. The color gradient represents the D-efficiency, with green indicating higher efficiency and red representing lower efficiency. The gradient is caped at 0.5, while all values below 0.5 show the same color (red). The grey areas denote settings where no solution could be determined.

Appendix F.

Manual for the Shiny Application

The following is the manual as published with the Shiny application on Zenodo.[123]

This manual is based on Version 1.0 of the Shiny Application 'Find Optimal Timing of Samples using Beta Distribution'. We are currently in active development and therefore provide a disclaimer that you might face different issues throughout the use of the application and invite you to report feedback, improvement ideas or bugs to stefan.embacher@medunigraz.at.

General remarks

This framework is only applicable if you expect an initial increase in antibody concentration, followed by a decay. We assume that the antibody concentration stabilizes at a plateau with no further decay, which is a modelling simplification used at the stage of study design. Further, the omission of a lag after exposure (we assume an immediate increase) or that sampling times are provided up to two decimal places are unrealistic assumptions.
The framework asks for five initial information and you need to provide all five, there is no possibility to leave one out. Our framework optimizes three or four sampling times, depending on your choice of the baseline concentration.
We believe that days are the most reasonable time scale, however, the framework is usable on other time scales as well. Further, the approach is not limited to specific antibodies, provided their kinetics follow the assumed functional form. Initial information can be informed by literature, from numerically reported values, derived from equations or extracted from figures, and should be critically evaluated against expert knowledge. If numerical issues arise, we recommend to slightly adjust the input parameters.
In the following, we provide a step-by-step guide of the input parameters, highlighting practical recommendations and key implications, where relevant.

Input

The panel on the left side of the Shiny Application is used to specify the input and consists of a total of 5 tabs, which are described step by step in more detail in the following.

1. Initial Information

The **'Initial Information'** tab requires five obligatory key inputs:

- **Baseline Concentration ($\mathbf{A_0}$)**
  This specifies the antibody concentration at time $t = 0$. Please be aware that the choice of $A_0 = 0$ implies 3 samples, while the choice of $A_0 \neq 0$ implies 4 samples. We recommend to set $A_0 = 0$ only in the case you are sure that you face naïve patients. Otherwise a non-zero A_0 is recommended.

- **Time of Maximum ($\mathbf{t_{max}}$)**
  Specifies when the maximum antibody concentration occurs.

- **Maximum Concentration ($\mathbf{A_{max}}$)**
  The (mean) maximum antibody concentration.

- **Plateau Concentration ($\mathbf{A_{plat}}$)**
  The mean antibody concentration when reaching a steady state. Choose a reasonable value, e.g., the lower limit of detection (LLOD), LLOD/2 or a specific value, where the decay becomes very slow. Based on the results of your robustness analysis, we recommend to choose a non-zero value. Note: This is a modelling simplification.

- **Time of the Plateau ($\mathbf{t_{plat}}$)**
  Specifies when the plateau is reached. The choice of t_{plat} has an effect on the time limits, since it specifies the default upper limit.

2. Enter Variability

The tab **'Enter Variability'** requires input on the variability. The variance-covariance matrix follows an AR(1) structure where:

- σ (standard deviation) and ρ (correlation) are assumed to be known.
- These values (for now) do not affect optimal sampling times.
- They do affect numerical stability and convergence.

3. Options for Optimal Design

In the tab **'Options for Optimal Design'** one can specify restrictions on the sampling times.

- Lower limit: Default is Day 0.
- Upper limit: Default is t_{plat}.
- Key considerations:
  - Ranges adjust based on the value of t_{plat}.
  - The used algorithm does not enforce ordered sampling times.
    - If restrictions are not properly set, time-points may switch. For example: If the first sampling time is between Day 14-28 and the second is unrestricted, results could be $t_1 = 28$, $t_2 = 0$. We highly recommend to either restrict no sampling time, or all sampling times in an increasing order.
    - If $A_0 = 0$, restrictions on the fourth time-window have no effect.
    - If starting values fall outside the set limits, the starting values are adjusted to the closest allowed value.

4. Plot Options

Regarding the tab **'Plot Options'**: If the checkboxes are ticked, the limits set apply to the displayed plot. There are two checkboxes, one for the x-axis and one for the y-axis.

5. Default Settings

In the tab **'Default Settings'** there are currently two optional settings: The first one is the display length of the plateau, which is called 'x-axis tolerance'. If the x-axis limits are set manually this value is ignored. The second one specifies the extra space shown above the y-axis maximum, unless the limits are specified by the user. The y-axis tolerance is the proportion of the maximum additionally display.

Output

The output section of the Shiny application consists of three parts/panels.

- The first part provides a visualisation of the curve derived from the assumptions made through the input and the resulting beta distribution, while the resulting optimal sampling times are displayed as the dashed red lines.

- The second part highlights the key input parameters and displays the resulting optimal sampling times numerically. Additionally, the user can copy the input and the results to the clipboard for further use. In addition to the displayed values, all others, including input on variability, time restrictions and the resulting beta parameters, are also copied to ensure reproducibility.
- The third part contains a comprehensive text summary of all relevant information, formatted for convenient sharing. Users can copy this summary for collaboration with colleagues or for direct inclusion in publications, ensuring transparency and reproducibility of the results.

Appendix G.

