Bioinformatics, 36(2), 2020, 594–602
doi: 10.1093/bioinformatics/btz581
Advance Access Publication Date: 26 July 2019
Original Paper

Systems biology

Efficient parameterization of large-scale dynamic models based on relative measurements

Leonard Schmiester[1,2,†], Yannik Schälte[1,2,†], Fabian Fröhlich[1,2], Jan Hasenauer [1,2,3,*] and Daniel Weindl [1]

[1]Institute of Computational Biology, Helmholtz Zentrum München – German Research Center for Environmental Health, 85764 Neuherberg, Germany, [2]Center for Mathematics, Technische Universität München, 85748 Garching, Germany and [3]Faculty of Mathematics and Natural Sciences, University of Bonn, 53113 Bonn, Germany

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.
Associate Editor: Russell Schwartz

Received on March 15, 2019; revised on June 26, 2019; editorial decision on July 17, 2019; accepted on July 18, 2019

Abstract

Motivation: Mechanistic models of biochemical reaction networks facilitate the quantitative understanding of biological processes and the integration of heterogeneous datasets. However, some biological processes require the consideration of comprehensive reaction networks and therefore large-scale models. Parameter estimation for such models poses great challenges, in particular when the data are on a relative scale.

Results: Here, we propose a novel hierarchical approach combining (i) the efficient analytic evaluation of optimal scaling, offset and error model parameters with (ii) the scalable evaluation of objective function gradients using adjoint sensitivity analysis. We evaluate the properties of the methods by parameterizing a pan-cancer ordinary differential equation model (>1000 state variables, >4000 parameters) using relative protein, phosphoprotein and viability measurements. The hierarchical formulation improves optimizer performance considerably. Furthermore, we show that this approach allows estimating error model parameters with negligible computational overhead when no experimental estimates are available, providing an unbiased way to weight heterogeneous data. Overall, our hierarchical formulation is applicable to a wide range of models, and allows for the efficient parameterization of large-scale models based on heterogeneous relative measurements.

Availability and implementation: Supplementary code and data are available online at http://doi.org/10.5281/zenodo.3254429 and http://doi.org/10.5281/zenodo.3254441.

Contact: jan.hasenauer@uni-bonn.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

In systems biology, mechanistic ordinary differential equation (ODE) models are widely used to deepen the understanding of biological processes. Applications range from the description of signaling pathways (Klipp *et al.*, 2005) to the prediction of drug responses (Hass *et al.*, 2017) and patient survival (Fey *et al.*, 2015). With the availability of scalable computational methods and increasing computing power, larger and larger models have been developed to capture the intricacies of biological regulatory networks more accurately (Bouhaddou *et al.*, 2018; Fröhlich *et al.*, 2018). In Fröhlich *et al.* (2018), we demonstrated how such a large-scale

mechanistic model integrating various cancer-related signaling pathways is able to, e.g. predict the response of cancer cells to drug combinations based on measurements for single treatment responses, a task which is commonly not possible with statistical models. Overall, mechanistic models can pave the way to personalized medicine by integrating patient specific information, and thus creating virtual patients (Kühn and Lehrach, 2012; Ogilvie *et al.*, 2015).

Mechanistic ODE models usually contain parameters such as reaction rate constants and initial concentrations, which have to be inferred from experimental data. Parameter estimation for larger models is limited by (i) computational power for large numbers of required model simulations and gradient evaluations, as well as by (ii) the availability of data to infer parameter values. Scalable methods have been developed to address the problem of computational complexity, e.g. adjoint sensitivity analysis (Fröhlich *et al.*, 2017b; Fujarewicz *et al.*, 2005; Lu *et al.*, 2013) and parallelization (Fröhlich *et al.*, 2018; Penas *et al.*, 2015). Complementary, large-scale transcriptomics, proteomics and pharmacological datasets have been acquired and have been made publicly available in databases such as the Cancer Cell Line Encyclopedia (CCLE) (Barretina *et al.*, 2012), the Genomics of Drug Sensitivity in Cancer project (Eduati *et al.*, 2017) and the MD Anderson Cell Lines Project (MCLP) (Li *et al.*, 2017).

The available databases are rather comprehensive and cover already hundreds of cell-lines. Yet, those datasets are usually relative measurements and data often undergo some type of normalization, which has to be accounted for when linking mechanistic model simulations to the data. A commonly used approach is to introduce scaling and offset parameters in the model outputs (Degasperi *et al.*, 2017; Raue *et al.*, 2013; Weber *et al.*, 2011). However, this increases the dimensionality of the optimization problem and slows down optimization. Indeed, even a small number of scaling factors can result in a substantial drop of optimizer performance (Degasperi *et al.*, 2017). The precise reasons are yet to be understood.

To improve optimizer performance, Weber *et al.* (2011) developed a hierarchical optimization method which exploits the fact that for given dynamic parameters, the optimal scaling parameters can be computed analytically, which improved convergence and reduced computation time. The approach was generalized by Loos *et al.* (2018) to error model parameters and different noise distributions. However, the available approaches only considered scaling parameters, but not offset parameters. In addition, those approaches were not compatible with adjoint sensitivity analysis, but only with forward sensitivity analysis, which is computationally prohibitive for large-scale models.

