HELMHOLTZ
RESEARCH FOR GRAND CHALLENGES

HelmholtzZentrum münchen
German Research Center for Environmental Health

TECHNISCHE
UNIVERSITÄT
MÜNCHEN

Efficient parameter optimization for ordinary differential equation models of biological processes using semi-quantitative and qualitative data

Leonard Günter Schmiester

December 2020

TECHNISCHE UNIVERSITÄT MÜNCHEN

Fakultät für Mathematik

Efficient parameter optimization for ordinary differential equation models of biological processes using semi-quantitative and qualitative data

Leonard Günter Schmiester

Vollständiger Abdruck der von der Fakultät für Mathematik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende(r): Prof. Dr. Felix Krahmer

Prüfer der Dissertation:

1. Prof. Dr.-Ing. Jan P. Hasenauer
2. Prof. Dr. Christina Kuttler
3. Prof. Dr. Julio Banga

Die Dissertation wurde am 16.12.2020 bei der Technischen Universität München eingereicht und durch die Fakultät für Mathematik am 12.10.2021 angenommen.

Acknowledgments

First I want to thank Jan Hasenauer for his outstanding guidance and input, for the numerous fruitful discussions and for giving me the opportunity to work in his group. I also want to thank Daniel Weindl for all his great help, his support and for his never-ending patience.

I also want to thank my present and former colleagues and friends in the group of Jan Hasenauer and at the ICB. In particular: Aaron, Akshaya, Anandhi, Anna, Benni, Caro, Claudia, Dantong, Dennis, Dilan, Elba, Erika, Emad, Fabi, Fan, Jakob, Karolina, Lara, Lea, Linda, Lorenzo, Lukas, Marc, Marius, Paul, Phil, Philipp, Polina, Sabrina, Simon, Susanne, Tobi, Yannik, and everyone else who I forgot to mention.

A special thanks also to Marianne Antunes, Nina Fischer, Elisabeth Noheimer, Sabine Kunze and Anna Sacher for all their support with organisational tasks.

Furthermore, I want to thank the Institute of Computational Biology with its director Fabian Theis for providing such an inspiring research environment.

I also would like to thank my thesis committee members Christina Kuttler and Julio Banga for all their input and the valuable discussions.

Additionally, I am grateful to all my collaborators and cooperation partners, Adrian Hauber, Alejandro Villaverde, Christian Tönsing, Clemens Kreutz, Frank Bergmann, Franz-Georg Wieland, Janine Egert, Jens Timmer, Julio Banga, Lorenz Adlung, Luisa Schwarzmüller, Lukas Refisch, Marcel Schilling, Marcus Rosenblatt, Sven Sahle, Svenja Kemmer, Tacio Camba, Ursula Klingmüller, Wolfgang Müller and the whole CanPathPro team. It was great working with all of you.

Finally, I want to deeply thank Sarina, my family, in particular my parents and my brother, and my friends for all their support and for sharing all the good, and the bad, moments with me.

Abstract

Mathematical models are commonly used in systems biology to gain new insight into biological processes. In particular, ordinary differential equations (ODEs) are used to analyze dynamic processes in cell biology. ODE models often comprise unknown parameters which have to be inferred from experimental data. The ever-increasing amount of biological knowledge and diverse measurement data lead to the development of larger, more comprehensive models. While larger models and heterogeneous datasets can enable a holistic understanding of cellular interactions, they pose new challenges. The computational complexity of parameter estimation increases substantially with the size of the model. Furthermore, measurement processes often provide only semi-quantitative or even qualitative data which requires additional care when linking experimental data and model output.

In this thesis, we develop new algorithms for parameter estimation using heterogeneous datasets. First, we derive an approach for efficient parameter estimation of large-scale models using semi-quantitative data, meaning, that the data is relative to the absolute quantity. To this end, we combine the concepts of scalable adjoint sensitivity analysis and hierarchical optimization. We show that this leads to substantial reductions in computation time for a pan-cancer signaling model with thousands of unknown parameters and facilitates the unbiased integration of relative molecular and phenotypic datasets.

Then, we consider measurements that only provide qualitative information on the ordering of different datapoints. We derive improvements for the optimal scaling approach developed for parameter estimation using qualitative data. We provide several simplifications on this approach and develop an algorithm for computing gradients of the optimal scaling objective function. We apply these enhancements on different application examples showing large efficiency improvements compared to the standard optimal scaling approach.

To make parameter estimation methods, such as the ones developed in this thesis, available to a broader community and to improve reusability, interoperability and reproducibility, we established PEtab, a standardized format for the specification of parameter estimation problems. PEtab is supported by several commonly used computational toolboxes and we provide a Python library for easy manipulation and validation of PEtab problems.

The mathematical methods presented in this thesis enable the efficient integration of semi-quantitative and qualitative datasets with larger models and expand the amount of available data that can be used for parameter estimation which can facilitate deeper understand of biological systems.

Zusammenfassung

Mathematische Modelle werden in der Systembiologie häufig verwendet, um neue Einblicke in biologische Prozesse zu gewinnen. Insbesondere werden gewöhnliche Differentialgleichungen (GDGLen) verwendet, um dynamische Prozesse in der Zellbiologie zu analysieren. GDGL-Modelle enthalten oft unbekannte Parameter, die aus experimentellen Daten geschätzt werden müssen. Die ständig wachsende Menge an biologischem Wissen und vielfältigen Messdaten führt zur Entwicklung größerer, umfassenderer Modelle. Größere Modelle und heterogene Datensätze können zwar ein ganzheitliches Verständnis der zellulären Interaktionen ermöglichen, stellen aber neue Herausforderungen dar. Die notwendige Rechenleistung der Parameterschätzung nimmt mit der Größe des Modells erheblich zu. Darüber hinaus liefern Messverfahren oft nur semi-quantitative oder sogar qualitative Daten, was zusätzliche Sorgfalt bei der Verknüpfung von experimentellen Daten und Modelloutput erfordert.

In dieser Arbeit entwickeln wir neue Algorithmen zur Parameterschätzung unter Verwendung heterogener Datensätze. Zunächst leiten wir einen Ansatz zur effizienten Parameterschätzung von groß-skaligen Modellen unter Verwendung semi-quantitativer Daten, die relativ zur absoluten Messung sind, her. Zu diesem Zweck kombinieren wir die Konzepte der skalierbaren adjungierten Sensitivitätsanalyse und der hierarchischen Optimierung. Wir zeigen, dass dies bei einem Pan-Krebs-Signalmodell mit Tausenden von unbekannten Parametern zu erheblichen Verkürzungen der Berechnungszeit führt und die unvoreingenommene Integration relativer molekularer und phänotypischer Datensätze erleichtert.

Dann betrachten wir Messungen, die lediglich qualitative Informationen über die Anordnung der verschiedenen Datenpunkte liefern. Wir leiten Verbesserungen für den "optimal scaling" Ansatz her, der für die Parameterschätzung mit qualitativen Daten entwickelt wurde. Wir leiten mehrere Vereinfachungen für diesen Ansatz her und entwickeln einen Algorithmus zur Berechnung von Gradienten der optimal scaling Kostenfunktion. Wir wenden diese Verbesserungen auf verschiedene Anwendungsbeispiele an, die große Effizienzverbesserungen im Vergleich zum Standard optimal scaling Ansatz zeigen.

Um Parameterschätzmethoden, wie die in dieser Arbeit entwickelten, einer breiteren wissenschaftlichen Community zugänglich zu machen und um die Wiederverwendbarkeit, Interoperabilität und Reproduzierbarkeit zu verbessern, haben wir PEtab, ein standardisiertes

Format zur Spezifikation von Parameterschätzproblemen, entwickelt. PEtab wird von mehreren häufig verwendeten Toolboxen unterstützt, und wir stellen eine Python-Bibliothek zur einfachen Manipulation und Validierung von PEtab-Problemen zur Verfügung.

Die in dieser Arbeit vorgestellten mathematischen Methoden ermöglichen die effiziente Integration von semi-quantitativen und qualitativen Datensätze mit größeren Modellen und erweitern die Menge der verfügbaren Daten, die zur Parameterschätzung verwendet werden können, was ein tieferes Verständnis biologischer Systeme erleichtern kann.

Contents

Chapter 1

Introduction

The interplay of mathematics and biology has gained increasing attention over the last decades. The use of mathematical models in systems biology has enabled a profound understanding of the mechanisms underlying biological processes beyond what would have been possible only with biological experiments (Kitano, 2002). Mechanistic models, where the mathematical equations are based on knowledge about the underlying biochemical process, are particularly popular as they can predict latent variables and the dynamic behavior of the system (Adlung et al., 2017). Among the most widely used mechanistic models are those based on differential equations (Ingalls, 2013). These models have been used to describe, among others, cell metabolism (Smallbone & Mendes, 2013), epigenetics (Uzkudun et al., 2015; Zheng et al., 2012) and cell signaling (Bachmann et al., 2011; Bouhaddou et al., 2018; Fröhlich et al., 2018) and facilitated insights into diseases like cancer (Hass et al., 2017) and HIV (Perelson et al., 1996). Mechanistic models have been used as biomarkers (Fey et al., 2015), to model pathway cross-talk, and to predict cancer treatment outcomes (Fröhlich et al., 2018). Overall, mechanistic models can generate new biological knowledge and can pave the way towards more personalized medicine (Ogilvie et al., 2015). The large range of possible applications and the unique ability of mechanistic models to elucidate interpretable insights into cellular mechanisms explains the growing interest in using them to study biological processes and diseases.

1.1 Parameter estimation for ODE models

In this thesis, we consider mechanistic systems biology models based on ordinary differential equations (ODEs). These mechanistic ODE models comprise parameters such as reaction rate constants or initial concentrations of modeled species. The exact quantities of these parameters can often not directly be assessed by experimental procedures and are thus unknown. In order to still make reliable model predictions, these unknown parameters have to be inferred from experimental data. This is commonly done by optimizing an objective function, e.g. a likelihood or least-squares function, which quantifies the discrepancy of measurement and model output (Raue et al., 2013b). Various optimization methods exist

(Nocedal & Wright, 2006) a lot of which are implemented in different toolboxes (Egea et al., 2014; Hoops et al., 2006; Raue et al., 2015; Stapor et al., 2018b).

Parameter estimation for models of biological systems based on ODEs is a computationally challenging task, even for models with few modeled species. The objective functions that are optimized are usually non-convex and multi-modal. Furthermore, often no-closed form solution of the ODE is available and it has to be numerically integrated repeatedly during optimization. With increasing mechanistic knowledge on signaling pathways, larger and more comprehensive models can be generated describing pathway cross-talk (Fröhlich et al., 2018; Korkut et al., 2015) or even whole cells (Karr et al., 2012). This development is further supported by advances in data generation leading to large-scale datasets (Barretina et al., 2012; Li et al., 2017; Yang et al., 2013) and increasing computational resources enabling parallelization on high-performance computing clusters. The increasing model size and number of unknown parameters comes at the cost of higher computational requirements hindering successful parameter estimation (Babtie & Stumpf, 2017; Kapfer et al., 2019; Karr et al., 2015). This highlights the need for efficient and scalable approaches.

Usually, heterogeneous data types have to be integrated, which can further complicate parameter estimation. As there is often no direct correspondence of the measurement to a model state, more complex observation models are required. Different measurement techniques provide different amounts of information ranging from absolute over semi-quantitative data, to measurements which rather provide qualitative observations on the ordering of the datapoints (Pargett & Umulis, 2013) (Figure 1.1). While semi-quantitative data can generally refer to quantitative measurements whose relation to the absolute values is known, but non-trivial, in this thesis, we will specifically consider relative data, which is proportional to the underlying quantity.

Frequently employed techniques, like Western blotting (Renart et al., 1979) or flow cytometry (Herzenberg et al., 2006) measure cellular species, in most cases, relative to the absolute values. Additionally, for large-scale databases measurements are normalized and therefore also relative. Other experimental setups rather give information about the qualitative behavior, for example, if a species is up- or down-regulated after stimulation. This can be either, because the observation is inherently qualitative, like viability or inviability data (Mitra et al., 2018; Oguz et al., 2013), or because of unknown non-linear dependencies of the measured signal on the internal state of the system. In the latter case, only monotonicity is preserved which also yields qualitative data. Non-linear dependencies have been shown e.g. for Förster resonance energy transfer (FRET) (Birtwistle et al., 2011). They can

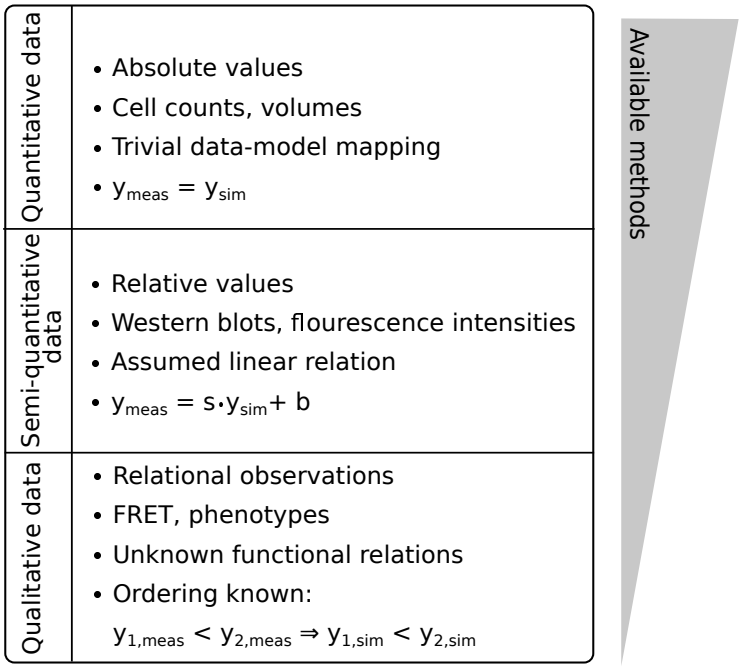


Figure 1.1: Overview of different data types ranging from quantitative to qualitative measurements. y_{meas} and y_{sim} denote measurement and simulation output respectively. s and b are scaling and offset parameters.

further occur due to detection thresholds and saturation effects, like for not properly set up Western blots (Butler et al., 2019).

In order to infer parameters from data that are not absolute, the non-trivial mapping between model simulation and measurement needs to be accounted for, which often increases the complexity of the parameter estimation problem and raises the need for tailored methods (Degasperi et al., 2017; Loos et al., 2018; Pargett et al., 2014; Weber et al., 2011).

1.2 Recent developments

The challenges in parameter estimation have led to new methodological developments. To cope with the increasing size of mathematical models, scalable gradient computation using adjoint sensitivities (cf. section 2.3.5) can be employed (Fröhlich et al., 2017). For relative data, either data-driven normalization (Degasperi et al., 2017), or scaling and offset parameters have been applied. For the latter, hierarchical approaches to reduce the dimensionality of the parameter search space by exploiting the problem structure were developed (Loos et al., 2018; Weber et al., 2011). The hierarchical approach has shown to

substantially improve optimizer efficiency. However, so far only scaling parameters, but not offsets were considered in the hierarchical approaches. Additionally, the combination with adjoint sensitivities remained an open issue.

For qualitative data, only a few methods exist to integrate them, and therefore they are often neglected. Still, they can provide valuable information to optimize parameters. The existing methods can roughly be divided into three approaches: (i) Oguz et al. (2013) used an objective function which quantified the number of correctly captured qualitative observations. (ii) Mitra et al. used qualitative data directly as inequality constraints imposed on the model output (Mitra et al., 2018, 2019). However, the approach suffers from hard to determine hyperparameters, which might influence optimization results. Mitra & Hlavacek (2020) developed a probabilistic distance measure for qualitative data. (iii) Pargett et al. used an *optimal scaling* method established in statistics (Shepard, 1962) and applied it to biological models (Pargett & Umulis, 2013; Pargett et al., 2014). Here, the optimal quantitative representation of the qualitative data is calculated in an hierarchical optimization problem. Solving this hierarchical problem is computationally demanding. Yet, the structure of this problem has not been analyzed in detail so far and it was unclear, if it can be simplified. Several studies have shown that gradient-based optimization is often more efficient than gradient-free optimization (Raue et al., 2013b; Schälte et al., 2018; Stapor et al., 2018b). Still, no reliable method to compute gradient information existed for optimization with qualitative data.

To make mathematical models and methods available to a broader community and to facilitate unbiased analysis, reproducibility and reusability are major concerns. Communities like the computational modeling in biology network (COMBINE) have made immense progress in this direction by developing standards for the description of mathematical and computational models in biology (Stanford et al., 2019), like the systems biology markup language (SBML) (Hucka et al., 2003). However, no standardized, widely adopted format for the definition of parameter estimation problems exists.

1.3 Contributions of this thesis

The contributions of this thesis focus on the following open problems and challenges:

(i) For large-scale ODE models, it is necessary to use scalable algorithms for gradient computation, like adjoint sensitivity analysis. The parameter search space for the optimization method is further increased when relative data is used, as this can require the introduction of scaling and offset parameters. A hierarchical method

exists to handle scaling parameters more efficiently. However, it remains unclear how offsets and a combination of scalings and offsets can be optimized hierarchically and how to combine the hierarchical method with adjoint sensitivity analysis, which is required for large-scale models.

(ii) Qualitative measurements have often been neglected even though they contain potentially important information to constrain parameters during optimization. Few methods exist to integrate qualitative data into the parameter estimation procedure. One promising approach is the optimal scaling method by Pargett et al. (2014). As for this method a constraint optimization problem has to be solved repeatedly it is computationally challenging and more efficient and reliable algorithms are needed.

(iii) Gradient-based optimization has shown to be advantageous compared to gradient-free methods (Schälte et al., 2018). Yet, so far no method exists to accurately compute gradients of the objective function, when qualitative data is used.

(iv) Standardized formats have proven to be necessary for reusability and interoperability. Henceforth, standards for the definition of computational models of biological processes have been established and widely adopted. In contrast, for parameter estimation problems no standard definition exists, which hinders reproducibility, reusability and interoperability.

The aforementioned issues are addressed and in the following, the contributions are delineated. The main contributions are:

(1) **Efficient parameter estimation of large-scale models using relative measurements.** We developed an approach to combine scalable adjoint sensitivity analysis with a hierarchical approach for efficient parameterization of scaling, offset and noise parameters. We demonstrated the superior convergence of local gradient-based optimization approaches when using the hierarchical method on a large-scale model of cancer signaling with thousands of parameters and provide an explanation for the impaired optimization when estimating these nuisance parameters without the hierarchical approach.

(2) **Efficient and robust parameter estimation for qualitative data using an optimal scaling approach.** We derived the theoretical foundation to reduce the parameter space for the inner problem of the optimal scaling approach. Additionally, we reparameterized the problem to facilitate the use of more efficient optimization methods. We showed that the reduction of the problem substantially improved

efficiency, while the reparameterization yielded an optimization problem that can be solved more robustly.

(3) **Gradient computation for the optimal scaling approach.** We derived formulas to calculate the gradient of the optimal scaling objective function with respect to the model parameters. This allows the use of gradient-based optimization techniques which, as we showed, are more efficient than gradient-free methods.

(4) **Interoperable specification of parameter estimation problems in systems biology.** We developed the PEtab format for the definition of parameter estimation problems. PEtab builds on the widely used SBML format (Hucka et al., 2003) and additional tab-separated value files for the definition of experimental data and an unambiguous link between measurement data and model observables. PEtab is implemented in several computational toolboxes, which makes mathematical methods, like the ones developed in this thesis, available to a broader scientific community.

Some of these contributions are already part of peer-reviewed publications, currently submitted to peer-reviewed journals or in preparation. Parts of the work in this thesis thus correspond or are to some extent similar to the following publications:

- **Schmiester, L.***, Schälte, Y.*, Fröhlich, F., Hasenauer, J. & Weindl, D. (2019). Efficient parameterization of large-scale dynamic models based on relative measurements. *Bioinformatics*, 36 (2), 594-602. (*equal contribution)

- **Schmiester, L.**, Weindl, D. & Hasenauer, J. (2020). Parameterization of mechanistic models from qualitative data using an efficient optimal scaling approach. *J. Math. Biol.*, 81, 603–623

- **Schmiester, L.**, Weindl, D. & Hasenauer, J. Efficient gradient-based parameter estimation for dynamic pathway models using qualitative data, *in preparation.*

- **Schmiester, L.***, Schälte, Y.*, Bergmann, F., Camba, T., Dudkin, E., Egert, J., Fröhlich, F., Fuhrmann, L., Hauber, A. L., Kemmer, S., Lakrisenko, P., Loos, C., Merkt, S., Müller, W., Pathirana, D., Raimúndez, E., Refisch, L., Rosenblatt, M., Stapor, P., Städter, P., Wang, D., Wieland, F.-G., Banga, J. R., Timmer, J., Villaverde, A. F., Sahle, S., Kreutz, C., Hasenauer, J., Weindl, D. (2020). PEtab–interoperable specification of parameter estimation problems in systems biology. *arXiv*, 2004.01154 [q-bio.QM]

- Fröhlich, F., Kessler, T., Weindl, D., Shadrin, A., **Schmiester, L.**, Hache, H., Muradyan, A., Schütte, M., Lim, J-H., Heinig, M., Theis, F., Lehrach, H., Wierling, C., Lange, B. & Hasenauer, J. (2019). Efficient parameter estimation enables the prediction of drug response using a mechanistic pan-cancer pathway model. *Cell Systems*, 7 (6), 567-579.

Other contributions of my doctoral research which are not included in this thesis are:

- Städter, P.*, Schälte, Y.*, **Schmiester, L.***, Hasenauer, J. & Stapor, P. (2020). Benchmarking of numerical integration methods for ODE models of biological systems. *bioRxiv*, 10.1101/2020.09.03.268276

- Stapor, P.*, Adlung, L.*, Tönsing, C.*, **Schmiester, L.**, Schwarzmüller, L., Wang, D., Timmer, J., Klingmüller, U., Hasenauer, J., Schilling, M. (2019). Cell-to-cell variability in JAK2/STAT5 pathway components and cytoplasmic volumes define survival threshold in erythroid progenitor cells. *bioRxiv*, 10.1101/866871

- Lines, G. T., Paszkowski, Ł., **Schmiester, L.**, Weindl, D., Stapor, P. & Hasenauer, J. (2019). Efficient computation of steady states in large-scale ODE models of biochemical reaction networks. *IFAC-PapersOnLine*, 52 (26), 32-37

- Stapor, P., **Schmiester, L.**, Weindl, D. & Hasenauer, J. (2019). Mini-batch optimization enables training of ODE models on large-scale datasets. *bioRxiv*, 10.1101/859884

Besides the contribution to these research articles, I also contributed to the development of several toolboxes which enable the reusability of the here developed methods:

- **pyPESTO** *Python Parameter EStimation TOolbox:* A Python-based optimization toolbox for systems biology models providing different optimization and uncertainty quantification algorithms. Contributions (2) and (3) are implemented in pyPESTO.

- **parPE** *parallel Parameter Estimation library:* A highly scalable optimization toolbox tailored for large-scale problems that can be used on high-performance computing clusters. The contribution (1) is implemented in parPE.

- **PEtab** *Parameter estimation problems in tabular format:* A standardized, interoperable format to define parameter estimation problems based on SBML (Hucka et al., 2003) and tab-separated value files, which is the basis for contribution (4). All case studies in this thesis were implemented in the PEtab format.

1.4 Outline

The remainder of the thesis is organized as follows: The notation and background on ODE modeling of biological processes as well as parameter inference is introduced in Chapter 2. Different data types, ranging from absolute over relative measurements to qualitative observations, and potential modeling approaches of these data are explained there. Chapters 3, 4 and 5 present the main contributions of this thesis. In Chapter 3, the combination of an efficient hierarchical approach for relative data and scalable gradient computation by using adjoint sensitivity analysis is derived. The benefit of this approach is then demonstrated on a large-scale model of cancer signaling. Chapter 4 starts with an introduction on parameter estimation for qualitative data using a so-called *optimal scaling* approach. Afterwards, the theoretical foundations for a more efficient and robust formulation of the optimal scaling approach are derived and a method for gradient computation of these problems is presented. The advantage of the newly developed methods is demonstrated on several application examples. In Chapter 5, a reusable pipeline for parameter estimation based on the PEtab format for specification of parameter estimation problems is introduced. The thesis is concluded in Chapter 6 with a summary of the main contributions and an outlook on promising further developments.

Chapter 2

Background

In the following, we introduce the notation and the relevant background that is needed throughout this thesis. In particular, we introduce the modeling of biological processes using ODEs and how to link model simulation to different types of measured data. Furthermore, we outline different methods for parameter inference and gradient computation. This chapter is based on my following publications, therefore, in parts similar or even identical to them:

- **Schmiester, L.***, Schälte, Y.*, Fröhlich, F., Hasenauer, J. & Weindl, D. (2019). Efficient parameterization of large-scale dynamic models based on relative measurements. *Bioinformatics*, 36 (2), 594-602.
- **Schmiester, L.**, Weindl, D. & Hasenauer, J. (2020). Parameterization of mechanistic models from qualitative data using an efficient optimal scaling approach. *J. Math. Biol.*, 81, 603–623
- **Schmiester, L.**, Weindl, D. & Hasenauer, J. Efficient gradient-based parameter estimation for dynamic pathway models using qualitative data, *in preparation.*

2.1 Mathematical modeling of biological systems

2.1.1 Ordinary differential equation models

Models based on ODEs are among the most widely used models for biological processes. Under the assumption of sufficiently abundant and homogeneously distributed biochemical species, ODE models often yield a good representation of the underlying process while maintaining computational tractability. In this thesis, we restricted ourselves to models based on ODEs and consider models of the form

$$\dot{x}(t, \theta, u) = f(x(t, \theta, u), \theta, u), \quad x(t_0, \theta, u) = x_0(\theta, u). \tag{2.1}$$

Here, the state vector $x(t, \theta, u) \in \mathbb{R}^{n_x}$ describes the concentrations of the modeled species. Examples are mRNAs, proteins or protein complexes. The derivative of x with respect

to the time $t \in \mathbb{R}$ is given by $\dot{x}$. The vector field $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ denotes the temporal evolution of the species. Here, n_x is the total number of modeled species, n_θ the number of parameters and n_u the number of known inputs. Unknown model parameters, such as reaction rate constants or initial concentrations, are described by $\theta \in \mathbb{R}^{n_\theta}$. $u \in \mathbb{R}^{n_u}$ are known inputs, e.g. different experimental conditions, such as drug concentrations or genetic profiles. Here, we only consider parameters θ and inputs u that are constant over time. We assume that the vector field f is Lipschitz continuous, which, according to the Picard-Lindelöf Theorem, guarantees local existence and uniqueness of the solution of (2.1) (Coddington & Levinson, 1955). As we consider ODE models of biological systems, which usually have a finite solution in the considered time interval, the assumption is appropriate for parameters θ in a biologically reasonable range.

2.1.2 Linking model output to measurements

To obtain predictive models, the unknown parameters have to be inferred from experimental data. To this end, model simulations need to be linked to measurements. As measurements often provide information not on a single species but rather on a combination of species, this is achieved by introducing an observation function $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$, which maps the states, parameters and inputs to observables $y(t, \theta, u) \in \mathbb{R}^{n_y}$ via

$$y(t, \theta, u) = h(x(t, \theta, u), \theta, u), \tag{2.2}$$

where n_y is the number of observables. The observables describe experimentally obtained measurements $\bar{y}$. These are usually subject to noise, which needs to be accounted for. This can be done using a noise model. Most frequently, i.i.d. additive Gaussian noise models are considered:

$$\bar{y}_{i_t,i_y,i_u} = y_{i_y}(t_{i_t}, \theta, u_{i_u}) + \varepsilon_{i_t,i_y,i_u}, \quad \text{with } \varepsilon_{i_t,i_y,i_u} \sim \mathcal{N}(0, \sigma^2_{i_t,i_y,i_u}), \tag{2.3}$$

with standard deviations σ, time index i_t, observable index i_y and condition index i_u. While Gaussian noise is the most widely used error model, other noise models have shown to be more robust in some cases, e.g. in the presence of outliers (Maier et al., 2017).

We denote the collection of all measurements as

$$\mathcal{D} = \{\bar{y}_{i_t,i_y,i_u}\}^{\substack{i_t=1,\dots,n_t \\ i_y=1,\dots,n_y \\ i_u=1,\dots,n_u}} \tag{2.4}$$

Biological experiments often provide measurements that are not absolute. In some cases, they are only proportional to the absolute concentrations of interest. In other cases, even the assumption of a linear relation does not hold true. Generally, non-absolute data can be modeled with an additional measurement process function $g : \mathbb{R}^{n_y} \times \mathbb{R}^{n_\xi} \to \mathbb{R}^{n_y}$

$$y(t, \theta, u, \xi) = g(h(x(t, \theta, u), \theta, u), \xi) \tag{2.5}$$

with observable parameters $\xi \in \mathbb{R}^{n_\xi}$. These parameters are usually not known and, therefore, also need to be estimated. Commonly used measurement techniques provide e.g. relative information leading to a linear transformation of the form

$$g(h(x(t, \theta, u), \theta, u), s, b) = s \cdot h(x(t, \theta, u), \theta, u) + b \tag{2.6}$$

with scaling and offset parameters s, b. Qualitative observations result in more complex and usually unknown non-linear transformations (Pargett & Umulis, 2013). This will be introduced in more detail in Chapters 3 and 4.

2.2 Maximum likelihood function

ODE models of biological processes of the form (2.1) usually comprise parameters θ, which are difficult to determine experimentally and are therefore unknown. Modeling of non-trivial measurement processes can introduce additional parameters ξ. These unknown parameters need to be inferred from experimental data which is commonly done by optimizing an objective function. Here, we consider the likelihood function, which is the probability of observing data $\mathcal{D}$, given the parameters θ, ξ:

$$L(\theta, \xi) = p(\mathcal{D}|\theta, \xi). \tag{2.7}$$

We assume i.i.d. additive Gaussian noise with standard deviation σ. In some cases, the standard deviations σ of the measurement error can be directly obtained, e.g. when experimental replicates are available. However, often σ is unknown and therefore also needs to be estimated along with θ and ξ. We denote the collection of model parameters θ, observable parameters ξ and unknown noise parameters σ as $\psi = (\theta, \xi, \sigma)$. The likelihood can then be written as

$$L(\psi) = \prod_{i_u}^{n_u} \prod_{i_y}^{n_y} \prod_{i_t}^{n_t} \frac{1}{\sqrt{2\pi}\sigma_{i_t,i_y,i_u}(\theta, \xi)} \exp\left\{ -\frac{1}{2} \left(\frac{\bar{y}_{i_t,i_y,i_u} - y_{i_y}(t_{i_t}, \theta, u_{i_u}, \xi)}{\sigma_{i_t,i_y,i_u}(\theta, \xi)} \right)^2 \right\}. \tag{2.8}$$

We are then interested in the maximum likelihood estimate ψ^{ML}, which maximizes the likelihood function

$$\psi^{\mathrm{ML}} = \arg\max_{\psi \in \Psi} L(\psi), \tag{2.9}$$

with the domain Ψ indicating the range of plausible parameter values. For numerical stability, usually the logarithm of the likelihood is considered (Raue et al., 2013a). Additionally, as the majority of optimization algorithms is implemented to minimize the objective function, the negative log-likelihood is used which is given by

$$\ell(\psi) = -\log(L(\psi)) = \frac{1}{2} \sum_{i_u}^{n_u} \sum_{i_y}^{n_y} \sum_{i_t}^{n_t} \ell_{i_t,i_y,i_u}(\psi), \tag{2.10}$$

with

$$\ell_{i_t,i_y,i_u}(\psi) = \log(2\pi\sigma^2_{i_t,i_y,i_u}(\theta,\xi)) + \left(\frac{\bar{y}_{i_t,i_y,i_u} - y_{i_y}(t_{i_t},\theta,u_{i_u},\xi)}{\sigma_{i_t,i_y,i_u}(\theta,\xi)} \right)^2. \tag{2.11}$$

Minimizing the negative log-likelihood ℓ yields the same optima as maximizing the likelihood L.

In the case of known measurement noise σ_{i_t,i_y,i_u}, we obtain a weighted least-squares function

$$J(\psi) = \frac{1}{2} \sum_{i_u}^{n_u} \sum_{i_y}^{n_y} \sum_{i_t}^{n_t} J_{i_t,i_y,i_u}(\psi) \tag{2.12}$$

with

$$J_{i_t,i_y,i_u}(\psi) = w_{i_t,i_y,i_u} \left(\bar{y}_{i_t,i_y,i_u} - y_{i_y}(t_{i_t},\theta,u_{i_u},\xi) \right)^2. \tag{2.13}$$

The weights are given by $w_{i_t,i_y,i_u} = 1/\sigma^2_{i_t,i_y,i_u}$. We neglected constant terms here, as they do not change the optima of the function. It has been shown that it is often advantageous to optimize log-transformed parameters (if positivity of the parameters can be assumed), as the objective function tends to be "more convex" in this case (Hass et al., 2019), in the sense that more randomly drawn parameters ψ^1, ψ^2 fulfill $J(\alpha\psi^1 + (1-\alpha)\psi^2) \leq \alpha J(\psi^1) + (1-\alpha)J(\psi^2), \forall \alpha \in [0,1]$.