Code

G.1. Solve for Beta Parameters

```
###############
#
# Code for simulations
#
# Define Functions to solve for beta parameters
#
###############

# load needed packages

library(nleqslv)
library(tidyverse)

####################################
#define functions ----
####################################

# Function to solve for beta parameters

#-------------------------------------------------#
  ### Function where we expect antibodies to start in 0
#-------------------------------------------------#

  #better to provide good starting values
  solve_beta <- function(t_max, A_max, t_plat, A_plat, x_start =
     c(1.5,6,1800)){
```

```
#First define system of equations based on input
beta_systemofequations <- function(x){
  # x[1] is alpha
  # x[2] is beta
  # x[3] is t_obs

  # For derivation see material (using values and property of
      modus)
  y <- numeric(3)
  y[1] <- (x[1]-1)/(x[1]+x[2]-2) - t_max/x[3]
  y[2] <- (1/beta(x[1],x[2]))*(t_max/x[3])^(x[1]-1)*(1- t_max
     /x[3])^(x[2]-1) - A_max
  y[3] <- (1/beta(x[1],x[2]))*(t_plat/x[3])^(x[1]-1)*(1- t_
     plat/x[3])^(x[2]-1) - A_plat
  y
}

#now solve system
result <- nleqslv(x = x_start,
                  fn = beta_systemofequations,
                  method = "Broyden",
                  global = "dbldog",
                  xscalm = "auto",
                  control=list(maxit = 1000, allowSingular =
                     TRUE))

#check if function converged
#check if function converged
if(result$termcd != 1){
   print("NLEQSLV did not converge with Broyden, use Newton!"
      )
   print(result$message)

  result_N <- nleqslv(x = x_start,
                      fn = beta_systemofequations,
                      method = "Newton",
                      global = "dbldog",
                      xscalm = "auto",
                      control=list(maxit = 1000,
                         allowSingular = TRUE))
```

```
  #check if function converged
  if(result_N$termcd != 1){
    #  print("NLEQSLV did not converge with Newton as well!")
    #  print(result_N$message)

    solution <- data.frame("alpha" = NA,
                           "beta" = NA,
                           "t_obs" = NA,
                           "converged_solving" = 0)
  }

  if(result_N$termcd == 1){
    #prepare output
    alpha <- result_N$x[1]
    beta <- result_N$x[2]
    t_obs <- result_N$x[3]

    solution <- data.frame("alpha" = alpha,
                           "beta" = beta,
                           "t_obs" = t_obs,
                           "converged_solving" = 1)
  }
}

# If function converged assign values
if(result$termcd == 1){
#prepare output
alpha <- result$x[1]
beta <- result$x[2]
t_obs <- result$x[3]

solution <- data.frame("alpha" = alpha,
                       "beta" = beta,
                       "t_obs" = t_obs,
                       "converged_solving" = 1)
}
# output

return(solution)
```

```
}

####------------------------------------------------####
### Function where we want the antibodies to not start in zero
####------------------------------------------------####

solve_beta_notzero <- function(A_0, t_max, A_max, t_plat, A_
   plat, x_start = c(1.5,6,1800,10)){

  #First define system of equations based on input
  beta_systemofequations <- function(x){
    # x[1] is alpha
    # x[2] is beta
    # x[3] is t_obs
    # x[4] is c

    # For derivation see material (using values and property of
        modus)
    y <- numeric(4)
    y[1] <- (x[1]-1)/(x[1]+x[2]-2) - (t_max + x[4])/x[3]
    y[2] <- (1/beta(x[1],x[2]))*(x[4]/x[3])^(x[1]-1)*(1-(x[4]/x
       [3]))^(x[2]-1) - A_0
    y[3] <- (1/beta(x[1],x[2]))*((t_max + x[4])/x[3])^(x[1]-1)*
       (1- (t_max + x[4])/x[3])^(x[2]-1) - A_max
    y[4] <- (1/beta(x[1],x[2]))*((t_plat + x[4])/x[3])^(x[1]-1)
       *(1- ((t_plat + x[4])/x[3]))^(x[2]-1) - A_plat
    y
  }

  #now solve system
  #now solve system
  result <- nleqslv(x = x_start,
                    fn = beta_systemofequations,
                    method = "Broyden",
                    global = "dbldog",
                    xscalm = "auto",
                    control=list(maxit = 1000, allowSingular =
                       TRUE))
```

```
#check if function converged
if(result$termcd != 1){
  print("NLEQSLV did not converge with Broyden, use Newton!")
  print(result$message)

  result_N <- nleqslv(x = x_start,
                      fn = beta_systemofequations,
                      method = "Newton",
                      global = "dbldog",
                      xscalm = "auto",
                      control=list(maxit = 1000,
                         allowSingular = TRUE))

  if(result_N$termcd != 1){
      print("NLEQSLV did not converge with Newton as well!")
      print(result_N$message)

    solution <- data.frame("alpha" = NA,
                           "beta" = NA,
                           "t_obs" = NA,
                           "constant" = NA,
                           "converged_solving" = 0)
  }

  if(result_N$termcd == 1){
    #prepare output
    alpha <- result_N$x[1]
    beta <- result_N$x[2]
    t_obs <- result_N$x[3]
    constant <- result_N$x[4]

    solution <- data.frame("alpha" = alpha,
                           "beta" = beta,
                           "t_obs" = t_obs,
                           "constant" = constant,
                           "converged_solving" = 1)
  }

}
```

```
    # If function converged assign values
    if(result$termcd == 1){
      #prepare output
      alpha <- result$x[1]
      beta <- result$x[2]
      t_obs <- result$x[3]
      constant <- result$x[4]

      solution <- data.frame("alpha" = alpha,
                             "beta" = beta,
                             "t_obs" = t_obs,
                             "constant" = constant,
                             "converged_solving" = 1)
    }


    # output

    return(solution)

  }

####-------------------------------------------------------------#
#Function to determine the parameters of the beta distribution
#Depending on whether we start in zero or not
####-------------------------------------------------------------#

beta_parameters <- function(A_0_input, t_max_input, A_max_input,
   t_plat_input, A_plat_input){

  #check if starting in zero
  if(A_0_input == 0){
    #solve equations
    para <- solve_beta(t_max = t_max_input,
                       A_max = A_max_input,
                       t_plat = t_plat_input,
                       A_plat = A_plat_input)
```

```
  #set parameters
  dt_tmp <- data.frame(A_0 =A_0_input,
                       t_max = t_max_input,
                       A_max = A_max_input,
                       t_plat = t_plat_input,
                       A_plat = A_plat_input,
                       alpha = para$alpha,
                       beta_value = para$beta,
                       t_obs = para$t_obs,
                       constant = 0,
                       converged_solving = para$converged_
                          solving) #constant is zero if we start
                           in 0


}

#check if starting in zero
if(A_0_input != 0){

  #solve equations
  para <- solve_beta_notzero(A_0 = A_0_input,
                             t_max = t_max_input,
                             A_max = A_max_input,
                             t_plat = t_plat_input,
                             A_plat = A_plat_input)

  #set parameters
  dt_tmp <- data.frame(
    A_0 = A_0_input,
    t_max = t_max_input,
    A_max = A_max_input,
    t_plat = t_plat_input,
    A_plat = A_plat_input,
    alpha = para$alpha,
    beta_value = para$beta,
    t_obs = para$t_obs,
    constant = para$constant,
    converged_solving = para$converged_solving
  )
```

```
  }

  #If converged check parameters
  if(dt_tmp$converged_solving == 1){
    #check if alpha > 1 and beta > 1
    if(dt_tmp$alpha <= 1 || dt_tmp$beta_value <= 1 || dt_tmp$t_
       obs < t_plat_input || dt_tmp$constant < 0){
      print("WARNING: Assumptions not fullfilled")
      dt_tmp$beta_parameters_valid <- 0
    }

    if(dt_tmp$alpha > 1 && dt_tmp$beta_value > 1 && dt_tmp$t_obs
       > t_plat_input && dt_tmp$constant >= 0){
      dt_tmp$beta_parameters_valid <- 1
    }
  }

  #If not converged add NA to data.frame for dimensionality
  if(dt_tmp$converged_solving == 0){
    dt_tmp$beta_parameters_valid <- NA
  }
  #print(dt_tmp)
  return(dt_tmp) #give this data frame as return

}
```