Here, we (i) analyze the problems caused by the introduction of scaling factors and (ii) extend the hierarchical optimization method introduced by Loos *et al.* (2018) to be used in combination with adjoint sensitivity analysis. Furthermore, we derive the governing equations to not only include scaling parameters, but also offset parameters and the combination of both as well as error model parameters in the case of additive Gaussian noise. Our method is more general and achieves a better scaling behavior than the existing ones (Loos *et al.*, 2018; Weber *et al.*, 2011). We apply it to estimate parameters for the large-scale pan-cancer signaling model from Fröhlich *et al.* (2018). First, we use simulated relative and absolute data to compare the performance of the standard and the novel hierarchical approach and to demonstrate the loss of information associated with using only relative data. Second, we use measured data to estimate model parameters, compare the performance of different optimization algorithms, and show how the performance of each of them improves with our hierarchical optimization approach.

2 Materials and methods

2.1 Mechanistic modeling

We consider ODE models of biochemical processes of the form

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

The state vector $x(t,\theta,u) \in \mathbb{R}^{n_x}$ denotes the concentrations of involved species, the vector field $f(x,\theta,u) \in \mathbb{R}^{n_x}$ describes the temporal evolution of the states, the vector $\theta \in \mathbb{R}^{n_\theta}$ unknown parameters, the vector $u \in \mathbb{R}^{n_u}$ differential experimental conditions and $x_0 \in \mathbb{R}^{n_x}$ the parameter- and condition-dependent states at initial time t_0.

An observation function h maps the system states to observables $y(t,\theta,u) \in \mathbb{R}^{n_y}$, via

$$y(t,\theta,u) = h(x(t,\theta,u),\theta,u).$$

Experimental data $\mathcal{D} = \{\bar{y}_{i_t,i_y,i_u}\}_{(i_t,i_y,i_u)\in I}$ corresponding to the observables are time-discrete and subject to measurement noise $\epsilon \in \mathbb{R}^{n_y}$,

$$\bar{y}_{i_t,i_y,i_u} = h_{i_y}(x(t_{i_t},\theta,u_{i_u}),\theta) + \epsilon_{i_t,i_y,i_u},$$

indexed over a finite index set I of time points i_t, observables i_y and experimental conditions i_u. We assume the measurement noise to be normally distributed and independent for all datapoints, i.e. $\epsilon_{i_t,i_y,i_u} \sim \mathcal{N}(0,\sigma^2_{i_t,i_y,i_u})$.

2.2 Relative measurements

Frequently, experiments provide measurement data only in a relative form, in arbitrary units, rather than as absolute concentrations. Thus, to compare model and data, the observables need to be rescaled. While the rescaling is usually incorporated in h and θ, here we use an explicit formulation. Since these cover a broad range of measurement types, we assume that we have scaling factors s and offsets b such that simulations and measured data are related via

$$\bar{y}_{i_t,i_y,i_u} = s_{i_t,i_y,i_u} \cdot \tilde{h}_{i_y}(x(t_{i_t},\theta,u_{i_u}),\theta) + b_{i_t,i_y,i_u} + \epsilon_{i_t,i_y,i_u},$$

in which $\tilde{h}(x,\theta)$ denotes the mapping to unscaled observables.

Scaling factors s_{i_t,i_y,i_u} and offsets b_{i_t,i_y,i_u}, but also noise parameters σ_{i_t,i_y,i_u}, in the setting considered here the standard deviations of Gaussian distributions, are often shared between some datapoints, e.g. for time series measurements, or for data taken under the same experimental conditions. In the following, we summarize all different scaling, offset and noise parameters in vectors $s \in \mathbb{R}^{n_s}$, $b \in \mathbb{R}^{n_b}$ and $\sigma \in \mathbb{R}^{n_\sigma}$, respectively, and refer to them as static parameters, to distinguish them from the original parameters θ, henceforth called dynamic parameters, since they affect the dynamics of the simulated states. The static parameters are often unknown and thus have to be estimated along with the dynamic parameters.

2.3 Parameter estimation problem with relative data

To infer the unknown parameters θ, s, b and σ, we maximize the likelihood

$$L(\theta,s,b,\sigma) = \prod_i \pi(\bar{y}_i \,|\, s_i \cdot \tilde{h}_i(\theta) + b_i, \sigma_i),$$

of observing the experimental data $\mathcal{D} = \{\bar{y}_i\}_{i\in I}$ given parameters θ, s, b, σ, where for simplicity of presentation we employ a general index set $i \in I$ over time points, observables and experimental conditions. π denotes the conditional probability of observing $\bar{y}_i$ given

simulation $y_i = s_i \cdot \tilde{h}_i(\theta) + b_i$ and noise parameters σ_i. For Gaussian noise, we have

$$\pi(\bar{y}_i \mid y_i, \sigma_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(\bar{y}_i - y_i)^2}{2\sigma_i^2}\right).$$

Instead of maximizing L directly, it is equivalent and numerically often preferable to minimize the negative log-likelihood $\min_{\theta,s,b,\sigma} J(\theta, s, b, \sigma)$ with $J = -\log L$. Assuming Gaussian noise, J becomes

$$J(\theta, s, b, \sigma) = \frac{1}{2}\sum_i \left[\log(2\pi\sigma_i^2) + \frac{(\bar{y}_i - (s_i\tilde{h}_i(\theta) + b_i))^2}{\sigma_i^2}\right], \quad (1)$$

which will henceforth be referred to as objective function.

2.4 Hierarchical optimization

In this section, we generalize the hierarchical optimization approach introduced by Loos *et al.* (2018) to (1), allowing for scaling, offset and noise parameters simultaneously, and we outline how hierarchical optimization can be combined with adjoint sensitivities.