2.3 ODE-constrained optimization

The unknown model parameters are estimated by solving an ODE-constrained optimization problem. These problems are usually non-convex, multi-modal and potentially high-dimensional. Different approaches exist to handle the ODE constraints. The multiple-shooting approach to solve this problem, splits the time interval into sub-intervals (Bock &

Plitt, 1984). The ODE constraints can also be relaxed by approximating the ODE, e.g. using spline functions, and optimizing the coefficients of the approximation together with the parameters (Chung et al., 2017). In contrast, the reduced or single-shooting approach does not handle the ODE as constraints. Instead, the initial value problem is solved on the whole time interval, to evaluate the objective function. In this thesis, we will focus on the single-shooting formulation approach using implicit ODE solvers. To this end, the numerical solution to the ODE needs to be calculated repeatedly for different parameter vectors during optimization.

The parameter estimation approaches are often looked at either from a frequentist or a Bayesian point-of-view (Raue et al., 2013a). The first one aims to optimize an objective function to obtain a point estimate, while the latter estimates the posterior distribution of the parameters using sampling-based approaches like Markov-chain Monte-Carlo methods (Ballnus et al., 2017). Here, we will focus mainly on the frequentist perspective.

2.3.1 Optimization problems with box-constraints

Parameter estimation problems usually consist of minimizing an objective function $\mathcal{J}$, here we consider (2.10) or (2.12), i.e. $\mathcal{J} = \{\ell, J\}$, given some constraints on the parameters. In systems biology, frequently simple box-constraints are considered, as often biologically plausible boundaries, but no further information on the relation of parameters is known. This leads to an optimization problem of the form

$$\begin{aligned} &\min_{\psi} \mathcal{J}(\psi) \\ &\text{s.t. } l \leq \psi \leq u \end{aligned} \tag{2.14}$$

with lower and upper bounds $l, u \in \mathbb{R}^{n_\psi}$. Under the assumption of a continuously differentiable objective function, necessary optimality conditions can be derived. If ψ^* is a local minimizer of (2.14), then it holds that (Boyd & Vandenberghe, 2004)

$$\frac{\partial \mathcal{J}(\psi^*)}{\partial \psi_k} = \begin{cases} \geq 0, & \text{if } \psi_k^* = l_k \\ \leq 0, & \text{if } \psi_k^* = u_k \\ = 0, & \text{else.} \end{cases} \tag{2.15}$$

When the bounds are sufficiently wide such that the optimum lies within them, or no bounds are present, the optimality condition simplifies to $\frac{\partial \mathcal{J}(\psi^*)}{\partial \psi_k} = 0$. In addition, a

sufficient condition for a local minimum is given by the positive definiteness of the Hessian $\nabla^2 \mathcal{J}(\psi^*)$.

2.3.2 Optimization problems with inequality constraints

In some settings, more general inequality constraints can occur, e.g. when incorporating qualitative measurements (Mitra et al., 2018; Pargett et al., 2014). Therefore, we consider the optimization problem

$$\begin{aligned} &\min_{\psi} \mathcal{J}(\psi) \\ &\text{s.t. } q_{i_q}(\psi) \leq 0, \text{ for } i_q = 1, \ldots, n_q \end{aligned} \tag{2.16}$$

with n_q inequality constraints $q_{i_q} : \mathbb{R}^{n_\psi} \to \mathbb{R}$. Optimality conditions can also be derived for (2.16). We consider the Lagrangian function for this problem which is defined as

$$\mathcal{L}(\psi, \mu) = \mathcal{J}(\psi) + \mu^T q(\psi), \tag{2.17}$$

with Lagrange multipliers $\mu \in \mathbb{R}_+^{n_q}$. With this, a necessary optimality condition is as follows (Nocedal & Wright, 2006): If $\mathcal{J}$ and q are continuously differentiable functions and ψ^* is a local optimum of (2.16) then there exist Lagrange multipliers $\mu \in \mathbb{R}_+^{n_q}$, such that

$$\begin{aligned} \nabla_\psi \mathcal{L}(\psi^*, \mu) = \nabla_\psi \mathcal{J}(\psi^*, \mu) + \mu^T \nabla_\psi q(\psi^*) &= 0 \\ \mu_{i_q} q_{i_q}(\psi^*) &= 0 \\ q_{i_q}(\psi^*) &\leq 0 \\ \mu_{i_q} &\geq 0, \end{aligned} \tag{2.18}$$

for $i_q = 1, \ldots, n_q$. If the objective function $\mathcal{J}$ is convex, the necessary optimality conditions (2.18) are also sufficient (Boyd & Vandenberghe, 2004). Sufficient conditions for a local optimum can also be derived in the general case (Fiacco, 1976), under the assumption of twice continuously differentiable functions $\mathcal{J}$ and q in the neighborhood of ψ^*. Then ψ^* is a minimizing point, if μ exists such that (2.18) is satisfied and the second-order conditions hold:

$$m^T \nabla_\psi^2 \mathcal{L}(\psi^*, \mu) m > 0 \tag{2.19}$$

for all $m \neq 0$ such that

$$\begin{aligned} &m^T \nabla_\psi q_i(\psi^*) \geq 0, \forall i, \text{ with } q_i(\psi^*) = 0 \\ &m^T \nabla_\psi q_i(\psi^*) = 0, \forall i, \text{ with } \mu_i > 0. \end{aligned} \tag{2.20}$$

If q reduces to box constraints or no constraints are present, the necessary and sufficient conditions are equivalent to the ones obtained in Section 2.3.1.

2.3.3 Global optimization

To find the optimal parameters of the problems (2.14) and (2.16), numerical optimization has to be performed. Numerous methods and algorithm were developed to this end (Nocedal & Wright, 2006). Usually, one is interested in the globally optimal parameters. Different strategies exist to find the global optimum of a multi-modal, non-linear objective function. Evolutionary (Bäck, 1996; Runarsson & Yao, 2000) and swarm-based (Kennedy, 2011; Yang, 2010) algorithms are inspired by nature and biology. Simulated annealing is another technique based on principles from physics (Goffe et al., 1994; Kirkpatrick et al., 1983).

Hybrid approaches try to benefit both from global and local approaches by combining them. Among the most popular methods are metaheuristics combined with local searches as well as multi-start local optimization (Fröhlich et al., 2019; Villaverde et al., 2018). Scatter-search algorithms (Egea et al., 2007; Penas et al., 2015; Villaverde et al., 2012) are examples of metaheuristics. These combine a global search phase with local optimizations at selected points. Multi-start methods aim to find the global optimum by repeatedly performing local optimizations initialized at different randomly chosen starting points (Raue et al., 2013b). They can be analyzed by means of the different values found from the local optimizations, giving rise not only to the global optimum but also to different local optima of the objective function. Hybrid methods can be further accelerated and scaled to large problems using parallelization, e.g. over the different local searches (Penas et al., 2015, 2017).

In the context of parameter estimation for ODE models in systems biology, there are several case studies trying to identify well-working approaches and establish guidelines (Hass et al., 2019; Schälte et al., 2018; Villaverde et al., 2018). These analyses only consider box-constrained optimization problems of the form (2.14) but no general inequality constraints as these occur less frequently.

2.3.4 Local optimization

Hybrid global-local methods like multi-start as well as metaheuristics combined with local searches have shown to perform reasonably well in most of the common systems biology applications (Villaverde et al., 2018). Still, the efficiency of the employed local optimization algorithms plays a crucial role for successful parameter estimation. Various methods exist

to find local minima (Nocedal & Wright, 2006). These incorporate different information like the objective function value, gradients or potentially even higher-order derivatives to iteratively update the parameter vector. Often, gradient-based optimization algorithms are preferred (Raue et al., 2013b; Schälte et al., 2018; Stapor et al., 2018b). However, they rely on the availability of efficient and robust ways to calculate gradients. In cases, where this is not possible, gradient-free algorithms, which only rely on the objective function to be evaluable, can be valuable alternatives. In the following, we briefly review some commonly used gradient-free and gradient-based optimization methods.

Gradient-free Optimization

For gradient-free optimizers, theoretical results on convergence to an optimal value can only sometimes be obtained (Rios & Sahinidis, 2013). Local search algorithms can be divided into direct and model-based methods. The direct method pattern search uses exploratory moves for local search of an improving direction and pattern moves for larger steps (Hooke & Jeeves, 1961). Simplex based methods like the Nelder-Mead algorithm (Nelder & Mead, 1965) use trial points to form a simplex and then extrapolate the objective function to replace the worst trial point with a new one. Model-based algorithms build a surrogate model of the objective function which is then optimized. Linear (Powell, 1994) as well as quadratic models (Powell, 2009) are often used in this context.

Most gradient-free methods have been developed for unconstrained or box-constrained problems. Still, some methods exist to handle more general constraints, e.g. the model-based approach developed by Powell (1994). Additionally, penalty and augmented Lagrangian methods can be used which combine the objective function and the constraints into either a penalty function of the form $\mathcal{J} + \lambda q^2$ or which are based on the Lagrangian function (2.17).

Gradient-based Optimization

Most gradient-based optimization algorithms iteratively update the parameter vector, yielding a sequence of (monotonically) decreasing objective function values. For this, a direction and a step length needs to be computed (Fröhlich et al., 2019). Typical directions are steepest descent (Curry, 1944), which is only based on the gradient, or (quasi-) Newton directions which use either explicitly computed Hessian information, or approximations thereof, like SR1 (Byrd et al., 1996), (L-)BFGS (Fletcher & Powell, 1963; Nocedal, 1980) or conjugate gradient methods (Branch et al., 1999). The strategies to determine the parameters of the next iterate are often divided into line-search (Wächter & Biegler, 2006) and trust-region methods (Sorensen, 1982; Yuan, 2015). They mainly differ in the order in

which they determine the direction and step length. Line-search strategies first compute a direction and then determine an appropriate step length. A common procedure for step length calculation is the backtracking line-search scheme, where a relatively large step size is initially chosen, which is iteratively shrunk until an improvement in the objective function value is observed. Trust-region methods proceed conversely, by first determining a trust region in which a model function of the objective function is constructed. The trust region defines the maximal step length. Then an appropriate direction is calculated. Algorithms based on combinations of trust-region and line-search steps have also been developed (Waltz et al., 2006). For many gradient-based algorithms, theoretical results under which convergence can be guaranteed exist (Nocedal & Wright, 2006).

Constraints can be incorporated in different ways. Again penalty and augmented Lagrangian methods can be used (Nocedal & Wright, 2006). Sequential quadratic programming (SQP) approaches solve a quadratic subproblem at each iterate using the Lagrangian (Schittkowski, 2011). The subproblem in SQP usually uses a linearization of the constraints. Interior point methods use barrier functions which result in infinite function values at the boundaries (Byrd et al., 1999).

2.3.5 Gradient calculation

Different strategies exist to calculate gradient information of the objective function. In the following, we consider the negative log-likelihood function ℓ, but the approaches can also be applied to weighted least squares functions. Most frequently used are (i) finite difference schemes, (ii) forward sensitivity analysis and (iii) adjoint sensitivity analysis (Fröhlich et al., 2019). But also other approaches like automatic differentiation exist (Rackauckas et al., 2018). In the following, we restrict ourselves to the trivial observable transformation ($g(h) = h$), i.e. the observable function (2.2) with no observable parameters ($\psi = \theta$). Gradient computation for relative and qualitative data will be introduced in chapters 3 and 4 respectively.

Finite difference methods calculate an approximation of the gradient of the objective function based on

$$\frac{d\ell(\theta)}{d\theta_k} \approx \frac{\ell(\theta + ae_k) - \ell(\theta - be_k)}{a + b}, \tag{2.21}$$

with unit vectors e_k and $a, b \geq 0$. Typical choices are forward differences ($b = 0$), backward differences ($a = 0$) or central differences ($a = b \neq 0$). While they are easy to implement irrespective of the function, they rely on step sizes a and b which need to be defined a

priori, and error control is difficult. The computational cost of finite difference methods scales linearly with the number of parameters.

Sensitivity analysis methods are based on analytically differentiating the negative log-likelihood function (2.10) yielding

$$\frac{d\ell(\theta)}{d\theta_k} = \sum_{i_u}^{n_u} \sum_{i_y}^{n_y} \sum_{i_t}^{n_t} \frac{\partial \ell_{i_t,i_y,i_u}(\theta)}{\partial \theta_k} + \frac{\partial \ell_{i_t,i_y,i_u}(\theta)}{\partial y_{i_y}} s_k^{y_{i_y}}(t_{i_t}, \theta, u_{i_u}), \tag{2.22}$$

where $s_k^{y_{i_y}}$ denotes the observable sensitivity which is the derivative of the observable y_{i_y} with respect to θ_k. The gradient of ℓ can then be computed once the observable sensitivities are derived.

Forward sensitivity analysis aims to calculate the observable sensitivities via their dependency on the state sensitivities $s_k^x(t_{i_t}, \theta, u_{i_u})$

$$s_k^{y_{i_y}}(t_{i_t}, \theta, u_{i_u}) = \frac{\partial h_{i_y}}{\partial x}(x(t_{i_t}, \theta, u_{i_u}), \theta, u_{i_u}) \cdot s_k^x(t_{i_t}, \theta, u_{i_u}) + \frac{\partial h_{i_y}}{\partial \theta_k}(x(t_{i_t}, \theta, u_{i_u}), \theta, u_{i_u}) \tag{2.23}$$

State sensitivities can be obtained by differentiating the ODE (2.1) with respect to θ which yields

$$\begin{aligned} \dot{s}_k^x(t, \theta, u) &= \frac{\partial f}{\partial x}(x(t, \theta, u), \theta, u) \cdot s_k^x(t, \theta, u) + \frac{\partial f}{\partial \theta_k}(x(t, \theta, u), \theta, u) \\ s_k^x(t_0, \theta, u) &= \frac{dx_0}{d\theta_k}(\theta, u). \end{aligned} \tag{2.24}$$

This system of ODEs can be solved along with the original ODE leading to the augmented system

$$\begin{aligned} \dot{x}(t, \theta, u) &= f(x(t, \theta, u), \theta, u) \\ \dot{s}_k^x(t, \theta, u) &= \frac{\partial f}{\partial x}(x(t, \theta, u), \theta, u) \cdot s_k^x(t, \theta, u) + \frac{\partial f}{\partial \theta_k}(x(t, \theta, u), \theta, u) \end{aligned} \tag{2.25}$$

with initial conditions

$$\begin{aligned} x(t_0, \theta, u) &= x_0(\theta, u) \\ s_k^x(t_0, \theta, u) &= \frac{dx_0}{d\theta_k}(\theta, u). \end{aligned} \tag{2.26}$$

Forward sensitivities can consequently be obtained by solving a problem of dimension $n_x(1 + n_\theta)$.

Adjoint sensitivity analysis is an alternative which circumvents evaluating the state sensitivities (Fröhlich et al., 2017; Kokotovic & Heller, 1967). To this end, an adjoint state $p : [t_0, t_{n_t}] \mapsto \mathbb{R}^{n_x}$ is introduced which is integrated backward in time. The adjoint state is chosen such that it is zero for $t > t_{n_t}$ and satisfies the following backward differential

equation on $(t_j, t_{j+1}]$:

$$\begin{aligned} \dot{p}(t) &= -\frac{\partial f}{\partial x}(x(t,\theta,u),\theta,u) \cdot p(t) \\ p(t_{j+1}) &= \lim_{t \searrow t_{j+1}} p(t) - \left(\frac{\partial \ell_{j+1}(\theta)}{\partial y} \frac{\partial h}{\partial x}(x(t,\theta,u),\theta,u) \right)^T \end{aligned} \tag{2.27}$$

With this, the gradient of the objective function can be written as

$$\frac{d\ell(\theta)}{d\theta_k} = \sum_{i_u}^{n_u} \sum_{i_y}^{n_y} \sum_{i_t}^{n_t} \left[\frac{\partial \ell_{i_t,i_y,i_u}(\theta)}{\partial \theta_k} + \frac{\partial \ell_{i_t,i_y,i_u}(\theta)}{\partial y_{i_y}} \frac{\partial h}{\partial \theta_k} \right] - p(t_0)^T \frac{dx_0(\theta, u_{i_u})}{d\theta_k} - \int_{t_0}^{t_{n_t}} p(t)^T \frac{\partial f}{\partial \theta_k} dt \tag{2.28}$$

This can be computed irrespective of the state sensitivities, except for at $t = 0$, which can usually be easily calculated. Calculation of the gradient here depends on the adjoint state p instead of the state sensitivities, whose dimension is n_x and does not depend on the number of parameters n_θ. The adjoint approach yields two systems of size n_x that need to be solved. Additionally, n_θ quadratures have to be solved. However, this is computationally much less demanding than solving the ODE and therefore often negligible. For larger ODE models, adjoint sensitivity analysis has shown to be much more efficient than finite differences and forward sensitivity analysis (Fröhlich et al., 2017; Fröhlich et al., 2018; Kapfer et al., 2019).

2.4 Implementation

The computational analysis in the following chapters of this thesis were performed using different toolboxes which are listed in the following:

- **AMICI** (Fröhlich et al., 2017) provides efficient algorithms for numerical integration of ODEs and sensitivity calculation by interfacing the SUNDIALS solver suite (Hindmarsh et al., 2005).
- **PESTO** (Stapor et al., 2018b) is a Matlab toolbox to solve parameter estimation problems. It interfaces multiple gradient-based and gradient-free optimization algorithms.
- **pyPESTO** (Schälte et al., 2020) is a Python toolbox for parameter estimation providing similar functionality to PESTO.

- **parPE** (Schmiester et al., 2019) is a high performance computing library for large-scale parameter estimation problems using massive parallelization.
- **PEtab** (Schmiester et al., 2020a) defines the specification of parameter estimation problems in a standardized format and provides a Python library for validation of problems defined in the PEtab format.

All of these tools are available under permissive licenses and are maintained on Github. Contribution (1) is implemented in parPE and contributions (2) and (3) are part of pyPESTO. PEtab is part of the contribution (4) of this thesis.

Chapter 3

Efficient parameter estimation for large-scale models with relative data

Signal transduction models often only consider isolated pathways, neglecting cross-talk between different pathways. With increasing biological knowledge and computation power, larger and more comprehensive models have been developed including hundreds to thousands of species and unknown parameters (Bouhaddou et al., 2018; Chen et al., 2009; Fröhlich et al., 2018). Large models like the pan-cancer model introduced by Fröhlich et al. (2018) have been used to accurately predict the response of cancer cell-lines to drug combination treatments and can pave the way to more personalized medicine. However, with increasing model size, more advanced parameter estimation methods are needed. The main challenges for parameter estimation of large-scale models are (i) the computational complexity of repeated model simulation and gradient calculation and (ii) the availability of large amounts of measurement data to train the model on.

Adjoint sensitivity analysis can be employed to decrease computational complexity (see Chapter 2.3.5) (Fröhlich et al., 2017). Furthermore, parallelization reduces the wall-time for optimization (Fröhlich et al., 2018; Penas et al., 2015). Complementary, advances in experimental measurement techniques have enabled the generation of large-scale pharmaceutical, transcriptomic and proteomic datasets (Barretina et al., 2012; Eduati et al., 2017; Li et al., 2017). These datasets typically contain relative measurements and unknown scaling and offset parameters are needed to map the model output to the measured quantities which increases the dimensionality of the optimization problem. As the measurements are noise corrupted, additionally, noise parameters have to be introduced. If no reliable values for the measurement noise are available, e.g. because of low numbers of experimental replicates, these parameters also need to be estimated. For scaling and noise parameters, a hierarchical optimization method has been developed to reduce the dimension of the parameter search space by analytically calculating the conditionally optimal scaling and noise parameters during optimization of the model parameters (Loos et al., 2018; Weber et al., 2011).

In addition to scaling parameters, offsets are frequently needed, e.g. to model background noise or when certain normalization, such as median subtraction, is applied. Still, so far offset parameters have not been considered in the hierarchical optimization approaches. Gradient calculation using forward sensitivity analysis can easily be combined with hierarchical optimization (Loos et al., 2018). However, the combination with adjoint sensitivity analysis remained unclear as the adjoint state explicitly depends on the observable parameters, hindering the application of this approach to large-scale models.

In this chapter, we (i) derive the analytical solutions of the optimal observable and noise parameters for the combination of offset and scaling parameters for additive Gaussian noise and (ii) extend the hierarchical optimization approach to be used in combination with adjoint sensitivity analysis. We apply this approach to the large-scale pan-cancer signaling model from Fröhlich et al. (2018). We first provide a conceptual explanation of the impaired optimizer performance when scaling parameters are introduced, which can be improved by using hierarchical optimization. We then show that the hierarchical approach yields speedups of more than one order of magnitude and that it can facilitate the unbiased integration of heterogeneous datasets for parameter inference. Therefore, the method presented in this chapter addresses the open challenge (i) stated in Section 1.3.

This chapter is based on and in part identical to the following publication:

- **Schmiester, L.***, Schälte, Y.*, Fröhlich, F., Hasenauer, J. & Weindl, D. (2019). Efficient parameterization of large-scale dynamic models based on relative measurements. *Bioinformatics*, 36 (2), 594-602.

3.1 Background

Often, measurements are relative to the concentrations of the considered species (Degasperi et al., 2017; Loos et al., 2018; Weber et al., 2011) or underwent some type of unknown and irreversible normalization (Barretina et al., 2012; Li et al., 2017). In these cases, the model simulation needs to be rescaled to compare it with the data. This is often either done by data-driven normalization (Degasperi et al., 2017), where e.g. a measurement relative to a control is modeled by dividing with the simulation of the control, or by the introduction of scaling and offset parameters. While the first approach does not change the dimension of the optimization problem, the statistical interpretation can be difficult as it results in non-trivial measurement error distributions (Loos et al., 2018). Therefore the latter approach is often preferred.

3.1.1 Relative measurements

When measurement techniques with a non-absolute, but linear relationship to the underlying quantities are employed, the model output can be mapped to the measurement by using scaling $s \in \mathbb{R}^{n_s}$ and offset parameters $b \in \mathbb{R}^{n_b}$ via the measurement process function

$$g(h_{i_y}(x(t_{i_t}, \theta, u_{i_u}), \theta, u_{i_u}), \xi_{i_t,i_y,i_u}) = s_{i_t,i_y,i_u} \cdot h_{i_y}(x(t_{i_t}, \theta, u_{i_u}), \theta, u_{i_u}) + b_{i_t,i_y,i_u}. \tag{3.1}$$

The observable parameters are then the collection of scaling and offset parameters $\xi_{i_t,i_y,i_u} = (s_{i_t,i_y,i_u}, b_{i_t,i_y,i_u})$.

In the following, we collect w.l.o.g. the indices for observables i_y, timepoints i_t and inputs i_u in one general index $i = (i_t, i_y, i_u) \in \mathcal{I}$, where $\mathcal{I}$ is the collection of all datapoint indices. Usually, different measurements share the same scaling, offset or noise parameters, e.g. if the data is collected in the same experiment. We introduce the index sets

$$\mathcal{I}^s_{i_s}, \mathcal{I}^b_{i_b}, \mathcal{I}^\sigma_{i_\sigma} \subset \mathcal{I}, \tag{3.2}$$

for $i_s = 1, \ldots, n_s$, $i_b = 1, \ldots, n_b$ and $i_\sigma = 1, \ldots, n_\sigma$, with n_s, n_b, n_σ denoting the numbers of scaling, offset and sigma parameters, respectively. These sets contain all indices of datapoints which are associated with the same observable or noise parameter. Common assumptions which we will also follow here are

$$\{\mathcal{I}^s_{i_s}\}_{i_s} = \{\mathcal{I}^b_{i_b}\}_{i_b} \tag{3.3}$$

$$\forall i_s \exists i_\sigma : \mathcal{I}^s_{i_s} \subset \mathcal{I}^\sigma_{i_\sigma}, \tag{3.4}$$

i.e. offset and scaling parameters share the same datapoints and datapoints that share a scaling parameter also share the noise parameter. If σ is know, e.g. if it can be calculated from experimental replicates, the second assumption can be dropped.

Notation: For ease of notation we will skip some dependencies of h and write $h_i(\theta) = h_{i_y}(x(t_{i_t}, \theta, u_{i_u}), \theta, u_{i_u})$ in the following.

3.1.2 Hierarchical optimization

The optimization problem of finding the maximum likelihood estimate for relative data is given by

$$\min_{\theta,s,b,\sigma} \ell(\theta, s, b, \sigma), \tag{3.5}$$

with the negative log-likelihood function

$$\ell(\theta, s, b, \sigma) = \frac{1}{2} \sum_{i \in \mathcal{I}} \log(2\pi\sigma_i^2) + \left(\frac{\bar{y}_i - s_i \cdot h_i(\theta) + b_i}{\sigma_i} \right)^2. \tag{3.6}$$

The standard optimization approach aims at numerically optimizing all parameters simultaneously, which results in an increased dimensionality of the optimization problem compared to absolute data. The hierarchical approach developed by Loos et al. (2018); Weber et al. (2011) makes use of the underlying structure of the optimization problem by splitting it into an outer optimization, where the dynamic parameters θ are optimized, and an inner optimization for the remaining observable and noise parameters. The existing approaches only considered trivial offsets $b = 0$. The optimization problem can then be written as

$$\min_{\theta} \ell(\theta, s(\theta), \sigma(\theta)) \tag{3.7}$$

$$\text{s.t. } (s(\theta), \sigma(\theta)) = \arg\min_{s,\sigma} \ell(\theta, s, \sigma). \tag{3.8}$$

This problem can be solved hierarchically by estimating the model parameters θ numerically in the outer optimization (3.7) and at each iteration of the outer optimization, calculating the conditionally optimal observable parameters s and noise parameters σ in the inner optimization (3.8). While splitting the optimization problem into two subproblems seems more involved, the inner problem does not involve numerically solving the ODE and analytical formulas for the optimal scaling and noise parameters can be derived. This can greatly improve computation times, as it reduces the dimensionality of the numerical optimization problem. The analytical formulas are calculated using the optimality conditions from Section 2.3.1 and are given by (see Loos et al. (2018); Weber et al. (2011) for the derivations)

$$s_{i_s}(\theta) = \frac{\sum_{i \in \mathcal{I}^s_{i_s}} \bar{y}_i \cdot h_i(\theta)}{\sum_{i \in \mathcal{I}^s_{i_s}} h_i(\theta)^2} \tag{3.9}$$

$$\sigma^2_{i_\sigma}(\theta) = \frac{\sum_{i \in \mathcal{I}^\sigma_{i_\sigma}} (\bar{y}_i - s_i h_i(\theta))^2}{|\mathcal{I}^\sigma_{i_\sigma}|}, \tag{3.10}$$

with $|\mathcal{I}^\sigma_{i_\sigma}|$ denoting the cardinality of $\mathcal{I}^\sigma_{i_\sigma}$.

The gradient of the outer problem can be calculated with the optimal scaling and noise parameters conditioned on θ via (see Loos et al. (2018); Weber et al. (2011))

$$\frac{\partial \ell(\theta, s, b, \sigma)}{\partial \theta_k} = \sum_{i \in \mathcal{I}} \frac{\bar{y}_i - s_i(\theta) h_i(\theta)}{\sigma_i^2(\theta)} \cdot s_i(\theta) \cdot s_k^{y_i}(\theta), \tag{3.11}$$

where here, $s_k^{y_i}$ denotes the observable sensitivities for the unscaled observables, i.e. with $s = 1$, which can be obtained by forward sensitivity analysis.

3.2 Hierarchical optimization for large-scale models

Large-scale models require a large amount of measurement data for parameter estimation. These measurements are usually relative and both scaling and offset parameters often need to be introduced. Here we first derive analytical formulas for the hierarchical optimization method when offsets are present in addition to scalings. Then we outline an approach to combine hierarchical optimization with adjoint sensitivity analysis.

3.2.1 Analytical formulas for optimal offset and scaling parameters

We extend the approach from Weber et al. (2011) and Loos et al. (2018) to the general case of relative data of the form (3.1) combining offset b and scaling parameters s. Including offset parameters, the hierarchical optimization problem is given by

$$\begin{aligned} &\min_{\theta} \ell(\theta, s(\theta), b(\theta), \sigma(\theta)) \\ &\text{s.t. } (s(\theta), b(\theta), \sigma(\theta)) = \arg\min_{s,b,\sigma} \ell(\theta, s, b, \sigma). \end{aligned} \tag{3.12}$$

Similar to the derivations in Loos et al. (2018); Weber et al. (2011), analytical formulas for the conditionally optimal scaling and offset parameters can be calculated via the necessary optimality conditions from Section 2.3.1. By differentiating (3.6) w.r.t. s or b, we obtain for scaling parameters

$$0 = \frac{\partial \ell(\theta, s, b, \sigma)}{\partial s_{i_s}} = -\sum_{i \in \mathcal{I}^s_{i_s}} \frac{\bar{y}_i - (s_{i_s} h_i(\theta) + b_i)}{\sigma_i^2} h_i(\theta) \tag{3.13}$$

and likewise for offset parameters

$$0 = \frac{\partial \ell(\theta, s, b, \sigma)}{\partial b_{i_b}} = -\sum_{i \in \mathcal{I}^b_{i_b}} \frac{\bar{y}_i - (s_i h_i(\theta) + b_{i_b})}{\sigma_i^2}. \tag{3.14}$$

Solving these equations for s and b respectively yields

$$s_{i_s} = \left(\sum_{i \in \mathcal{I}^s_{i_s}} \frac{h_i^2(\theta)}{\sigma_i^2} \right)^{-1} \left(\sum_{i \in \mathcal{I}^s_{i_s}} \frac{(\bar{y}_i - b_i) h_i(\theta)}{\sigma_i^2} \right) \tag{3.15}$$

$$b_{i_b} = \left(\sum_{i \in \mathcal{I}^b_{i_b}} \frac{1}{\sigma_i^2} \right)^{-1} \left(\sum_{i \in \mathcal{I}^b_{i_b}} \frac{\bar{y}_i - s_i h_i(\theta)}{\sigma_i^2} \right). \tag{3.16}$$

If only scalings or offsets are to be estimated in the inner optimization loop, e.g. if there are no offsets ($b = 0$) or no scalings ($s = 1$) present, these formulas can readily be used and we are done. If both parameters have to be estimated, then formula (3.15) can be inserted into (3.16). With assumption (3.3) we have $i_s = i_b$ and $\mathcal{I}^s_{i_s} = \mathcal{I}^b_{i_b}$. This then gives

$$\left[1 - \left(\sum_i \frac{1}{\sigma_i^2} \right)^{-1} \left(\frac{\sum_i \frac{h_i(\theta)}{\sigma_i^2} \sum_j \frac{h_j(\theta)}{\sigma_j^2}}{\sum_i \frac{h_i^2(\theta)}{\sigma_i^2}} \right) \right] b_{i_b} = \left(\sum_i \frac{1}{\sigma_i^2} \right)^{-1} \left(\sum_i \frac{\bar{y}_i}{\sigma_i^2} - \frac{\sum_i \frac{\bar{y}_i h_i(\theta)}{\sigma_i^2} \sum_j \frac{h_j(\theta)}{\sigma_j^2}}{\sum_i \frac{h_i^2(\theta)}{\sigma_i^2}} \right), \tag{3.17}$$

where we always sum over i or $j \in \mathcal{I}^s_{i_s}$. Dividing by the factor in front of b on the left hand side results in

$$b = \frac{\left(\sum_i \frac{1}{\sigma_i^2} \right)^{-1} \left(\sum_i \frac{\bar{y}_i}{\sigma_i^2} - \frac{\sum_i \frac{\bar{y}_i h_i(\theta)}{\sigma_i^2} \sum_j \frac{h_j(\theta)}{\sigma_j^2}}{\sum_i \frac{h_i^2(\theta)}{\sigma_i^2}} \right)}{\left[1 - \left(\sum_i \frac{1}{\sigma_i^2} \right)^{-1} \left(\frac{\sum_i \frac{h_i(\theta)}{\sigma_i^2} \sum_j \frac{h_j(\theta)}{\sigma_j^2}}{\sum_i \frac{h_i^2(\theta)}{\sigma_i^2}} \right) \right]}. \tag{3.18}$$

The scaling parameters s can then be calculated by inserting (3.18) into equation (3.15).