G.2. Solve for Optimal Times

```
###############
#
# Code for simulations
#
# Define Functions to solve for optimal design
#
###############

# load needed packages
library(optimx)
library(matlib)
#----------------------------------------#
# Define functions ----
#----------------------------------------#
####-----------------------#
# derivatives of mu
####-----------------------#

dmu_derivatives <- function(t, starting_zero = TRUE){

  if(starting_zero == TRUE){

    #derivatives of mu if we start in zero, ie. c=0
    dmuda <- ((1-t/t_obs)^(beta_value-1)*(t/t_obs)^(alpha-1)* log
       (t/t_obs)-(1-t/t_obs)^(beta_value-1)*(t/t_obs)^(alpha-1)*(
       digamma(alpha)-digamma(alpha+beta_value)))/beta(alpha,beta_
       value)
    dmudb <- ((1-t/t_obs)^(beta_value-1)*(t/t_obs)^(alpha-1)* log
       (1-t/t_obs)-(1-t/t_obs)^(beta_value-1)*(t/t_obs)^(alpha-1)*
       (digamma(beta_value)-digamma(alpha+beta_value)))/beta(alpha
       ,beta_value)
    dmudtobs <- ((alpha-1)*t*(1-t/t_obs)^(beta_value-1)*(t/t_obs)
       ^(alpha-2)+(beta_value-1)*t*(1-t/t_obs)^(beta_value-2)*(t/t
       _obs)^(alpha-1))/(t_obs^2*beta(alpha,beta_value))

    res <- c(dmuda,dmudb,dmudtobs)
  }
```

```
  if(starting_zero == FALSE){
    #derivatives of mu if we do not start in zero, which means we
        have an additional derivative
    # dmudc is derivative of mu after the constant
    # dmuda is derivative of mu after alpha
    # dmudb is the derivative of mu after beta
    # dmudtobs is the derviative after the scaling factor

    dmudc <- ((alpha-1)*(1-((t+constant)/t_obs))^(beta_value-1)*
       ((t+constant)/t_obs)^(alpha-2)-(beta_value-1)*(1-((t+
       constant)/t_obs))^(beta_value-2)*((t+constant)/t_obs)^(
       alpha-1))/(beta(alpha,beta_value)*t_obs)
    dmuda <- ((1-(constant +t)/t_obs)^(beta_value-1)*((constant +
       t)/t_obs)^(alpha-1)* log((constant +t)/t_obs)-(1-(constant
       +t)/t_obs)^(beta_value-1)*((constant +t)/t_obs)^(alpha-1)*(
       digamma(alpha)-digamma(alpha+beta_value)))/beta(alpha,beta_
       value)
    dmudb <- ((1-(constant +t)/t_obs)^(beta_value-1)*((constant +
       t)/t_obs)^(alpha-1)* log(1-(constant +t)/t_obs)-(1-(
       constant +t)/t_obs)^(beta_value-1)*((constant +t)/t_obs)^(
       alpha-1)*(digamma(beta_value)-digamma(alpha+beta_value)))/
       beta(alpha,beta_value)
    dmudtobs <- ((alpha-1)*(constant +t)*(1-(constant +t)/t_obs)
       ^(beta_value-1)*((constant +t)/t_obs)^(alpha-2)+(beta_value
       -1)*(constant +t)*(1-(constant +t)/t_obs)^(beta_value-2)*((
       constant +t)/t_obs)^(alpha-1))/(t_obs^2*beta(alpha,beta_
       value))

    res <- c(dmudc,dmuda,dmudb,dmudtobs)
  }

  return(res)
}

#----------------------------------------#
# Function for sigma inverse ----
#----------------------------------------#
```

```
#start with very simple case
# Sigma_inverse <- function(s, starting_zero = TRUE){
#
#   if(starting_zero == TRUE){
#     Sigma_inverse_return <- matrix(data = c(1/s^2,0,0,0,1/s
   ^2,0,0,0,1/s^2),nrow=3,ncol= 3)
#   }
#
#   if(starting_zero == FALSE){
#     Sigma_inverse_return <- matrix(data = c(1/s^2,0,0,0,0,1/s
   ^2,0,0,0,0,1/s^2,0,0,0,0,1/s^2),nrow=4,ncol= 4)
#   }
#
#   return(Sigma_inverse_return)
# }

# Function to compute the analytical inverse of AR(1) covariance
   matrix
ar1_inverse_analytical <- function(n, rho) {
  # Precompute the scaling factor
  scaling_factor <- 1 / (1 - rho^2)

  # Initialize the inverse matrix as a zero matrix
  Sigma_inv <- matrix(0, n, n)

  # Fill in the tridiagonal structure
  Sigma_inv[1,1] <- scaling_factor
  Sigma_inv[n,n] <- scaling_factor
  if (n > 1) {
    diag(Sigma_inv[-1, ]) <- -rho * scaling_factor  # Lower
       diagonal
    diag(Sigma_inv[, -1]) <- -rho * scaling_factor  # Upper
       diagonal
  }

  # Adjust the second-to-last rows and columns for the full
     tridiagonal structure
  for (i in 2:(n-1)) {
    Sigma_inv[i, i] <- (1 + rho^2) * scaling_factor  # Diagonal
       correction
```

```
  }

  return(Sigma_inv)
}

#extension to AR(1) covariance structure
Sigma_inverse <- function(s, rho, starting_zero = TRUE){


  if(starting_zero == TRUE){
     diags <- matrix(data = c(1/s^2,0,0,0,1/s^2,0,0,0,1/s^2),nrow
        =3,ncol= 3)
     Sigma_inverse_return <- ar1_inverse_analytical(n=3,rho) %*%
        diags
  }

  if(starting_zero == FALSE){
    diags <- matrix(data = c(1/s^2,0,0,0,0,1/s^2,0,0,0,0,1/s
       ^2,0,0,0,0,1/s^2),nrow=4,ncol= 4)
    Sigma_inverse_return <- ar1_inverse_analytical(n=4,rho) %*%
       diags
  }

  return(Sigma_inverse_return)


}

####---------------------------------#
# Function to calculate FIM/optimality criteria
#---------------------------------#

#Use -ln(x+1) to make sure we fulfil convexity property and do
   not have problems with zero
FIM_beta_knownvar <- function(t,starting_zero = TRUE){

  if(starting_zero == TRUE){
    #t is vector of 3 times
    t1 <- t[1]
    t2 <- t[2]
```

```
  t3 <- t[3]
  #H matrix
  H <- matrix(data = c(dmu_derivatives(t1, starting_zero = TRUE
     ),
                       dmu_derivatives(t2, starting_zero = TRUE
                          ),
                       dmu_derivatives(t3, starting_zero = TRUE
                          )), nrow = 3, ncol = 3)
  FIM <- t(H) %*% Sigma_inverse(s, rho, starting_zero = TRUE) %
     *% H

  #Use -ln(x+1) to make sure we fulfil convexity property and
     do not have problems with zero
  res <- -log(det(FIM)+1)

}

if(starting_zero == FALSE){
  # We want four time points
  t1 <- t[1]
  t2 <- t[2]
  t3 <- t[3]
  t4 <- t[4]

  #Then get  the H matrix (see own notes)
  H <- matrix(data = c(dmu_derivatives(t1, starting_zero =
     FALSE),
                       dmu_derivatives(t2, starting_zero =
                          FALSE),
                       dmu_derivatives(t3, starting_zero =
                          FALSE),
                       dmu_derivatives(t4, starting_zero =
                          FALSE)), nrow = 4, ncol = 4)
  FIM <- t(H) %*% Sigma_inverse(s, rho, starting_zero = FALSE)
     %*% H

  #Use -ln(x+1) to make sure we fulfil convexity property and
     do not have problems with zero
  res <- -log(det(FIM)+1)
```

```
  }
  return(res)
}

####----------------------------------------#
# Function to find optimal time points ----
#----------------------------------------#

#function to find optimal design
#for now ignore limits, just set upper limit t_plat
FIM_to_optimaldesign <- function(funct,starting_zero = TRUE){

  if(starting_zero == TRUE){

    optim_sol <- hjn(par = c(1,t_max,t_plat), #choosen as
       reasonable startuing times
          fn = funct,
          starting_zero = TRUE,
          lower = c(0.001,0.001,0.001),
          upper = c(t_plat,t_plat,t_plat), #Upper limit is where
             plateau is expected
          control=list(maxfeval = 100000,
                       stepsize = 1.5))
  }

  if(starting_zero == FALSE){
    optim_sol <- hjn(par = c(1,t_max,t_max + (t_plat - t_max)/2,
       t_plat),
          fn = funct,
          starting_zero = FALSE,
          lower = c(0.001,0.001,0.001,0.001),
          upper = c(t_plat,t_plat,t_plat,t_plat), #Upper limit is
              where plateau is expected
          control=list(maxfeval = 100000,
                       stepsize = 1.5))

  }
  return(optim_sol)
}
```

```
####--------------------------------------#
# data frame for simulations ----
#--------------------------------------#

solve_optimal_design <- function(starting_zero_input,s, rho){

  dt_tmp <- FIM_to_optimaldesign(funct = FIM_beta_knownvar,
                                 starting_zero = starting_zero_
                                    input)

  solution <- data.frame("standard_deviation" = s,
                         "rho" = rho,
                         "starting_in_zero" = starting_zero_input
                            ,
                         "time1" = sort(dt_tmp$par)[1],
                         "time2" = sort(dt_tmp$par)[2],
                         "time3" = sort(dt_tmp$par)[3],
                         "time4" = sort(dt_tmp$par)[4],
                         "converged_optim" = ifelse(dt_tmp$
                            convergence == 0,
                                                    1,
                                                    0))

  return(solution)
  }

#-----------------------------------------------------#
# Function for D-efficiency ----
#-----------------------------------------------------#

#need determinants not -ln
FIM_for_eff_knownvar <- function(t,starting_zero = TRUE){

  if(starting_zero == TRUE){
    #t is vector of 3 times
    t1 <- t[1]
    t2 <- t[2]
    t3 <- t[3]
    #H matrix
    H <- matrix(data = c(dmu_derivatives(t1, starting_zero = TRUE
```

```
      ),
                        dmu_derivatives(t2, starting_zero = TRUE
                           ),
                        dmu_derivatives(t3, starting_zero = TRUE
                           )), nrow = 3, ncol = 3)
    FIM <- t(H) %*% Sigma_inverse(s, rho, starting_zero = TRUE) %
       *% H

    #Use -ln(x+1) to make sure we fulfil convexity property and
       do not have problems with zero
    res <- det(FIM)

  }

  if(starting_zero == FALSE){
    # We want four time points
    t1 <- t[1]
    t2 <- t[2]
    t3 <- t[3]
    t4 <- t[4]

    #Then get  the H matrix (see own notes)
    H <- matrix(data = c(dmu_derivatives(t1, starting_zero =
       FALSE),
                        dmu_derivatives(t2, starting_zero =
                           FALSE),
                        dmu_derivatives(t3, starting_zero =
                           FALSE),
                        dmu_derivatives(t4, starting_zero =
                           FALSE)), nrow = 4, ncol = 4)
    FIM <- t(H) %*% Sigma_inverse(s, rho, starting_zero = FALSE)
       %*% H

    #Use -ln(x+1) to make sure we fulfil convexity property and
       do not have problems with zero
    res <- det(FIM)

  }
  return(res)
}
```

```
#need determinants for both designs and p denotes the number of
   parameters, i.e. the number of timepoints in our case
D_eff <- function(det_selected_design,det_optimal_design, p){
  eff <- (det_selected_design/det_optimal_design)^(1/p)
  return(eff)
}
```