The standard approach to handle the static parameters is to consider the extended parameter vector (θ, s, b, σ) and to optimize all its elements simultaneously. However, the increased dimension makes the optimization problem in general harder to solve. Instead, we can make use of the specific problem structure of (1) more efficiently by splitting the optimization problem into an outer problem where we optimize the dynamic parameters θ, and an inner problem where we optimize the static parameters s, b and σ, conditioned on θ. That is, we compute

$$\min_\theta \hat{J}(\theta) \quad \text{with} \quad \hat{J}(\theta) := J(\theta, s(\theta), b(\theta), \sigma(\theta)), \quad (2)$$

in which

$$(s(\theta), b(\theta), \sigma(\theta)) = \arg\min_{s,b,\sigma} \; J(\theta, s, b, \sigma). \quad (3)$$

It can be shown that global optima of the standard optimization problem are preserved in the hierarchical problem (Supplementary Material, Section 1).

2.4.1 Analytic expressions for the optimal scaling, offset and noise parameters

In general, an inner optimization problem like (3) needs to be solved numerically. However, under certain conditions one can give analytic expressions for the optimal static parameters, which renders solving the inner problem computationally very cheap. The analytic expressions are based on evaluating the necessary condition for a local minimum in s, b, σ given θ,

$$\nabla_{s,b,\sigma} J(\theta, s, b, \sigma) = 0. \quad (4)$$

Here, we extend the available results by Weber *et al.* (2011) and Loos *et al.* (2018).

We define index sets $I^s_\alpha, I^b_\beta, I^\sigma_\gamma \subset I$ for $\alpha = 1, \ldots, n_s$, $\beta = 1, \ldots, n_b$, $\gamma = 1, \ldots, n_\sigma$, with n_s, n_b and n_σ indicating the number of scaling, offset and noise parameters. The index sets indicate which datapoints share static parameters, e.g. all datapoints $\bar{y}_i$ with $i \in I^s_\alpha$ share a scaling parameter. In order to derive analytic formulas, we will in the following assume that $\{I^s_\alpha\}_\alpha = \{I^b_\beta\}_\beta$, i.e. that scaling and offset parameters are shared among the same datapoints, and that for all α there exists γ such that $I^s_\alpha \subset I^\sigma_\gamma$, i.e. that datapoints sharing the scaling (and offset) parameter share also the noise parameter. Furthermore, we also allow for any of the s, b or σ to be fixed (e.g. $s = 1$ when no scaling factor is necessary) or estimated as dynamic parameters. For an extended discussion and derivations of the below formulas see the Supplementary Material, Section 3.

First, we consider single scaling parameters s_α and offset parameters b_β. Without loss of generality, we reduce the objective (1) to only include relevant summands. Then, (4) yields

$$s_\alpha(\theta) = \left(\sum_{i\in I_\alpha} \frac{\tilde{h}_i^2}{\sigma_i^2}\right)^{-1} \left(\sum_{i\in I_\alpha} \frac{(\bar{y}_i - b_i)\tilde{h}_i}{\sigma_i^2}\right), \quad (5a)$$

$$b_\beta(\theta) = \left(\sum_{i\in I_\beta} \frac{1}{\sigma_i^2}\right)^{-1} \left(\sum_{i\in I_\beta} \frac{\bar{y}_i - s_i\tilde{h}_i}{\sigma_i^2}\right). \quad (5b)$$

If either the s_i or b_i are no static parameters, we are done by just inserting those values in the respective other formula. If both are to be optimized as static parameters, in which case by assumption $s_i \equiv s_\alpha, b_i \equiv b_\beta$, we can proceed by inserting (5a) into (5b), which yields non-interdependent formulas, see the Supplementary Material, Section 3.1. Note that the noise parameters drop out of the formulas if all values coincide, as is our assumption in the case that we want to estimate the noise parameters hierarchically as well. Thus, in either case $s_\alpha(\theta)$ and $b_\beta(\theta)$ can now be readily computed. Note that for the special case $b = 0$ we recover the formula from Loos *et al.* (2018).

Second, for a given single noise parameter σ_γ, we consider without loss of generality an objective function (1) reduced to indices I_γ, while s_i and b_i can be arbitrary. The objective considered here will typically contain multiple sums of the type discussed for the scalings and offsets. As s and b are known already at this stage, (4) immediately gives

$$\sigma_\gamma^2(\theta) = \left(\sum_{i\in I_\gamma} 1\right)^{-1} \left(\sum_{i\in I_\gamma} (\bar{y}_i - (s_i\tilde{h}_i + b_i))^2\right).$$

Note that a problem occurs when the rescaled simulations match the measured data exactly, since then $\sigma^2 = 0$. In this case, the noise parameter and thus the objective function is unbounded in the standard and the hierarchical formulation, so that measures to deal with this case have to be taken, e.g. by specifying a lower bound for σ_γ.

Inspection of the Hessian $\nabla^2_{s,b,\sigma} J(\theta, s, b, \sigma)$ shows that the found stationary points indeed are minima (see Supplementary Material, Section 3).

2.4.2 Combining hierarchical optimization and adjoint sensitivity analysis

In optimization, the objective function gradient is of considerable value, because it gives the direction of steepest descent in the objective function landscape. Recent studies indicated that optimization methods using gradients tend to outperform those which do not (Schälte *et al.*, 2018; Villaverde *et al.*, 2018). In Loos *et al.* (2018), hierarchical optimization was performed using objective gradients computed via forward sensitivity analysis. However, for large-scale models adjoint sensitivity analysis has shown to be orders of magnitude faster (Fröhlich *et al.*, 2018), because essentially here the evaluation of state sensitivities is circumvented by defining an adjoint state $p \in \mathbb{R}^{n_x}$ which does not scale in the number of parameters (Fröhlich *et al.*, 2017b). For a derivation of the adjoint equation see also the Supplementary Material, Sections 2.3 and 3.3.