Analytical formulas for the standard deviations σ can similarly be derived by differentiating ℓ w.r.t. σ^2

$$\begin{aligned} 0 = \frac{\partial \ell(\theta, s, b, \sigma)}{\partial \sigma_{i_\sigma}^2} &= \frac{1}{2} \sum_{i \in \mathcal{I}^\sigma_{i_\sigma}} \frac{1}{\sigma_{i_\sigma}^2} - \frac{(\bar{y}_i - (s_i h_i(\theta) + b_i))^2}{(\sigma_{i_\sigma}^2)^2} \\ \Rightarrow \sigma_{i_\sigma}^2 &= \frac{\sum_{i \in \mathcal{I}^\sigma_{i_\sigma}} (\bar{y}_i - (s_i h_i(\theta) + b_i))^2}{|\mathcal{I}^\sigma_{i_\sigma}|}. \end{aligned} \tag{3.19}$$

Here we assumed, that the factor on the left hand side of (3.17), σ and $\sum_i h_i^2$ are all non-zero. Otherwise, the data is not sufficient to uniquely determine the observable and noise parameters and the data or the modeling approach should be revised. Calculation of the second derivatives reveals that the sufficient optimality conditions are fulfilled (see

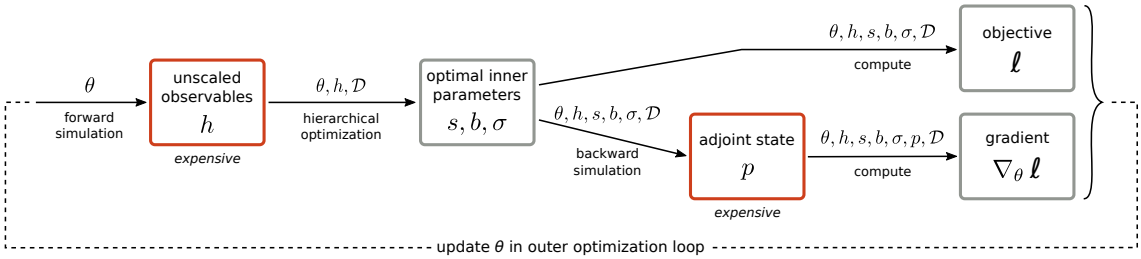


Figure 3.1: Illustration of the proposed combination of the hierarchical approach with adjoint sensitivity analysis. The parameters θ are optimized in an outer loop. In the inner loop, first the unscaled observables are computed. From this the optimal inner parameters are calculated analytically. The inner loop is then completed by simulating the adjoint state, if gradients are to be computed. The figure is a modified version of Figure 1 in the author's publication (Schmiester et al., 2019).

Section 2.3.1) and s, b and σ are indeed (global) minima:

$$\begin{aligned}
\frac{\partial^2 \ell(\theta, s, b, \sigma)}{\partial^2 s_{i_s}} &= \sum_{i \in \mathcal{I}^s_{i_s}} \frac{h_i^2(\theta)}{\sigma_i^2} > 0 \\
\frac{\partial^2 \ell(\theta, s, b, \sigma)}{\partial^2 b_{i_b}} &= \sum_{i \in \mathcal{I}^b_{i_b}} \frac{1}{\sigma_i^2} > 0 \\
\frac{\partial^2 \ell(\theta, s, b, \sigma)}{\partial^2 \sigma^2_{i_\sigma}} &= \frac{|\mathcal{I}^\sigma_{i_\sigma}|}{2\sigma^4_{i_\sigma}} > 0
\end{aligned} \tag{3.20}$$

3.2.2 Combining adjoint sensitivity analysis and hierarchical optimization

The hierarchical approach can easily be combined with forward sensitivity analysis by

(i) simulating the ODE with forward sensitivities for the unscaled observable

(ii) calculating the optimal inner parameters

(iii) computing the gradient via (3.11) (Loos et al., 2018).

Employing adjoint sensitivity analysis, we can reformulate the gradient of the objective function to (see Section 2.3.5)

$$\frac{\partial \ell(\theta, s(\theta), b(\theta), \sigma(\theta))}{\partial \theta_k} = -\sum_{i \in \mathcal{I}} \frac{\bar{y}_i - (s_i h_i(\theta) + b_i)}{\sigma_i^2} s_i \frac{\partial h(\theta)}{\partial \theta_k} - p(t_0)^T \frac{\partial x_0(\theta)}{\partial \theta_k} - \int_{t_0}^{t_{n_t}} p^T \frac{\partial f(\theta)}{\partial \theta_k} dt. \tag{3.21}$$

For relative data, we observe that the initial condition of the adjoint state p explicitly depends on the observable parameters. p satisfies $p(t) = 0$ for $t > t_{n_t}$ and on $(t_j, t_{j+1}]$, p

Table 3.1: Overview of optimization approaches. Objective-based optimization refers to methods that only rely on the full objective function in contrast to least-squares based optimization which is based on the residuals. Scalable here refers to the scaling behavior w.r.t. the number of parameters. This table is a modified version of Table 1 in the supplementary data of Schmiester et al. (2019).

	scalable	handle observable parameters efficiently	objective based optimization	least-squares based optimization	improved convergence
standard, forward			✓	✓	
standard, adjoint	✓		✓		
hierarchical, forward		✓	✓	✓	✓
hierarchical, adjoint	✓	✓	✓		✓

fulfills the backward differential equation

$$\begin{aligned} \dot{p} &= -\frac{\partial f^T}{\partial x} p \\ p(t_{j+1}, \theta) &= \lim_{t \to t_{j+1}} p(t) + \sum_{i: t_i = t_j} \frac{\bar{y}_i - s_i h_i(\theta) + b_i}{\sigma_i^2} s_i \frac{\partial \tilde{h}_i(\theta)}{\partial x}, \end{aligned} \tag{3.22}$$

with the scaled observable $\tilde{h}(\theta) = s \cdot h(\theta) + b$ and the sum only taken over summands involving t_i. As the adjoint state depends on the observable and noise parameters, the approach for forward sensitivities is not applicable in the adjoint case. Here, we propose a new scheme that circumvents this problem (Figure 3.1). In the outer optimization loop, the parameters θ are iteratively updated as is also done by Loos et al. (2018). In the inner loop, we first simulate the unscaled observable without sensitivities and calculate the conditionally optimal parameters $s(\theta), b(\theta)$ and $\sigma(\theta)$, which can be done using the analytical formulas derived in section 3.2.1. We then treat the optimal inner parameters as fixed inputs and simulate the adjoint state p of the scaled observable which can be used to calculate the gradients. This scheme enables the efficient handling of observable and noise parameters using the hierarchical optimization while using scalable gradient calculation via adjoint sensitivity analysis.

An overview of the advantages and disadvantages of the different methods is depicted in Table 3.1.

3.3 Implementation

We implemented the hierarchical approach and the combination with adjoint sensitivity analysis as part of the C++ library parPE. parPE provides functionality for parameter estimation, exploiting parallelization of ODE simulations and multistarts. Internally, it uses AMICI (Fröhlich et al., 2017) for ODE simulation and sensitivity calculation and interfaces multiple optimization algorithms for parameter estimation. For ease of implementation, we also calculated the forward simulation of the ODE in the adjoint state step (Figure 3.1). In addition to the optimization routines provided by parPE, we also implemented a Matlab objective function which uses hierarchical optimization that can be optimized with the Matlab toolbox PESTO (Stapor et al., 2018b). All code used for the study in this chapter is available via Zenodo at `http://doi.org/10.5281/zenodo.3254429` and `http://doi.org/10.5281/zenodo.3254441`.

3.4 Application to a large-scale pan-cancer model

Here, we consider the pan-cancer model developed by Fröhlich et al. (2018). The model comprises 1396 biochemical species, mainly proteins and protein complexes, and 4232 unknown parameters. Seven cancer drugs are implemented enabling simulation of different drug treatments. The model can be individualized to specific cancer cell-lines using genetic profiles and gene expression data. Solely trained on drug response data for single-drug treatments, the model demonstrated promising performance to predict the outcome of drug combinations (Fröhlich et al., 2018). Recently, it was shown that the model can be used to optimize drug combination treatments (Schmucker et al., 2020). We anticipated that additional data on the molecular level could provide complementary information during parameter estimation to improve predictive power even further and reduce parameter uncertainties. Motivated by this, we set out to integrate additional measurements used for model training.

3.4.1 Integration of heterogeneous datasets

The pan-cancer model was individualized using sequencing data from the Cancer Cell Line Encyclopedia (CCLE) (Barretina et al., 2012). So far, for model training, only a subset of the viability data from CCLE was used (*Dataset 1*, Table 3.2). This viability dataset contained measurements on 96 cancer cell-lines in response to 7 drugs at 8 drug concentrations (Barretina et al., 2012). The measurements are normalized to the respective control, and therefore, relative. To take this into account, we used cell-line specific scaling

parameters ($s_{\text{cell-line}_{i_s}}$). This yields the observable

$$y_{\text{viability}_i} = s_{\text{cell-line}_{i_s}} \cdot h_{\text{viability}_i} + \varepsilon_{\text{viability}_i}, \text{ with } i \in \mathcal{I}^s_{i_s}, \tag{3.23}$$

where $h_{\text{viability}}$ is a combination of weighted species that are assumed to determine cell viability (Fröhlich et al., 2018).

We complemented the viability dataset with molecular measurements from the MD Anderson Cell Lines Project (MCLP) (Li et al., 2017) (*Dataset 2*, Table 3.2), which contains reverse phase protein array (phospho-)proteomic measurements for various untreated cancer cell-lines. We identified 32 proteins and 16 phospho-proteins measured that were also covered by the model and in total 54 cell-lines which overlapped with the 96 cell-lines from *dataset 1*. In the MCLP database, measurements are normalized across cell-lines and across all proteins by subtracting the respective median from the $\log_2$-transformed measured values. We modeled this using cell-line specific offsets $b_{\text{cell-line}_{j_b}}$ and protein-specific offsets $b_{\text{protein}_{i_b}}$ yielding the observable

$$y_{\text{protein}_i,\text{cell-line}_j} = \log_2(h_{\text{protein}_i,\text{cell-line}_j}) + b_{\text{cell-line}_{j_b}} + b_{\text{protein}_{i_b}} + \varepsilon_{\text{protein}_i,\text{cell-line}_j}. \tag{3.24}$$

Here, h is the simulated absolute protein concentration

$$h_{\text{protein}_i,\text{cell-line}_j} = \sum_{l \in I_{\text{protein}_i}} k_l x_l. \tag{3.25}$$

The index set I_{protein_i} refers to the species that include protein_i and k_l is the respective stoichiometric multiplicity.

Table 3.2: Datasets used for parameter estimation. The number of observable and noise parameters of certain classes is indicated, followed by the number of parameters that are computed analytically in the hierarchical setting in parentheses. The table is taken from the author's publication (Schmiester et al., 2019).

	Dataset 1 (CCLE)	**Dataset 2 (MCLP)**
# datapoints	5281	1799
# cell-lines	96	54
# observables	1	48
# scalings	96(96)	0
# offsets	0	102(48)
# noise parameters	1(1)*	48(48)

*The noise parameter is set to one if dataset 1 is considered individually.

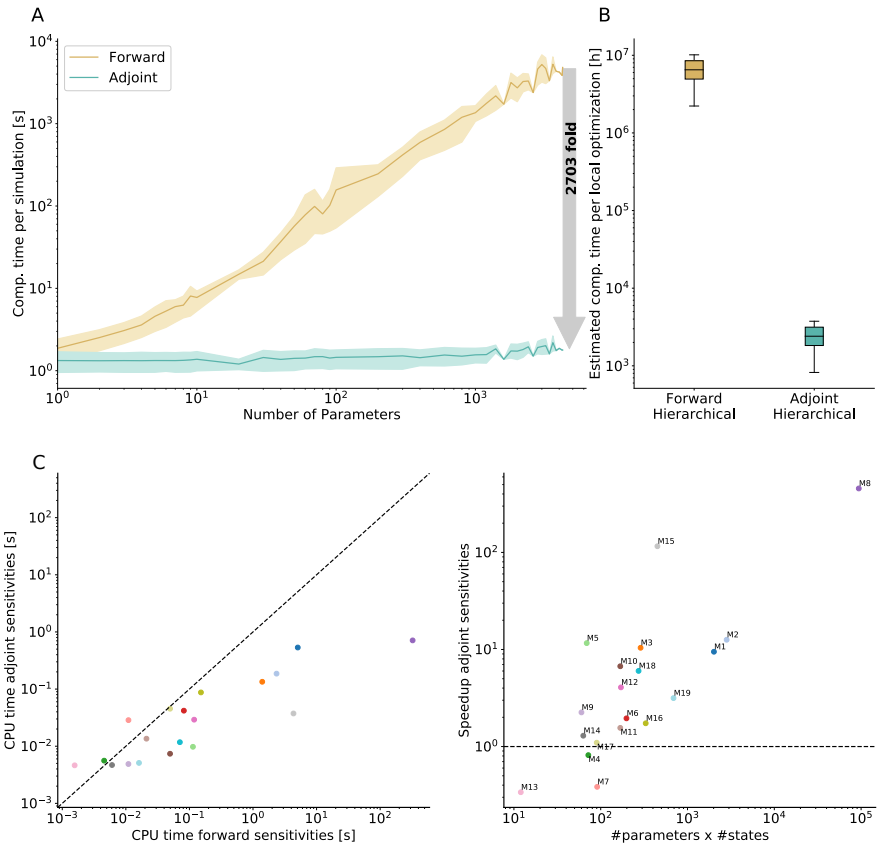


Figure 3.2: Comparison of computation times for forward and adjoint sensitivity analysis. (A): Estimated computation times when sensitivities are computed for varying numbers of parameters for the pan-cancer model. (B): Estimated computation times per local optimization of the full pan-cancer model based on the CPU times of previous optimizations. (C): Computation times and speedups for a collection of 19 models taken from `https://github.com/Benchmarking-Initiative/Benchmark-Models-PEtab`. Subfigures A and B are modified versions of the Supplementary Figure S1 of the author's publication (Schmiester et al., 2019).

The integration of viability and molecular measurements provides information on two different levels, which potentially improves the reliability of the model. However, it requires a substantial number of observable and noise parameters (Table 3.2).

3.4.2 Adjoint sensitivity analysis facilitates parameterization of large-scale models

First, we analyzed the computation times for adjoint sensitivity analysis compared to forward sensitivities (Figure 3.2). As computation times for forward sensitivities for the pan-cancer model were prohibitively large, we extrapolated the time needed for a full gradient

evaluation using a randomly sampled subset of the different simulation conditions. We then calculated forward and adjoint sensitivities for a subset of the parameters ranging from one to all dynamic parameters. This yielded an estimated speedup for adjoint sensitivity analysis of a factor of 2703 (Figure 3.2A). From this, we estimated the computation times needed for a full optimization run resulting in approximately 700 years of CPU time, if forward sensitivity analysis would be employed. The computation times were estimated based on the number of gradient evaluations of previous optimizations with adjoint sensitivity analysis. We also compared the computation times for a collection of 19 models derived from a benchmark collection of parameter estimation problems established in Hass et al. (2019) ranging from small to medium-sized models (Figure 3.2, cf. Section 5.2.4 for details on the models). For this collection, we observed that adjoint sensitivities were computationally less demanding for 16 out of the 19 models with substantial speedups in most cases, indicating that the here derived combination of hierarchical optimization with adjoint sensitivity analysis can benefit a broad range of models.

3.4.3 Evaluation of standard and hierarchical optimization using simulated data

It is a priori not clear how much the introduction of scaling, offset and noise parameters influence optimizer performance. In Degasperi et al. (2017) it was observed on two examples that scaling parameters led to inferior optimization results compared to the data-driven normalization that was also employed in Fröhlich et al. (2018). To analyse the influence of scaling parameters systematically, we first used simulated data for parameter estimation. The simulated dataset we considered was a simulation of the same conditions and observables as in *dataset* 1 using parameters from previous optimization runs. To obtain a realistic setting, we added normally distributed measurement noise to the simulated data. The noise levels were based on the estimated standard deviations obtained from the previous optimization. We generated simulated datasets for relative ($\bar{y}_{\text{rel}}$) and absolute ($\bar{y}_{\text{abs}}$) data via

$$\begin{aligned} \bar{y}_{\text{rel}} &= \hat{s} \cdot h(\hat{\theta}) + \varepsilon_{\text{rel}}, \quad \varepsilon_{\text{rel}} \sim \mathcal{N}(0, \hat{\sigma}^2_{\text{viability}}) \\ \bar{y}_{\text{abs}} &= h(\hat{\theta}) + \varepsilon_{\text{abs}}, \quad \varepsilon_{\text{abs}} \sim \mathcal{N}\left(0, \left(\frac{\hat{\sigma}_{\text{viability}}}{\hat{s}}\right)^2\right), \end{aligned} \tag{3.26}$$

where $\hat{\theta}, \hat{s}, \hat{\sigma}$ are parameters obtained from optimization with relative viability and proteomic data (*dataset 1* and *dataset 2*). Using simulated data with known true parameters $\hat{\theta}, \hat{s}, \hat{\sigma}$ allowed to compare the goodness-of-fit and assess the information associated with relative data compared to absolute data.

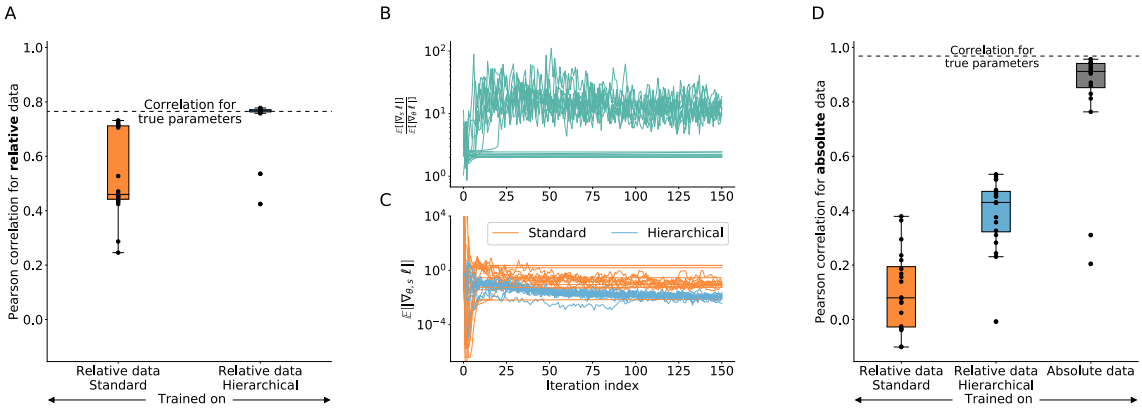


Figure 3.3: Parameter estimation results using a simulated version of *dataset 1.* (A): Pearson correlation coefficients for relative training data using parameters obtained from 20 local optimization runs using standard and hierarchical optimization. (B): Ratio of the average gradient contribution for scaling parameters against dynamic parameters using the standard optimization. (C): Expected gradient for standard and hierarchical optimization. Only parameters that were optimized numerically in the outer loop were considered. (D): Pearson correlation coefficients for absolute data using parameters from optimization on relative (left and middle) and absolute (right) data. This figure is a modified version of Figure 2 of the author's publication (Schmiester et al., 2019).

Hierarchical optimization outperforms standard optimization

We performed parameter estimation for the simulated, relative dataset with standard and hierarchical optimization using the gradient-based Interior Point OPTimizer (Ipopt) (Wächter & Biegler, 2006). To this end, we performed parameter estimation for 20 local starts using up to 150 optimizer iterations and analyzed the Pearson correlation between data and simulation for the best parameters of the 20 starts.

The correlation coefficients obtained from hierarchical optimization were substantially better than the standard optimization for almost all runs (Figure 3.3A). Most local optimizations using the hierarchical approach achieved similar correlations compared to the ones using the true parameters that were used to simulate the dataset. As no optimized parameter found much better correlations than the true parameters, overfitting could not be observed in this analysis. In contrast to hierarchical optimization, the standard optimization resulted in correlations far lower than the correlation with the true parameters.

Disproportional contribution of scalings to gradient can explain improved optimizer convergence

The substantial improvement using hierarchical optimization cannot solely be explained by the reduced dimensionality of the parameter search space. In fact, the hierarchical optimization decreased the number of optimization parameters by only 2%. We therefore hypothesized that the scaling parameters are particularly relevant. To assess this, we evaluated the average absolute values of the gradient of the objective function for scaling parameters ($E[|\nabla_s \ell|]$) and dynamic parameters ($E[|\nabla_\theta \ell|]$). Investigating the ratio ($E[|\nabla_s \ell|]/E[|\nabla_\theta \ell|]$) revealed that in most runs the objective function is one order of magnitude more sensitive to the scaling parameters compared to dynamic parameters (Figure 3.3B). Consequently, removing the scalings from optimization yielded smaller total absolute gradient values which decreased faster compared to the standard optimization (Figure 3.3C).

Data normalization leads to loss of information

We next analyzed the information loss when using relative compared to absolute data. We performed parameter estimation using the absolute simulated data. To guarantee comparability, we used the same initial parameters as for the optimization with relative data. Additionally, we predicted the absolute data using previous parameters estimated on relative data and compared the correlation coefficients to the parameter estimation results from absolute data (Figure 3.3D). As expected, the parameters obtained from training on relative data yielded correlations far lower than those obtained from the true parameters, suggesting that information is lost in the normalization process. Interestingly, the hierarchical optimization again yielded improved correlations compared to the standard optimization, indicating the improved convergence properties.

3.4.4 Hierarchical optimization improves efficiency for all tested optimizers

We showed that the hierarchical optimization outperformed standard optimization on a synthetic dataset using the Ipopt optimizer. To provide a more comprehensive assessment of this result, we performed parameter estimation on the measured viability data from *dataset 1*. We considered four different, commonly used, gradient-based optimizers for this analysis, which implement different optimization algorithms:

(i) Ipopt (Wächter & Biegler, 2006): Interior-point line-search algorithm using the Armijo condition for acceptance of line-search steps.

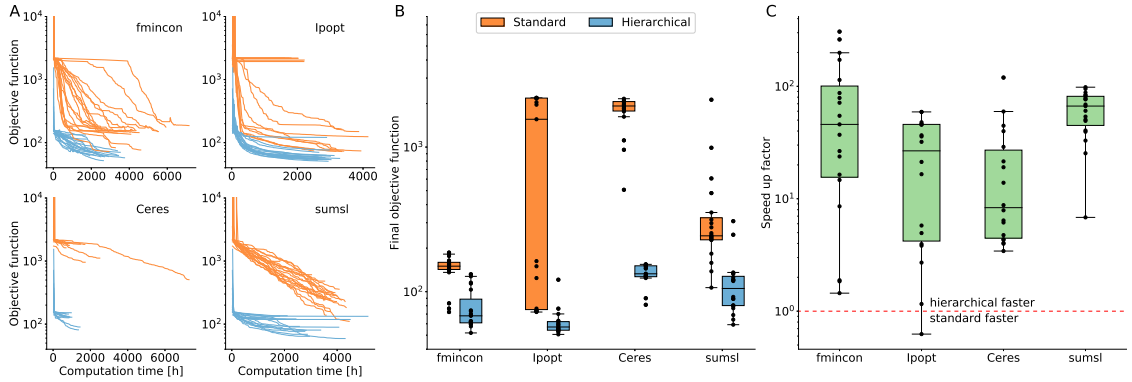


Figure 3.4: Parameter estimation results for standard and hierarchical optimization using *dataset 1*. 20 local optimizations were performed each with a maximum of 150 iterations. (A): Optimizer trajectories for all four employed optimization algorithms. (B): The final objective function values found by the different optimizers using standard and hierarchical optimization. (C): Speedup of the hierarchical approach compared to the standard approach. The speedup is defined by the time the hierarchical approach needs to find the best objective function value found by standard optimization. This figure is taken from Figure 3 of the author's publication (Schmiester et al., 2019).

(ii) Ceres (`http://ceres-solver.org`): Line-search algorithm using the strong Wolfe condition for acceptance of line-search steps.

(iii) sumsl (Gay, 1983): Trust-region algorithm using a quadratic model.

(iv) fmincon (with the optimizer option `interior-point`) (`https://de.mathworks.com/help/optim/ug/fmincon.html`): Interior-point trust-region method based on the algorithm described in Byrd et al. (2000).

As Ceres and sumsl did not natively support parameter bounds, we used a naive implementation returning failed objective function evaluations, if bounds were violated, which led to a rejection of a trial point outside of the bounds. We again performed 20 local optimizations with a maximum of 150 iterations for all optimizers and the standard and hierarchical approach. To be able to compare the results between the standard and hierarchical approach, the optimizations were started each from the same initial parameters for all optimizers.

We investigated the trajectories of the local optimizations, which is the evolution of the objective function over computation time. The hierarchical approach consistently achieved better objective function values given the same computation time with a generally lower variability between starts (Figure 3.4A). The hierarchical approach found substantially better final objective function values than the standard approach, across all employed

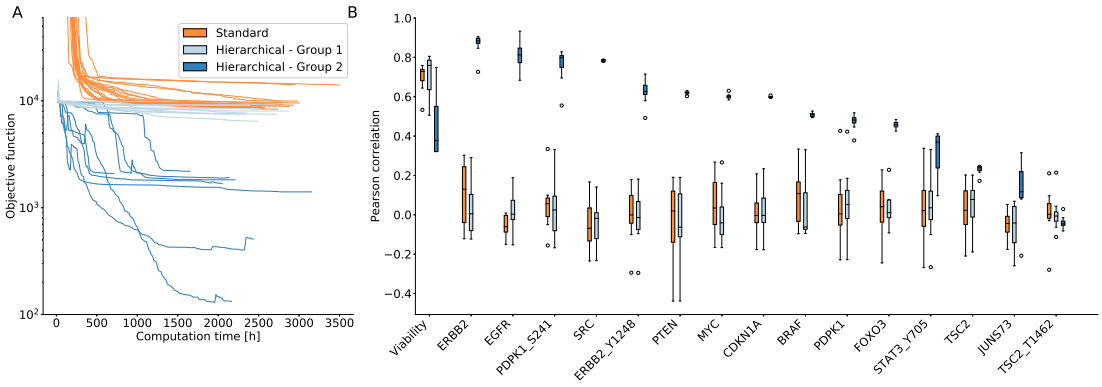


Figure 3.5: Parameter estimation results using both *dataset 1* and *dataset 2* for model training with Ipopt and 20 local optimization runs. (A): Optimizer trajectories for standard and hierarchical optimization. The two groups obtained from hierarchical optimization are indicated with different shades of blue. (B): Pearson correlation coefficients between model simulation and measured data for standard optimization and both groups of parameters found by hierarchical optimization. Only observables with at least 55 datapoints were considered. The figure is a modified version of the Figure 4 of the authors publication. Schmiester et al. (2019)

optimizers (Figure 3.4B). To quantify the computational benefit from using the hierarchical optimization, we calculated the computation time required by the hierarchical approach to reach the best objective function value of the standard approach (Figure 3.4C). For all but one local optimization, the hierarchical approach was computationally more efficient with average speedups between one and two orders of magnitude.

The performance of different optimization algorithms has so far not been assessed on models of this size. Our results indicate that there are large differences in optimizer performance and the choice of the optimizer is crucial for successful parameter estimation. We found that for the considered model and dataset, Ceres stopped prematurely in most cases, sumsl progressed slower than Ipopt and fmincon especially for standard optimization. Ipopt and fmincon appeared to be most efficient (Figure 3.4A & B). Interestingly, sumsl showed the largest improvement when using hierarchical optimization.

3.4.5 Hierarchical optimization enables integration of heterogeneous data

So far, the pan-cancer model was only trained on viability measurements (*dataset 1*) which provide limited information about molecular mechanisms. To improve this we complemented *dataset 1* with the (phospho-)proteomic measurements from MCLP (*dataset 2*). To guarantee an unbiased weighting of the different datasets, we introduced noise parameters for the newly included observables which resulted in 48 additional parameters for

standard deviations (one per (phospho-)protein) that needed to be estimated. Additionally, *dataset 2* introduced cell-line and observable specific offset parameters. As the hierarchical optimization method only supports optimizing one of these offsets analytically, we chose the observable-specific offsets and estimated (i) the cell-line specific scaling parameters of the viability observable (ii) the observable specific offset parameters of the (phospho-)proteomic dataset (iii) the noise parameters of both datasets hierarchically, which resulted in 193 parameters that were calculated in the inner optimization loop (Table 3.2).

We only performed optimization using Ipopt, as it was among the best performing optimization algorithms (Figure 3.4). Multi-start local optimization revealed again the superior convergence behavior of the hierarchical approach compared to the standard approach (Figure 3.5A). For most local optimizations, the hierarchical approach yielded better objective function values already after few iterations compared to the final values of the standard approach. Inspection of the final objective function values showed that all starts of the standard approach found similar values around 10^4. In contrast, the objective function values obtained from hierarchical optimization can be split into two groups. Group 1 yielded similar values as the standard approach of $\ell \approx 10^4$. Group 2 reached much better values with $\ell < 3 \times 10^3$.

The two groups of parameter vectors resulted in a substantially different agreement of model simulation with the experimental data (Figure 3.5B). The parameters in Group 1 obtained from standard and hierarchical optimization were able to fit the viability data but failed to describe the protein measurements with median correlation coefficients smaller than 0.2. The parameters from Group 2, which were only reached using the hierarchical optimization, showed a good agreement for both viability and protein measurements. Accordingly, only the hierarchical approach was able to find parameters that balanced the fit of the two datasets, thereby achieving a successful integration and a better overall description of the data.

3.5 Summary and discussion

Large-scale mechanistic models can provide a comprehensive representation of the underlying biological processes. To render these models predictive, they have to be trained on large heterogeneous and often relative datasets and efficient methods for model training are urgently needed. In this chapter, we extended the hierarchical optimization framework from Weber et al. (2011) and Loos et al. (2018) to allow for efficient handling of offset parameters in addition to scaling and noise parameters. To this end, we derived analytical formulas for the conditionally optimal observable parameters when offset and scaling parameters

are present. We then developed an approach to combine hierarchical optimization with scalable gradient computation using adjoint sensitivity analysis, which is essential for gradient-based optimization of large-scale models.

We evaluated the hierarchical optimization method on a large-scale pan-cancer model by Fröhlich et al. (2018). We first compared the hierarchical optimization to the standard optimization approach on simulated data which revealed substantially improved optimizer convergence using the hierarchical method. Only the hierarchical approach was able to obtain parameters which provided a similar goodness-of-fit compared to the true parameters. We identified that the contribution of the scaling parameters to the gradient is disproportionally large compared to the dynamic parameters, which could explain the large benefit of the hierarchical method. The numerical stiffness which can arise from this for numerical optimization methods is the first conceptual explanation of the vast improvements achieved by hierarchical methods. We then used real experimental data to analyze the optimizer behavior for the standard and hierarchical methods using four different commonly used optimization algorithms. We obtained median speedups of more than one order of magnitude, irrespective of the employed optimizer. Consequently, the hierarchical method resulted in superior objective function values given the same amount of computation time for all tested cases. We subsequently complemented the viability dataset already used in Fröhlich et al. (2018) with measurements on the molecular level for model training. These heterogeneous datasets could be integrated in an unbiased way by hierarchically estimating noise parameters with almost no computational overhead. Only the hierarchical method was able to obtain parameters which yielded a good fit of both the viability and the (phospho-)proteomic dataset. Consequently, the hierarchical approach is crucial for successful parameter estimation of large-scale mechanistic models using heterogeneous relative measurements.

So far, we only considered normally distributed measurement noise and other noise models cannot yet be handled by the here developed approach. It was shown that other noise models can be more robust e.g. to outliers while still maintaining good optimization convergence (Maier et al., 2017). An extension of this approach to other noise models would therefore be of interest, even if no analytical formulas for the inner problem can be derived. The current hierarchical framework is able to handle individual scalings, offsets or a combination thereof per observable. Already the (phospho-)proteomic dataset considered here for the large-scale model required two different offset parameters per observable and a more flexible approach handling multiple parameters would be valuable and could improve convergence even further. Also more complex observable transformations could benefit from hierarchical optimization. Recently, we and others have combined mechanistic ODE

models with mini-batch or stochastic gradient descent optimization, which only use a randomly chosen subset of the full dataset in each iteration to update the parameters (Stapor et al., 2019; Yuan et al., 2019). A combination of mini-batch and hierarchical optimization would require a more careful choice of the mini-batches but could accelerate optimization especially for very large datasets. We extended the training data for the pan-cancer model to also include measurements on the molecular level. However, we consider this study to be a proof-of-concept and the here considered datasets are not sufficient to obtain high-quality estimates of the model parameters. For more biology-driven analyses additional molecular data needs to be integrated to improve the predictive power of the model. With the advance of high-throughput technologies more large-scale datasets are being generated, like the cancer proteomic atlas (Li et al., 2013) or the datasets provided by Frejno et al. (2017); Gholami et al. (2013). Additionally, an updated version of the MCLP was recently published including proteomic measurements after drug treatment (Zhao et al., 2020).

In conclusion, we developed a flexible hierarchical optimization algorithm to handle different observable parameters for relative data and combined it with scalable adjoint sensitivity analysis for gradient computation. This approach facilitated efficient parameterization of large-scale mechanistic models and allowed an unbiased integration of heterogeneous datasets with almost no computation overhead.

Chapter 4

Robust and efficient parameter estimation using qualitative data

In Chapter 3, we have shown that models can be calibrated efficiently on relative measurements by introducing scaling and offset parameters and using adjoint sensitivity analysis combined with a hierarchical approach to reduce the dimensionality of the optimization problem. In some cases, however, one cannot safely assume proportionality between measurement and the underlying quantity of interest and instead, the measurement only provides a qualitative observation. Frequently encountered reasons for this are, that (i) the measurement is fundamentally qualitative, such as categorical characterizations or phenotypic observations (Chen et al., 2004) or (ii) an unknown non-linear relationship of the measured signal on the internal state of the system exist, as can be the case, among others, for Förster resonance energy transfer (FRET) data (Birtwistle et al., 2011) or stained images (Brooks et al., 2012; Pargett et al., 2014). Furthermore, non-linear relationships can occur due to detection limits and saturation effects (Butler et al., 2019). In these cases, an exact quantitative information on the considered system is not available. Yet, monotonicity between the measured species and the detected signal can be assumed. Therefore, if the measurement noise is neglected, the ordering of the datapoints is still preserved, resulting in qualitative observations. Even though qualitative data is less informative than quantitative data, it can still be valuable to infer parameters. However, qualitative observations are often neglected and most established parameter estimation toolboxes do not support to integrate them.