G.3. Create Scenarios

```
###############
#
# Code for simulations
#
# Define Scenaria
#
###############

rm(list=ls())

# Start with defining scenaria

### set values
#fast increase
t_max_fast <- 30
A_max_fast <- 10

#slow increase
t_max_slow <- 70
A_max_slow <- 10

#low plateau
t_plat_low <- 365
A_plat_low <- 1 #0.5 resulted in convergence issues for scenario
   1

#high_plateau
t_plat_high <- 365
A_plat_high <- 2.5

#starting values A(0)
A_0_low <- 1
A_0_high <- 5


#Starting in zero
scenario1 <- data.frame(A_0 = 0,
```

```
                        t_max = t_max_fast,
                        A_max = A_max_fast,
                        t_plat = t_plat_low,
                        A_plat = A_plat_low,
                        description = "Start␣in␣zero,␣faster␣
                           increase,␣lower␣plateau")

scenario2 <- data.frame(A_0 = 0,
                        t_max = t_max_slow,
                        A_max = A_max_slow,
                        t_plat = t_plat_low,
                        A_plat = A_plat_low,
                        description = "Start␣in␣zero,␣slower␣
                           increase,␣lower␣plateau")

scenario3 <- data.frame(A_0 = 0,
                        t_max = t_max_fast,
                        A_max = A_max_fast,
                        t_plat = t_plat_high,
                        A_plat = A_plat_high,
                        description = "Start␣in␣zero,␣faster␣
                           increase,␣higher␣plateau")

scenario4 <- data.frame(A_0 = 0,
                        t_max = t_max_slow,
                        A_max = A_max_slow,
                        t_plat = t_plat_high,
                        A_plat = A_plat_high,
                        description = "Start␣in␣zero,␣slower␣
                           increase,␣higher␣plateau")

#Starting low
scenario5 <- data.frame(A_0 = A_0_low,
                        t_max = t_max_fast,
                        A_max = A_max_fast,
                        t_plat = t_plat_low,
                        A_plat = A_plat_low,
                        description = "Starting␣low,␣faster␣
                           increase,␣lower␣plateau")
```

```
scenario6 <- data.frame(A_0 = A_0_low,
                        t_max = t_max_slow,
                        A_max = A_max_slow,
                        t_plat = t_plat_low,
                        A_plat = A_plat_low,
                        description = "Starting␣low,␣slower␣
                           increase,␣lower␣plateau")

scenario7 <- data.frame(A_0 = A_0_low,
                        t_max = t_max_fast,
                        A_max = A_max_fast,
                        t_plat = t_plat_high,
                        A_plat = A_plat_high,
                        description = "Starting␣low,␣faster␣
                           increase,␣higher␣plateau")

scenario8 <- data.frame(A_0 = A_0_low,
                        t_max = t_max_slow,
                        A_max = A_max_slow,
                        t_plat = t_plat_high,
                        A_plat = A_plat_high,
                        description = "Starting␣low,␣slower␣
                           increase,␣higher␣plateau")

#Starting high
scenario9 <- data.frame(A_0 = A_0_high,
                        t_max = t_max_fast,
                        A_max = A_max_fast,
                        t_plat = t_plat_low,
                        A_plat = A_plat_low,
                        description = "Starting␣high,␣faster␣
                           increase,␣lower␣plateau")

scenario10 <- data.frame(A_0 = A_0_high,
                        t_max = t_max_slow,
                        A_max = A_max_slow,
                        t_plat = t_plat_low,
                        A_plat = A_plat_low,
                        description = "Starting␣high,␣slower␣
                           increase,␣lower␣plateau")
```

```
scenario11 <- data.frame(A_0 = A_0_high,
                         t_max = t_max_fast,
                         A_max = A_max_fast,
                         t_plat = t_plat_high,
                         A_plat = A_plat_high,
                         description = "Starting␣high,␣faster␣
                            increase,␣higher␣plateau")

scenario12 <- data.frame(A_0 = A_0_high,
                         t_max = t_max_slow,
                         A_max = A_max_slow,
                         t_plat = t_plat_high,
                         A_plat = A_plat_high,
                         description = "Starting␣high,␣slower␣
                            increase,␣higher␣plateau")

########################################
# put scenaria in list (might be useful)

scenaria <- list(scenario1,
                 scenario2,
                 scenario3,
                 scenario4,
                 scenario5,
                 scenario6,
                 scenario7,
                 scenario8,
                 scenario9,
                 scenario10,
                 scenario11,
                 scenario12)

########################################-
# Save ----
########################################-

rm(A_0_high,
   A_0_low,
   t_max_fast,
```

```
    t_max_slow,
    t_plat_high,
    t_plat_low,
    A_plat_high,
    A_plat_low,
    A_max_fast,
    A_max_slow)

save(scenaria,
     file=paste0("R/Simulation/Rdata/scenaria",Sys.Date(),".RData
        "))
```