Whether hierarchical optimization can be combined with adjoint sensitivity analysis so far remained unclear. Unlike the forward sensitivity equations, the adjoint state depends on the data and the scaled observables and thus requires knowledge of the static parameters. Therefore, the approaches by Weber *et al.* (2011) and Loos

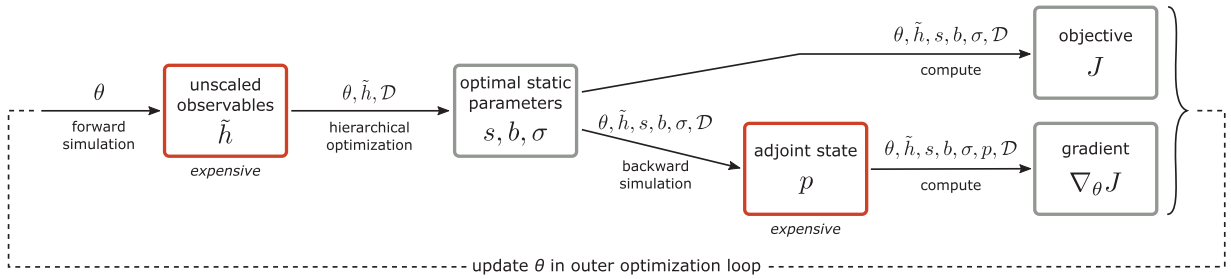

Fig. 1. Illustration of the hierarchical optimization scheme using adjoint sensitivities. In the outer loop, θ is updated by the employed iterative gradient-based optimization method. When a new value of θ is proposed, an inner loop is entered, in which the optimal static parameters are computed for the given θ, and objective function value and gradient are returned before exiting the inner loop. Here, the solution of the inner problem is shown in detail. The red boxes involve the simulation of ODEs and are thus usually computationally more expensive. If the gradient is not required in some optimizer iteration, the adjoint and gradient steps can be omitted. Note that the dependence of s, b, σ, p, J and ∇J on θ is in the setting considered in this study only indirect via $\tilde{h}$, while in general also an explicit dependence is possible. (Color version of this figure is available at *Bioinformatics* online.)

et al. (2018) of first simulating the state trajectory $x(t, \theta, u)$ as well as all required sensitivities, and then computing optimal static parameters in order to compute $\hat{J}$ and $\nabla\hat{J}$ without further simulations, are not applicable.

To combine hierarchical optimization and adjoint sensitivity analysis, we derived the scheme illustrated in Figure 1.

In an outer optimization loop, iteratively new dynamic parameters θ are proposed. For each such θ, in an inner loop we compute the corresponding conditionally optimal static parameters $s(\theta), b(\theta), \sigma(\theta)$, which here involves just an analytic calculation. Only after we have obtained the static parameters, do we simulate the adjoint state p allowing to efficiently calculate the objective function gradient. As the derivatives of the objective function with respect to the optimal static parameters are zero, i.e. $\nabla_{s,b,\sigma} J = 0$, since we solve the inner subproblem exactly, we can prove that this scheme provides the correct objective function gradient $\nabla\hat{J}$. For a more detailed discussion and derivation of the adjoint-hierarchical approach, we refer to the Supplementary Material, Section 2. An overview over the properties of the different hierarchical optimization approaches is provided in the Supplementary Table S1.

2.5 Implementation

We implemented the proposed method in MATLAB and C++. A custom parallelized objective function implementation was used to decrease the wall time (see Supplementary Material, Sections 4.5.2 and 4.5.3). The modular implementation can be adopted to work with other Systems Biology Markup Language (Hucka *et al.*, 2003) models as described in Supplementary Material, Section 4.6. Model simulation and gradient evaluation using the proposed scheme were performed using AMICI (Fröhlich *et al.*, 2017a). Parameter estimation was performed using multi-start local optimization. The starting points were sampled from a uniform distribution. The initial dynamic parameters were identical for the standard and hierarchical optimization, where initial static parameters only had to be chosen for the standard approach. We considered different local optimization methods (see Section 3) and ran all for a maximum of 150 iterations (see Supplementary Material, Section 4.5.1 for more details). The complete code and data are available at http://doi.org/10.5281/zenodo.3254429 and http://doi.org/10.5281/zenodo.3254441.

3 Results

In this study, we considered the pan-cancer signaling pathway model developed by Fröhlich *et al.* (2018). This model comprises 1396 biochemical species (1228 dynamic states and 168 constant species) and 4232 unknown parameters, and can be individualized to specific cancer cell-lines using genetic profiles and gene expression data. Fröhlich *et al.* (2018) demonstrated a promising performance of the model in drug response prediction, but molecular insights were limited by non-identifiabilities. Motivated by these results, we set out to parameterize this model using additional data.

3.1 Mapping multiple datasets to a large-scale model of cancer signaling

For model calibration, we considered two datasets. *Dataset 1* is the training data studied by Fröhlich *et al.* (2018). These are viability measurements for 96 cancer cell-lines in response to 7 drugs at 8 drug concentrations available in the CCLE (Barretina *et al.*, 2012). The viability measurements are normalized to the respective control. To account for this normalization, Fröhlich *et al.* (2018) simulated the model for the treated condition and the control, and the simulations were then divided by each other. This corresponds to the method proposed by Degasperi *et al.* (2017). However, this approach is not applicable if multiple observables need to be considered, e.g. when incorporating additional data types, or when more complex data normalizations are applied. Therefore, we reformulated the model output and replaced the normalization with the control by a cell-line specific scaling ($s_{\text{cell-line}_j}$). This yields the observation model

$$y_{\text{viability}_i} = s_{\text{cell-line}_j} \tilde{h}_{\text{viability}_i} + \epsilon_{\text{viability}_i}$$

with i indexing the datapoints belonging to cell-line j. The measurement noise is assumed to be normally distributed, $\epsilon_{\text{viability}_i} \sim \mathcal{N}(0, \sigma^2_{\text{viability}})$.