Some exceptions exist, where a general framework for parameter estimation using qualitative data in systems biology was developed. (i) Oguz et al. (2013) optimized the number of qualitative observations that were correctly captured by the model. (ii) Mitra et al. used qualitative observations as static penalty functions which were combined with a least-squares approach for quantitative data (Mitra et al., 2018). The method was subsequently implemented in the pyBioNetFit toolbox (Mitra et al., 2019). While the approach is comparably easy to implement, it relies on weights, which have to be defined prior to parameter estimation and which are hard to determine. Mitra & Hlavacek (2020) proposed a likelihood function that was used for Bayesian uncertainty quantification. However,

it remains unclear, if the statistical model that was used is appropriate for qualitative measurements and if parameter estimation using this function is hindered by gradients close to zero for parameters far away from an optimum, which is often the case when optimization is initialized at random parameters. (iii) Pargett & Umulis (2013) and Pargett et al. (2014) used a method termed *optimal scaling* established in statistics (Shepard, 1962) and applied it to models of biological systems. Here, the optimal quantitative representation of the qualitative observation is calculated. In this context, the quantitative representation is termed *surrogate data*. This results, similar to the approach from Chapter 3, in a hierarchical optimization problem, where the outer problem optimizes the parameters of the dynamic model and the inner problem is constrained by the measured qualitative behavior, such that inconsistencies of the model simulation with the qualitative measurement data are penalized (cf. Figure 4.1). While the optimal scaling approach is deeply grounded in statistical theory, it is computationally demanding, as a constrained optimization problem has to be solved repeatedly. The approaches (i)-(iii) facilitate the extraction of information about the model parameters from qualitative experimental data. Yet, the objective functions are either intrinsically discontinuous or an analytical formulation for the objective function gradient was unknown. Accordingly, only gradient-free optimization methods could be employed.

In this chapter, we build upon the optimal scaling approach by Pargett et al. (2014). We propose two reformulations of the optimal scaling problem that (i) reduce the dimensionality of the optimization problem and (ii) simplify the constraints of the problem. While the first reformulation improves computational efficiency, the second reformulation can be solved more robustly using a larger set of optimization algorithms. We derive proofs that the reformulations conserve the optimal points. These reformulations address the challenge (ii) stated in Chapter 1. We then derive an algorithm to calculate gradient information on the optimal scaling objective function, which facilitates the use of gradient-based optimization algorithms and tackles challenge (iii). We show on several application examples that the reformulations and the gradient-based approach yield more reliable results and substantially improve computational efficiency.

This Chapter is based on and in part identical with the following publications:

- **Schmiester, L.**, Weindl, D. & Hasenauer, J. (2020). Parameterization of mechanistic models from qualitative data using an efficient optimal scaling approach. *J. Math. Biol.*, 81, 603–623

- **Schmiester, L.**, Weindl, D. & Hasenauer, J. Efficient gradient-based parameter estimation for dynamic pathway models using qualitative data, *in preparation.*

4.1 Background

In this section, we introduce the optimal scaling approach used for parameter estimation of biological systems with qualitative measurements (Pargett et al., 2014). We provide the necessary notation and the optimization problem arising in this approach which is the basis for the extensions that are developed in the following sections.

4.1.1 Qualitative measurements

Qualitative data often arises, when the underlying measurement process function g is non-linear and unknown (Pargett & Umulis, 2013). Here, we denote qualitative readouts as $z(\theta, t, u)$. Then, for the transformation g from the underlying quantity y to the readout z, only monotonicity can be assumed. The qualitative measurements are potentially subject to noise

$$\bar{z}_{i_t,i_y,i_u} = z_{i_y}(\theta, t_{i_t}, u_{i_u}) + \nu_{i_t,i_y,i_u} \tag{4.1}$$

with measurement errors $\nu_{i_t,i_y,i_u} \sim \mathcal{N}(0, \sigma^2_{i_t,i_y,i_u})$, assuming normally distributed noise. Analog to Chapter 3, we use the general index $i = (i_t, i_y, i_u) \in \mathcal{I}$, with the number of datapoints $N = |\mathcal{I}|$ in the following. Qualitative measurements can either be ordered, i.e. $\bar{z}_i\{<,>\}\bar{z}_j$ or they can be indistinguishable, $\bar{z}_i \approx \bar{z}_j$. We use a similar notation as in Pargett et al. (2014) and introduce categories $\mathcal{C}_{i_k}$, with $i_k = 1, \ldots, n_k$. Qualitative measurements belonging to the same category are indistinguishable, i.e. $\bar{z}_i, \bar{z}_j \in \mathcal{C}_{i_k} \Rightarrow \bar{z}_i \approx \bar{z}_j$, and measurements belonging to different categories are ordered with respect to the ordering of the category. Without loss of generality, we assume that the categories are ordered as $\mathcal{C}_1 \prec \mathcal{C}_2 \cdots \prec \mathcal{C}_{n_k}$. We denote the index of the category to which the readout $\bar{z}_i$ belongs as $i_k(i)$. An example of qualitative data is illustrated in the upper left figure of Figure 4.1B.

4.1.2 Optimal scaling approach for parameter estimation with qualitative data

To estimate unknown model parameters, the optimal scaling approach considers the weighted least-squares function (2.12). The approach addresses the issue that no quantitative values for the measurements are available by introducing so-called surrogate data $\tilde{y}_i$, $i \in \mathcal{I}$. A surrogate datapoint is the optimal quantitative representation of the qualitative observation. Therefore, it provides the best agreement with the model simulation within the constraints provided by the qualitative data which is given by the ordering of the categories (Figure 4.1B). As a category $\mathcal{C}_{i_k}$ describes the set of observations that cannot be distinguished, it can be quantitatively represented by an interval $[l_{i_k}, u_{i_k}]$, with lower bound l_{i_k} and upper bound u_{i_k}. The surrogate datapoint $\tilde{y}_i$ affiliated with category $\mathcal{C}_{i_k(i)}$

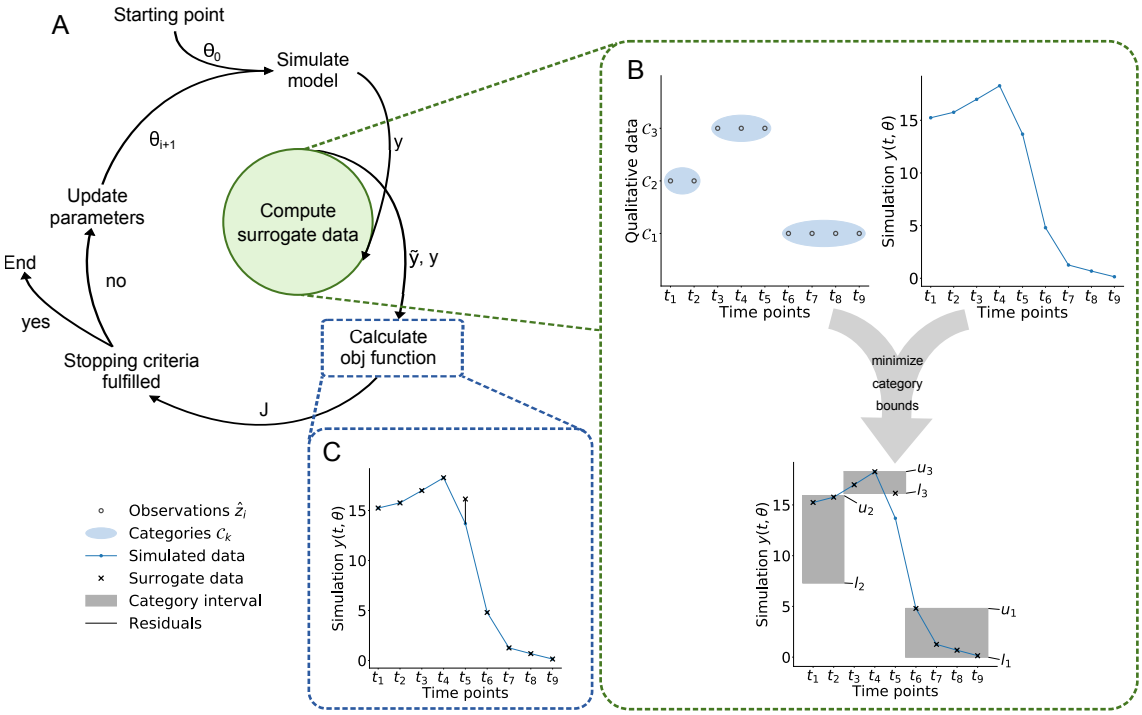


Figure 4.1: Illustration of the optimal scaling approach. (A): Individual steps of an optimization run. (B): Schematic of surrogate data calculation for a given simulation results $y(t, \theta)$ and set of qualitative data (with three categories). The interval between the optimized lower and upper bounds of the categories are indicated by grey areas. (C): Schematic of residuals used in the objective function for the parameter optimization. This figure is taken from Figure 1 of the author's publication (Schmiester et al., 2020b).

can then be placed anywhere within the respective interval. As the surrogate data and the interval bounds are a priori unknown, they are subject to optimization resulting in the problem

$$\begin{aligned}(\tilde{y}(\theta), l(\theta), u(\theta)) = \arg\min_{\tilde{y},l,u} \sum_{i\in\mathcal{I}} w_i(\tilde{y}_i - y_i(\theta))^2 \\ \text{s.t. } l_{i_k(i)} \leq \tilde{y}_i \leq u_{i_k(i)}, \quad i \in \mathcal{I} \\ u_{i_k} \leq l_{i_k+1}, \qquad i_k = 1, \ldots, n_k - 1.\end{aligned} \tag{4.2}$$

The first constraint here guarantees that the surrogate data is within the bounds imposed by the category and the second constraint assures that the ordering of the categories is fulfilled. In practice, the weights w are often chosen such that violations of the qualitative data are penalized independently of the scale of the simulation $y(\theta)$, e.g. by using $w_i = 1/y_i(\theta)$. If the simulated data matches the qualitative observations perfectly, the objective function becomes zero. Similarly to the hierarchical approach for relative data (Chapter 3), the model parameters θ can be estimated by nesting problem (4.2) into the optimization of θ

(Figure 4.1). This yields the optimization problem, which minimizes the distance between simulation and optimal surrogate data (Figure 4.1C):

$$\begin{aligned} &\min_{\theta} \sum_{i \in \mathcal{I}} w_i (\tilde{y}_i - y_i(\theta))^2 \\ &\text{s.t. } (\tilde{y}(\theta), l(\theta), u(\theta)) \quad \text{solve (4.2).} \end{aligned} \tag{4.3}$$

The observable parameters, in this case, are then the surrogate data, the lower bounds and the upper bounds. As pointed out in Pargett et al. (2014), the surrogate data can easily be computed from the bounds $l(\theta), u(\theta)$ using the observations that

(Case 1) If the model simulation $y_i(\theta)$ is smaller than the lower bound $l_{i_k(i)}(\theta)$, the surrogate data are set to the smallest feasible value to minimize the difference, i.e. the lower bound $\tilde{y}_i(\theta) = l_{i_k(i)}(\theta)$.

(Case 2) If the model simulation $y_i(\theta)$ is larger than the upper bound $u_{i_k(i)}(\theta)$, the surrogate data are set to the largest feasible value to minimize the difference, i.e. the upper bound $\tilde{y}_i(\theta) = u_{i_k(i)}(\theta)$.

(Case 3) If the model simulation $y_i(\theta)$ is in the interval $[l_{i_k(i)}(\theta), u_{i_k(i)}(\theta)]$, then the surrogate data are set to $\tilde{y}_i(\theta) = y_i(\theta)$. In this case, the error is zero.

With this, the surrogate data can be calculated analytically using the construction rule:

$$\tilde{y}_i(\theta) = \begin{cases} l_{i_k(i)}(\theta) & \text{, if } y_i(\theta) < l_{i_k(i)}(\theta) \\ u_{i_k(i)}(\theta) & \text{, if } u_{i_k(i)}(\theta) < y_i(\theta) \\ y_i(\theta) & \text{, otherwise.} \end{cases} \tag{4.4}$$

The inner optimization problem (4.2) can be simplified using this construction rule yielding the optimization problem

$$\begin{aligned} (l(\theta), u(\theta)) = \arg\min_{l,u} \sum_{i \in \mathcal{I}} w_i \left(\max\{0, l_{i_k(i)} - y_i(\theta)\}^2 + \max\{0, y_i(\theta) - u_{i_k(i)}\}^2 \right) \\ \text{s.t. } l_k \leq u_{i_k}, \qquad i_k = 1, \ldots, n_k \\ u_{i_k} \leq l_{i_k+1}, \quad i_k = 1, \ldots, n_k - 1. \end{aligned} \tag{4.5}$$

Here, only the lower and upper bounds need to be estimated numerically reducing the optimization parameters from $N + 2n_k$ to $2n_k$. In the objective function from (4.5), one of the two terms is always zero. For Case 2 and 3, $\max\{0, l_{i_k(i)} - y_i(\theta)\}^2$ vanishes and for Case 1 and 3, $\max\{0, y_i(\theta) - u_{i_k(i)}\}^2$ vanishes. Consequently, for Case 3 both terms

vanish and the respective summand is zero. This problem provides the same solution for the lower and upper bounds as (4.2). Hence, in the hierarchical optimization problem (4.3) we can solve for (4.5) instead of (4.2).

In the optimization problems (4.2) and (4.5), the qualitative data provide only limited information about the first lower bound l_1 of category $\mathcal{C}_1$ and the last upper bound u_{n_k} of category $\mathcal{C}_{n_k}$. The lower bound l_1 may be set to any value smaller or equal to the minimum of $y_i(\theta)$, $l_1 \leq \min_i y_i(\theta)$, and the upper bound u_{n_k} may be set to any value greater or equal to the maximum of $y_i(\theta)$, $u_{n_k} \geq \max_i y_i(\theta)$.

It commonly occurs, that data is collected in different experimental setups or for different biological species, where the qualitative relationship within a group of measurements is known, but between the different groups (e.g. different observables), the relations are unknown. This can be described by splitting the index set $\mathcal{I}$ into disjoint subsets $\mathcal{I}_\mathfrak{g}, \mathfrak{g} = 1, \ldots, n_\mathfrak{g}$, with the number of different groups $n_\mathfrak{g}$. The optimal scaling approach can be easily extended to this case by solving $n_\mathfrak{g}$ optimization problems of the form

$$\begin{aligned}(l(\theta), u(\theta)) = \arg\min_{l,u} \sum_{i \in \mathcal{I}_\mathfrak{g}} w_i \left(\max\{0, l_{i_k(i)} - y_i(\theta)\}^2 + \max\{0, y_i(\theta) - u_{i_k(i)}\}^2 \right) \\ \text{s.t. } l_k \leq u_{i_k}, \qquad i_k = 1, \ldots, n_k(\mathfrak{g}) \\ u_{i_k} \leq l_{i_k+1}, \quad i_k = 1, \ldots, n_k(\mathfrak{g}) - 1.\end{aligned} \tag{4.6}$$

The overall objective function value can then be calculated by summing up over the $n_\mathfrak{g}$ values obtained by solving (4.6). In the following, we will derive formulas for the case of one group, i.e. $n_\mathfrak{g} = 1$ and $\mathcal{I}_\mathfrak{g} = \mathcal{I}$, but the approaches can be trivially extended to the case of multiple groups as described above.

4.2 Reformulation of the optimal scaling problem

Compared to parameter estimation with quantitative data, the optimal scaling approach for qualitative data is computationally substantially more demanding as during optimization of the model parameters θ, a constraint optimization problem has to be solved numerically in each iteration. This is often time-consuming and efficient approaches for solving this are needed. Here, we propose two reformulations of the optimal scaling problem to simplify and accelerate the surrogate data optimization.

4.2.1 Optimal scaling problem can be reduced

We first derive a reformulation of the optimal scaling problem that is based on the empirical observation, that the gaps between lower and upper bounds $l_{i_k+1} - u_{i_k}$ of two adjacent categories are usually estimated as small as possible. Intuitively, this can be explained by the fact that the surrogate data can be placed anywhere within the intervals but not in the gaps between intervals. Increasing the gap will therefore either leave the objective function unchanged or even result in increased objective function values. This lead to the first result

Lemma 4.2.1. *The optimization problem* (4.5) *possesses an optimal solution* (l^*, u^*) *with* $u_k^* = l_{i_k+1}^*$ *for* $i_k = 1, \ldots, n_k - 1$.

Proof. Let us assume that there is an optimal solution (l', u'), such that the gap between lower and upper bound of two adjacent categories is larger than zero. W.l.o.g., we assume $u'_{i'_k} < l'_{i'_k+1}$ for some i'_k. For all observations belonging to category $\mathcal{C}_{i'_k}$, i.e. all indices i with $i_k(i) = i'_k$, it has to hold that

$$y_i(\theta) - u'_{i_k(i)} = y_i(\theta) - u'_{i'_k} < 0. \tag{4.7}$$

Otherwise, the objective function could be decreased by setting $u'_{i_k(i)} = l'_{i_k(i)+1}$ as then the summand

$$\begin{aligned} \max\{0, y_i(\theta) - u'_{i_k(i)}\}^2 &= \max\{0, y_i(\theta) - u'_{i'_k}\}^2 \\ &> \max\{0, y_i(\theta) - l'_{i_k(i)+1}\}^2 \end{aligned} \tag{4.8}$$

of the objective function would decrease. This would be a contradiction to the assumption that (l', u') is an optimal solution. As $y_i(\theta) - u'_{i'_k} < 0$, the corresponding summands of the objective function are zero, $\max\{0, y_i(\theta) - u'_{i'_k}\}^2 = 0$. Increasing $u'_{i'_k}$ to $l'_{i'_k+1}$ does not change this. □

An illustration of the idea behind lemma 4.2.1 is depicted in Figure 4.2. Lemma 4.2.1 implies that at least one solution of the optimization problem (4.5) exists with vanishing gaps between adjacent categories. This result can be used to replace the optimization problem (4.5) with a reduced problem:

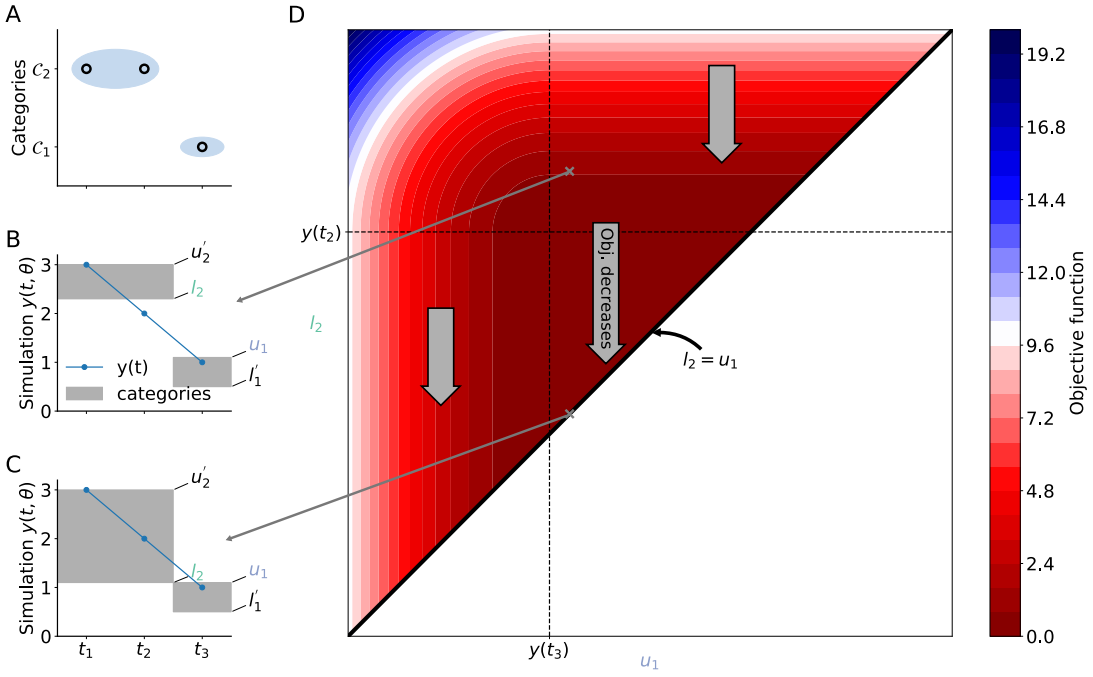


Figure 4.2: Illustration of the reduction of the optimal scaling optimization problem. (A): Example of qualitative data. (B & C): Different category intervals and surrogate data with non-zero gaps (B) and zero gap size (C). (D): Objective function landscape for the optimal scaling problem for the upper bound of $\mathcal{C}_1$ and the lower bound of $\mathcal{C}_2$. The optimal solution with zero gaps is given by the diagonal. This figure is taken from Figure 2 of the author's publication (Schmiester et al., 2020b).

Theorem 4.2.2. *An optimal solution of the optimization problem* (4.5) *is given by the solution to the problem*

$$\begin{aligned} u(\theta) = \arg\min_{u} \sum_{i \in \mathcal{I}} w_i \left(\max\left\{0, u_{i_k(i)-1} - y_i(\theta)\right\}^2 + \max\left\{0, y_i(\theta) - u_{i_k(i)}\right\}^2 \right) \\ s.t.\ u_{i_k} \leq u_{i_k+1}, \quad i_k = 0, \ldots, n_k - 1, \end{aligned} \tag{4.9}$$

for $u_0 = \min_i y_i(\theta)$ *and* $u_{n_k} = \max_i y_i(\theta)$*, and setting* $l_{i_k+1}(\theta) = u_{i_k}(\theta)$ *for* $i_k = 1, \ldots, n_k - 1$.

Proof. As a solution to (4.5) is given with minimal gaps between the category intervals, i.e. $u_{i_k} = l_{i_k+1}$, the reduced problem (4.9) can be obtained by substituting l_{i_k+1} with u_{i_k} and removing trivially fulfilled constraints. With this substitution, the first summand of (4.5), $\max\left\{0, l_{i_k(i)} - y_i(\theta)\right\}^2$ is replaced by $\max\left\{0, u_{i_k(i)-1} - y_i(\theta)\right\}^2$ yielding problem (4.9). □

Here, u_0 is not a bound of an additional category, but just an auxiliary variable for ease of notation. As only the upper bounds u have to be optimized numerically, this reformulation reduced the number of optimization variables by a factor of two.

4.2.2 Optimal scaling problem can be reformulated as box-constrained problem

The reduction introduced in the last chapter reduces the optimization variables and should therefore improve the computation time needed to solve the optimal scaling problem. Still, the algorithms that can be used for optimization are limited as the problem contains linear inequality constraints. As multiple algorithms only support simple box-constraints (see Section 2.3) and in our experience these algorithms often perform better, it could be valuable to simplify the constraints of the optimal scaling problem. We therefore introduce the vector of differences between adjacent upper bounds $d_{i_k} \coloneqq u_{i_k} - u_{i_k-1}$. For $l_{i_k} = u_{i_k-1}$, d_{i_k} is equivalent to the length of the interval belonging to category $\mathcal{C}_{i_k}$. With this we can rewrite the upper bounds of the categories as

$$u_{i_k} = u_0 + \sum_{i'_k=1}^{i_k} d_{i'_k}. \tag{4.10}$$

The auxiliary variable u_0 can be set to some value lower or equal to the minimum of $y_i(\theta)$, e.g. $u_0 = \min_i y_i(\theta)$. Replacing u with the new parameters d yields the reparameterized optimization problem

$$\begin{aligned} d(\theta) = \arg\min_{d} \sum_{i\in\mathcal{I}} w_i \left(\max\left\{0, u_0 + \sum_{i'_k=1}^{i_k(i)-1} d_{i'_k} - y_i(\theta)\right\}^2 + \max\left\{0, y_i(\theta) - u_0 - \sum_{i'_k=1}^{i_k(i)} d_{i'_k}\right\}^2 \right) \\ \text{s.t. } d_{i_k} \geq 0, \quad i_k = 1, \ldots, n_k - 1. \end{aligned} \tag{4.11}$$

The reparameterized problem (4.11) is equivalent to solving (4.9), only that the linear inequality constraints are replaced by positivity constraints. Hence, we can use a broader spectrum of optimization algorithms for the reparameterized problem.

4.2.3 Minimal category and gap sizes

As we have shown, an optimal solution to the optimal scaling problem is obtained with minimal gaps between intervals, i.e. $u_{i_k} = l_{i_k+1}$. With this, a surrogate datapoint at the upper bound of category i_k has a negligible difference to a datapoint at the lower bound of

category $i_k + 1$. Therefore, Pargett et al. (2014) enforced minimal gaps $g \in \mathbb{R}_+$ between intervals and minimal interval sizes $s \in \mathbb{R}_+$. With this, the optimization problem (4.5) can be modified to the problem

$$\begin{aligned}(l(\theta), u(\theta)) = \arg\min_{l,u} \sum_{i \in \mathcal{I}} w_i \left(\max\{0, l_{i_k(i)} - y_i(\theta)\}^2 + \max\{0, y_i(\theta) - u_{i_k(i)}\}^2 \right) \\ \text{s.t. } l_{i_k} + s \leq u_{i_k}, \qquad i_k = 1, \ldots, n_k \\ u_{i_k} + g \leq l_{i_k+1}, \quad i_k = 1, \ldots, n_k - 1.\end{aligned} \tag{4.12}$$

Similar to the previous section, optimization problem (4.12) can be reduced and reparameterized. In this case, an optimal solution is given for $l_{i_k+1} = u_{i_k} + g$, which is a straightforward extension to Lemma (4.2.1). Additionally, one can fix the auxiliary variable u_0 and the upper bound of the last category u_{n_k} to $u_0 = \min_i y_i(\theta) - n_k(g + s)$ and $u_{n_k} = \max_i y_i(\theta) + n_k(g + s)$, as increasing the interval $[u_0, u_{n_k}]$ further cannot lead to improved objective function values. The reduced optimization problem is then given by the following result:

Theorem 4.2.3. *An optimal solution of the optimization problem* (4.12) *is obtained by solving*

$$\begin{aligned}u(\theta) = \arg\min_{u} \sum_{i \in \mathcal{I}} w_i \left(\max\{0, u_{i_k(i)-1} + g - y_i(\theta)\}^2 + \max\{0, y_i(\theta) - u_{i_k(i)}\}^2 \right) \\ s.t.\ u_{i_k} + g + s \leq u_{i_k+1}, \quad i_k = 0, \ldots, n_k - 1,\end{aligned} \tag{4.13}$$

for $u_0 = \min_i y_i(\theta) - n_k(g + s)$ *and* $u_{n_k} = \max_i y_i(\theta) + n_k(g + s)$*, and setting* $l_{i_k+1}(\theta) = u_{i_k}(\theta) + g$ *for* $i_k = 1, \ldots, n_k - 1$.

The proof of Theorem 4.2.3 is analog to Theorem 4.2.2, which is the special case with $g = s = 0$. We can again reparameterize the problem by introducing $d_{i_k} = u_{i_k} - (u_{i_k-1} + g + s)$. This leads to the reparameterized optimization problem

$$\begin{aligned}d(\theta) = \arg\min_{d} \sum_{i \in \mathcal{I}} w_i \left(\max\left\{0, u_0 + \sum_{i'_k=1}^{i_k(i)-1} (d_{i'_k} + g + s) - y_i(\theta)\right\}^2 \right. \\ \left. + \max\left\{0, y_i(\theta) - u_0 + \sum_{i'_k=1}^{i_k(i)} (d_{i'_k} + g + s)\right\}^2 \right) \\ \text{s.t. } d_{i_k} \geq 0, \quad i_k = 1, \ldots, n_k - 1,\end{aligned} \tag{4.14}$$

with $u_0 = \min_i y_i(\theta) - n_k(g + s)$.

We analyzed the properties of the different optimization problems and found

Theorem 4.2.4. *The optimization problems* (4.5)*,* (4.9)*,* (4.11)*,* (4.12)*,* (4.13) *and* (4.14) *are all convex.*

Proof. The objective functions of the respective optimization problems are sums of convex functions of the lower bounds l, the upper bounds u and/or the differences d. As the sum of convex functions is itself convex (Boyd & Vandenberghe, 2004, Section 3.2), the overall objective function is convex. In combination with linear inequality constraints, this implies that the optimization problem is convex. □

As convex problems only possess one optimum, local optimizations should converge to the global optimum.

4.3 Evaluation of reformulations on application examples

To assess the performance of the standard, reduced and reparameterized formulations on the optimization of the surrogate data, we evaluated the different formulations on several application examples.

4.3.1 Model overview

For our analysis, we considered one smaller toy model for illustration and two realistic application examples (Table 4.1). T1 is a model of RAF inhibition (Figure 4.3A) where we assume active RAF (dimerized and not fully inhibited) to be measured (Mitra et al., 2018). M1 models the hetero- and homo-dimerization of the transcription factors STAT5A and STAT5B and was calibrated using data of phosphorylated STAT5B as well as phoshporylated and total STAT5A (Boehm et al., 2014). M2 describes IL13-induced signaling of the JAK2/STAT5 pathway in lymphoma cell-lines (Raia et al., 2011). The application examples were taken from a benchmark collection of parameter estimation problems, based on the collection established by (Hass et al., 2019) (cf. Chapter 5). The quantitative experimental data that was provided in the collection was transformed to qualitative observations based on their ordering, where we assumed datapoints to be indistinguishable if the numerical values were equal.

Table 4.1: Key numbers of the different considered models and datasets for comparison of the reformulations.

Model	**T1**	**M1**	**M2**
# state variables	6	8	14
# parameters	2	6	18
# observables	1	3	8
# datapoints	9	48	205
# categories	2–9	3×16	6–38
Description	RAF inhibition	STAT5 dimerization	IL13-induced signaling
Reference	Mitra et al. (2018)	Boehm et al. (2014)	Raia et al. (2011)

4.3.2 Implementation

We implemented the optimal scaling approach into the pyPESTO toolbox (Schälte et al., 2020). This implementation allows computing category bounds of the optimal scaling problem using

- the standard optimization problem (4.12)
- the reduced optimization problem (4.13)
- the reduced reparameterized optimization problem (4.14).

For solving the inner optimization problem we used the SLSQP algorithm implemented in SciPy (Jones et al., 2001) for the standard and reduced optimization and the L-BFGS-B algorithm from SciPy for the reduced reparameterized problem. The ODEs were simulated using the AMICI toolbox (Fröhlich et al., 2017).

For the comparison of the standard formulation with the reformulations, we choose hyperparameters based on the recommendations from Pargett et al. (2014). Minimal gaps between categories and minimal interval sizes were set to

$$\begin{aligned} s &= \max\left\{\frac{\max_i y_i(\theta)}{2n_k+1}, \epsilon\right\} \\ g &= \max\left\{\frac{\max_i y_i(\theta)}{4(n_k-1)+1}, \epsilon\right\}, \end{aligned} \tag{4.15}$$

with model-dependent lower bounds ϵ. As weights, Pargett et al. (2014) proposed a combination of the sum of model simulations and the net variation of the simulations,

which penalizes flat simulations:

$$w_i = \left(\frac{1}{2} \sum_{j \in \mathcal{I}} |y_j(\theta)| + \sum_{j \in \mathcal{I} \setminus \{1\}} |y_j(\theta) - y_{j-1}(\theta)| + \gamma \right)^{-1} \quad \text{with} \quad \gamma = 10^{-10}. \tag{4.16}$$

The code for the evaluation of the different formulations is made available at Zenodo under `https://doi.org/10.5281/zenodo.3561952`.

4.3.3 Results

Convexity and optimality of optimal scaling approach

We first verified the validity of our theoretical findings that

- the inner optimization problem is convex
- the standard, reduced and reparameterized formulations have the same optimal objective function value.

To analyze the convexity of the inner problem, we performed multi-start local optimization for the model T1 with 3 categories and the reduced formulation (Figure 4.3B). All local optimizations found the same objective function value which is in line with the theoretical finding that the problem is indeed convex. As the model only comprises two unknown model parameters, we evaluated the objective function for a large number of parameter combinations and inspected the objective function landscape for a dataset with 3 and 9 categories for all three formulations (Figure 4.3C & D). For all formulations the numeric values for the objective function were identical, confirming the second theoretical finding.

Reformulations improve scalability of the optimal scaling approach

As the inner optimization problem is convex, performing one local optimization is sufficient to find the global optimum. Still, the computational complexity increases linearly with the number of categories (and therefore observable parameters) (Figure 4.3E). The decrease of optimization parameters in the inner problem when using the reduced formulations resulted in generally lower computation times. For the reduced and the reparameterized formulation, the computation times are comparable with slight improvement using the reparameterized formulation. One explanation for this is the different optimization algorithms that can be employed in the reparameterized case.