G.4. Robustness Analysis - Single Parameter Misspecification

```
###############
#
# Code for simulations
#
# Run Simulation for a single scenario and a single
   misspecification
#
##################

####------------------------#
# load functions ----
####------------------------#

source("R/Simulation/Rfile/00_Functions_solving_beta_20241016.R")
source("R/Simulation/Rfile/00_Functions_solving_optimaldesign_
   20250225.R")

#---------------------------#
# Define function ----
#---------------------------#

#Input: the selected scenario
#       the parameter we want to vary
#       N Feinheit des grids
simulate_scenario_single_para <- function(scenario_input,
   parameter_missspecify, N){

  #Give error if wrong parameter is set
  if(!(parameter_missspecify %in% c("A_0","A_max","A_plat","t_max
     ","t_plat"))){
    stop("Not␣an␣correct␣parameter␣misspecification")
  }

  #need to set environments
  environment(solve_optimal_design) <- environment()
  environment(FIM_to_optimaldesign) <- environment()
  environment(FIM_beta_knownvar) <- environment()
```

```
environment(dmu_derivatives) <- environment()
environment(FIM_for_eff_knownvar) <- environment()
environment(dmu_derivatives) <- environment()

####---------####
# Part 1: get "true" values
####---------####

#set standard deviation (assumed) to be known
s <- 0.25
rho <- 0.73

A_0_true <- scenario_input$A_0
t_max_true <- scenario_input$t_max
A_max_true <- scenario_input$A_max
t_plat_true <- scenario_input$t_plat
A_plat_true <- scenario_input$A_plat

#set if its starting in zero
starting_zero_input <- (A_0_true == 0)

#solve beta
beta_values_true <- beta_parameters(A_0_input = A_0_true,
                                    t_max_input = t_max_true,
                                    A_max_input = A_max_true,
                                    t_plat_input = t_plat_true,
                                    A_plat_input = A_plat_true)

#set values for functions
alpha <- beta_values_true$alpha
beta_value <- beta_values_true$beta_value
t_obs <- beta_values_true$t_obs
constant <- beta_values_true$constant

#solve optimal design
A_0 <- A_0_true
A_max <- A_max_true
t_max <- scenario_input$t_max
t_plat <- t_plat_true
A_plat <- A_plat_true
```

```
optimal_design_values_true <- solve_optimal_design(starting_
   zero_input,s, rho)

#data frame with all necessary values
vals_true_scenario <- cbind(beta_values_true, optimal_design_
   values_true)


if(is.na(vals_true_scenario$time4)){
  determinant_true_scenario <- FIM_for_eff_knownvar(
  c(
  vals_true_scenario$time1,
  vals_true_scenario$time2,
  vals_true_scenario$time3
  ),starting_zero = TRUE)
}

if(!is.na(vals_true_scenario$time4)){
  determinant_true_scenario <- FIM_for_eff_knownvar(
  c(
  vals_true_scenario$time1,
  vals_true_scenario$time2,
  vals_true_scenario$time3,
  vals_true_scenario$time4
  ),starting_zero = FALSE)
}
print("Optimal␣Scenario␣done")
####-------------####
# Part 2: Specify missspecification
####-------------####

if(parameter_missspecify == "A_0"){
  if(A_0_true != 0){
    A_0_seq <- seq(max(0,A_0_true-2), A_0_true+2, length=N)
  }

  if(A_0_true == 0){
    A_0_seq <- rep(A_0_true,N)
  }
```

```
}

if(parameter_missspecify == "A_max"){
  A_max_seq <- seq(A_max_true - 2, A_max_true + 2 , length= N)
}

if(parameter_missspecify == "A_plat"){
  A_plat_seq <- seq(max(0,A_plat_true-2),A_plat_true +2, length
      = N)
}

if(parameter_missspecify == "t_max"){
  t_max_seq <- seq(t_max_true - 14, t_max_true + 14, length=N)
}

if(parameter_missspecify == "t_plat"){
  t_plat_seq <- seq(t_plat_true - 50 , t_plat_true + 50, length
      = N)
}

####--------------------####
# Simulate ----
####--------------------####

solution <- c()

if(parameter_missspecify == "A_0"){
  # Fix all except one
  t_max <- t_max_true
  A_max <- A_max_true
  t_plat <- t_plat_true
  A_plat <- A_plat_true

  for(i in 1:N){
    print(paste("Iteration",i,"of",N))

    #set varying parameter
    A_0 <- A_0_seq[i]

    #calculate beta values
```

```
  beta_values_tmp <- beta_parameters(A_0_input = A_0,
                                     t_max_input = t_max,
                                     A_max_input = A_max,
                                     t_plat_input = t_plat,
                                     A_plat_input = A_plat)

  #dataframe if optim does not solve for optimal design, set
     convergence to 0
  solution_tmp <- cbind(beta_values_tmp,
                        data.frame("standard_deviation" = s,
                                   "rho" = rho,
                                   "starting_in_zero" =
                                      starting_zero_input,
                                   "time1" = NA,
                                   "time2" = NA,
                                   "time3" = NA,
                                   "time4" = NA,
                                   "converged_optim" = 0))

  tryCatch({
    #set values for functions
    alpha <- beta_values_tmp$alpha
    beta_value <- beta_values_tmp$beta_value
    t_obs <- beta_values_tmp$t_obs
    constant <- beta_values_tmp$constant


    #solve optimal design
    optimal_design_values <- solve_optimal_design(starting_
       zero_input,s, rho)

    #data frame with all necessary values
    solution_tmp<- cbind(beta_values_tmp, optimal_design_
       values)
  },  error=function(e){print("optimal␣design␣error")})

  solution <- rbind(solution,solution_tmp)
}

}
```

```
if(parameter_missspecify == "A_max"){
  # Fix all except one
  A_0 <- A_0_true
  t_max <- t_max_true
  t_plat <- t_plat_true
  A_plat <- A_plat_true

  for(i in 1:N){
    print(paste("Iteration",i,"of",N))


    #set varying parameter
    A_max <- A_max_seq[i]

    #calculate beta values
    beta_values_tmp <- beta_parameters(A_0_input = A_0,
                                       t_max_input = t_max,
                                       A_max_input = A_max,
                                       t_plat_input = t_plat,
                                       A_plat_input = A_plat)

    #dataframe if optim does not solve for optimal design, set
       convergence to 0
    solution_tmp <- cbind(beta_values_tmp,
                          data.frame("standard_deviation" = s,
                                     "rho" = rho,
                                     "starting_in_zero" =
                                        starting_zero_input,
                                     "time1" = NA,
                                     "time2" = NA,
                                     "time3" = NA,
                                     "time4" = NA,
                                     "converged_optim" = 0))

    tryCatch({
      #set values for functions
      alpha <- beta_values_tmp$alpha
      beta_value <- beta_values_tmp$beta_value
```

```
      t_obs <- beta_values_tmp$t_obs
      constant <- beta_values_tmp$constant


      #solve optimal design
      optimal_design_values <- solve_optimal_design(starting_
         zero_input,s, rho)

      #data frame with all necessary values
      solution_tmp<- cbind(beta_values_tmp, optimal_design_
         values)
    }, error=function(e){print("optimal design error")})

    solution <- rbind(solution,solution_tmp)
  }
}

if(parameter_missspecify == "A_plat"){
  # Fix all except one
  A_0 <- A_0_true
  t_max <- t_max_true
  A_max <- A_max_true
  t_plat <- t_plat_true

  for(i in 1:N){
    print(paste("Iteration",i,"of",N))

    #set varying parameter
    A_plat <- A_plat_seq[i]

    #calculate beta values
    beta_values_tmp <- beta_parameters(A_0_input = A_0,
                                       t_max_input = t_max,
                                       A_max_input = A_max,
                                       t_plat_input = t_plat,
                                       A_plat_input = A_plat)

    #dataframe if optim does not solve for optimal design, set
       convergence to 0
    solution_tmp <- cbind(beta_values_tmp,
```

```
                              data.frame("standard_deviation" = s,
                                         "rho" = rho,
                                         "starting_in_zero" =
                                            starting_zero_input,
                                         "time1" = NA,
                                         "time2" = NA,
                                         "time3" = NA,
                                         "time4" = NA,
                                         "converged_optim" = 0))

    tryCatch({
      #set values for functions
      alpha <- beta_values_tmp$alpha
      beta_value <- beta_values_tmp$beta_value
      t_obs <- beta_values_tmp$t_obs
      constant <- beta_values_tmp$constant


      #solve optimal design
      optimal_design_values <- solve_optimal_design(starting_
         zero_input,s, rho)

      #data frame with all necessary values
      solution_tmp<- cbind(beta_values_tmp, optimal_design_
         values)
    },  error=function(e){print("optimal␣design␣error")})

    solution <- rbind(solution,solution_tmp)
  }
}

if(parameter_missspecify == "t_max"){
  # Fix all except one
  A_0 <- A_0_true
  A_max <- A_max_true
  t_plat <- t_plat_true
  A_plat <- A_plat_true

  for(i in 1:N){
    print(paste("Iteration",i,"of",N))
```

```
#set varying parameter
t_max <- t_max_seq[i]

#calculate beta values
beta_values_tmp <- beta_parameters(A_0_input = A_0,
                                   t_max_input = t_max,
                                   A_max_input = A_max,
                                   t_plat_input = t_plat,
                                   A_plat_input = A_plat)

#dataframe if optim does not solve for optimal design, set
   convergence to 0
solution_tmp <- cbind(beta_values_tmp,
                      data.frame("standard_deviation" = s,
                                 "rho" = rho,
                                 "starting_in_zero" =
                                    starting_zero_input,
                                 "time1" = NA,
                                 "time2" = NA,
                                 "time3" = NA,
                                 "time4" = NA,
                                 "converged_optim" = 0))

tryCatch({
  #set values for functions
  alpha <- beta_values_tmp$alpha
  beta_value <- beta_values_tmp$beta_value
  t_obs <- beta_values_tmp$t_obs
  constant <- beta_values_tmp$constant


  #solve optimal design
  optimal_design_values <- solve_optimal_design(starting_
     zero_input,s, rho)

  #data frame with all necessary values
  solution_tmp<- cbind(beta_values_tmp, optimal_design_
     values)
},  error=function(e){print("optimal␣design␣error")})
```

```
    solution <- rbind(solution,solution_tmp)
  }
}

if(parameter_missspecify == "t_plat"){
  # Fix all except one
  A_0 <- A_0_true
  t_max <- t_max_true
  A_max <- A_max_true
  A_plat <- A_plat_true

  for(i in 1:N){
    print(paste("Iteration",i,"of",N))

    #set varying parameter
    t_plat <- t_plat_seq[i]

    #calculate beta values
    beta_values_tmp <- beta_parameters(A_0_input = A_0,
                                       t_max_input = t_max,
                                       A_max_input = A_max,
                                       t_plat_input = t_plat,
                                       A_plat_input = A_plat)

    #dataframe if optim does not solve for optimal design, set
       convergence to 0
    solution_tmp <- cbind(beta_values_tmp,
                          data.frame("standard_deviation" = s,
                                     "rho" = rho,
                                     "starting_in_zero" =
                                        starting_zero_input,
                                     "time1" = NA,
                                     "time2" = NA,
                                     "time3" = NA,
                                     "time4" = NA,
                                     "converged_optim" = 0))

    tryCatch({
      #set values for functions
```

```
      alpha <- beta_values_tmp$alpha
      beta_value <- beta_values_tmp$beta_value
      t_obs <- beta_values_tmp$t_obs
      constant <- beta_values_tmp$constant


      #solve optimal design
      optimal_design_values <- solve_optimal_design(starting_
         zero_input,s, rho)

      #data frame with all necessary values
      solution_tmp<- cbind(beta_values_tmp, optimal_design_
         values)
    }, error=function(e){print("optimal design error")})

    solution <- rbind(solution,solution_tmp)
  }

}

####----------------------####
# Get D_efficiency ----
####----------------------####

#reset alpha, beta etc. to true values
alpha <- beta_values_true$alpha
beta_value <- beta_values_true$beta_value
t_obs <- beta_values_true$t_obs
constant <- beta_values_true$constant

#initialize D_effieciency and determinant
D_efficiency <- c()
determinant_selected_design <- c()

#Cacluclate D_efficiency for all
for(i in 1:nrow(solution)){

  #If we have 3 time points use this
  if(is.na(solution[i,]$time4)){
    determinant_selected_design[i] <- FIM_for_eff_knownvar(
```

```
    c(
    solution[i,]$time1,
    solution[i,]$time2,
    solution[i,]$time3
    ),starting_zero = TRUE)

    p_set <- 3
  }

  #if we have 4 time points use this
  if(!is.na(solution[i,]$time4)){
    determinant_selected_design[i] <- FIM_for_eff_knownvar(
    c(
    solution[i,]$time1,
    solution[i,]$time2,
    solution[i,]$time3,
    solution[i,]$time4
    ),starting_zero = FALSE)

    p_set <- 4
  }

  #get D_effieciency
  D_eff_tmp <- D_eff(det_selected_design = determinant_selected
     _design[i],
                     det_optimal_design = determinant_true_
                        scenario,
                     p = p_set)

  #save d_efficiency
  D_efficiency[i] <- D_eff_tmp
}

#bind columns to get full output
simulation_result <- cbind(solution,determinant_selected_design
   ,D_efficiency)

####----------------####
# return results ----
####----------------####
```

```
  return(simulation_result)
}


##################
#
# File for simulation
#
##################

rm(list=ls())

load("R/Simulation/Rdata/scenaria2024-07-15.RData")

library(parallel)
library(foreach)

##################
#
#################

misspec_vars <- c("A_0","A_max","t_max","A_plat","t_plat")

N_set <- 1001

#######------------########
# Set up cluster for parallel ----
#######------------########

n.cores <- parallel::detectCores(logical = FALSE) - 1

#create the cluster
my.cluster <- parallel::makeCluster(
  n.cores,
  type = "PSOCK"
)

#check cluster definition (optional)
print(my.cluster)
```

```
#register it to be used by %dopar%
doParallel::registerDoParallel(cl = my.cluster)

#check if it is registered (optional)
foreach::getDoParRegistered()

#how many workers are available? (optional)
foreach::getDoParWorkers()

####--------------------------------####
# Run Simulations parallelized ----
####--------------------------------####

results <- foreach(i = 1:length(scenaria)) %dopar% {
  load("R/Simulation/Rdata/scenaria2024-07-15.RData")

  source("R/Simulation/Rfile/02_Simulation_single_scenario_
     singlemisspec_20250225.R")
  res_tmp <- list()
  for(j in 1:length(misspec_vars)) {
    # Do not use Scenario in name for easier accesability in
       presenting results
    #name_dataframe <- paste0("Scenario",i,"Misspec",misspec_vars
       [j])
    name_dataframe <- paste0("Misspec",misspec_vars[j])

    dt_tmp <- simulate_scenario_single_para(scenario_input =
       scenaria[[i]],
                                            parameter_missspecify
                                                = misspec_vars[[j
                                               ]],
                                            N = N_set)

    #assign(name_dataframe,dt_tmp)
    #res_tmp <- append(res_tmp,dt_tmp)
    res_tmp[[length(res_tmp)+1]] <- dt_tmp
    names(res_tmp)[length(res_tmp)] <- name_dataframe
  }
  return(res_tmp)
```

```
}

#### Close cluster again
parallel::stopCluster(cl = my.cluster)

#Use this to check if stopping worked
#Should result in: "description class mode text isopen can read
   can write"
showConnections()

####-----------------------------------------------####
# Save results ----
####-----------------------------------------------####

save(results,
     file = paste0("R/Simulation/Rdata/Results_Simulation",Sys.
        Date(),".Rdata"))
```