We complemented the viability measurements employed with molecular measurements to refine the parameter estimates. *Dataset 2* contains reverse phase protein array (phospho-)proteomic data for various cancer cell-lines taken from the MCLP (Li *et al.*, 2017). We developed a pipeline which (i) maps the measured protein levels to the state variables of the model and (ii) employs the mapping to construct observables (see Supplementary Material, Section 4.1 for more details). We identified 32 proteins and 16 phosphoproteins measured that were also covered by the model. In total, 54 out of the 96 considered cell-lines were included in the MCLP (*dataset 2* in Table 1). In the MCLP database, measurements are normalized across cell-lines and across all proteins by subtracting the respective median from the log2-transformed measured values (see *Level 4 data* in https://tcpaportal.org/mclp/#/faq). Therefore, we included

one cell-line specific offset ($b_{\text{cell-line}_j}$) and one protein specific offset (b_{protein_i}), yielding the observation model

$$y_{\text{protein}_i,\text{cell-line}_j} = \log_2(\tilde{h}_{\text{protein}_i,\text{cell-line}_j}) + b_{\text{cell-line}_j} + b_{\text{protein}_i} + \epsilon_{\text{protein}_i,\text{cell-line}_j},$$

normally distributed measurement noise $\epsilon_{\text{protein}_{i,j}} \sim \mathcal{N}(0, \sigma^2_{\text{protein}_i})$ and the simulated absolute protein level

$$\tilde{h}_{\text{protein}_i,\text{cell-line}_j} = \sum_{l \in I_{\text{protein}_i}} k_l x_l.$$

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

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 observation parameters (Table 1).

3.2 Evaluation of standard and hierarchical optimization using simulated data

A priori it is not clear which influence scaling, offset and noise parameter have on optimizer performance. However, Degasperi *et al.* (2017) observed in two examples that the use of scalings lead to inferior optimizer behavior compared to the normalization-based approach which was also used by Fröhlich *et al.* (2018). Thus, before estimating parameters using real measured data from CCLE and MCLP, we first used simulated data. To get realistic data, we simulated the model for the same experimental conditions that were provided in *dataset 1* and added normally distributed noise to the simulations (see Supplementary Material, Section 4.7 for a detailed description of the data generation and an analysis of the simulated data). The simulation of experimental data allowed us to (i) compare the goodness-of-fit of estimated and true parameter and to (ii) assess the information associated with relative data.

Table 1. Datasets used for parameter estimation

	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)[a]	48 (48)

Note: The number of static parameters of certain classes is indicated, followed by the number of parameters which are computed analytically in the hierarchical setting in parentheses.

[a]The noise parameter is set to one if dataset 1 is considered individually.

3.2.1 Hierarchical optimization facilitates convergence

To compare standard and hierarchical optimization, we employed both approaches for the analysis of simulated, noisy relative data. For local optimization we employed the Interior Point OPTimizer (Ipopt) (Wächter and Biegler, 2006). As metric we considered the Pearson correlations between data and simulation for each of the optimized parameter vectors and the true parameter vector. The Pearson correlation reflected the objective function value (Supplementary Fig. S3) but was easier to interpret.

The hierarchical optimization achieved substantially better correlations between simulation and data than the standard optimization (Fig. 2A). Furthermore, variability between different local optimization runs was reduced. Indeed, all but two optimizer runs using hierarchical optimization achieved correlations similar to the correlation observed for the true parameters, indicating a good model fit and—in contrast to the standard optimization—a good convergence. No run found a substantially better scoring fit than the true parameters, which would indicate over-fitting.

3.2.2 Scalings have a pronounced influence on the objective function value

Hierarchical optimization decreases the effective dimension of the optimization problem. However, as the number of parameters decreases for the considered problem only by 2%—this does not explain the substantially improved convergence—the scaling factors might be particularly relevant. To assess this, we evaluated the average absolute values of the objective function gradient for scaling