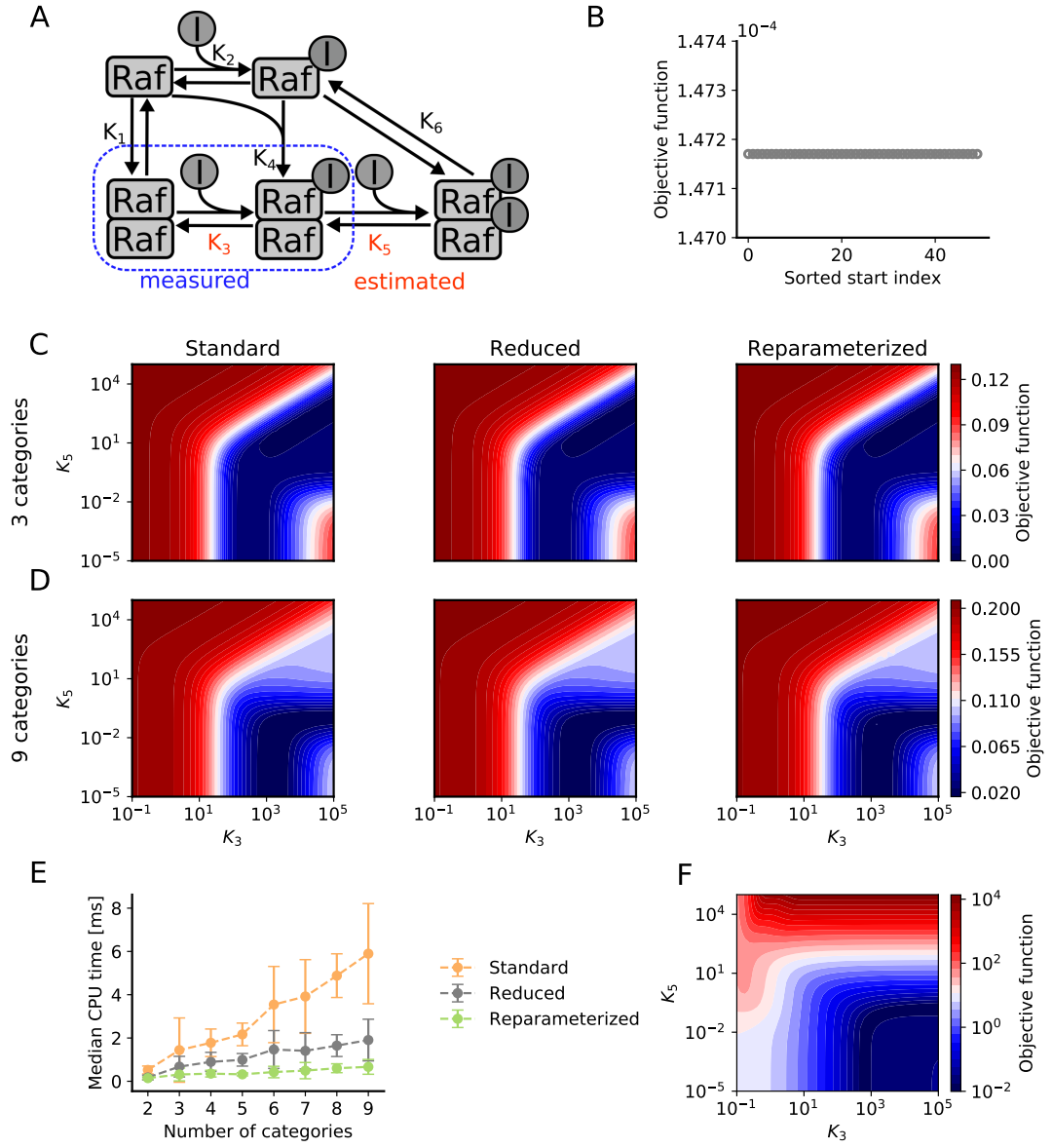


Figure 4.3: Comparison of the standard and reduced formulations to calculate the optimal surrogate data for model T1. (A): Illustration of the model. (B): Waterfall plot for multi-start local optimization of the inner problem with the reduced formulation for data with three categories. The inner problem was solved at the true model parameters with $K_3 = 4000$ and $K_5 = 0.1$. (C & D): Objective function landscapes for T1 with the dataset discretized in 3 categories (C) and 9 categories (D). (E): Median computation times to solve the inner optimization problem for different numbers of categories and the different formulations of the optimal scaling problem. (F): Objective function landscape for quantitative data. This figure is taken from Figure 3 of the author's publication (Schmiester et al., 2020b).

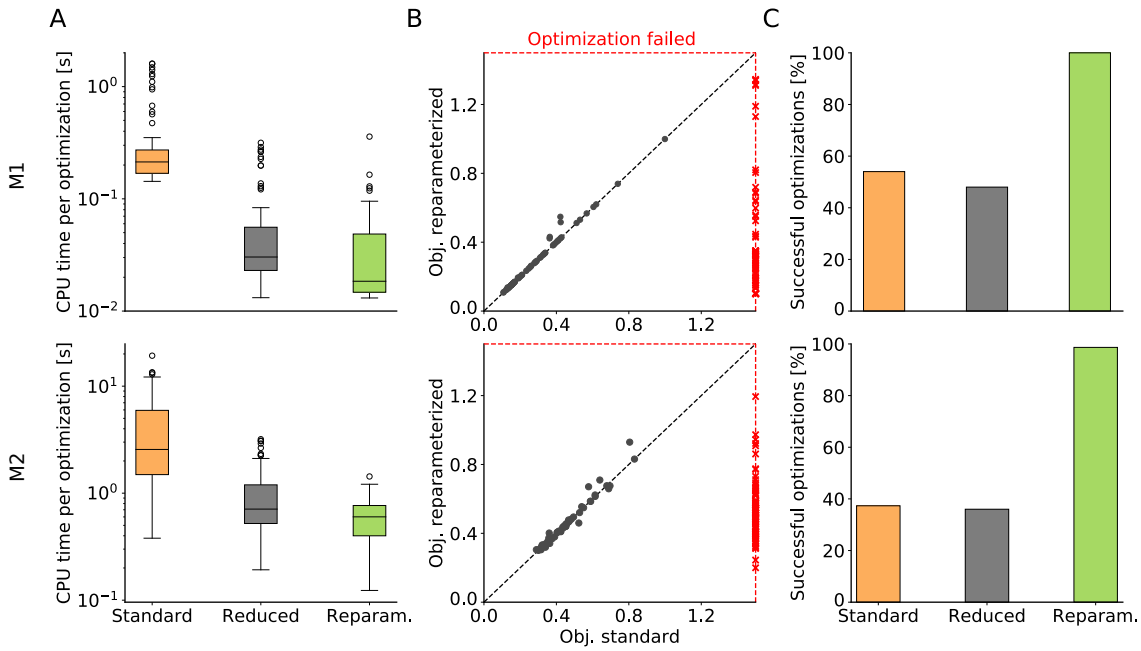


Figure 4.4: Evaluation of efficiency and robustness of the inner optimization problem tested on 150 randomly sampled parameter vectors. (A): Computation times for the three different formulations for models M1 and M2. (B): Objective function values for the standard and reparameterized formulations. Failed optimizations are indicated as red crosses. (C): Percentage of optimizations that successfully returned appropriate optimal scaling bounds. This figure is a modified version of Figure 5 of the author's publication (Schmiester et al., 2020b).

Qualitative data can have similar information content as quantitative data

To assess the information contained in the qualitative data, we additionally calculated the objective function landscape for the model T1 with quantitative measurements (Figure 4.3F). To assure comparability, we used the same objective function as for qualitative data and replaced the surrogate data with the measured quantitative values. While the objective function landscape for qualitative data depends on the number of categories (Figure 4.3C & D), for a sufficiently large number, it possesses similar characteristics as the landscape for quantitative measurements. This corroborates the finding from Mitra & Hlavacek (2020) that enough qualitative measurements can be similarly informative as quantitative data.

Reformulations improve efficiency and robustness of the optimal scaling approach

We proposed two reformulations of the optimal scaling approach which aim to improve different issues of the inner optimization problem: (i) the reduced formulation only possesses half of the optimization variables and should consequently improve the computation

time and (ii) the reparameterized formulation facilitates the use of a broader range of optimization algorithms. To evaluate the impact of the reformulations on realistic application examples, we solved the inner optimization problem for the models M1 and M2 for all three formulations. For each model and formulation, we randomly sampled 150 parameters and optimized the surrogate data.

The computation times for the parameters, for which all formulations successfully solved the optimization problem revealed that the CPU time for the reduced and reparameterized formulations were substantially decreased compared to the standard formulation (Figure 4.4A). The reparameterized formulation was slightly faster than the reduced formulation, which could again be due to the more efficient optimization algorithms that can be employed. The reparameterized formulation yielded median and mean speedups of 11.5 and 18.9 respectively for the model M1 and 4.2 and 7.4 for the model M2 compared to the standard formulation.

Although the optimization problem is convex, the different formulations sometimes resulted in different objective function values (Figure 4.4B). Surprisingly, with the standard and reduced formulations, the optimization problem could often not be solved successfully (Figure 4.4C). Numerical optimization only provided appropriate category bounds for around 50 % of the cases for M1 and only less than 40 % of the cases for M2. In contrast, for the reparameterized formulation we observed success rates of 100 %.

Reformulations improve overall parameter estimation performance

The here developed reformulations resulted in reduced computation times and improved robustness for solving the inner optimization problem. However, solving the inner problem is only one step during parameter estimation of the model parameters θ. We therefore performed parameter estimation for the different formulations again for models M1 and M2 to assess the overall performance of the different formulations. To this end, we used multi-start optimization with around 100 local optimizations of the gradient-free optimizers Nelder-Mead and Powell implemented in SciPy.

The optimization results yielded generally similar final objective function values between the different formulations (Figure 4.5A). For all cases except for M1 with the Nelder-Mead optimizer, the reparameterized formulation achieved slightly better values than the other formulations. A possible explanation for this is the improved robustness to solve the inner problem.

The decreased computation times for solving the inner optimization problem (Figure 4.4A) could also be observed for the overall optimization (Figure 4.5B). Again, the reduced

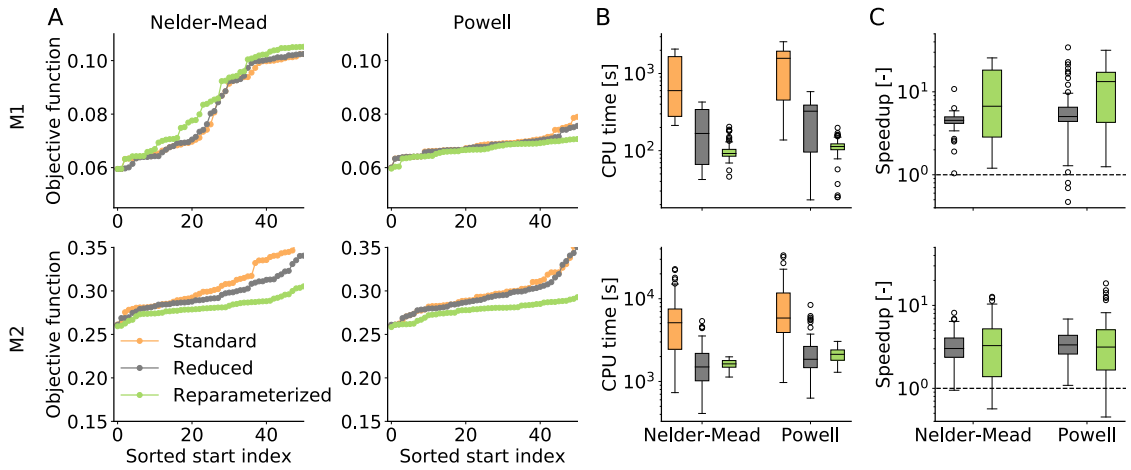


Figure 4.5: Overall optimization results for M1 and M2 using the Nelder-Mead and Powell optimizers with around 100 local optimizations. (A): Waterfall plots for all models and optimizers. Only the best 50 starts are shown. (B): Computation times for the different models and optimizers. (C): Speedup defined by the computation time for the standard optimization divided by the times for the reduced or reparameterized formulations. Above the dashed line the use of the reformulation was computationally more efficient and below the use of the standard formulation. This figure is a modified version of Figure 6 of the author's publication (Schmiester et al., 2020b).

and the reparameterized formulations yielded similar computation times. Interestingly, the reparameterized formulation resulted in smaller variations between the different local optimizations. Quantifying the speedups compared to the standard formulations, we observed on average a 5-10 fold reduction in computation time with the reduced and reparameterized formulations (Figure 4.5C).

Optimal scaling approach yields good agreement between model simulation and measurement

It has so far not been comprehensively assessed if the optimal scaling approach applied to systems biology models yields meaningful estimates for the parameters that achieve a good agreement between model simulation and measured qualitative data. To investigate this, we simulated the models for the overall best found parameters and compared the simulation to the optimal surrogate data (Figure 4.6).

For both models, the best parameters resulted in good fits of the model with the surrogate data (Figure 4.6A & B). For the model M1, the best 10 parameters consistently achieved good correlations of 0.90–0.98 for the three observables (Figure 4.6C). For the model M2, we observed larger differences between the fits of the different parameters (Figure 4.6D),

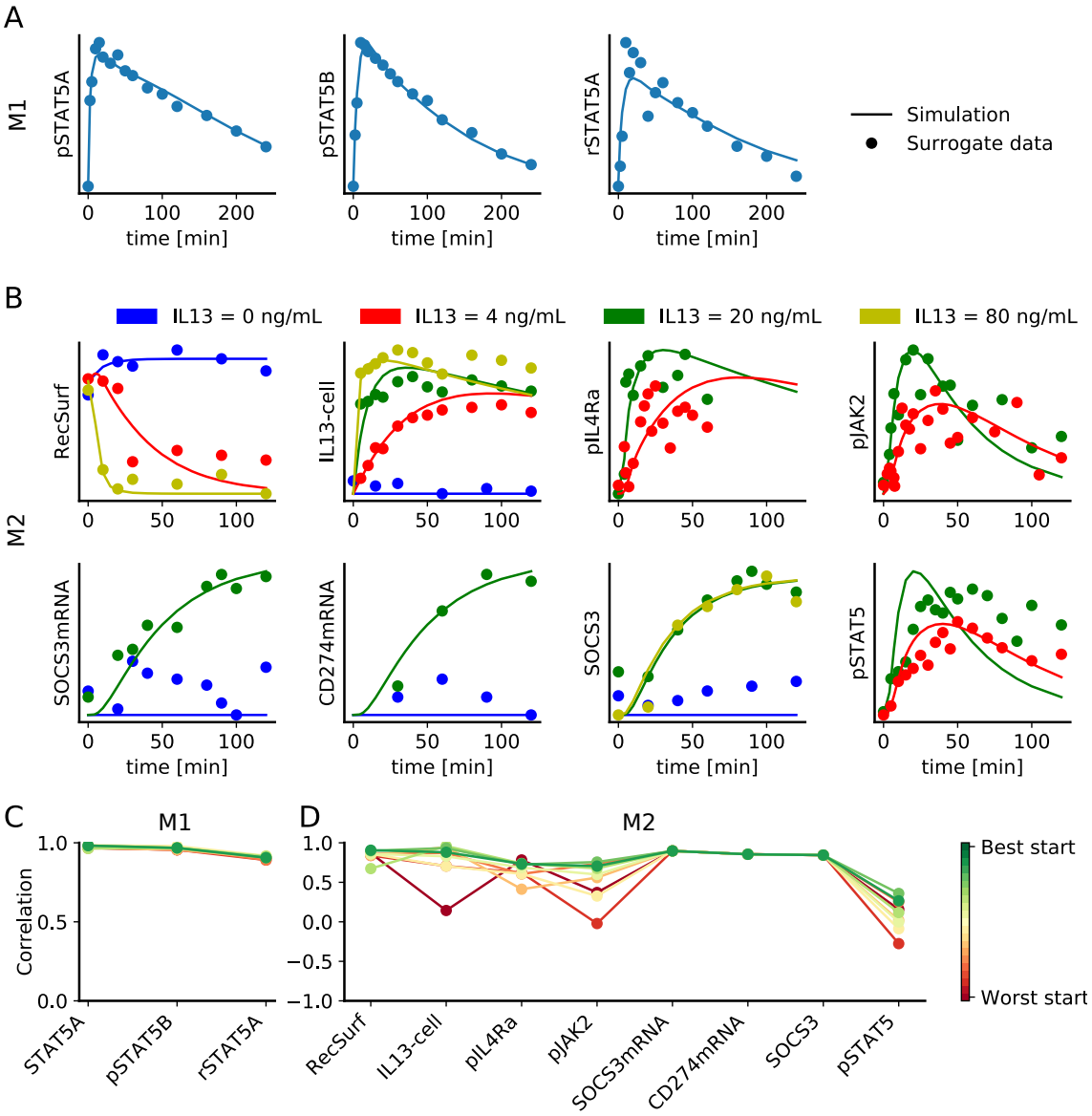


Figure 4.6: Fits of the model simulation and the surrogate data using the overall best found parameters. (A & B): Model simulation and optimal surrogate data for the models M1 (A) and M2 (B). (C & D): Pearson correlation coefficients for the 10 best starts for model M1 (C) and M2 (D). This figure is a modified version of Fig. 7 of the author's publication (Schmiester et al., 2020b).

indicating that among the 10 best starts not all converged to the same point in parameter space. Still, for the best fit, a good agreement between model simulation and surrogate data is achieved with an average correlation across observables of 0.85.

4.4 Gradient computation for the optimal scaling objective function

The reformulations introduced in Section 4.2 accelerated the solution to the inner optimal scaling problem and facilitated the use of a broader set of optimization algorithms for this sub-problem. However, they did not influence the outer optimization problem. As many optimization algorithms rely on the availability of gradient information (see Section 2.3.4), the set of available optimization algorithms for the outer optimization problem could be extended by computing the derivative of the optimal scaling objective function with respect to the model parameters θ. In this section, we derive a semi-analytical algorithm for this based on ideas from bi-level optimization theory (Fiacco, 1976).

4.4.1 Semi-analytical gradient computation scheme

To derive formulas for gradient computation, we consider the original full optimal scaling problem (4.3) with no minimal category and gap sizes first. For ease of notation, we rewrite the optimization problem in matrix-vector notation. For this, we denote the collection of all inner optimization variables by $\xi = (\tilde{y}, l, u)^T \in \mathbb{R}^{n_\xi}$ and the vector of simulations by $\bar{\xi}(\theta) = (y(\theta), 0, 0)^T \in \mathbb{R}^{n_\xi}$ which is filled with zeros such that ξ and $\bar{\xi}$ have the same dimensions. Here, $n_\xi = N + 2n_k$ is the number of inner optimization variables. With this, we can define the objective function J as

$$J(\theta, \xi) = \left(\xi - \bar{\xi}(\theta)\right)^T W \left(\xi - \bar{\xi}(\theta)\right). \tag{4.17}$$

Here, the weight matrix W is given by

$$W = \begin{pmatrix} \mathrm{diag}(w) & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{n_\xi \times n_\xi}.$$

W is augmented with zeros such that the bounds l and u do not contribute to the objective function J. This then yields the optimization problem

$$\min_\theta J(\theta, \xi(\theta)) \tag{4.18}$$

$$\begin{aligned} \text{s.t. } \xi(\theta) &= \arg\min_\xi J(\theta, \xi) \\ \text{with } C\xi &\leq 0, \end{aligned} \tag{4.19}$$

in which $C \in \mathbb{R}^{n_c \times n_\xi}$ encodes the inequality constraints of the inner problem, with the total number of constraints n_c. As before, the outer problem aims to optimize the model parameter vector $\theta \in \mathbb{R}^{n_\theta}$ and the inner problem optimizes, conditioned on θ, the vector

containing the surrogate data and interval bounds $\xi \in \mathbb{R}^{n_\xi}$. W was augmented with zeros, such that the dimensions of W and C are consistent, which is necessary for the following calculations. We are now interested in the derivative of the objective function J with respect to θ, which is given via

$$\frac{dJ(\theta, \xi(\theta))}{d\theta_k} = \frac{\partial J(\theta, \xi)}{\partial \theta_k} + \frac{\partial J(\theta, \xi)}{\partial \xi} \frac{\partial \xi(\theta)}{\partial \theta_k}, \tag{4.20}$$

for $k = 1, \ldots, n_\theta$. The first parts, $\frac{\partial J(\theta,\xi)}{\partial \theta_k}$ and $\frac{\partial J(\theta,\xi)}{\partial \xi}$, can be easily computed:

$$\frac{\partial J(\theta, \xi)}{\partial \theta_k} = -2(\xi - \bar{\xi}(\theta))^T W \frac{\partial \bar{\xi}(\theta)}{\partial \theta_k} \tag{4.21}$$

and

$$\frac{\partial J(\theta, \xi)}{\partial \xi} = 2(\xi - \bar{\xi}(\theta))^T W. \tag{4.22}$$

The last part that needs to be calculated is the derivative of the observable parameters with respect to the model parameters $\frac{\partial \xi(\theta)}{\partial \theta_k}$. To obtain this, we consider the Lagrangian function for the inner optimization problem

$$\mathcal{L}(\xi, \mu) = J(\theta, \xi) + \mu^T C \xi. \tag{4.23}$$

The necessary optimality conditions for the inner optimal scaling problem (4.19) are then given by (see Section 2.3.2)

$$\nabla_\xi \mathcal{L}(\xi, \mu) = 2(\xi(\theta) - \bar{\xi}(\theta))^T W + \mu(\theta)^T C = 0 \tag{4.24}$$

$$\mu_i(\theta) C_i \xi(\theta) = 0 \tag{4.25}$$

$$C_i \xi(\theta) \leq 0 \tag{4.26}$$

$$\mu_i(\theta) \geq 0 \tag{4.27}$$

for $i = 1, \ldots, n_c$, where C_i is the i-th row of the matrix C. As the inner optimal scaling problem is convex (Theorem 4.2.4), these conditions are necessary and sufficient for a mimimum. Given the optimal values of ξ, the Lagrange multiplier μ can be obtained by solving this system. To obtain the derivatives of ξ w.r.t. θ, we calculate the derivatives of equations (4.24) and (4.25) w.r.t. θ_k:

$$2\left(\frac{\partial \xi(\theta)}{\partial \theta_k} - \frac{\partial \bar{\xi}(\theta)}{\partial \theta_k}\right)^T W + \frac{\partial \mu(\theta)^T}{\partial \theta_k} C = 0 \tag{4.28}$$

$$\frac{\partial \mu_i(\theta)}{\partial \theta_k} C_i \xi(\theta) + \mu_i(\theta) C_i \frac{\partial \xi(\theta)}{\partial \theta_k} = 0. \tag{4.29}$$

This yields a linear system of equations that needs to be solved for every parameter θ_k:

$$\begin{pmatrix} 2W & C^T \\ \operatorname{diag}(\mu(\theta))C & \operatorname{diag}(C\xi(\theta)) \end{pmatrix} \begin{pmatrix} \frac{\partial \xi(\theta)}{\partial \theta_k} \\ \frac{\partial \mu(\theta)}{\partial \theta_k} \end{pmatrix} = \begin{pmatrix} 2W \frac{\partial \bar{\xi}(\theta)}{\partial \theta_k} \\ 0 \end{pmatrix}. \tag{4.30}$$

This linear system can be solved to calculate $\frac{\partial \xi(\theta)}{\partial \theta_k}$. With this, an algorithm for calculating the gradients of the objective function for given parameters θ is as follows:

(i) Simulate the ODE with sensitivities to obtain $\bar{\xi}(\theta)$ and $\frac{\partial \bar{\xi}(\theta)}{\partial \theta}$.

(ii) Calculate optimal surrogate data $\tilde{y}$ and category bounds l, u using one of the formulations introduced in Section 4.2 ((4.12), (4.14) or (4.14)).

(iii) Solve the optimality conditions (4.24) – (4.27) for the Lagrange multiplier $\mu(\theta)$.

(iv) Solve the linear system of equations (4.30) to obtain $\frac{\partial \xi(\theta)}{\partial \theta}$.

(v) Evaluate the gradient $\frac{dJ(\theta, \xi(\theta))}{d\theta}$ of the objective function via (4.20).

4.4.2 Parameter-dependent weights and category and gap sizes

As discussed in Section 4.2.3, it is often beneficial to impose minimal sizes for the category intervals and the gaps between the intervals. These are often chosen by incorporating the simulation $y(\theta)$, making them dependent on the model parameters. Additionally, the weights of the least squares function are also commonly chosen such that they depend on the simulation and therefore on the parameters θ. Assuming, that $s(\theta)$, $g(\theta)$ and $W(\theta)$ are differentiable functions, it is again possible to derive formulas for the objective function gradients. Collecting the minimal category and gap sizes in the vector $d(\theta)$, we can rewrite the optimal scaling problem to

$$\min_{\theta} J(\theta, \xi(\theta)) \tag{4.31}$$

$$\begin{aligned} \text{s.t. } \xi(\theta) &= \arg\min_{\xi} J(\theta, \xi) \\ \text{with } C\xi + d(\theta) &\leq 0, \end{aligned} \tag{4.32}$$

with

$$J(\xi, \theta) = \left(\xi - \bar{\xi}(\theta)\right)^T W(\theta) \left(\xi - \bar{\xi}(\theta)\right). \tag{4.33}$$

With this, $\frac{\partial J(\theta,\xi)}{\partial \theta_k}$ can be calculated via

$$\frac{\partial J(\theta,\xi)}{\partial \theta_k} = (\xi - \bar{\xi}(\theta))^T \left(\frac{\partial W(\theta)}{\partial \theta_k}(\xi - \bar{\xi}(\theta)) - 2W(\theta)\frac{\partial \bar{\xi}(\theta)}{\partial \theta_k} \right). \tag{4.34}$$

Again, with the Lagrangian function $\mathcal{L}(\xi,\mu) = J(\theta,\xi) + \mu^T C\xi$, the optimality conditions for the problem (4.32) are

$$\nabla_\xi \mathcal{L}(\xi,\mu) = 2(\xi(\theta) - \bar{\xi}(\theta))^T W(\theta) + \mu(\theta)^T C = 0 \tag{4.35}$$

$$\mu_i(\theta)(C_i\xi(\theta) + d_i(\theta)) = 0 \tag{4.36}$$

$$C_i\xi(\theta) + d_i(\theta) \leq 0 \tag{4.37}$$

$$\mu_i(\theta) \geq 0, \tag{4.38}$$

for $i = 1, \ldots, n_c$. Again, we can calculate the derivative of the first two equations w.r.t. θ_k:

$$2\left(\frac{\partial \xi(\theta)}{\partial \theta_k} - \frac{\partial \bar{\xi}(\theta)}{\partial \theta_k} \right)^T W(\theta) + 2(\xi(\theta) - \bar{\xi}(\theta))^T \frac{\partial W(\theta)}{\partial \theta_k} + \frac{\partial \mu(\theta)^T}{\partial \theta_k} C = 0 \tag{4.39}$$

$$\frac{\partial \mu_i(\theta)}{\partial \theta_k}(C_i\xi(\theta) + d_i(\theta)) + \mu_i(\theta)\left(C_i \frac{\partial \xi(\theta)}{\partial \theta_k} + \frac{\partial d_i(\theta)}{\partial \theta_k} \right) = 0, \tag{4.40}$$

which yields the linear system of equations

$$\begin{pmatrix} 2W(\theta) & C^T \\ \mathrm{diag}(\mu(\theta))C & \mathrm{diag}(C\xi(\theta) + d(\theta)) \end{pmatrix} \cdot \begin{pmatrix} \frac{\partial \xi(\theta)}{\partial \theta} \\ \frac{\partial \mu(\theta)}{\partial \theta} \end{pmatrix} = \begin{pmatrix} 2W(\theta)\frac{\partial \bar{\xi}(\theta)}{\partial \theta_k} - \frac{\partial W(\theta)}{\partial \theta_k}(\xi(\theta) - \bar{\xi}(\theta)) \\ -\mathrm{diag}(\mu(\theta))\frac{\partial d(\theta)}{\partial \theta_k} \end{pmatrix}. \tag{4.41}$$

As (4.30) and (4.41) are sparse linear systems, they can be solved efficiently.

4.5 Evaluation of gradient-based approach on application examples

Here, we evaluate the gradient calculation framework and compare it to gradient-free optimization on several application examples.

4.5.1 Model overview

As the reformulations introduced in the last sections reduced the computational complexity substantially, it was feasible to extend the set of application examples. We therefore added three more examples. M3 models the impact of vaccination on infectious diseases dynamics (Rahman et al., 2016). M4 describes an oscillatory network of transcriptional regulators (Elowitz & Leibler, 2000). M5 models the RAF-MEK-ERK signaling pathway (Fiedler

Table 4.2: Key numbers of the additional models and datasets for parameter estimation.

Model	**T1**	**M3**	**M4**	**M5**
# state variables	6	7	8	6
# parameters	2	9	18	12
# observables	2	1	1	8
# datapoints	18	23	58	72
# categories	2×3	19	43	9–11
Description	RAF inhibition	Infectious disease dynamics	Transcriptional regulation	RAF-MEK-ERK signaling
Reference	Mitra et al. (2018)	Rahman et al. (2016)	Elowitz & Leibler (2000)	Fiedler et al. (2016)

et al., 2016). Again, the additional models were taken from the benchmark collection introduced by Hass et al. (2019). An overview of the additional models and datasets used for parameter estimation is given in Table 4.2. To test the accuracy of the computed gradients on a toy model resembling most realistic examples, we added data for a second observable to the model T1.

4.5.2 Implementation

The gradient-based approach is again implemented in the pyPESTO toolbox. Gradients are calculated using the here described approach automatically, if a gradient-based optimizer is employed. As we have shown that the reduced and reparameterized formulation to calculate the surrogate data was the most robust and efficient approach, we will restrict the following analysis to this formulation. Again, the model is simulated and sensitivities are calculated using AMICI (Fröhlich et al., 2017). In the previous chapter, we chose the hyperparameters based on recommendations by Pargett et al. (2014). Based on the experience from our analysis, we revised the choice made there and slightly adapted some of the hyperparameters. As the weights are required to be differentiable w.r.t. θ, we set them to

$$w_i(\theta) = \left(\sum_{i \in \mathcal{I}} y_i(\theta) + \gamma \right)^{-1} \quad \text{with} \quad \gamma = 10^{-10}. \tag{4.42}$$

Since the application examples considered here all have non-zero simulations, this is equivalent to the first part of the weights (4.16). The derivative of $w(\theta)$, which is needed for gradient computation, is given by

$$\frac{\partial w(\theta)}{\partial \theta_k} = \frac{-\sum_{i \in \mathcal{I}} \frac{\partial y_i(\theta)}{\partial \theta_k}}{\left(\sum_{i \in \mathcal{I}} y_i(\theta) \right)^2}. \tag{4.43}$$

As minimal interval and gap sizes, we chose

$$\begin{aligned} s(\theta) &= \frac{\max_i y_i(\theta)}{2n_k + 1} \\ g(\theta) &= \frac{\max_i y_i(\theta)}{4(n_k - 1) + 1} + \epsilon. \end{aligned} \tag{4.44}$$

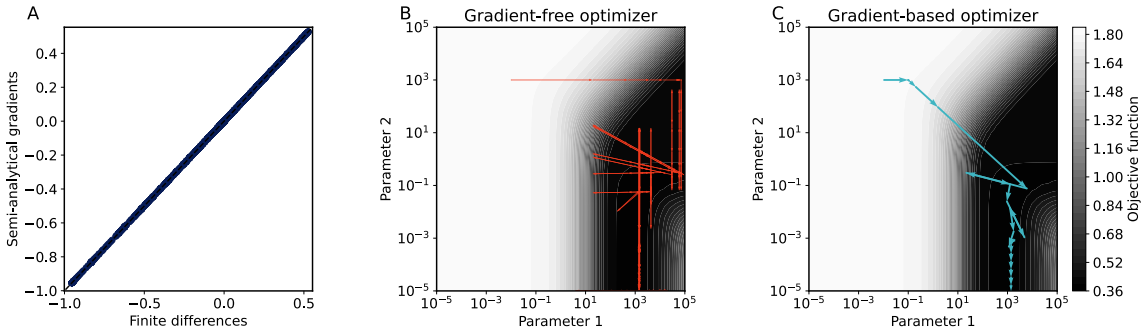


Figure 4.7: Illustration of gradient-based approach using the model T1. (A): Gradients evaluated on randomly sampled parameters using the semi-analytical approach and a central finite differences scheme. (B): Objective function landscape and trajectory of a gradient-free optimizer. (C): Objective function landscape and trajectory of a gradient-based optimizer.

The derivatives of s and g are given via

$$\begin{aligned} \frac{\partial s(\theta)}{\partial \theta_k} &= \frac{\frac{\partial y_j(\theta)}{\partial \theta_k}}{2n_k + 1} \\ \frac{\partial g(\theta)}{\partial \theta_k} &= \frac{\frac{\partial y_j(\theta)}{\partial \theta_k}}{4(n_k - 1) + 1}, \end{aligned} \tag{4.45}$$

with $j = \arg\max_i y_i(\theta)$ being the index of the maximal simulation.