G.5. Robustness Analysis - Double Parameter Misspecification

```
###############
#
# Code for simulations
#
# Run Simulation for a single scenario and a double
   misspecification
#
##################


####------------------------#
# load functions ----
####------------------------#

source("R/Simulation/Rfile/00_Functions_solving_beta_20241016.R")
source("R/Simulation/Rfile/00_Functions_solving_optimaldesign_
   20250225.R")

#---------------------------#
# Define function ----
#---------------------------#

simulate_scenario_double_para <- function(scenario_input,
   parameter_missspecify, N){

  #Give error if not 2 parameters are chosen
  if(length(parameter_missspecify) != 2){
    stop("Length␣of␣misspecified␣parameters␣is␣not␣2")
  }

  #Give error if selected parameters are not named correctly
  for(i in 1:length(parameter_missspecify)){
    if(!(parameter_missspecify[i] %in% c("A_0","A_max","A_plat","
       t_max","t_plat"))){
      stop("Not␣an␣correct␣parameter␣misspecification,␣maybe␣
         spelling")
```

```
  }
}

#need to set environments
environment(solve_optimal_design) <- environment()
environment(FIM_to_optimaldesign) <- environment()
environment(FIM_beta_knownvar) <- environment()
environment(dmu_derivatives) <- environment()
environment(FIM_for_eff_knownvar) <- environment()
environment(dmu_derivatives) <- environment()

####---------####
# Part 1: get "true" values
####---------####

#set standard deviation (assumed) to be known
s <- 0.25
rho <- 0.73

A_0_true <- scenario_input$A_0
t_max_true <- scenario_input$t_max
A_max_true <- scenario_input$A_max
t_plat_true <- scenario_input$t_plat
A_plat_true <- scenario_input$A_plat

#set if its starting in zero
starting_zero_input <- (A_0_true == 0)

#solve beta
beta_values_true <- beta_parameters(A_0_input = A_0_true,
                                    t_max_input = t_max_true,
                                    A_max_input = A_max_true,
                                    t_plat_input = t_plat_true,
                                    A_plat_input = A_plat_true)

#set values for functions
alpha <- beta_values_true$alpha
beta_value <- beta_values_true$beta_value
t_obs <- beta_values_true$t_obs
constant <- beta_values_true$constant
```

```
#solve optimal design
A_0 <- A_0_true
A_max <- A_max_true
t_max <- t_max_true
t_plat <- t_plat_true
A_plat <- A_plat_true

optimal_design_values_true <- solve_optimal_design(starting_
   zero_input,s, rho)

#data frame with all necessary values
vals_true_scenario <- cbind(beta_values_true, optimal_design_
   values_true)


if(is.na(vals_true_scenario$time4)){
  determinant_true_scenario <- FIM_for_eff_knownvar(
  c(
  vals_true_scenario$time1,
  vals_true_scenario$time2,
  vals_true_scenario$time3
  ), starting_zero = TRUE)
}

if(!is.na(vals_true_scenario$time4)){
  determinant_true_scenario <- FIM_for_eff_knownvar(
  c(
  vals_true_scenario$time1,
  vals_true_scenario$time2,
  vals_true_scenario$time3,
  vals_true_scenario$time4
  ), starting_zero = FALSE)
}
print("Optimal Scenario done")

####-------------####
# Part 2: Specify missspecification
####-------------####
```

```
misspecified_values_1 <- c()
misspecified_values_2 <- c()

#set first misspecified parameter

if(parameter_missspecify[1] == "A_0"){
  if(A_0_true != 0){
    misspecified_values_1 <- seq(max(0,A_0_true-2), A_0_true+2,
        length=N)
  }

  if(A_0_true == 0){
    misspecified_values_1 <- rep(A_0_true,N)
  }
}

if(parameter_missspecify[1] == "A_max"){
  misspecified_values_1 <- seq(A_max_true - 2, A_max_true + 2 ,
      length= N)
}

if(parameter_missspecify[1] == "A_plat"){
  misspecified_values_1 <- seq(max(0,A_plat_true-2),A_plat_true
      +2, length = N)
}

if(parameter_missspecify[1] == "t_max"){
  misspecified_values_1 <- seq(t_max_true - 14, t_max_true +
     14, length=N)
}

if(parameter_missspecify[1] == "t_plat"){
  misspecified_values_1 <- seq(t_plat_true - 50 , t_plat_true +
      50, length = N)
}

#set second misspecified parameter

if(parameter_missspecify[2] == "A_0"){
  if(A_0_true != 0){
```

```
    misspecified_values_2 <- seq(max(0,A_0_true-2), A_0_true+2,
        length=N)
  }

  if(A_0_true == 0){
    misspecified_values_2 <- rep(A_0_true,N)
  }
}

if(parameter_missspecify[2] == "A_max"){
  misspecified_values_2 <- seq(A_max_true - 2, A_max_true + 2 ,
      length= N)
}

if(parameter_missspecify[2] == "A_plat"){
  misspecified_values_2 <- seq(max(0,A_plat_true-2),A_plat_true
      +2, length = N)
}

if(parameter_missspecify[2] == "t_max"){
  misspecified_values_2 <- seq(t_max_true - 14, t_max_true +
     14, length=N)
}

if(parameter_missspecify[2] == "t_plat"){
  misspecified_values_2 <- seq(t_plat_true - 50 , t_plat_true +
      50, length = N)
}

####--------------------####
# Simulate ----
####--------------------####

solution <- c()

  # Fix all first and replace in simulation step
  A_0 <- A_0_true
  t_max <- t_max_true
  A_max <- A_max_true
  t_plat <- t_plat_true
```

```
A_plat <- A_plat_true

#vary first parameter
for(i in 1:N){
  print(paste("Iteration",i,"of",N))

  #set varying parameter
  if(parameter_missspecify[1] == "A_0"){
    A_0 <- misspecified_values_1[i]
  }
  if(parameter_missspecify[1] == "A_max"){
    A_max <- misspecified_values_1[i]
  }
  if(parameter_missspecify[1] == "t_max"){
    t_max <- misspecified_values_1[i]
  }
  if(parameter_missspecify[1] == "A_plat"){
    A_plat <- misspecified_values_1[i]
  }
  if(parameter_missspecify[1] == "t_plat"){
    t_plat <- misspecified_values_1[i]
  }

  #vary second parameter
  for(j in 1:N){
    #set varying parameter
    if(parameter_missspecify[2] == "A_0"){
      A_0 <- misspecified_values_2[j]
    }
    if(parameter_missspecify[2] == "A_max"){
      A_max <- misspecified_values_2[j]
    }
    if(parameter_missspecify[2] == "t_max"){
      t_max <- misspecified_values_2[j]
    }
    if(parameter_missspecify[2] == "A_plat"){
      A_plat <- misspecified_values_2[j]
    }
    if(parameter_missspecify[2] == "t_plat"){
      t_plat <- misspecified_values_2[j]
```

```
}


#calculate beta values
beta_values_tmp <- beta_parameters(A_0_input = A_0,
                                   t_max_input = t_max,
                                   A_max_input = A_max,
                                   t_plat_input = t_plat,
                                   A_plat_input = A_plat)

#dataframe if optim does not solve for optimal design,
   set convergence to 0
solution_tmp <- cbind(beta_values_tmp,
                      data.frame("standard_deviation" = s
                         ,
                                 "rho" = rho,
                                 "starting_in_zero" =
                                    starting_zero_input,
                                 "time1" = NA,
                                 "time2" = NA,
                                 "time3" = NA,
                                 "time4" = NA,
                                 "converged_optim" = 0))

tryCatch({
  #set values for functions
  alpha <- beta_values_tmp$alpha
  beta_value <- beta_values_tmp$beta_value
  t_obs <- beta_values_tmp$t_obs
  constant <- beta_values_tmp$constant


  #solve optimal design
  optimal_design_values <- solve_optimal_design(starting_
     zero_input,s, rho)

  #data frame with all necessary values
  solution_tmp<- cbind(beta_values_tmp, optimal_design_
     values)
},  error=function(e){print("optimal␣design␣error")})
```

```
    solution <- rbind(solution,solution_tmp)
  }
}

####----------------------####
# Get D_efficiency ----
####----------------------####

#reset alpha, beta etc. to true values
alpha <- beta_values_true$alpha
beta_value <- beta_values_true$beta_value
t_obs <- beta_values_true$t_obs
constant <- beta_values_true$constant

#initialize D_effieciency and determinant
D_efficiency <- c()
determinant_selected_design <- c()

#Cacluclate D_efficiency for all
for(i in 1:nrow(solution)){

  #If we have 3 time points use this
  if(is.na(solution[i,]$time4)){
    determinant_selected_design[i] <- FIM_for_eff_knownvar(
    c(
    solution[i,]$time1,
    solution[i,]$time2,
    solution[i,]$time3
    ), starting_zero = TRUE)
    p_set <- 3
  }

  #if we have 4 time points use this
  if(!is.na(solution[i,]$time4)){
    determinant_selected_design[i] <- FIM_for_eff_knownvar(
    c(
    solution[i,]$time1,
    solution[i,]$time2,
    solution[i,]$time3,
```

```
        solution[i,]$time4
        ), starting_zero = FALSE)
        p_set <- 4
      }

      #get D_effieciency
      D_eff_tmp <- D_eff(det_selected_design = determinant_
         selected_design[i],
                         det_optimal_design = determinant_true_
                            scenario,
                         p = p_set)

      #save d_efficiency
      D_efficiency[i] <- D_eff_tmp
    }

    #bind columns to get full output
    simulation_result <- cbind(solution,determinant_selected_
       design,D_efficiency)

    ####----------------####
    # return results ----
    ####----------------####

    return(simulation_result)
}
```