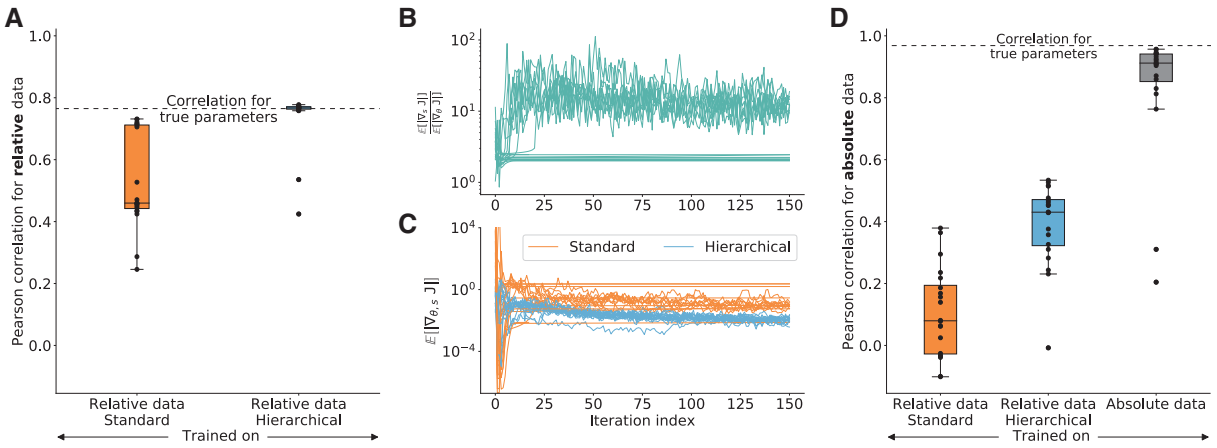

Fig. 2. Convergence of standard and hierarchical optimization. Parameter estimation results using a simulated version of *dataset 1* from Table 1 with the Ipopt optimizer. For all evaluations 20 optimizer runs were performed. (**A**) Pearson correlation of relative training data and corresponding model simulation after training on relative data using standard and hierarchical optimization. Dashed line shows the correlation that is achieved using the true parameters used to generate the training data. (**B**) Ratio of the average gradient contribution for scaling parameters against dynamic parameters using the standard optimization for all optimizer runs along their trajectory. (**C**) Expected gradient for standard and hierarchical optimization. Only the parameters, that were optimized numerically, were taken into account. (**D**) Pearson correlation of absolute data and corresponding model simulation after training on (left and middle) relative data and (right) on absolute data

parameters ($E[|\nabla_s J|]$) and dynamic parameters ($E[|\nabla_\theta J|]$). Indeed, the evaluation of the ratio ($E[|\nabla_s J|]/E[|\nabla_\theta J|]$) revealed that the objective function is in most optimizer runs 10 times more sensitive to scaling parameters than to dynamic parameters (Fig. 2B). This indicates that the elimination of the scaling factors will improve the conditioning of the optimization problem. As the condition number of an optimization problem has a pronounced influence on the convergence rate (Boyd and Vandenberghe, 2004, Chapter 9.3), the removal of the scaling factors can substantially improve the convergence rate. Accordingly, the average absolute value of the gradient decreases for the hierarchical optimization faster than for the standard optimization (Fig. 2C).

An inspection of the optimizer trajectories revealed that for the standard optimization some optimizer runs show flat trajectories of the objective function, while still having a comparably large gradient (Fig. 2C and Supplementary Fig. S4). For these runs, the contribution of the scalings became small (flat lines in Fig. 2B), which might be due to a valley in the objective function landscape defined by the scaling parameters, where the optimizer got stuck. Such valleys are eliminated in the hierarchical optimization.

3.2.3 Normalization results in information loss

To assess the influence of information loss associated with the use of relative data, we performed optimization using simulated absolute data. For comparison, we predicted the absolute values using the parameters inferred with relative data (see Supplementary Material, Section 4.7.3). As expected, we found that the prediction of absolute data from relative data yields a correlation far from one (Fig. 2D), implying that information is lost in the normalization process. Interestingly, hierarchical optimization again outperformed standard optimization. A potential reason is that the improved convergence of the optimizer allows for the extraction of more information from the relative data.

3.3 All tested local optimization methods profit from hierarchical formulation

To provide a thorough comparison of the performance of standard and hierarchical optimization, we assessed it for different local optimization algorithms on the measured viability data (*dataset 1*). We considered four commonly used or open-source optimizers: Ipopt (Wächter and Biegler, 2006), Ceres (http://ceres-solver.org), sumsl (Gay, 1983) and fmincon (https://de.mathworks.com/help/optim/ug/fmincon.html). These optimizers use different updating schemes, e.g. based on line-search or trust-region methods.

We assessed the performance by studying the evolution of objective function values over computation time and optimizer iterations. Given the same computational budget, the hierarchical optimization consistently achieved better objective function values for all considered optimization algorithms and for almost all runs (Fig. 3A and Supplementary Fig. S5). Furthermore, the objective function at the maximum number of iterations was substantially better for hierarchical optimization than standard optimization, and there was in general a lower variability (Fig. 3B). Given this result, we determined the computation time required by the hierarchical optimization to achieve the final objective function value of the standard optimization and computed the resulting speed-up (Fig. 3C). Except for one start of Ipopt, the hierarchical optimization was always faster with a median speed-up between one and two orders of magnitude. Since a single local optimization run required several thousand hours of computation time, the efficiency improvement achieved using hierarchical optimization is crucial. Indeed, the hierarchical optimization only needed tens to hundreds of computation hours to find the same objective function values for which the standard optimization required thousands of computation hours (Supplementary Fig. S6).

As the performance of optimization algorithms has so far mostly been evaluated for ODE models with tens and hundreds of unknown parameters (Hass *et al.*, 2019; Villaverde *et al.*, 2018), we used our results for a first comparison on a large-scale ODE model. We found that for the considered problem (i) Ceres always stopped prematurely, (ii) sumsl progressed (at least for the standard optimization) slower than Ipopt and fmincon and (iii) fmincon and Ipopt reached the best objective function values and appeared to be most efficient (Fig. 3A and B).