4.5.3 Results

Semi-analytical approach yields accurate gradients

To assess the accuracy of the derived scheme for gradient computation, we first considered the toy model T1 and compared the computed gradients to those obtained via finite difference calculation. We therefore evaluated the gradient of the objective function on a large number of different parameter vectors, which revealed almost identical gradients for the different approaches for all tested points (Figure 4.7A).

Gradient-based optimization reduces computation time

To illustrate the differences between gradient-free and gradient-based optimization, we optimized parameters for the model T1 using a gradient-free and a gradient-based optimizer starting from the same initial parameters. As this model only contains two parameters, we inspected the whole objective function landscape and the respective optimizer trajectories (Figure 4.7B & C). While the gradient-free optimizer is based on a rather naive scheme to update the parameters (Figure 4.7B), which often moves along sub-optimal directions, the

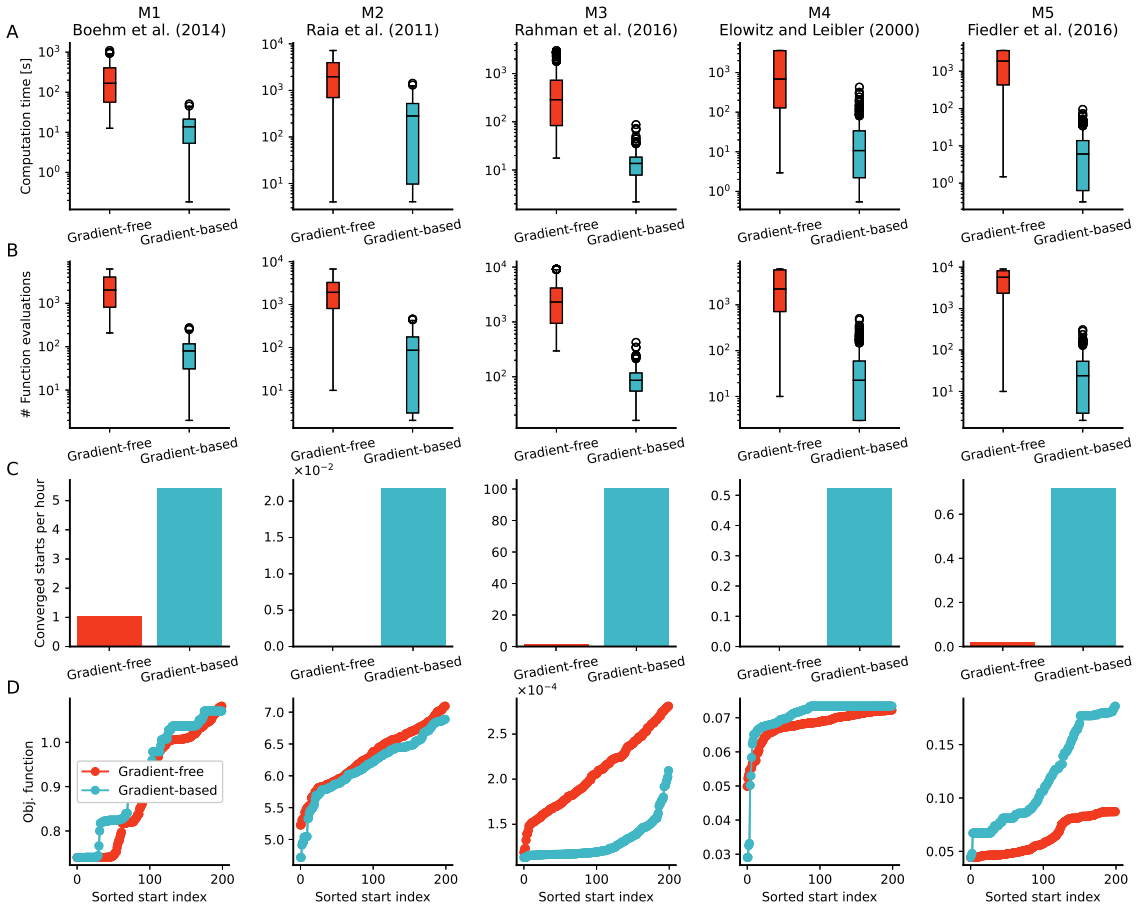


Figure 4.8: Optimization results for all models using gradient-free and gradient-based optimization for 500 local optimizations. (A): Computation times until the optimizer terminates per local optimization. (B): Number of function evaluations per local optimization. (C): Converged starts per hour. A start is considered converged if the absolute difference to the overall best value is less than 10^{-4}. (D): Waterfall plots for all 5 considered models using gradient-free and gradient-based optimization. Best 200 starts out of a total of 500 are shown.

gradient-based optimizer moves towards the optimal point within a few function evaluations (Figure 4.7C).

To assess the performance of gradient-based and gradient-free optimization in a realistic setting, we performed multi-start local optimization for the application examples M1–M5 using 500 local optimizations each. The optimization results revealed substantially reduced computation times using gradient-based optimization (Figure 4.8A). Depending on the considered model, we observed median speedups roughly between one and two orders of magnitude. Especially for the larger models, one of the main stopping criteria of the

gradient-free optimizer was a computation time limit that we implemented to guarantee computational feasibility. As none of the gradient-based optimization runs reached this upper bound, the improvement in computation time would have been even larger without this limit.

As illustrated in Figure 4.7B & C, the gradient-based optimizer used a more sophisticated scheme to update parameters during optimization which resulted in reduced numbers of objective function evaluations required to converge to a local optimum. The substantial reduction in function evaluations could also be observed for all here employed models (Figure 4.8B). Even though a single evaluation is more costly when gradient information needs to be calculated, the reduced number of evaluations outweighs this, explaining the improved computation times.

Along with the computation times, we additionally incorporated the final objective function values into the analysis. To this end, we considered the number of local optimization runs which converged to the overall best objective function value (Figure 4.7C). The analysis showed that the gradient-based optimizer yielded substantially improved efficiency for all considered models. In particular, this was the case even for the models M1 and M5, for which the gradient-free optimizer found the optimal objective function value more often than the gradient-based optimizer (Figure 4.7D).

Gradient-based optimization yields improved model fits

The waterfall plots for the different considered models showed that the gradient-based optimization consistently yielded equal (M1, M3 and M5) or even better (M2 and M4) final objective function values compared to gradient-free optimization (Figure 4.7D). To assess, if the different best parameters yielded substantially different fits of the model with the qualitative data, we simulated the models for the best parameters of both optimizations and compared the simulation to the optimal surrogate data. This revealed large improvements of the parameters obtained from gradient-based optimization for the model M4 (Figure 4.9A & B). In fact, only the parameters found by gradient-based optimization captured the oscillatory behavior of the qualitative data correctly. For the model M2, smaller improvements for some observables could be achieved using gradient-based optimization (Figure 4.9C & D).

Gradient-based optimization enables uncertainty quantification

Qualitative data is often considered to be less informative than quantitative data. As this can result in parameter non-identifiabilities, it is important to assess parameter uncertainties

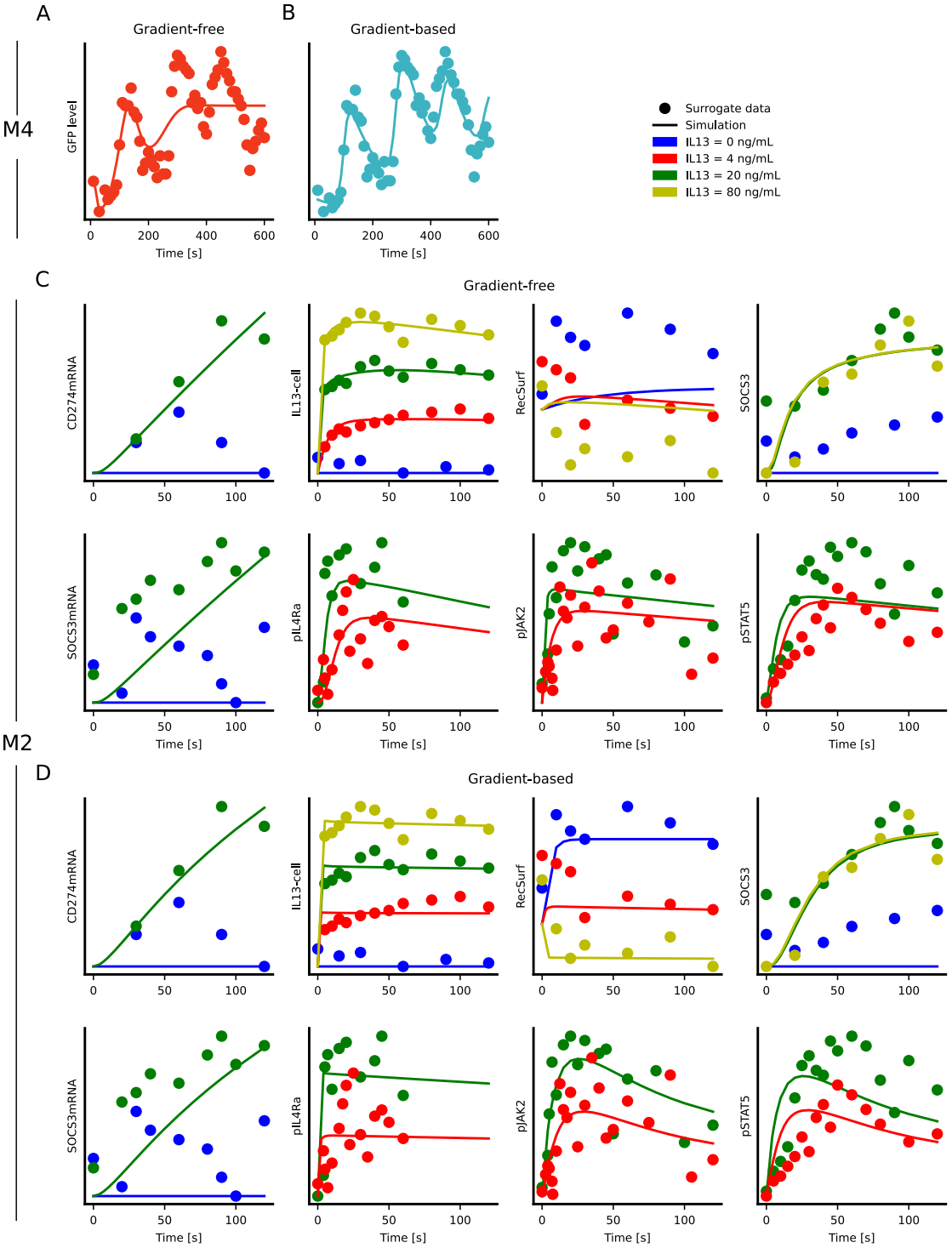


Figure 4.9: Model simulation and optimal surrogate data for the best parameters from gradient-free and gradient-based optimization. (A & B): Model fits for the model M4 for gradient-free (A) and gradient-based (B) optimization. (C & D): Model fits for the model M2 for gradient-free (C) and gradient-based (D) optimization.

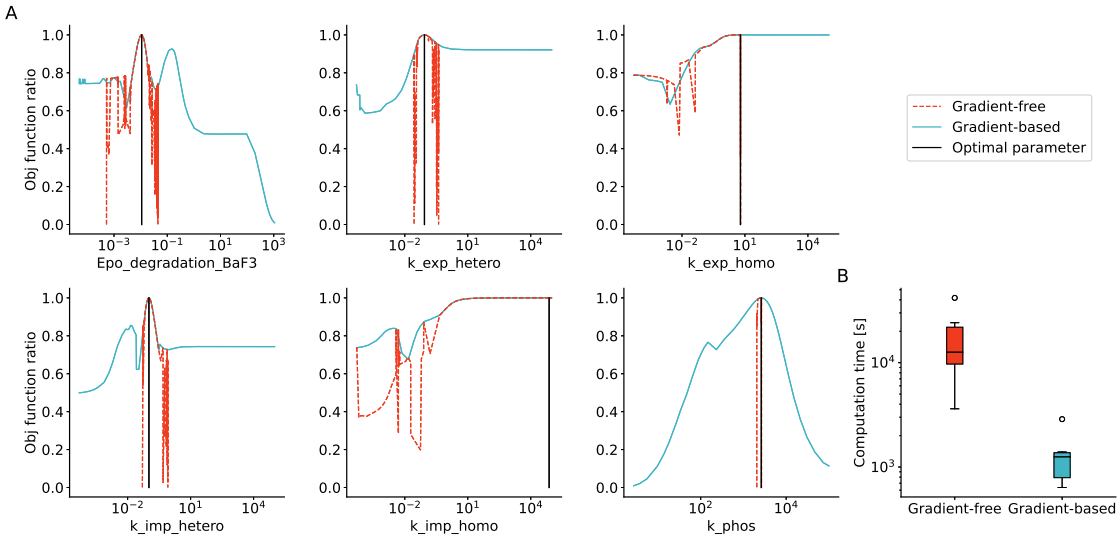


Figure 4.10: Objective function profiles for the model M1. (A): Profiles using gradient-free and gradient-based optimization for all six parameters. (B): Computation times for the profiles per parameter for gradient-free and gradient-based optimization.

when using qualitative measurements. For uncertainty analysis, we used objective function profiles analogously to profile likelihoods in the case of a likelihood function (Raue et al., 2009). We exemplarily calculated profiles for the model M1 (Figure 4.10A). The gradient-based approach yielded mostly smooth profiles indicating that several parameters of this model can be identified using the qualitative dataset. In contrast, the gradient-free approach resulted in multiple discontinuities in the profiles indicating impaired optimizer performance. This shows that only the gradient-based approach was able to yield meaningful profiles for this model. In addition to the improved profile results, the gradient-based approach required on average an order of magnitude less computation time than the gradient-free optimizer (Figure 4.10B).

4.6 Overall performance

In this chapter, we introduced several modifications and extensions to the standard optimal scaling approach for qualitative data. First, we reformulated the inner optimization problem, and then we developed an algorithm for gradient computation for the outer optimization problem. To assess the combined efficiency improvement by these adaptations, we compared the computation times for parameter estimation of the standard approach with a gradient-free optimizer with the reparameterized and reduced approach using a gradient-based optimizer (Figure 4.11). To this end, we ran multi-start local optimization for the models

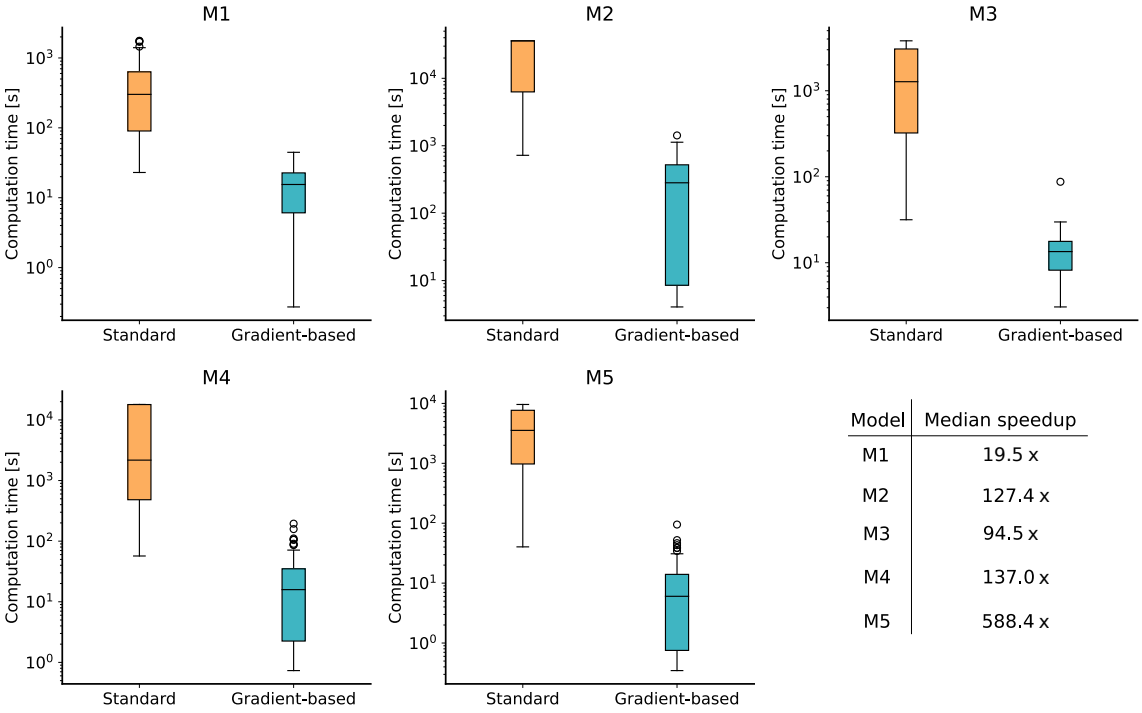


Figure 4.11: Computation times for multi-start local optimization using the standard formulation and a gradient-free optimizer and the reduced and reparameterized formulation with a gradient-based optimizer. 100 starts with a maximal wall-time of 5 hours (10 hours for model M2) were performed. Median speedup factors of the gradient-based approach are shown in the lower right table.

M1-M5 for 100 starts. Analysis of the computation times per local optimization revealed speedup factors of two orders of magnitude for most models using the methods developed in this thesis.

4.7 Summary and discussion

Qualitative measurements can be an important resource for parameter estimation of mechanistic models. Yet, only few, computationally demanding methods exist to integrate such data and it is therefore often neglected by modelers. In this chapter, we built upon the already established optimal scaling approach (Pargett & Umulis, 2013; Pargett et al., 2014; Shepard, 1962). As this approach is based on the repeated solution of a constrained optimization problem, fast and reliable methods are required. Here, we proposed several improvements to the optimal scaling approach to facilitate robust and efficient parameter estimation for qualitative data. We first investigated the structure of the inner optimization problem and exploited it to derive two reformulations. The first reformulation aimed to

improve computation times by reducing the dimensionality of the problem. The second proposed reformulation simplified the optimization problem by transforming the initial linear inequality constraints into simpler box-constraints. This enabled the use of a larger set of possible optimization algorithms. As gradient-based optimization has shown to often outperform gradient-free algorithms, we additionally developed a framework for semi-analytical gradient computation for the objective function arising in the optimal scaling approach. This method was derived from the derivatives of the necessary conditions for a local optimum (see chapter 2.3.2).

The different methods proposed in this chapter were extensively evaluated on different application examples. These examples were taken from a collection of published models and data to enable a realistic test setting. We first showed that the reduced formulation yielded decreased computation times, with an average speedup factor of 5-10 compared to the standard optimization approach. Considering success rates, we observed that the reparameterized formulation largely improved the robustness of solving the inner optimization problem. Both results were found consistently across multiple models and gradient-free optimization algorithms. We next evaluated the here proposed gradient computation algorithm. We demonstrated that this algorithm yielded accurate gradients and that the gradient-based optimizers resulted in computation times reduced by more than one order of magnitude for all five considered models. In addition to the reduction in computation time, the gradient-based optimization found better objective function values in two cases, which resulted in improved fits of the model with the measured qualitative data. We concluded with computing objective function profiles, which indicated that only the gradient-based approach could be reliably used. The gradient-based optimization not only resulted in more reliable profiles compared to the gradient-free optimization but also reduced computation times by an order of magnitude. In total, the combination of the reformulations and the gradient calculation lead to a reduction of computation time by two orders of magnitude for most models.

Still, open questions remain which could further improve efficiency and reliability of the optimal scaling method. The approach includes the choice of several hyperparameters, like the weights w and the minimal sizes for category intervals and gaps between intervals. While these values are already chosen in an adaptive way, such that the modeler does not need to determine them prior to parameter estimation, we observed that the literature values were often not ideal and a more thorough analysis of these parameters would be valuable to investigate the impact on optimization results. Several further improvements could be made regarding efficiency. Parallelization could be exploited to speed up the solution of the inner problem as well as the gradient computation, especially when multiple

optimal scaling problems need to be solved. Additionally, optimization algorithms tailored to the underlying problem could be employed. For the gradient computation, the problem can be split into active and inactive constraints, which can be exploited to improve efficiency (Kolstad & Lasdon, 1990). While we used the gradient-based approach to calculate objective function profiles and investigate uncertainties, an objective function which can be easily interpreted statistically could be beneficial. Uncertainty analysis is of great importance, as one can expect that qualitative data is less informative than quantitative data. A first attempt has been made by Mitra & Hlavacek (2020), where a Bayesian formulation has been proposed. A proper statistical formulation can also benefit the combination of quantitative and qualitative data, which is the most frequent setting and which is important to improve parameter identifiability. While here, we only used qualitative data, it could be combined with quantitative data using the optimal scaling approach by formulating a similar objective function also for the quantitative measurements.

To conclude, the integration of qualitative data is a valuable extension to the existing parameter estimation methods, as this can increase the amount of available training data, and we contributed to the feasibility of this by substantially improving the efficiency and robustness of the optimal scaling approach. We additionally provided easy-to-use, open-source implementations to make this method applicable to a broader scientific community.

Chapter 5

PEtab – Interoperable specification of parameter estimation problems of biological systems

The reproducibility crisis has been identified as a major concern that can hinder fast and reliable scientific advances (Casadevall & Fang, 2010; Prinz et al., 2011). In computational sciences, a growing community aims at improving reproducibility, reusability and interoperability, which resulted in various standards, tools and guidelines (Peng, 2011; Sandve et al., 2013; Stanford et al., 2019; Waltemath & Wolkenhauer, 2016; Waltemath et al., 2011b; Wilkinson et al., 2016). In the area of mathematical modeling in systems biology, different standards have been developed to specify biochemical models (Stanford et al., 2019), such as the Systems Biology Markup Language (SBML) (Hucka et al., 2003), CellML (Cuellar et al., 2003) or the BioNetGen Language (BNGL) (Harris et al., 2016). Various software tools, implementing diverse methods and algorithms, support these model specification formats (Balsa-Canto & Banga, 2011; Choi et al., 2018; Fröhlich et al., 2017; Hoops et al., 2006; Mitra et al., 2019; Raue et al., 2015; Stapor et al., 2018b). The development of model specification standards have additionally facilitated the generation of large databases containing hundreds of systems biology models that can be used for method benchmarking or model development (Le Novère et al., 2006; Olivier & Snoep, 2004).

Other standards move a step beyond model definition and also include simulation results or experimental data. The Simulation Experiment Description Markup Language (SED-ML) builds on top of the model specifications in e.g. SBML, and allows for a machine-readable definition of simulation experiments based on XML (Waltemath et al., 2011a). It includes repeated simulation tasks that can be used e.g. to encode parameter scans. The phraSED-ML format is a human-readable extension of SED-ML (Choi et al., 2016). Going into a similar direction, the Systems Biology Results Markup Language (SBRML) was developed to associate models with experimental data and define simulation experiments. Like SED-ML, SBRML is based on XML and can also be used for parameter scans. The SBTab format is a collection of table-based conventions for the definition of models and experimental data which is human-readable and -writable (Lubitz et al., 2016).

While these standards facilitate reusability and reproducibility of computational models, simulation results and to some extent experimental data, parameter estimation is so far not in the scope of any of these available formats. Necessary information, such as the definition of a noise model is missing, hindering the unambiguous definition of parameter estimation problems. Consequently, toolboxes for parameter estimation currently use their custom input formats, making it difficult and error-prone for a user to switch between toolboxes and benefit from the different methods implemented in different tools.

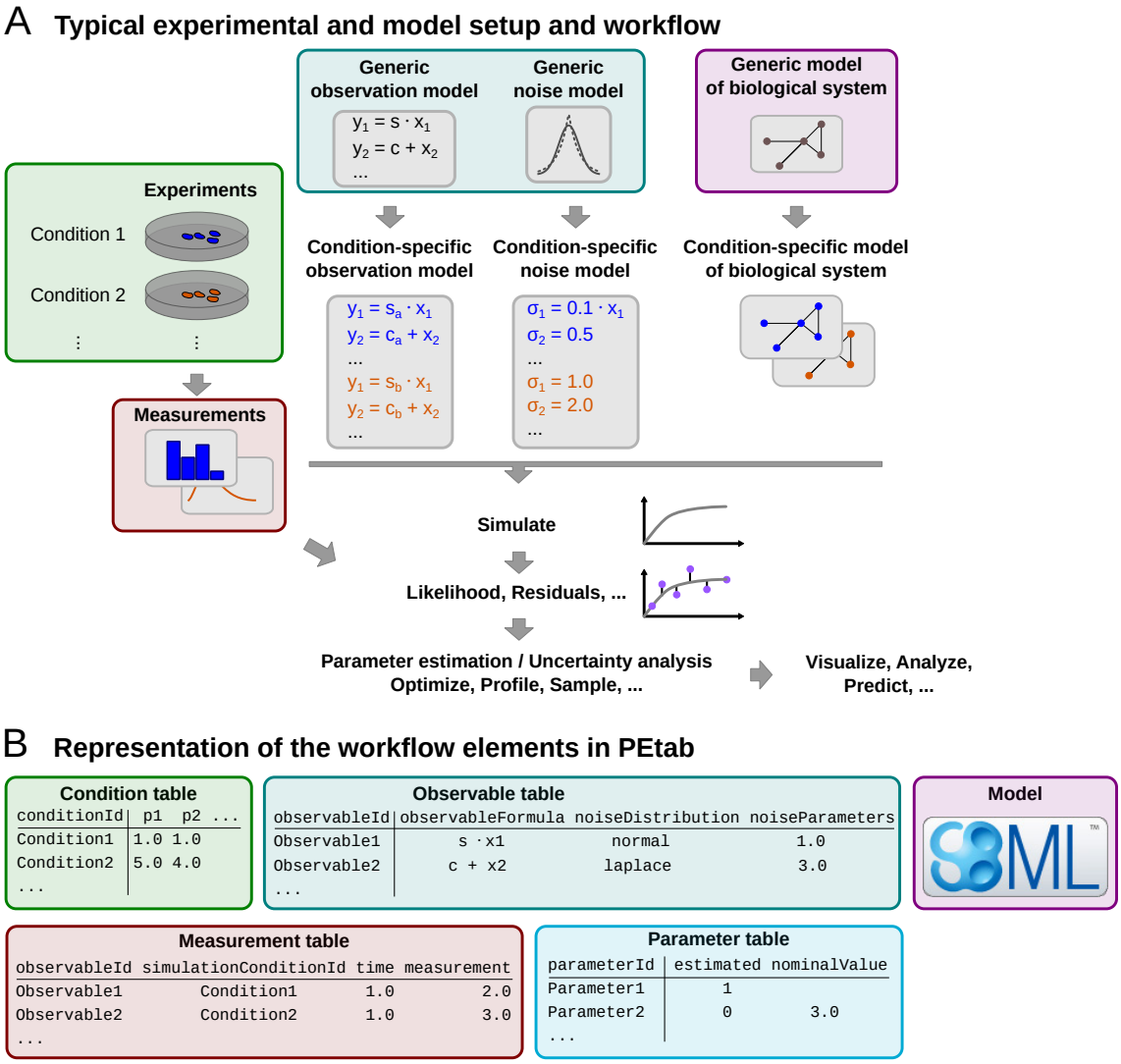
To improve this, we developed the PEtab format for a standardized and interoperable description of parameter estimation problems. PEtab extends SBML for model specification with tabular files for the definition of observables, noise models, experimental data, their mapping to the model output as well as parameters in an unambiguous way. PEtab support is so far implemented in eight different toolboxes. Additionally, a Python library is provided for easy reading, writing, manipulation and validation of problems specified in the PEtab format. Therefore, this chapter addresses challenge (iv) stated in Chapter 1.3.

This chapter is based on and in part identical to the following publication:

- **Schmiester, L.***, Schälte, Y.*, Bergmann, F., Camba, T., Dudkin, E., Egert, J., Fröhlich, F., Fuhrmann, L., Hauber, A. L., Kemmer, S., Lakrisenko, P., Loos, C., Merkt, S., Müller, W., Pathirana, D., Raimúndez, E., Refisch, L., Rosenblatt, M., Stapor, P., Städter, P., Wang, D., Wieland, F.-G., Banga, J. R., Timmer, J., Villaverde, A. F., Sahle, S., Kreutz, C., Hasenauer, J., Weindl, D. (2020). PEtab–interoperable specification of parameter estimation problems in systems biology. *arXiv*, 2004.01154 [q-bio.QM]

5.1 Specification of PEtab problems

A common workflow in data-based modeling begins with a mathematical model of a biological process and experimental data that need to be linked (Figure 5.1A and Chapter 2). PEtab consists of multiple files that were designed to cover the typical elements of this workflow (Figure 5.1B). To this end, measurement data and a mathematical model have to be specified. Then the measurements are linked to the model through observation functions. Since measurements are usually noise corrupted, an error model needs to be defined. Data is often collected under different experimental conditions that need to be accounted for during model simulation and a generic model has to be modified for each condition. Finally, parameters subject to optimization have to be specified. With this, sufficient information is available, to define e.g. a likelihood or least-squares objective



Condition table

conditionId	p1	p2	...
Condition1	1.0	1.0	
Condition2	5.0	4.0	
...			

Observable table

observableId	observableFormula	noiseDistribution	noiseParameters
Observable1	s ·x1	normal	1.0
Observable2	c + x2	laplace	3.0
...			

Measurement table

observableId	simulationConditionId	time	measurement
Observable1	Condition1	1.0	2.0
Observable2	Condition2	1.0	3.0
...			

Parameter table

parameterId	estimated	nominalValue
Parameter1	1	
Parameter2	0	3.0
...		

Figure 5.1: Illustration of a typical parameter estimation workflow in PEtab. (A): Example of a common setting in data-based modeling. A mathematical model is usually trained on measurement data derived from different experimental conditions, which are linked to the model via observables. A generic model can be adapted to the different experimental perturbations. (B): A simplified illustration of how the different workflow elements are covered by the different PEtab files (not all table columns are shown here). This figure is taken from Figure 1 of the author's publication (Schmiester et al., 2020a).

function which can be used for parameter estimation and uncertainty quantification. As we intended to develop a modular, machine- and human-readable and -writable format, we decided on tab-separated value files (TSV) for the specifications of the parameter estimation problems with standardized row and column identifiers. The different PEtab files are outlined in Figure 5.2 and described in more detail in the following:

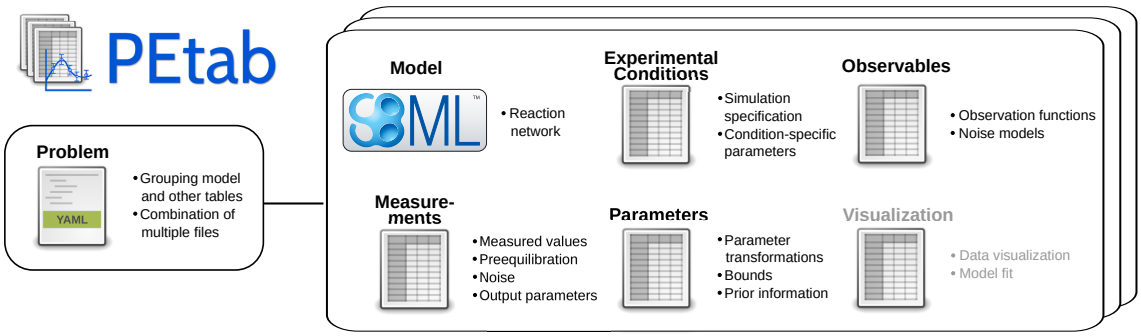


Figure 5.2: Overview of the different PEtab files. The PEtab files aim to define different features needed for parameter estimation. The PEtab files can be grouped in a YAML-based problem file. A visualization file can be included optionally. This figure is taken from Figure 2 of the author's publication (Schmiester et al., 2020a).

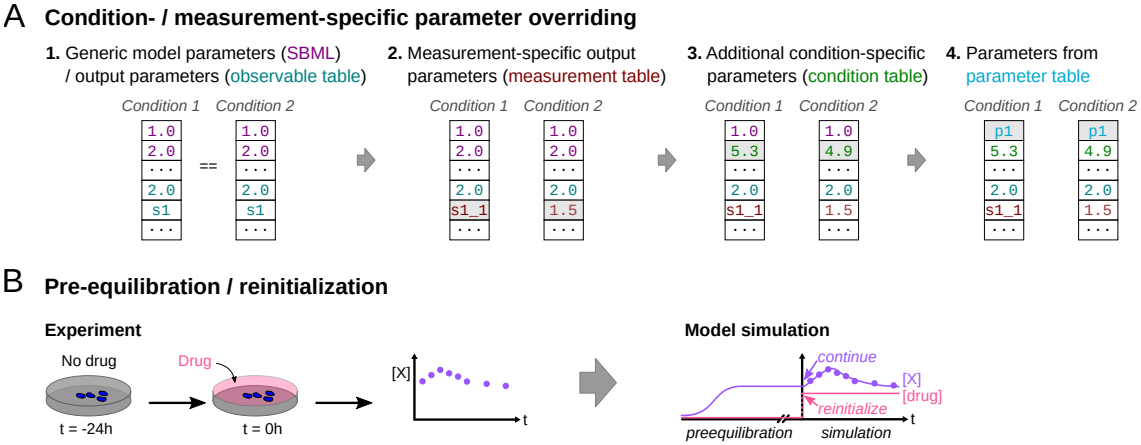


Figure 5.3: Hierarchy of parameters and pre-equilibration in the PEtab format. (A): Illustration of possibilities and precedence of parameter overriding at different stages. Simulation parameter vectors are shown for two simulation conditions. A generic model parameterization can be overridden in a condition- and measurement-specific manner to account for different model inputs or observational model parameters. Grey background indicates the newly overridden parameters in each step. Individual parameters can be set to specific values or marked to be estimated (as here p1). (B): Biological experiments often consist of a pre-equilibration experiment, where, under some "baseline" condition, the system is assumed to be in equilibrium (e.g., here depicted for after 24h incubation) before a perturbation is applied. This can be modeled by simulating the ODE until a steady-state is reached for the pre-equilibration system and then re-initializing the model states to simulate the perturbation. This figure is taken from Figure 3 of the author's publication (Schmiester et al., 2020a).