Execute:

```
##################
#
# File for simulation for double misspecifications
#
##################

rm(list=ls())

load("R/Simulation/Rdata/scenaria2024-07-15.RData")

library(parallel)
```

```
library(foreach)

##################
#
#################

#Should be 10, beacuse Binkoeff(5,2) = 10
misspec_vars_pairs <- list(c("A_0","A_max"),
                           c("A_0","t_max"),
                           c("A_0","A_plat"),
                           c("A_0","t_plat"),
                           c("A_max","t_max"),
                           c("A_max","A_plat"),
                           c("A_max","t_plat"),
                           c("t_max","A_plat"),
                           c("t_max","t_plat"),
                           c("A_plat","t_plat"))


#need to reduce number of runs
N_set <- 101

#######-------------########
# Set up cluster for parallel ----
#######-------------########

n.cores <- parallel::detectCores(logical = FALSE) - 1

#create the cluster
my.cluster <- parallel::makeCluster(
  n.cores,
  type = "PSOCK"
)

#check cluster definition (optional)
print(my.cluster)

#register it to be used by %dopar%
doParallel::registerDoParallel(cl = my.cluster)
```

```
#check if it is registered (optional)
foreach::getDoParRegistered()

#how many workers are available? (optional)
foreach::getDoParWorkers()

####--------------------------------####
# Run Simulations parallelized ----
####--------------------------------####

start.time <- Sys.time()
results <- foreach(i = 1:length(scenaria)) %dopar% {
  load("R/Simulation/Rdata/scenaria2024-07-15.RData")

  source("R/Simulation/Rfile/05_Simulation_single_scenario_
     doublemisspec_20250313.R")
  res_tmp <- list()
  for(j in 1:length(misspec_vars_pairs)) {
    # Do not use Scenario in name for easier accesability in
       presenting results
    #name_dataframe <- paste0("Scenario",i,"Misspec",misspec_vars
       [j])
    name_dataframe <- paste0("Misspec",misspec_vars_pairs[[j
       ]][1],misspec_vars_pairs[[j]][2])

    dt_tmp <- simulate_scenario_double_para(scenario_input =
       scenaria[[i]],
                                            parameter_missspecify
                                                = misspec_vars_
                                               pairs[[j]],
                                            N = N_set)

    #assign(name_dataframe,dt_tmp)
    #res_tmp <- append(res_tmp,dt_tmp)
    res_tmp[[length(res_tmp)+1]] <- dt_tmp
    names(res_tmp)[length(res_tmp)] <- name_dataframe
  }
  return(res_tmp)
}
end.time <- Sys.time()
```

```
time.taken <- end.time - start.time
time.taken

#### Close cluster again
parallel::stopCluster(cl = my.cluster)

#Use this to check if stopping worked
#Should result in: "description class mode text isopen can read
   can write"
showConnections()

####-------------------------------------------####
# Save results ----
####-------------------------------------------####

save(results,
     file = paste0("R/Simulation/Rdata/Results_Simulation_double_
        misspec",Sys.Date(),".Rdata"))
```

G.6. Numerical Sensitivity of σ and ρ

```
#### make a mini robustness analysis to check how much of an
   effect variability and correlation have numerically
rm(list=ls())
#load functions

source("R/Simulation/Rfile/00_Functions_solving_beta_20241016.R")
source("R/Simulation/Rfile/00_Functions_solving_optimaldesign_
   20250225.R")

#set values for sigma and rho

sigma_vals <- seq(from = 0.2, to = 2, by=0.2)
rho_vals <- seq(from=0,to=0.9,by=0.2)

# Create all combinations
param_grid <- expand.grid(sigma = sigma_vals, rho = rho_vals)

#load scenarios
load("R/Simulation/Rdata/scenaria2024-07-15.RData")

#define function to vary variability
simulate_scenario_variability_single <- function(scenario_input,
   sigma_input, rho_input){

  #need to set environments
  environment(solve_optimal_design) <- environment()
  environment(FIM_to_optimaldesign) <- environment()
  environment(FIM_beta_knownvar) <- environment()
  environment(dmu_derivatives) <- environment()
  environment(FIM_for_eff_knownvar) <- environment()
  environment(dmu_derivatives) <- environment()

  ####---------####
  # Part 1: get "true" values
  ####---------####

  #set standard deviation (assumed) to be known
```

```
s <- sigma_input
rho <- rho_input

A_0_true <- scenario_input$A_0
t_max_true <- scenario_input$t_max
A_max_true <- scenario_input$A_max
t_plat_true <- scenario_input$t_plat
A_plat_true <- scenario_input$A_plat

#set if its starting in zero
starting_zero_input <- (A_0_true == 0)

#solve beta
beta_values_true <- beta_parameters(A_0_input = A_0_true,
                                    t_max_input = t_max_true,
                                    A_max_input = A_max_true,
                                    t_plat_input = t_plat_true,
                                    A_plat_input = A_plat_true)

#set values for functions
alpha <- beta_values_true$alpha
beta_value <- beta_values_true$beta_value
t_obs <- beta_values_true$t_obs
constant <- beta_values_true$constant

#solve optimal design
A_0 <- A_0_true
A_max <- A_max_true
t_max <- scenario_input$t_max
t_plat <- t_plat_true
A_plat <- A_plat_true

optimal_design_values_true <- solve_optimal_design(starting_
   zero_input,s, rho)

#data frame with all necessary values
vals_true_scenario <- cbind(beta_values_true, optimal_design_
   values_true)
```

```
  if(is.na(vals_true_scenario$time4)){
    determinant_true_scenario <- FIM_for_eff_knownvar(
    c(
    vals_true_scenario$time1,
    vals_true_scenario$time2,
    vals_true_scenario$time3
    ), starting_zero = TRUE)
  }

  if(!is.na(vals_true_scenario$time4)){
    determinant_true_scenario <- FIM_for_eff_knownvar(
    c(
    vals_true_scenario$time1,
    vals_true_scenario$time2,
    vals_true_scenario$time3,
    vals_true_scenario$time4
    ), starting_zero = FALSE)
  }
  print("Optimal␣Scenario␣done")
  return(vals_true_scenario)
}

list_results <- list()
for(i in 1:length(scenaria)){
# Apply the function and bind rows (each is a 1x19 vector)
try({results_mat <- do.call(rbind, apply(param_grid, 1, function(
   row) {
  print(row)
  sigma_input <- as.numeric(row["sigma"])
  rho_input <- as.numeric(row["rho"])
  simulate_scenario_variability_single(scenaria[[i]],sigma_input,
      rho_input)
}))})

# Combine param_grid and the function results
final_df <- cbind(param_grid, results_mat)
list_results[[i]] <- final_df
}

####### Simulation done now evaluate results
```

```
dt_timedeviation <- data.frame(
  scenario = character(),
  time1_deviation = numeric(),
  time2_deviation = numeric(),
  time3_deviation = numeric(),
  time4_deviation = numeric())

for(i in 1:12){
scenario_tmp <- list_results[[i]]

time1_deviation <- max(scenario_tmp$time1)-min(scenario_tmp$time1
   )
time2_deviation <- max(scenario_tmp$time2)-min(scenario_tmp$time2
   )
time3_deviation <- max(scenario_tmp$time3)-min(scenario_tmp$time3
   )
time4_deviation <- max(scenario_tmp$time4)-min(scenario_tmp$time4
   )

dt_timedeviation <- rbind(dt_timedeviation,tibble("scenario" = i,
    time1_deviation,time2_deviation,time3_deviation,time4_
   deviation))
}

knitr::kable(dt_timedeviation, format = "latex")
```