3.4 Hierarchical optimization enables integration of heterogeneous data

As the information about molecular mechanisms provided by viability measurements (*dataset 1*) are limited, we complemented it using

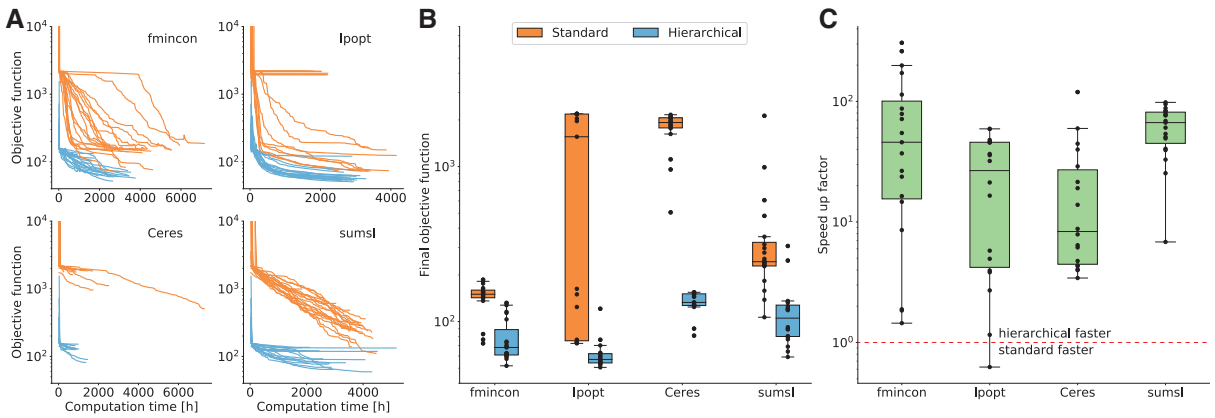

Fig. 3. Computational efficiency of standard and hierarchical optimization for multiple optimization algorithms. (**A**) Optimizer trajectories for fmincon, Ipopt, Ceres and sumsl using standard and the hierarchical optimization. Since the noise parameter was set to 1 for these runs, the constant term in the objective function was omitted. *dataset 1* from Table 1 was used. Fmincon runs were performed on different systems and using a different implementation than the other optimizers, so that absolute computation times are not comparable. (**B**) Boxplots of final objective function values obtained after 150 iterations by the different optimizers using standard and hierarchical optimization. (**C**) Speed-up of the hierarchical optimization compared to the standard optimization. The speed-up is defined by the computation time the hierarchical optimization required to find the final objective function value of the standard optimization for every local optimization (or vice versa if the standard optimization finds a better final value). The dashed red line shows the point at which standard and hierarchical are equally fast. (Color version of this figure is available at *Bioinformatics* online.)

the (phospho-)protein measurements (*dataset 2*). An unbiased weighting was ensured by introducing error model parameters (i.e. standard deviations) for the individual observables and estimating them along with the remaining parameters. In hierarchical optimization, (i) the error model parameters, (ii) the cell-line specific scaling of the viability measurements and (iii) the observable-specific offsets of the log-transformed protein measurements are optimized analytically (Table 1). The analytic optimization of the cell-line specific offsets of the log-transformed protein measurements is not supported by the approach as the error model parameters and the offsets have to share the same datapoints.

We performed multi-start local optimization for the combined dataset using Ipopt. Again, the hierarchical optimization was computationally much more efficient and reached better objective function values than the standard optimization (Fig. 4A). For the standard optimization, all starts yielded objective function values of $\sim 10^4$. For the hierarchical optimization, we observed runs yielding objective values similar to those for standard optimization denoted by Group 1, with $J \approx 10^4$ as well as runs which provided much better objective function values, i.e. Group 2, $J < 3 \times 10^3$. The optimized parameter vectors obtained using standard optimization runs and hierarchical optimization runs in Group 1 were able to fit the viability measurements but failed to describe the protein data (Fig. 4B). In contrast, the optimized parameter vectors obtained using hierarchical optimization runs in Group 2 show a good fit for viability and most protein measurements (Fig. 4B). Accordingly, only hierarchical optimization runs managed to balance the fit of the datasets, thereby achieving an integration and a better overall description of the data.

While the computation time for forward sensitivities scales linearly with the number of parameters, it stays constant for adjoint sensitivities, leading to an approximately 2700-fold speed-up for the here considered model (Supplementary Fig. S1A). With this, we estimated the computation time of a full optimization using a forward-hierarchical approach (Loos *et al.*, 2018; Weber *et al.*, 2011) to be in the order of 10^6–10^7 h (>1000 years) (see Supplementary Fig. S1B), which is roughly three orders of magnitude slower than the adjoint-hierarchical approach. In summary, the adjoint-hierarchical approach outperformed in all regards standard and forward optimization.

4 Discussion

Parameterization of large-scale mechanistic models is a challenging task requiring new approaches. Here, we combine the concept of hierarchical optimization (Loos *et al.*, 2018; Weber *et al.*, 2011) with adjoint sensitivities (Fröhlich *et al.*, 2017b; Fujarewicz *et al.*, 2005; Lu *et al.*, 2013). This is crucial when parameterizing large-scale models for which the use of forward sensitivities is computationally prohibitive. Additionally, we derived more general formulas for hierarchically optimizing a combination of scaling and offset parameters as well as noise parameters.

The proposed method is intended for cases where relative measurements are available or measurement uncertainties are not known. We would like to emphasize that it is neither able to, nor meant to, make absolute measurements or assessment of measurement uncertainties obsolete. It comes as no surprise that absolute measurements contain much more information than relative measurements, as we illustrated using a synthetic dataset. Introducing additional output parameters will increase degrees of freedom, and therefore, should be a deliberate modeling decision, based on the requirements of the data at hand. Whenever it is possible to obtain absolute measurements with manageable overhead, this would be the preferred way to go. In cases where calibration curves or similar data are available, relative data can and should be converted to absolute data before parameter estimation. Along the same lines, all measurement uncertainties would ideally be known beforehand. However, in many datasets this information is absent or only inaccurate estimates based on very low sample sizes are available (Raue *et al.*, 2013). A common approach is then estimating error model parameters along with kinetic model parameters. However, this will blur inadequacies of the model and the data. Independently of the hierarchical approach, such estimates for noise parameters provide the noise level under which the given model and data would be the most plausible, but not necessarily an accurate estimate of the true levels of measurement noise. Therefore, in an ideal world, the proposed method would not be necessary, and offset, scaling and noise parameter would be known prior to parameter estimation. However, in reality this is not the case for most current (large-scale) datasets, and thus, the respective parameters need to be estimated.