Model (SBML): The model file describes the biological process and the dynamics of the modeled species $x(t, \theta, u)$. It has to be defined in the SBML format, which is among the

most widely used model formats in systems biology. Any existing SBML model can be used without further adaptation. The algorithm used for model simulation is not specified within PEtab and can be freely chosen by the user and employed toolbox.

Observables (TSV): The observable file defines the model output $y(t, \theta, u)$ in terms of mathematical formulas of the parameters and species and assigns a unique identifier to them. Observable parameters, such as offsets and scalings (see Chapter 3) can be included by specific keywords. Additionally, a noise model has to be defined. The most commonly used noise models, normal and Laplace distributed noise, are currently supported.

Experimental conditions (TSV): The experimental condition file describes the different setups under which the data was collected, such as different external stimuli or genetic backgrounds. These are used as inputs u for model simulation and allow for a hierarchical specification of model properties (Fig 5.3A). If simulation conditions are used for pre-equilibration, i.e. that some experiment started from the equilibrium reached for another condition, specific model states can be marked for re-initialization (Fig 5.3B).

Measurements (TSV): The measurement file specifies the experimental data and links it to the model observables via the observable identifiers. The different experimental conditions under which the measurements are taken are defined using condition identifiers establishing a link to the experimental conditions file. This allows for a unique mapping of measurement data to model output. Observable parameters, that are specific to individual measurements can additionally be provided.

Parameters (TSV): The parameter file lists all parameters that need to be estimated and can also include other known parameters. Additionally, lower and upper bounds as well as parameter transformations (such as linear or logarithmic) can be specified. If prior information is available for the parameters, it can be included and used for Bayesian inference or sampling of initial parameters for optimization.

Visualization (TSV): The visualization file specifies the combination of simulation and measurement for plotting purposes. The file allows for different types of figures, such as time-course or dose-response curves and can be interpreted by the PEtab Python library described later.

PEtab problem file (YAML): The problem file collects all other above-mentioned files together. With this, multiple measurement or model files can be combined into one parameter estimation problem.

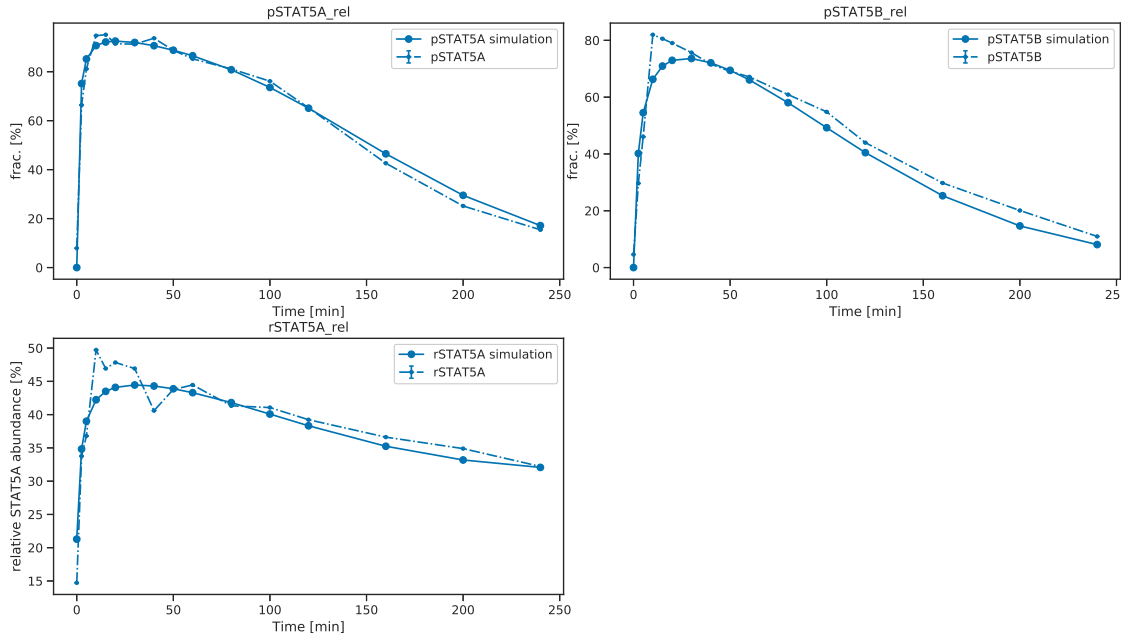


Figure 5.4: Example of the visualization routines provided in the PEtab library. Shown are the simulations and measurements for the three observables of the model developed by Boehm et al. (2014) using the optimal parameters provided in Hass et al. (2019). The plots were created using the `plot_data_and_simulation` function from the PEtab visualization routines without further post-processing.

To encode additional information which is not directly covered by PEtab, it is possible to extend the existing files. For example, the hierarchical optimization (Section 3) and the optimal scaling approach (Section 4) can be defined by marking parameters as hierarchical in the parameters file and assigning keywords, such as "scaling" or "offset", to define, how they are calculated during optimization.

5.2 PEtab implementation

We provided several supporting tools and information which enable easy usability of PEtab. These implementations are maintained open-source under `https://github.com/PEtab-dev` and are described in detail in the following.

5.2.1 PEtab documentation and library

We implemented a Python-based library which facilitates easy reading, writing and modifying of PEtab problems. Additionally, the library can be used for validation of existing

PEtab files and provides hints on miss-specifications. Commonly, the quality of the parameter estimation results are validated by plotting the fits of the model to the data. Therefore, the PEtab library contains several visualization routines, which take the PEtab visualization files as input and can be used for easy analyzes of the parameter estimation results. An example of the visualization for the model by Boehm et al. (2014) (M4 in Table 5.3) is given in Figure 5.4. Furthermore, the PEtab library provides functionality to package PEtab files into COMBINE archives (Bergmann et al., 2014). We also provided a documentation describing the details of the different files in a concise yet comprehensive manner (`https://petab.readthedocs.io/en/latest/`). This includes all necessary and optional information that can be defined in PEtab as well as the use of the library and the visualization routines.

5.2.2 PEtab support

To render PEtab useful for a broader scientific community, it needs to be supported by parameter estimation software tools. Together with multiple research groups focused on toolbox and method development, we implemented PEtab support in eight commonly used systems biology toolboxes, namely COPASI (Hoops et al., 2006), Data2Dynamics (Raue et al., 2015), MEIGO (Egea et al., 2014), dMod (Kaschek et al., 2019), pyPESTO (Schälte et al., 2020), pyABC (Klinger et al., 2018), parPE (Schmiester et al., 2019) and AMICI (Fröhlich et al., 2018). These toolboxes implement a broad range of different methods and features, including model generation, model simulation, parameter inference using optimization and sampling and profile calculation (Table 5.1). As depicted in Table 5.1, the different tools focus on different aspects with varying strengths and weaknesses. Profiting from the features of different toolboxes is in practice often time-consuming and error-prone as the parameter estimation problem has to be re-implemented in the respective input formats. With the interoperable input format PEtab, the modelers can more easily switch between toolboxes and benefit from a broader set of computational methods.

5.2.3 PEtab test suite

To support further toolbox developers who consider implementing PEtab import into their tools and to validate the correctness of the existing PEtab importers, we set up a test suite of different toy problems and reference values. Various different tests are included to verify if a tool supports the different PEtab features, ranging from basic simulations of the model to less common features such as pre-equilibration, transformation of observables and data

Table 5.1: Non-exhaustive overview of different features and methods implemented in the computational toolboxes that support PEtab. Darker colors indicate more accurate, scalable, or broader functionality compared to basic implementations. This table is taken from the Table 1 of the author's publication (Schmiester et al., 2020a).

	COPASI	D2D	dMod	MEIGO	AMICI	parPE	pyABC	pyPESTO
Interface / Language	Graphical interface	MATLAB	R	MATLAB	C++, Python, MATLAB	C++	Python	Python
Model construction	Advanced	Basic	Basic	No	No	No	No	No
Model simulation	Accurate	Accurate, Scalable	Accurate, Scalable	Uses AMICI	Accurate, Scalable	Uses AMICI	Uses AMICI	Uses AMICI
Gradient computation	Approximative	Accurate	Accurate	Uses AMICI	Accurate, Scalable	Uses AMICI	No	Uses AMICI
Gradient-free (global) parameter estimation	Multiple algorithms	Basic	No	Metaheuristic algorithms	No	No	No	Basic
Gradient-based parameter estimation	Basic	Multiple local optimizers	Multiple local optimizers	Metaheuristic algorithms	No	Multiple local optimizers	No	Multiple local optimizers
Parameter profile likelihood	Basic	Advanced	Advanced	No	No	No	No	Advanced
Prediction profile likelihood	Basic	Advanced	Advanced	No	No	No	No	No
Parameter sampling	No	Basic	No	No	No	No	Adaptive SMC algorithms	Multiple MCMC algorithms
Simulation of stochastic models	Multiple algorithms	No	No	No	No	No	No	No
Parameter inference for stochastic models	Basic	No	No	Basic	No	No	Scalable	Basic
Particular strengths	Advanced modeling Strong allrounder Graphical interface	Powerful gradient based optimization Advanced profiling Strong allrounder	Powerful gradient based optimization Strong allrounder	Powerful metaheuristic optimization	Highly scalable simulation & gradient comp.	Highly scalable optimization for large-scale models	Scalable likelihood-free inference	Allrounder Multiple MCMC samping methods

Table 5.2: Overview of supported PEtab features in different tools, based on the test cases implemented in the PEtab test suite. The first character indicates whether computing simulated data is supported and simulations are correct (✓) or not (-). The second character indicates whether computing χ^2 values of residuals are supported and correct (✓) or not (-). The third character indicates whether computing likelihoods is supported and correct (✓) or not (-). This table is taken from the Table 2 of the author's publication (Schmiester et al., 2020a).

Test-case	**AMICI**	**Copasi**	**D2D**	**dMod**	**MEIGO**	**parPE**	**pyABC**	**pyPESTO**
Basic simulation	✓✓✓	✓- -	✓✓✓	✓✓✓	✓✓✓	- - ✓	✓✓✓	✓✓✓
Multiple simulation conditions	✓✓✓	✓- -	✓✓✓	✓✓✓	✓✓✓	- - ✓	✓✓✓	✓✓✓
Numeric initial compartment sizes in condition table	- - -	✓- -	✓✓✓	✓✓✓	✓✓✓	- - -	- - -	- - -
Numeric initial concentration in condition table	✓✓✓	✓- -	✓✓✓	✓✓✓	✓✓✓	- - ✓	✓✓✓	✓✓✓
Numeric noise parameter overrides in measurement table	✓✓✓	✓- -	✓✓✓	✓✓✓	✓✓✓	- - ✓	✓✓✓	✓✓✓
Numeric observable parameter overrides in measurement table	✓✓✓	✓- -	✓✓✓	✓✓✓	✓✓✓	- - ✓	✓✓✓	✓✓✓
Observable transformations to log scale	✓- ✓	✓- -	✓✓✓	✓✓-	✓✓✓	- - ✓	✓- ✓	✓- ✓
Observable transformations to log10 scale	✓- ✓	✓- -	✓✓✓	✓✓-	✓✓✓	- - ✓	✓- ✓	✓- ✓
Parametric initial concentrations in condition table	✓✓✓	✓- -	✓✓✓	✓✓✓	✓✓✓	- - ✓	✓✓✓	✓✓✓
Parametric noise parameter overrides in measurement table	✓✓✓	✓- -	✓✓✓	✓✓✓	✓✓✓	- - ✓	✓✓✓	✓✓✓
Parametric observable parameter overrides in measurement table	✓✓✓	✓- -	✓✓✓	✓✓✓	✓✓✓	- - ✓	✓✓✓	✓✓✓
Parametric overrides in condition table	✓✓✓	✓- -	✓✓✓	✓✓✓	✓✓✓	- - ✓	✓✓✓	✓✓✓
Partial pre-equilibration	✓✓✓	- - -	✓✓✓	✓✓✓	✓✓✓	- - ✓	✓✓✓	✓✓✓
Pre-equilibration	✓✓✓	✓- -	✓✓✓	✓✓✓	✓✓✓	- - ✓	✓✓✓	✓✓✓
Replicate measurements	✓✓✓	✓- -	✓✓✓	✓✓✓	✓✓✓	- - ✓	✓✓✓	✓✓✓
Time-point specific overrides in the measurement table	- - -	- - -	✓✓✓	✓✓✓	✓✓✓	- - -	- - -	- - -

e.g. on a log-scale, or time-point specific parameters in the measurement table (Table 5.2). The tests include checks for correct simulations as well as χ^2 and likelihood values.

5.2.4 Collection of PEtab example problems

In addition to the test suite, we implemented a set of currently 20 parameter estimation problems in the PEtab format. These problems are largely based on a benchmark collection of parameter estimation problems by Hass et al. (2019), which was originally provided in another, less well supported format. The problems all include published models and experimental data to facilitate a realistic setting for method testing and were already used

Table 5.3: Overview of key numbers of the benchmark models that are implemented in the PEtab format. The collection will be extended gradually and an up-to-date version is publicly available at `https://github.com/Benchmarking-Initiative/Benchmark-Models-PEtab`.

	# parameters	# states	# datapoints	Reference
M1	56	36	1873	Alkan et al. (2018)
M2	113	25	542	Bachmann et al. (2011)
M3	72	4	27132	Beer et al. (2014)
M4	6	8	48	Boehm et al. (2014)
M5	23	3	111	Borghans et al. (1997)
M6	22	9	43	Brännmark et al. (2010)
M7	13	7	77	Bruno et al. (2016)
M8	188	500	120	Chen et al. (2009)
M9	12	5	22	Crauste et al. (2017)
M10	21	8	58	Elowitz & Leibler (2000)
M11	28	6	72	Fiedler et al. (2016)
M12	19	9	144	Fujita et al. (2010)
M13	3	4	16	Perelson et al. (1996)
M14	9	7	23	Rahman et al. (2016)
M15	6	75	18	Salazar-Cavazos et al. (2020)
M16	30	11	286	Schwen et al. (2015)
M17	15	6	135	Sneyd & Dufour (2002)
M18	39	7	135	Weber et al. (2015)
M19	46	15	60	Zheng et al. (2012)
M20	9	16	289	Blasi et al. (2016)

for the method evaluations in the Chapters 3 and 4 of this thesis. Table 5.3 provides an overview of the currently implemented parameter estimation problems, which will be continuously extended by additional models. The existing 20 models already cover a large spectrum of features, such as model size and number of datapoints (Table 5.3) and model a broad range of different biological processes. In addition to method development and testing, they can also be used as templates for the creation of new PEtab problems and to check the validity of import functionality of toolboxes.

5.3 Summary and discussion

Reproducibility, reusability and interoperability are important aspects for the development of computational models and mathematical algorithms. Therefore, we developed PEtab, a standardized and interoperable specification of parameter estimation problems. PEtab builds upon the well established SBML format for model specification and extends it with

TSV files for the definition of measurement data, experimental conditions, observables and parameters. Unique identifiers are used to link these entities with each other to provide a full and unambiguous definition of the most common parameter estimation problems.

The PEtab format is described in a comprehensive documentation, including use-cases and examples. Additionally, we provided a Python-based library to simplify the generation and validation of PEtab problems. Along with editing and validating features, the library also provides routines for common visualizations of model simulation and measured data to inspect the quality of the parameter estimation results. Several well-known systems biology toolboxes with hundreds of cumulative users already support PEtab. With PEtab, one can benefit from the distinct model development, parameter optimization and uncertainty quantification methods implemented in these tools. Furthermore, we provided a test suite of PEtab toy problems that can assist the implementation and validation of PEtab support in additional toolboxes. Next to the test suite, we established a collection of currently 20 parameter estimation problems in the PEtab format that can be used for method benchmarking and as templates for the generation of PEtab problems. This collection already enabled the comprehensive analysis of the mathematical methods developed in this thesis on realistic application examples.

PEtab was designed to cover the most common features of parameter estimation problems. Still, as a multitude of tailored parameter estimation algorithms exist, which require different information on the underlying problem, it does not cover every aspect. Currently, PEtab only supports model definition in the SBML format. While some other formats, such as SBtab (Lubitz et al., 2016) or Antimony (Smith et al., 2009) can also be used indirectly via their respective SBML converters, it would be valuable to extend PEtab support to models defined for example in CellML (Cuellar et al., 2003) or rule-based formats (Harris et al., 2016). PEtab is currently intended to define one parameter estimation problem. However, it could be extended to cover the most common model selection problems, e.g. via an additional file specifying the different models. The PEtab parameters file allows for simple bound constraints. As in some cases, more complex inequality constraints on the parameters can arise it would be valuable to extend the constraints in PEtab to a more general case. The parameter estimation problem using qualitative data introduced in the previous chapter was defined using a non-standard extension of PEtab, tailored to the optimal scaling approach. However, a larger comparison of the approach with other methods, such as the ones developed by Mitra et al. (2018) and Mitra & Hlavacek (2020), would require a more general definition of qualitative observations.

To summarize, we developed the PEtab format for the specification of parameter estimation problems, which we anticipate will be helpful for modelers and method developers to simplify workflows and enable less biased comparison of newly developed methods and algorithms.

Chapter 6

Summary and conclusion

Mathematical models can be used to integrate the ever-increasing amount of heterogeneous experimental data to gain new insights into biological systems. The work of this thesis focused on challenges that arose during parameter estimation of mathematical models based on ODEs, when the training data is non-absolute (see Section 1.3). To address these challenges, we first introduced the necessary mathematical background and notation in Chapter 2. Based on this, new methods and algorithms were developed in Chapters 3, 4 and 5 to tackle the open problems.

Chapter 3 described the efficient integration of relative and heterogeneous data for parameter estimation of large-scale models. To this end, we developed a framework to combine two concepts: (i) scalable gradient computation using adjoint sensitivity analysis and (ii) a hierarchical optimization method, which reduces the dimensionality of the optimization problem by analytically calculating conditionally optimal observable and noise parameters. To make the hierarchical approach more flexible, we extended it to the general case of simultaneously occurring scaling and offset parameters. The combination of adjoint sensitivity analysis and hierarchical optimization enabled the calibration of a large-scale pan-cancer signaling model with thousands of unknown parameters with computation times reduced by more than one order of magnitude compared to standard optimization. We identified the large contribution of the scaling parameters to the gradient of the objective function as a possible explanation for the substantial improvements obtained using the hierarchical optimization approach. Subsequently, we showed how the hierarchical method could be used for an unbiased integration of measurements on the phenotypic and molecular level with negligible computational overhead. The method was additionally implemented in the open-source toolbox parPE, making it freely available to other researchers.

In Chapter 4, we considered the case of integrating qualitative observations, i.e. measurements, where only the ordering is known, for model calibration. For this, we built upon the optimal scaling method established in statistics, which minimizes the distance between model simulation and a quantitative representation of the qualitative data. Therefore, this consists of repeatedly solving an optimization problem with linear inequality constraints. We derived the theoretical foundation to reduce the number of optimization variables of

this problem by a factor of two. Additionally, we provided a reparameterization of the optimization problem yielding simpler bound constraints. We showed that the problem could be solved more robustly and with substantially reduced computation times using the reformulations. In addition, we developed an algorithm for semi-analytical calculation of the gradient of the objective function arising in the optimal scaling approach. This enabled the use of more efficient gradient-based optimization algorithms for parameter estimation. Indeed, we found that these algorithms yielded optimal parameters in a fraction of the computation time that was required by gradient-free optimization methods. The combination of the here developed methods lead to a reduction of computation times by two orders of magnitude. Both, the reformulations and the algorithm for gradient calculation were implemented in the parameter estimation toolbox pyPESTO and comprehensively tested on several application examples.

In Chapter 5, the problem of a lack of reproducibility, reusability and interoperability in the area of parameter estimation in systems biology was addressed. We achieved this, in collaboration with various other research groups, by developing PEtab, a standardized and interoperable format for the specification of parameter estimation problems. PEtab is based on the widely adopted SBML format for model definition and extends it with tab-separated value files providing all necessary information for an unambiguous definition of parameter estimation problems. We provided a Python library for using and validating PEtab problems and implemented support for PEtab problems in several parameter estimation toolboxes. We additionally provided a test-suite to aid the implementation of PEtab support and a database of examples of published parameter estimation problems in the PEtab format.

The methods derived in this thesis opened up possibilities for further developments and extensions. These include methodological extensions such as the combination of minibatch optimization (Stapor et al., 2019) with hierarchical optimization or the extension of the hierarchical approach to a more general inner optimization problem and other error models. An example of another inner optimization problem arose in the optimal scaling approach for qualitative data and it remained an open question, whether this can be combined with adjoint sensitivity analysis to scale it up to models of similar size as the pan-cancer model used for the hierarchical optimization. A further possible extension of the optimal scaling approach would be the derivation of second-order derivatives which could be used for parameter estimation and profile calculation (Stapor et al., 2018a). As parameter estimation using qualitative data in systems biology is a comparably young area of research, it has so far not been assessed, which objective function is most appropriate. For example, a formulation providing a statistical interpretation for qualitative data could be beneficial.

First steps have been made towards this direction by Mitra & Hlavacek (2020). A thorough assessment of different methods, such as the ones available for qualitative data, requires a large test-suite of different parameter estimation problems. PEtab, and, in particular the collection of problems in this format, provide a suitable framework for such benchmarks. The full potential of this collection and the interoperability of PEtab has not yet been fully realized. Among others, a possible use case of this would be the comparison of different parameter estimation toolboxes developed by different research groups and the identification of well-working optimization algorithms across platforms and programming languages. Further developments of PEtab also depend on how it is adopted by the community and what type of not yet supported features will be requested. Already now, an additional toolbox, SBML2Julia (Lang et al., 2020), has implemented full PEtab support.

Complementary to the method development it is highly relevant to apply these methods in more biology-driven research. For the pan-cancer model, it will be important to increase the training data further to improve parameter identifiability and predictive power of the model. Large-scale datasets have been generated providing e.g. relative measurements of drug responses on the protein level (Zhao et al., 2020) or CRISPR-Cas9 essentiality screens (Behan et al., 2019) which could be leveraged for this. Possible applications are in-silico drug screens or the identification of new drug targets. Additionally, one could incorporate prior information on the parameters from public databases such as BRENDA (Schomburg et al., 2002). Besides integrating more data, the coverage of further relevant pathways will also be important to improve model predictions. Qualitative data has so far only been used in a few application examples to infer parameters. With the advance of novel methods, this is likely to increase in the future. More often, qualitative observations have been used for model validation, e.g. by Tan et al. (2017), and it would be interesting to include this data also for model training. While the here developed methods were applied to mechanistic ODE models, they are also applicable to a broader range of models.

To conclude this thesis, we developed efficient methods for parameter estimation of mathematical models using different experimental data types, such as relative or qualitative measurements. We showed that these methods could be used on realistic application examples to substantially reduce computation times and enable the integration of larger datasets with comprehensive large-scale models. This demonstrates that the methods developed in this thesis can facilitate a deeper, holistic understanding of biological processes on a systems level.

Bibliography

Lorenz Adlung, Sandip Kar, Marie-Christine Wagner, Bin She, Sajib Chakraborty, Jie Bao, Susen Lattermann, Melanie Boerries, Hauke Busch, Patrick Wuchter, Anthony D Ho, Jens Timmer, Marcel Schilling, Thomas Höfer, & Ursula Klingmüller. Protein abundance of AKT and ERK pathway components governs cell type-specific regulation of proliferation. *Mol. Syst. Biol.*, 13(1): 904, 2017. ISSN 1744-4292. doi: 10.15252/msb.20167258.

Ozan Alkan, Birgit Schoeberl, Millie Shah, Alexander Koshkaryev, Tim Heinemann, Daryl C. Drummond, Michael B. Yaffe, & Andreas Raue. Modeling chemotherapy-induced stress to identify rational combination therapies in the dna damage response pathway. *Science Signaling*, 11(540), 2018. ISSN 1945-0877. doi: 10.1126/scisignal.aat0229.

Ann C. Babtie & Michael P. H. Stumpf. How to deal with parameters for whole-cell modelling. *J. R. Soc. Interface*, 14(133), 2017. ISSN 1742-5689. doi: 10.1098/rsif.2017.0237.

J. Bachmann, A. Raue, M. Schilling, M. E. Böhm, C. Kreutz, D. Kaschek, H. Busch, N. Gretz, W. D. Lehmann, J. Timmer, & U. Klingmüller. Division of labor by dual feedback regulators controls JAK2/STAT5 signaling over broad ligand range. *Mol. Syst. Biol.*, 7(1):516, July 2011.

T. Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, New York and Oxford, 1996.

B. Ballnus, S. Hug, K. Hatz, L. Görlitz, J. Hasenauer, & F. J. Theis. Comprehensive benchmarking of Markov chain Monte Carlo methods for dynamical systems. *BMC Syst Biol*, 11(63):63, 2017. doi: 10.1186/s12918-017-0433-1.

E. Balsa-Canto & J. R. Banga. AMIGO, a toolbox for advanced model identification in systems biology using global optimization. *Bioinformatics*, 27(16):2311–2313, Aug. 2011.

J. Barretina, G. Caponigro, N. Stransky, K. Venkatesan, A. A. Margolin, S. Kim, C. J. Wilson, J. Lehár, G. V. Kryukov, D. Sonkin, A. Reddy, M. Liu, L. Murray, M. F. Berger, J. E. Monahan, P. Morais, J. Meltzer, A. Korejwa, J. Jané-Valbuena, F. A. Mapa, J. Thibault, E. Bric-Furlong, P. Raman, A. Shipway, I. H. Engels, J. Cheng, G. K. Yu, J. Yu, P. Aspesi, Jr, M. de Silva,

K. Jagtap, M. D. Jones, L. Wang, C. Hatton, E. Palescandolo, S. Gupta, S. Mahan, C. Sougnez, R. C. Onofrio, T. Liefeld, L. MacConaill, W. Winckler, M. Reich, N. Li, J. P. Mesirov, S. B. Gabriel, G. Getz, K. Ardlie, V. Chan, V. E. Myer, B. L. Weber, J. Porter, M. Warmuth, P. Finan, J. L. Harris, M. Meyerson, T. R. Golub, M. P. Morrissey, W. R. Sellers, R. Schlegel, & L. A. Garraway. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603–607, Mar. 2012. doi: 10.1038/nature11003.

Ralf Beer, Konrad Herbst, Nikolaos Ignatiadis, Ilia Kats, Lorenz Adlung, Hannah Meyer, Dominik Niopek, Tania Christiansen, Fanny Georgi, Nils Kurzawa, Johanna Meichsner, Sophie Rabe, Anja Riedel, Joshua Sachs, Julia Schessner, Florian Schmidt, Philipp Walch, Katharina Niopek, Tim Heinemann, Roland Eils, & Barbara Di Ventura. Creating functional engineered variants of the single-module non-ribosomal peptide synthetase indc by t domain exchange. *Mol. BioSyst.*, 10:1709–1718, 2014. doi: 10.1039/C3MB70594C.

Fiona M Behan, Francesco Iorio, Gabriele Picco, Emanuel Gonçalves, Charlotte M Beaver, Giorgia Migliardi, Rita Santos, Yanhua Rao, Francesco Sassi, Marika Pinnelli, et al. Prioritization of cancer therapeutic targets using crispr–cas9 screens. *Nature*, 568(7753):511–516, 2019.

Frank T. Bergmann, Richard Adams, Stuart Moodie, Jonathan Cooper, Mihai Glont, Martin Golebiewski, Michael Hucka, Camille Laibe, Andrew K. Miller, David P. Nickerson, Brett G. Olivier, Nicolas Rodriguez, Herbert M. Sauro, Martin Scharm, Stian Soiland-Reyes, Dagmar Waltemath, Florent Yvon, & Nicolas Le Novère. Combine archive and omex format: one file to share all information to reproduce a modeling project. *BMC bioinformatics*, 15:369, December 2014. ISSN 1471-2105. doi: 10.1186/s12859-014-0369-z.

Marc R Birtwistle, Alexander von Kriegsheim, Katarzyna Kida, Juliane P Schwarz, Kurt I Anderson, & Walter Kolch. Linear approaches to intramolecular förster resonance energy transfer probe measurements for quantitative modeling. *PloS one*, 6(11):e27823, 2011.

T. Blasi, C. Feller, J. Feigelman, J. Hasenauer, A. Imhof, F. J. Theis, P. B. Becker, & C. Marr. Combinatorial histone acetylation patterns are generated by motif-specific reactions. *Cell Systems*, 2(1):49–58, Jan. 2016.

H. H. G. Bock & K. J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proc. of the 9th IFAC World Congress, Budapest, Hungary*, pages 242–247, July 1984.

Martin E Boehm, Lorenz Adlung, Marcel Schilling, Susanne Roth, Ursula Klingmueller, & Wolf D Lehmann. Identification of isoform-specific dynamics in phosphorylation-dependent stat5 dimerization by quantitative mass spectrometry and mathematical modeling. *Journal of Proteome Research*, 13(12):5685–5694, 2014.

JoséA.M. Borghans, Geneviève Dupont, & Albert Goldbeter. Complex intracellular calcium oscillations a theoretical exploration of possible mechanisms. *Biophysical Chemistry*, 66(1):25 – 41, 1997. ISSN 0301-4622. doi: https://doi.org/10.1016/S0301-4622(97)00010-0.

M. Bouhaddou, A. M. Barrette, A. D. Stern, R. J. Koch, M. S. DiStefano, E. A. Riesel, L. C. Santos, A. L. Tan, A. E. Mertz, & M. R. Birtwistle. A mechanistic pan-cancer pathway model informed by multi-omics data interprets stochastic cell fate responses to drugs and mitogens. *PLoS Comput. Biol.*, 14(3):e1005985, Mar. 2018.

S. Boyd & L. Vandenberghe. *Convex Optimisation*. Cambridge University Press, UK, 2004.

Mary Ann Branch, Thomas F Coleman, & Yuying Li. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM J. Sci. Comput.*, 21(1):1–23, 1999. doi: 10.1137/s1064827595289108.

C. Brännmark, R. Palmer, S. T. Glad, G. Cedersund, & P. Strålfors. Mass and information feedbacks through receptor endocytosis govern insulin signaling as revealed using a parameter-free modeling framework. *J. Biol. Chem.*, 285(26):20171–20179, June 2010.

Alexander Brooks, Wei Dou, Xiaoying Yang, Tara Brosnan, Michael Pargett, Laurel A Raftery, & David M Umulis. Bmp signaling in wing development: A critical perspective on quantitative image analysis. *FEBS letters*, 586(14):1942–1952, 2012.

Mark Bruno, Julian Koschmieder, Florian Wuest, Patrick Schaub, Mirjam Fehling-Kaschek, Jens Timmer, Peter Beyer, & Salim Al-Babili. Enzymatic study on AtCCD4 and AtCCD7 and their potential to form acyclic regulatory metabolites. *Journal of Experimental Botany*, 67(21):5993–6005, 10 2016. ISSN 0022-0957. doi: 10.1093/jxb/erw356.

Trent AJ Butler, Jonathan W Paul, Eng-Cheng Chan, Roger Smith, & Jorge M Tolosa. Misleading westerns: Common quantification mistakes in western blot densitometry and proposed corrective measures. *BioMed research international*, 2019, 2019.

Richard H Byrd, Humaid Fayez Khalfan, & Robert B Schnabel. Analysis of a symmetric rank-one trust region method. *SIAM J. Optim.*, 6(4):1025–1039, 1996. doi: 10.1137/s1052623493252985.

Richard H. Byrd, Mary E. Hribar, & Jorge Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9:877–900, 1999.

Richard H. Byrd, Jean Charles Gilbert, & Jorge Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Math. Program.*, 89(1):149–185, Nov 2000. ISSN 1436-4646. doi: 10.1007/PL00011391.

Arturo Casadevall & Ferric C. Fang. Reproducible science. *Infection and Immunity*, 78(12):4972–4975, 2010. ISSN 0019-9567. doi: 10.1128/IAI.00908-10.

Katherine C Chen, Laurence Calzone, Attila Csikasz-Nagy, Frederick R Cross, Bela Novak, & John J Tyson. Integrative analysis of cell cycle control in budding yeast. *Molecular biology of the cell*, 15(8):3841–3862, 2004.