G.7. Sensitivity Analysis of Starting Values for Beta Distribution

```
#### make a mini simulation study to check if solutions are
   global or local

rm(list=ls())
#load functions

source("R/Simulation/Rfile/00_Functions_solving_beta_20250701.R")
source("R/Simulation/Rfile/00_Functions_solving_optimaldesign_
   20250225.R")


#load scenarios
load("R/Simulation/Rdata/scenaria2024-07-15.RData")

###-----------------------------------------------------------###
#### For beta
#-------------------------------------------------------------#

rm(solve_beta)
rm(solve_beta_notzero)

#-----------------------------------------------------#
### Function where we expect antibodies to start in 0
#-----------------------------------------------------#

#removed xstart from function input
#better to provide good starting values
solve_beta <- function(t_max, A_max, t_plat, A_plat){

  #First define system of equations based on input
  beta_systemofequations <- function(x){
    # x[1] is alpha
    # x[2] is beta
    # x[3] is t_obs

    # For derivation see material (using values and property of
```

```
    modus)
  y <- numeric(3)
  y[1] <- (x[1]-1)/(x[1]+x[2]-2) - t_max/x[3]
  y[2] <- (1/beta(x[1],x[2]))*(t_max/x[3])^(x[1]-1)*(1- t_max/x
    [3])^(x[2]-1) - A_max
  y[3] <- (1/beta(x[1],x[2]))*(t_plat/x[3])^(x[1]-1)*(1- t_plat
    /x[3])^(x[2]-1) - A_plat
  y
}

#now solve system
result <- nleqslv(x = x_start,
                  fn = beta_systemofequations,
                  method = "Broyden",
                  global = "dbldog",
                  xscalm = "auto",
                  control=list(maxit = 1000, allowSingular =
                     TRUE))

#check if function converged
#check if function converged
if(result$termcd != 1){
  print("NLEQSLV did not converge with Broyden, use Newton!")
  print(result$message)

  result_N <- nleqslv(x = x_start,
                      fn = beta_systemofequations,
                      method = "Newton",
                      global = "dbldog",
                      xscalm = "auto",
                      control=list(maxit = 1000, allowSingular
                         = TRUE))

  #check if function converged
  if(result_N$termcd != 1){
    #  print("NLEQSLV did not converge with Newton as well!")
    #  print(result_N$message)

    solution <- data.frame("alpha" = NA,
                           "beta" = NA,
```

```
                              "t_obs" = NA,
                              "converged_solving" = 0)
    }

    if(result_N$termcd == 1){
      #prepare output
      alpha <- result_N$x[1]
      beta <- result_N$x[2]
      t_obs <- result_N$x[3]

      solution <- data.frame("alpha" = alpha,
                              "beta" = beta,
                              "t_obs" = t_obs,
                              "converged_solving" = 1)
    }
  }

  # If function converged assign values
  if(result$termcd == 1){
    #prepare output
    alpha <- result$x[1]
    beta <- result$x[2]
    t_obs <- result$x[3]

    solution <- data.frame("alpha" = alpha,
                           "beta" = beta,
                           "t_obs" = t_obs,
                           "converged_solving" = 1)
  }
  # output

  return(solution)

}


#####-------------------------------------########
### Function where we want the antibodies to not start in zero
#####-------------------------------------########
```

```
#removed xstart from function input
solve_beta_notzero <- function(A_0, t_max, A_max, t_plat, A_plat)
   {

  #First define system of equations based on input
  beta_systemofequations <- function(x){
    # x[1] is alpha
    # x[2] is beta
    # x[3] is t_obs
    # x[4] is c

    # For derivation see material (using values and property of
       modus)
    y <- numeric(4)
    y[1] <- (x[1]-1)/(x[1]+x[2]-2) - (t_max + x[4])/x[3]
    y[2] <- (1/beta(x[1],x[2]))*(x[4]/x[3])^(x[1]-1)*(1-(x[4]/x
       [3]))^(x[2]-1) - A_0
    y[3] <- (1/beta(x[1],x[2]))*((t_max + x[4])/x[3])^(x[1]-1)*
       (1- (t_max + x[4])/x[3])^(x[2]-1) - A_max
    y[4] <- (1/beta(x[1],x[2]))*((t_plat + x[4])/x[3])^(x[1]-1)*
       (1- ((t_plat + x[4])/x[3]))^(x[2]-1) - A_plat
    y
  }

  #now solve system
  #now solve system
  result <- nleqslv(x = x_start,
                    fn = beta_systemofequations,
                    method = "Broyden",
                    global = "dbldog",
                    xscalm = "auto",
                    control=list(maxit = 1000, allowSingular =
                       TRUE))

  #check if function converged
  if(result$termcd != 1){
    print("NLEQSLV did not converge with Broyden, use Newton!")
    print(result$message)
```

```
  result_N <- nleqslv(x = x_start,
                      fn = beta_systemofequations,
                      method = "Newton",
                      global = "dbldog",
                      xscalm = "auto",
                      control=list(maxit = 1000, allowSingular
                         = TRUE))

  if(result_N$termcd != 1){
    print("NLEQSLV did not converge with Newton as well!")
    print(result_N$message)

    solution <- data.frame("alpha" = NA,
                           "beta" = NA,
                           "t_obs" = NA,
                           "constant" = NA,
                           "converged_solving" = 0)
  }

  if(result_N$termcd == 1){
    #prepare output
    alpha <- result_N$x[1]
    beta <- result_N$x[2]
    t_obs <- result_N$x[3]
    constant <- result_N$x[4]

    solution <- data.frame("alpha" = alpha,
                           "beta" = beta,
                           "t_obs" = t_obs,
                           "constant" = constant,
                           "converged_solving" = 1)
  }

}


# If function converged assign values
if(result$termcd == 1){
  #prepare output
  alpha <- result$x[1]
```

```
    beta <- result$x[2]
    t_obs <- result$x[3]
    constant <- result$x[4]

    solution <- data.frame("alpha" = alpha,
                           "beta" = beta,
                           "t_obs" = t_obs,
                           "constant" = constant,
                           "converged_solving" = 1)
  }


  # output

  return(solution)

}


##############################################
# make simulation for one

list_results_beta <- list()

for(j in 1:length(scenaria)){
scenario_tmp <- scenaria[[j]]

A_0_true <- scenario_tmp$A_0
t_max_true <- scenario_tmp$t_max
A_max_true <- scenario_tmp$A_max
t_plat_true <- scenario_tmp$t_plat
A_plat_true <- scenario_tmp$A_plat

#N simulation runs
N_simruns <- 10000

mat_starting_vals_beta <- c()
res_scenario_beta <- c()
for(i in 1:N_simruns){
```

```
if(scenario_tmp$A_0 == 0){
  x_start <- c(
    runif(1, min = 1.01, max = 100),       # >1
    runif(1, min = 1.01, max = 100),       # >1
    runif(1, min = 365.01, max = 50000)
  )
}

if(scenario_tmp$A_0 != 0){
  x_start <- c(
    runif(1, min = 1.01, max = 100),       # >1
    runif(1, min = 1.01, max = 100),       # >1
    runif(1, min = 365.01, max = 50000),    # >365
    runif(1, min = 0.01, max = 100)         # >0
  )
}

  try({
  res_tmp <- beta_parameters(A_0_input = A_0_true,
                  t_max_input = t_max_true,
                  A_max_input = A_max_true,
                  t_plat_input = t_plat_true,
                  A_plat_input = A_plat_true)

  mat_starting_vals_beta <- rbind(mat_starting_vals_beta,x_start)
  res_scenario_beta <- rbind(res_scenario_beta,res_tmp)
  })

}

res_all_single_scenario <- cbind(mat_starting_vals_beta,res_
   scenario_beta)

list_results_beta[[j]] <- res_all_single_scenario
}

###-----------------------------------------------------###
# Evaluate beta simulations ----
###-----------------------------------------------------###
```

```
for(i in 1:12){
  print(i)
  res_tmp <- list_results_beta[[i]] %>% filter(beta_parameters_
     valid == 1)
  print(max(res_tmp$alpha, na.rm=TRUE) - min(res_tmp$alpha, na.rm
     =TRUE))
  print(max(res_tmp$beta_value, na.rm=TRUE) - min(res_tmp$beta_
     value, na.rm=TRUE))
  print(max(res_tmp$t_obs, na.rm=TRUE) - min(res_tmp$t_obs, na.rm
     =TRUE))
  print(max(res_tmp$constant, na.rm=TRUE) - min(res_tmp$constant,
      na.rm=TRUE))

}
```