W. W. Chen, B. Schoeberl, P. J. Jasper, M. Niepel, U. B. Nielsen, D. A. Lauffenburger, & P. K. Sorger. Input–output behavior of ErbB signaling pathways as revealed by a mass action model trained against dynamic data. *Mol. Syst. Biol.*, 5(1): 239, Jan. 2009. doi: 10.1038/msb.2008.74.

Kiri Choi, Lucian P Smith, J Kyle Medley, & Herbert M Sauro. phrased-ml: A paraphrased, human-readable adaptation of sed-ml. *Journal of bioinformatics and computational biology*, 14(06):1650035, 2016.

Kiri Choi, J. Kyle Medley, Matthias König, Kaylene Stocking, Lucian Smith, Stanley Gu, & Herbert M. Sauro. Tellurium: An extensible python-based modeling environment for systems and synthetic biology. *Bio Systems*, 171:74–79, September 2018. ISSN 1872-8324. doi: 10.1016/j.biosystems.2018.07.006.

Matthias Chung, Justin Krueger, & Mihai Pop. Identification of microbiota dynamics using robust parameter estimation methods. *Mathematical biosciences*, 294: 71–84, 2017.

E. A. Coddington & N. Levinson. *Theory of ordinary differential equations*. McGraw-Hill, New York, 1955.

Fabien Crauste, Julien Mafille, Lilia Boucinha, Sophia Djebali, Olivier Gandrillon, Jacqueline Marvel, & Christophe Arpin. Identification of nascent memory CD8 T cells and modeling of their ontogeny. *Cell Systems*, 4(3):306–317, 2017.

Autumn A. Cuellar, Catherine M. Lloyd, Poul F. Nielsen, David P. Bullivant, David P. Nickerson, & Peter J. Hunter. An overview of CellML 1.1, a biological model description language. *Simulation*, 79(12):740–747, 2003. doi: 10.1177/0037549703040939.

Haskell B Curry. The method of steepest descent for non-linear minimization problems. *Quarterly of Applied Mathematics*, 2(3):258–261, 1944.

Andrea Degasperi, Dirk Fey, & Boris N Kholodenko. Performance of objective functions and optimisation procedures for parameter estimation in system biology models. *npj Syst Biol Appl*, 3(1):20, 2017. doi: 10.1038/s41540-017-0023-2.

Federica Eduati, Victoria Doldàn-Martelli, Bertram Klinger, Thomas Cokelaer, Anja Sieber, Fiona Kogera, Mathurin Dorel, Mathew J. Garnett, Nils Blüthgen, & Julio Saez-Rodriguez. Drug resistance mechanisms in colorectal cancer dissected with cell type-specific dynamic logic models. *Cancer Res.*, 77(12): 3364–3375, 2017. ISSN 0008-5472. doi: 10.1158/0008-5472.CAN-17-0078.

J. A. Egea, M. Rodriguez-Fernandez, J. R. Banga, & R. Marti. Scatter search for chemical and bio-process optimization. *J. Global Optim.*, 37(3):481–503, Oct. 2007. doi: 10.1007/s10898-006-9075-3.

J. A. Egea, D. Henriques, T. Cokelaer, A. F. Villaverde, A. MacNamara, D. P. Danciu, J. R. Banga, & J. Saez-Rodriguez. MEIGO: An open-source software suite based on metaheuristics for global optimization in systems biology and bioinformatics. *BMC Bioinf.*, 15(136), 2014. doi: 10.1186/1471-2105-15-136.

Michael B Elowitz & Stanislas Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, 2000.

D. Fey, M. Halasz, D. Dreidax, S. P. Kennedy, J. F. Hastings, N. Rauch, A. Garcia Munoz, R. Pilkington, M. Fischer, F. Westermann, W. Kolch, B. N. Kholodenko, & D. R. Croucher. Signaling pathway models as biomarkers: Patient-specific simulations of JNK activity predict the survival of neuroblastoma patients. *Sci. Signal.*, 8(408), Dec. 2015. doi: 10.1126/scisignal.aab0990.

Anthony V Fiacco. Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical programming*, 10(1):287–311, 1976.

A. Fiedler, S. Raeth, F. J. Theis, A. Hausser, & J. Hasenauer. Tailored parameter optimization methods for ordinary differential equation models with steady-state constraints. *BMC Syst. Biol.*, 10(80), Aug. 2016. doi: 10.1186/s12918-016-0319-7.

Roger Fletcher & Michael JD Powell. A rapidly convergent descent method for minimization. *Comp J*, 6(2):163–168, 1963. doi: 10.1093/comjnl/6.2.163.

Martin Frejno, Riccardo Zenezini Chiozzi, Mathias Wilhelm, Heiner Koch, Runsheng Zheng, Susan Klaeger, Benjamin Ruprecht, Chen Meng, Karl Kramer, Anna Jarzab, Stephanie Heinzlmeir, Elaine Johnstone, Enric Domingo, David Kerr, Moritz Jesinghaus, Julia Slotta-Huspenina, Wilko Weichert, Stefan Knapp, Stephan M Feller, & Bernhard Kuster. Pharmaco-

proteomic characterisation of human colon and rectal cancer. *Molecular Systems Biology*, 13(11):951, 2017. doi: 10.15252/msb.20177701.

F. Fröhlich, B. Kaltenbacher, F. J. Theis, & J. Hasenauer. Scalable parameter estimation for genome-scale biochemical reaction networks. *PLoS Comput Biol*, 13(1): e1005331, 2017. doi: 10.1371/journal.pcbi.1005331.

Fabian Fröhlich, Thomas Kessler, Daniel Weindl, Alexey Shadrin, Leonard Schmiester, Hendrik Hache, Artur Muradyan, Moritz Schütte, Ji-Hyun Lim, Matthias Heinig, Fabian J. Theis, Hans Lehrach, Christoph Wierling, Bodo Lange, & Jan Hasenauer. Efficient parameter estimation enables the prediction of drug response using a mechanistic pan-cancer pathway model. *Cell Syst.*, 7(6):567–579.e6, Dec. 2018. ISSN 2405-4712. doi: https://doi.org/10.1016/j.cels.2018.10.013.

Fabian Fröhlich, Carolin Loos, & Jan Hasenauer. Scalable inference of ordinary differential equation models of biochemical processes. In G. Sanguinetti & V. A. Huynh-Thu, editors, *Gene Regulatory Networks: Methods and Protocols*, volume 1883 of *Methods in Molecular Biology*, chapter 16, pages 385–422. Humana Press, 1 edition, 2019.

Kazuhiro A Fujita, Yu Toyoshima, Shinsuke Uda, Yu-ichi Ozaki, Hiroyuki Kubota, & Shinya Kuroda. Decoupling of receptor and downstream signals in the akt pathway by its low-pass filter characteristics. *Sci Signal*, 3(132):ra56, Jul 2010. doi: 10.1126/scisignal.2000810.

David M. Gay. Algorithm 611: Subroutines for unconstrained minimization using a model/trust-region approach. *ACM Trans. Math. Softw.*, 9(4):503–524, December 1983. ISSN 0098-3500. doi: 10.1145/356056.356066.

Amin Moghaddas Gholami, Hannes Hahne, Zhixiang Wu, Florian Johann Auer, Chen Meng, Mathias Wilhelm, & Bernhard Kuster. Global proteome analysis of the nci-60 cell line panel. *Cell Reports*, 4(3):609–620, Aug 2013. ISSN 2211-1247. doi: 10.1016/j.celrep.2013.07.018.

William L Goffe, Gary D Ferrier, & John Rogers. Global optimization of statistical functions with simulated annealing. *J. Econometrics*, 60(1-2):65–99, 1994. doi: 10.1016/0304-4076(94)90038-8.

Leonard A. Harris, Justin S. Hogg, José-Juan Tapia, John A. P. Sekar, Sanjana Gupta, Ilya Korsunsky, Arshi Arora, Dipak Barua, Robert P. Sheehan, & James R. Faeder. BioNetGen 2.2: advances in rule-based modeling. *Bioinformatics*, 32(21):3366–3368, 07 2016. ISSN 1367-4803. doi: 10.1093/bioinformatics/btw469.

Helge Hass, Kristina Masson, Sibylle Wohlgemuth, Violette Paragas, John E. Allen, Mark Sevecka, Emily Pace, Jens Timmer, Joerg Stelling, Gavin MacBeath, Birgit Schoeberl, & Andreas Raue. Predicting ligand-dependent

tumors from multi-dimensional signaling features. *npj Syst Biol Appl*, 3(1):27, 2017. doi: 10.1038/s41540-017-0030-3.

Helge Hass, Carolin Loos, Elba Raimúndez-Álvarez, Jens Timmer, Jan Hasenauer, & Clemens Kreutz. Benchmark problems for dynamic modeling of intracellular processes. *Bioinformatics*, 35(17):3073–3082, 01 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz020.

L. A. Herzenberg, J. Tung, W. A. Moore, L. A. Herzenberg, & D. R. Parks. Interpreting flow cytometry data: A guide for the perplexed. *Nat. Immunol.*, 7(7): 681–685, July 2006. doi: 10.1038/ni0706-681.

A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, & C. S. Woodward. SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers. *ACM T. Math. Software.*, 31(3):363–396, September 2005. doi: 10.1145/1089014.1089020.

Robert Hooke & Terry A Jeeves. "Direct Search" solution of numerical and statistical problems. *J. ACM*, 8(2):212–229, 1961. doi: 10.1145/321062.321069.

S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, & U. Kummer. COPASI – a COmplex PAthway SImulator. *Bioinformatics*, 22(24):3067–3074, 2006. doi: 10.1093/bioinformatics/btl485.

M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, I. I. Goryanin, W. J. Hedley, T. C. Hodgman, J.-H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. Le Novère, L. M. Loew, D. Lucio, P. Mendes, E. Minch, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, & J. Wang. The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003. doi: 10.1093/bioinformatics/btg015.

B. Ingalls. *Mathematical modelling in systems biology: An introduction*. MIT Press, 2013.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001.

Eva-Maria Kapfer, Paul Stapor, & Jan Hasenauer. Challenges in the calibration of large-scale ordinary differential equation models. *IFAC-PapersOnLine*, 52(26):58–64, Dec. 2019.

J. R. Karr, J. C. Sanghvi, D. N. Macklin, M. V. Gutschow, J. M. Jacobs, B. Bolival Jr, N. Assad-Garcia, J. I. Glass, & M. W. Covert. A whole-cell computational model predicts phenotype from genotype. *Cell*, 150(2):389–401, July 2012. doi: 10.1016/j.cell.2012.05.044.

J. R. Karr, A. H. Williams, J. D. Zucker, A. Raue, B. Steiert, J. Timmer, C. Kreutz, DREAM8 Parameter Estimation Challenge Consortium, S. Wilkinson, B. A. Allgood, B. M. Bot, B. R. Hoff, M. R. Kellen, M. W. Covert, G. A. Stolovitzky, & P. Meyer. Summary of the DREAM8 parameter estimation challenge: Toward parameter identification for whole-cell models. *PLoS Comput. Biol.*, 11(5):e1004096, May 2015. doi: 10.1371/journal.pcbi.10004096.

Daniel Kaschek, Wolfgang Mader, Mirjam Fehling-Kaschek, Marcus Rosenblatt, & Jens Timmer. Dynamic modeling, parameter estimation, and uncertainty analysis in R. *J. Stat. Softw.*, 88(10), 2019.

James Kennedy. Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer, 2011.

S. Kirkpatrick, C. D. Gelatt Jr, & M. P. M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983. doi: 10.1126/science.220.4598.671.

H. Kitano. Systems biology: A brief overview. *Science*, 295(5560):1662–1664, Mar. 2002.

Emmanuel Klinger, Dennis Rickert, & Jan Hasenauer. pyABC: distributed, likelihood-free inference. *Bioinformatics*, 34(20):3591–3593, Oct. 2018. doi: 10.1093/bioinformatics/bty361.

P. Kokotovic & J. Heller. Direct and adjoint sensitivity equations for parameter optimization. *IEEE T. Autom. Contr.*, 12(5):609–610, 1967. doi: 10.1109/tac.1967.1098670.

Charles D Kolstad & Leon S Lasdon. Derivative evaluation and computational experience with large bilevel mathematical programs. *Journal of optimization theory and applications*, 65(3):485–499, 1990.

Anil Korkut, Weiqing Wang, Emek Demir, Bülent Arman Aksoy, Xiaohong Jing, Evan J Molinelli, Özgün Babur, Debra L Bemis, Selcuk Onur Sumer, David B Solit, et al. Perturbation biology nominates upstream–downstream drug combinations in raf inhibitor resistant melanoma cells. *Elife*, 4:e04640, 2015.

Paul F. Lang, Sungho Shin, & Victor M. Zavala. Sbml2julia: interfacing sbml with efficient nonlinear julia modelling and solution tools for parameter optimization, 2020.

N. Le Novère, B. Bornstein, A. Broicher, M. Courtot, M. Donizelli, H. Dharuri, L. Li, H. Sauro, M. Schilstra, B. Shapiro, J. L. Snoep, & M. Hucka. BioModels Database: A free, centralized database of curated, published,

quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res.*, 34(Database issue):D689–D691, Jan. 2006.

J. Li, Y. Lu, R. Akbani, Z. Ju, P. L. Roebuck, W. Liu, J.-Y. Yang, B. M. Broom M, R. G. W. Verhaak, D. W. Kane, C. Wakefield, J. N. Weinstein, G. B. Mills, & H. Liang. TCPA: A resource for cancer functional proteomics data. *Nat. Methods*, 10(11):1046–1047, Nov. 2013. doi: 10.1038/nmeth.2650.

Jun Li, Wei Zhao, Rehan Akbani, Wenbin Liu, Zhenlin Ju, Shiyun Ling, Christopher P. Vellano, Paul Roebuck, Qinghua Yu, A. Karina Eterovic, Lauren A. Byers, Michael A. Davies, Wanleng Deng, Y. N. Vashisht Gopal, Guo Chen, Erika M. von Euw, Dennis Slamon, Dylan Conklin, John V. Heymach, Adi F. Gazdar, John D. Minna, Jeffrey N. Myers, Yiling Lu, Gordon B. Mills, & Han Liang. Characterization of human cancer cell lines by reverse-phase protein arrays. *Cancer Cell*, 31(2):225–239, 2017/05/08 2017. doi: 10.1016/j.ccell.2017.01.005.

Carolin Loos, Sabrina Krause, & Jan Hasenauer. Hierarchical optimization for the efficient parametrization of ODE models. *Bioinf.*, 34(24):4266–4273, July 2018. doi: 10.1093/bioinformatics/bty514.

Timo Lubitz, Jens Hahn, Frank T. Bergmann, Elad Noor, Edda Klipp, & Wolfram Liebermeister. SBtab: a flexible table format for data exchange in systems biology. *Bioinformatics*, 32(16):2559–2561, 04 2016. ISSN 1367-4803. doi: 10.1093/bioinformatics/btw179.

C. Maier, C. Loos, & J. Hasenauer. Robust parameter estimation for dynamical systems from outlier-corrupted data. *Bioinformatics*, 33(5):718–725, Mar. 2017. doi: 10.1093/bioinformatics/btw703.

Eshan D Mitra & William S Hlavacek. Bayesian Inference Using Qualitative Observations of Underlying Continuous Variables. *Bioinformatics*, 02 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa084. btaa084.

Eshan D Mitra, Raquel Dias, Richard G Posner, & William S Hlavacek. Using both qualitative and quantitative data in parameter identification for systems biology models. *Nature communications*, 9(1):3901, 2018.

Eshan D Mitra, Ryan Suderman, Joshua Colvin, Alexander Ionkov, Andrew Hu, Herbert M Sauro, Richard G Posner, & William S Hlavacek. PyBioNetFit and the biological property specification language. *iScience*, 19:1012–1036, September 2019. ISSN 2589-0042. doi: 10.1016/j.isci.2019.08.045.

John A Nelder & Roger Mead. A simplex method for function minimization. *Comput. J.*, 7(4):308–313, 1965. doi: 10.1093/comjnl/7.4.308.

Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980. doi: 10.2307/2006193.

Jorge Nocedal & Stephen Wright. *Numerical Optimization*. Springer Science & Business Media, 2006. doi: 10.1007/b98874.

L. A. Ogilvie, C. Wierling, T. Kessler, H. Lehrach, & B. M. H. Lange. Predictive modeling of drug treatment in the area of personalized medicine. *Cancer Informatics*, 14(S4):95–103, 2015. doi: 10.4137/CIN.S19330.

C. Oguz, T. Laomettachit, K. C. Chen, L. T. Watson, W. T. Baumann, & J. J. Tyson. Optimization and model reduction in the high dimensional parameter space of a budding yeast cell cycle model. *BMC Syst. Biol.*, 7(53), Jun. 2013. doi: 10.1186/1752-0509-7-53.

Brett G Olivier & Jacky L Snoep. Web-based kinetic modelling using JWS Online. *Bioinf.*, 20(13):2143–4, Sep 2004. doi: 10.1093/bioinformatics/bth200.

Michael Pargett & David M Umulis. Quantitative model analysis with diverse biological data: applications in developmental pattern formation. *Methods*, 62(1):56–67, 2013.

Michael Pargett, Ann E Rundell, Gregery T Buzzard, & David M Umulis. Model-based analysis for qualitative data: an application in drosophila germline stem cell regulation. *PLoS Comput. Biol.*, 10(3):e1003498, 2014.

David R Penas, Patricia González, José A Egea, Julio R Banga, & Ramón Doallo. Parallel metaheuristics in computational biology: An asynchronous cooperative enhanced scatter search method. *Procedia Comput. Sci.*, 51:630–639, 2015. doi: 10.1016/j.procs.2015.05.331.

David R Penas, Patricia González, Jose A Egea, Ramón Doallo, & Julio R Banga. Parameter estimation in large-scale systems biology models: a parallel and self-adaptive cooperative strategy. *BMC bioinformatics*, 18(1):52, 2017.

Roger D Peng. Reproducible research in computational science. *Science*, 334(6060): 1226–1227, 2011.

Alan S Perelson, Avidan U Neumann, Martin Markowitz, John M Leonard, & David D Ho. Hiv-1 dynamics in vivo: virion clearance rate, infected cell life-span, and viral generation time. *Science*, 271(5255):1582–1586, 1996.

Michael JD Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*, pages 51–67. Springer, 1994.

Michael JD Powell. The bobyqa algorithm for bound constrained optimization without derivatives. Technical report, Cambridge NA Report NA2009/06, University of Cambridge, Cambridge, 2009.

Florian Prinz, Thomas Schlange, & Khusru Asadullah. Believe it or not: how much can we rely on published data on potential drug targets? *Nature reviews Drug discovery*, 10(9):712–712, 2011.

Christopher Rackauckas, Yingbo Ma, Vaibhav Dixit, Xingjian Guo, Mike Innes, Jarrett Revels, Joakim Nyberg, & Vijay Ivaturi. A comparison of automatic differentiation and continuous sensitivity analysis for derivatives of differential equation solutions. *arXiv preprint arXiv:1812.01892*, 2018.

S M Ashrafur Rahman, Naveen K Vaidya, & Xingfu Zou. Impact of early treatment programs on hiv epidemics: An immunity-based mathematical model. *Mathematical biosciences*, 280:38—49, October 2016. ISSN 0025-5564. doi: 10.1016/j.mbs.2016.07.009.

Valentina Raia, Marcel Schilling, Martin Böhm, Bettina Hahn, Andreas Kowarsch, Andreas Raue, Carsten Sticht, Sebastian Bohl, Maria Saile, Peter Möller, et al. Dynamic mathematical modeling of il13-induced signaling in hodgkin and primary mediastinal b-cell lymphoma allows prediction of therapeutic targets. *Cancer research*, 71(3):693–704, 2011.

A. Raue, C. Kreutz, T. Maiwald, J. Bachmann, M. Schilling, U. Klingmüller, & J. Timmer. Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics*, 25(25):1923–1929, May 2009. doi: 10.1093/bioinformatics/btp358.

A. Raue, C. Kreutz, F. J. Theis, & J. Timmer. Joining forces of Bayesian and frequentist methodology: A study for inference in the presence of non-identifiability. *Philos T Roy Soc A*, 371(1984), 2013a. doi: 10.1098/rsta.2011.0544.

A. Raue, M. Schilling, J. Bachmann, A. Matteson, M. Schelke, D. Kaschek, S. Hug, C. Kreutz, B. D. Harms, F. J. Theis, U. Klingmüller, & J. Timmer. Lessons learned from quantitative dynamical modeling in systems biology. *PLoS ONE*, 8 (9):e74335, Sept. 2013b. doi: 10.1371/journal.pone.0074335.

A. Raue, B. Steiert, M. Schelker, C. Kreutz, T. Maiwald, H. Hass, J. Vanlier, C. Tönsing, L. Adlung, R. Engesser, W. Mader, T. Heinemann, J. Hasenauer, M. Schilling, T. Höfer, E. Klipp, F. J. Theis, U. Klingmüller, B. Schöberl, & J.Timmer. Data2Dynamics: a modeling environment tailored to parameter estimation in dynamical systems. *Bioinformatics*, 31(21):3558–3560, 2015. doi: 10.1093/bioinformatics/btv405.

J. Renart, J. Reiser, & G. R. Stark. Transfer of proteins from gels to diazobenzyloxymethyl-paper and detection with antisera: a method for studying antibody specificity and antigen structure. *Proc. Natl. Acad. Sci. USA*, 76(7):3116–3120, July 1979.

Luis Miguel Rios & Nikolaos V. Sahinidis. Derivative-free optimization: A review of algorithms and comparison of software implementations. *J. Global Optim.*, 56(3): 1247–1293, Jul 2013. ISSN 1573-2916. doi: 10.1007/s10898-012-9951-y.

Thomas P. Runarsson & Xin Yao. Stochastic ranking for constrained evolutionary optimization. *IEEE T Evolut Comput*, 4(3):284–294, 2000. doi: 10.1109/4235.873238.

Emanuel Salazar-Cavazos, Carolina Franco Nitta, Eshan D. Mitra, Bridget S. Wilson, Keith A. Lidke, William S. Hlavacek, & Diane S. Lidke. Multisite egfr phosphorylation is regulated by adaptor protein abundances and dimer lifetimes. *Molecular Biology of the Cell*, 31(7):695–708, 2020. doi: 10.1091/mbc.E19-09-0548.

Geir Kjetil Sandve, Anton Nekrutenko, James Taylor, & Eivind Hovig. Ten simple rules for reproducible computational research. *PLoS Comput Biol*, 9(10):e1003285, Oct 2013. doi: 10.1371/journal.pcbi.1003285.

Y. Schälte, P. Stapor, & J. Hasenauer. Evaluation of derivative-free optimizers for parameter estimation in systems biology. *to appear in Proc. of the Foundations of Syst. Biol. in Engin. (FOSBE)*, 2018.

Yannik Schälte, Fabian Fröhlich, Paul Stapor, Dantong Wang, Daniel Weindl, Leonard Schmiester, & Jan Hasenauer. ICB-DCM/pyPESTO: pyPESTO 0.0.11, March 2020. https://doi.org/10.5281/zenodo.3715448.

K. Schittkowski. A robust implementation of a sequential quadratic programming algorithm with successive error restoration. *Optimization Letters*, 5(2):283–296, 2011. doi: 10.1007/s11590-010-0207-9.

Leonard Schmiester, Yannik Schälte, Fabian Fröhlich, Jan Hasenauer, & Daniel Weindl. Efficient parameterization of large-scale dynamic models based on relative measurements. *Bioinformatics*, 36(2):594–602, 07 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz581.

Leonard Schmiester, Yannik Schälte, Frank T. Bergmann, Tacio Camba, Erika Dudkin, Janine Egert, Fabian Fröhlich, Lara Fuhrmann, Adrian L. Hauber, Svenja Kemmer, Polina Lakrisenko, Carolin Loos, Simon Merkt, Dilan Pathirana, Elba Raimúndez, Lukas Refisch, Marcus Rosenblatt, Paul L. Stapor, Philipp Städter, Dantong Wang, Franz-Georg Wieland, Julio R. Banga, Jens Timmer, Alejandro F. Villaverde, Sven Sahle, Clemens Kreutz, Jan Hasenauer, & Daniel Weindl. PEtab – interoperable specification of parameter estimation problems in systems biology. *arXiv preprint arXiv:2004.01154*, 2020a.

Leonard Schmiester, Daniel Weindl, & Jan Hasenauer. Parameterization of mechanistic models from qualitative data using an efficient optimal scaling approach. *J. Math. Biol.*, 81(2):603–623, July 2020b. doi: 10.1007/s00285-020-01522-w.

Robin Schmucker, Gabriele Farina, James Faeder, Fabian Froehlich, Ali Sinan Saglam, & Tuomas Sandholm. Combination treatment optimization using a pan-cancer pathway model. *bioRxiv*, 2020. doi: 10.1101/2020.07.05.184960.

Ida Schomburg, Antje Chang, & Dietmar Schomburg. Brenda, enzyme data and metabolic information. *Nucleic acids research*, 30(1):47–49, 2002.

Lars Ole Schwen, Arne Schenk, Clemens Kreutz, Jens Timmer, María Matilde Bartolomé Rodríguez, Lars Kuepfer, & Tobias Preusser. Representative sinusoids for hepatic four-scale pharmacokinetics simulations. *PLOS ONE*, 10(7):1–39, 07 2015. doi: 10.1371/journal.pone.0133653.

Roger N Shepard. The analysis of proximities: multidimensional scaling with an unknown distance function. i. *Psychometrika*, 27(2):125–140, 1962.

Kieran Smallbone & Pedro Mendes. Large-scale metabolic models: From reconstruction to differential equations. *Ind. Biotechnol.*, 9(4):179–184, 2013. doi: 10.1089/ind.2013.0003.

Lucian P. Smith, Frank T. Bergmann, Deepak Chandran, & Herbert M. Sauro. Antimony: a modular model definition language. *Bioinformatics*, 25(18):2452–2454, 09 2009. doi: 10.1093/bioinformatics/btp401.

James Sneyd & Jean-Francois Dufour. A dynamic model of the type-2 inositol trisphosphate receptor. *Proceedings of the National Academy of Sciences*, 99(4):2398–2403, 2002. ISSN 0027-8424. doi: 10.1073/pnas.032281999.

D. C. Sorensen. Newton's method with a model trust region modification. *SIAM J. Numer. Anal.*, 19(2):409–426, 1982. doi: 10.1137/0719026.

Natalie J Stanford, Martin Scharm, Paul D Dobson, Martin Golebiewski, Michael Hucka, Varun B Kothamachu, David Nickerson, Stuart Owen, Jürgen Pahle, Ulrike Wittig, et al. Data management in computational systems biology: exploring standards, tools, databases, and packaging best practices. In *Yeast Systems Biology*, pages 285–314. Springer, 2019.

Paul Stapor, Fabian Fröhlich, & Jan Hasenauer. Optimization and profile calculation of ODE models using second order adjoint sensitivity analysis. *Bioinformatics*, 34 (13):i151–i159, 2018a.

Paul Stapor, Daniel Weindl, Benjamin Ballnus, Sabine Hug, Carolin Loos, Anna Fiedler, Sabrina Krause, Sabrina Hross, Fabian Fröhlich, & Jan

Hasenauer. PESTO: Parameter EStimation TOolbox. *Bioinformatics*, 34(4):705–707, 2018b. doi: 10.1093/bioinformatics/btx676.

Paul Stapor, Leonard Schmiester, Christoph Wierling, Bodo M.H. Lange, Daniel Weindl, & Jan Hasenauer. Mini-batch optimization enables training of ode models on large-scale datasets. *bioRxiv*, 2019. doi: 10.1101/859884.

Philip M. Tan, Kyle S. Buchholz, Jeffrey H. Omens, Andrew D. McCulloch, & Jeffrey J. Saucerman. Predictive model identifies key network regulators of cardiomyocyte mechano-signaling. *PLOS Computational Biology*, 13(11):1–17, 11 2017. doi: 10.1371/journal.pcbi.1005854.

M. Uzkudun, L. Marcon, & J. Sharpe. Data-driven modelling of a gene regulatory network for cell fate decisions in the growing limb bud. *Mol. Syst. Biol.*, 11(7):815, July 2015. doi: 10.15252/msb.20145882.

Alejandro F Villaverde, Jose A Egea, & Julio R Banga. A cooperative strategy for parameter estimation in large scale systems biology models. *BMC systems biology*, 6 (1):75, 2012.

Alejandro F. Villaverde, Fabian Froehlich, Daniel Weindl, Jan Hasenauer, & Julio R Banga. Benchmarking optimization methods for parameter estimation in large kinetic models. *Bioinformatics*, page bty736, 2018.

A. Wächter & L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.*, 106(1): 25–57, 2006. doi: 10.1007/s10107-004-0559-y.

D. Waltemath, R. Adams, F. T. Bergmann, M. Hucka, F. Kolpakov A. K. Miller, I. I. Moraru, D. Nickerson, J. L. Snoep, & N. Le Novère. Reproducible computational biology experiments with SED-ML – The Simulation Experiment Description Markup Language. *BMC Syst. Biol.*, 5(198), Dec. 2011a. doi: 10.1186/1752-0509-5-198.

Dagmar Waltemath & Olaf Wolkenhauer. How modeling standards, software, and initiatives support reproducibility in systems biology and systems medicine. *IEEE Transactions on Biomedical Engineering*, 63(10):1999–2006, 2016.

Dagmar Waltemath, Richard Adams, Daniel A Beard, Frank T Bergmann, Upinder S Bhalla, Randall Britten, Vijayalakshmi Chelliah, Michael T Cooling, Jonathan Cooper, Edmund J Crampin, Alan Garny, Stefan Hoops, Michael Hucka, Peter Hunter, Edda Klipp, Camille Laibe, Andrew K Miller, Ion Moraru, David Nickerson, Poul Nielsen, Macha Nikolski, Sven Sahle, Herbert M Sauro, Henning Schmidt, Jacky L Snoep, Dominic Tolle, Olaf Wolkenhauer, & Nicolas Le Novère. Minimum information about a

simulation experiment (miase). *PLoS Comput. Biol.*, 7(4):e1001122, April 2011b. doi: 10.1371/journal.pcbi.1001122.

R.A. Waltz, J.L. Morales, J. Nocedal, & D. Orban. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Math. Program.*, 107(3):391–408, Jul 2006. ISSN 1436-4646. doi: 10.1007/s10107-004-0560-5.

P. Weber, J. Hasenauer, F. Allgöwer, & N. Radde. Parameter estimation and identifiability of biological networks using relative data. In S. Bittanti, A. Cenedese, & S. Zampieri, editors, *Proc. of the 18th IFAC World Congress*, volume 18, pages 11648–11653, Milano, Italy, Aug. 2011. doi: 10.3182/20110828-6-IT-1002.01007.

Patrick Weber, Mariana Hornjik, Monilola A. Olayioye, Angelika Hausser, & Nicole E. Radde. A computational model of pkd and cert interactions at the trans-golgi network of mammalian cells. *BMC Systems Biology*, 9(1):9, Feb 2015. ISSN 1752-0509. doi: 10.1186/s12918-015-0147-1.

Mark D Wilkinson, Michel Dumontier, I Jsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, Jildau Bouwman, Anthony J Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J G Gray, Paul Groth, Carole Goble, Jeffrey S Grethe, Jaap Heringa, Peter A C 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J Lusher, Maryann E Martone, Albert Mons, Abel L Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, & Barend Mons. The fair guiding principles for scientific data management and stewardship. *Sci Data*, 3:160018, Mar 2016. doi: 10.1038/sdata.2016.18.

W. Yang, J. Soares, P. Greninger, E. J. Edelman, H. Lightfoot, S. Forbes, N. Bindal, D. Beare, J. A. Smith, I. R. Thompson, S. Ramaswamy, P. A. Futreal, D. A. Haber, M. R. Stratton, C. Benes, U. McDermott, & M. J. Garnett. Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucl. Acids Res.*, 41(Database issue):D955–D961, Jan. 2013. doi: 10.1093/nar/gks1111.

X.S. Yang. *Nature-inspired metaheuristic algorithms*. Luniver Press, Bristol, UK, 2nd edition, 2010.

Bo Yuan, Ciyue Shen, Augustin Luna, Anil Korkut, Debora S. Marks, John Ingraham, & Chris Sander. Interpretable machine learning for perturbation biology. *bioRxiv*, https://doi.org/10.1101/746842, August 2019.

Ya-xiang Yuan. Recent advances in trust region algorithms. *Math. Program.*, 151(1): 249–281, 2015.

Wei Zhao, Jun Li, Mei-Ju M Chen, Yikai Luo, Zhenlin Ju, Nicole K Nesser, Katie Johnson-Camacho, Christopher T Boniface, Yancey Lawrence, Nupur T Pande, et al. Large-scale characterization of drug responses of clinically relevant proteins in cancer cell lines. *Cancer Cell*, 2020.

Y. Zheng, S. M. M. Sweet, R. Popovic, E. Martinez-Garcia, J. D. Tipton, P. M. Thomas, J. D. Licht, & N. L. Kelleher. Total kinetic analysis reveals how combinatorial methylation patterns are established on lysines 27 and 36 of histone H3. *Proc. Natl. Acad. Sci. USA*, 109(34):13549–13554, Aug. 2012. doi: 10.1073/pnas.1205707109.