For this reason, we developed this hierarchical optimization approach and demonstrated its advantages using a recently published

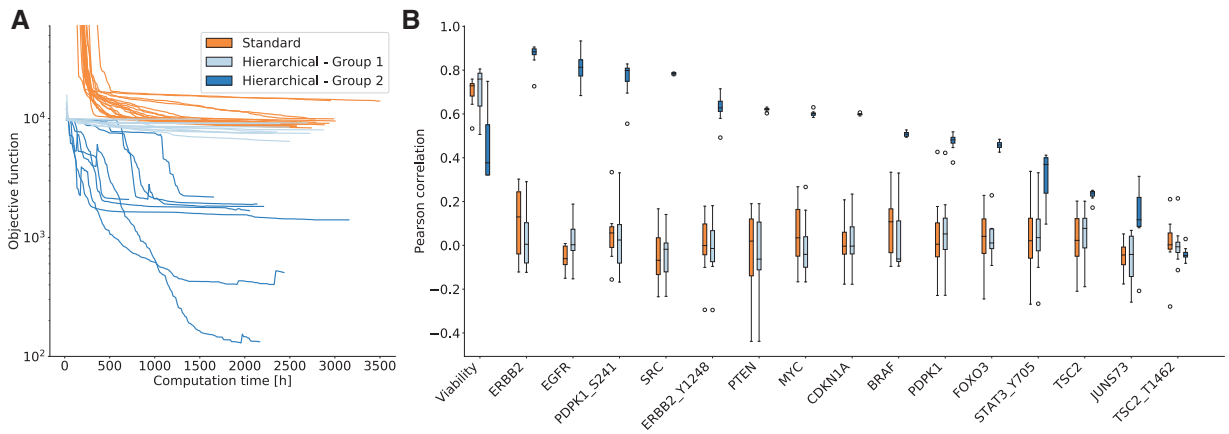

Fig. 4. Integration of heterogeneous data using hierarchical optimization. (**A**) Optimizer trajectories for standard and hierarchical optimization with *dataset 1 and 2* from Table 1 using Ipopt. The two groups found by the hierarchical optimization are indicated by different shades of blue. (**B**) Pearson correlations for all observables with at least 55 datapoints for all runs of the standard optimization and for the two groups found by the hierarchical optimization. For all observables, see Supplementary Figure S7. (Color version of this figure is available at *Bioinformatics* online.)

large-scale pan-cancer model and two published large-scale datasets. We obtained median speed-ups of more than one order of magnitude as compared to the conventional approach, irrespective of the employed optimizer. Given that the overall computation time is thousands of CPU hours, this improvement is substantial. Compared to simulating the ODE, the computation time needed to calculate the analytical formulas of the inner problem was five orders of magnitude faster, and therefore, negligible. While previous studies had already shown a reduced convergence rate when calibrating models to relative data (Degasperi *et al.*, 2017), we identified the large gradients with respect to the scalings as a possible explanation and established a flexible and easy way to circumvent them. The numerical stiffness which can arise from this for numerical optimization methods is the first conceptual explanation of the large improvements achieved by hierarchical methods (Loos *et al.*, 2018; Weber *et al.*, 2011).

In addition to the methodological contribution, we provide here the first proof-of-principle for the integration of multiple datasets using large-scale mechanistic models of cancer signaling. We showed for the example of viability and (phospho-)proteomic measurements that our optimization approach facilitates (i) data integration—where other methods failed—and (ii) an easy weighting of datasets. This is possible without computational overhead. The optimized noise parameters provide estimates for the measurement noise when no or only low numbers of replicates are available, as it is the case in many large-scale databases (e.g. CCLE and MCLP).

In this study, we used hierarchical optimization to estimate individual static parameters per observable. However, measurements may require multiple scaling and offset parameters per observable (e.g. the protein observables considered here), as well as arbitrary combinations thereof. In general, the problem of multiple scalings or offsets will always exist when multiple non-mutually inclusive normalizations need to be applied to model simulations, accounting for different experimental covariates like, e.g. output types, replication index, day-to-day variability or experimental devices. The current hierarchical framework cannot account for such settings. An extension to efficiently estimate all such parameters would thus presumably yield an even improved performance. Similarly, extending the optimization approach to other noise models would be of interest, even when the inner subproblem lacks an analytical solution. Of particular interest are distributions that are more robust to outliers, while still maintaining the good optimization convergence (Maier *et al.*, 2017).

Large-scale mechanistic models are of high value for systems biomedicine, since, as opposed to machine learning methods, they allow for mechanistic interpretation, analysis of latent variables and extrapolation to unseen conditions (Baker *et al.*, 2018; Fröhlich *et al.*, 2018). We consider this study to be a proof-of-concept for the integration of heterogeneous datasets into a mechanistic model and the efficient estimation of the unknown parameters. However, the here considered datasets are not sufficient to obtain high-quality estimates of the model parameters. Therefore, for future biology-driven analyses it will be valuable to include additional molecular measurements to improve the predictive power and the mechanistic interpretation of the model. With the advance of high-throughput technologies, more and more such large-scale datasets have been published. For example, the cancer proteomic atlas (Li *et al.*, 2013) or the datasets provided by Frejno *et al.* (2017) or Gholami *et al.* (2013) constitute rich sources of training data for future analyses. Our hierarchical optimization now allows for a much more efficient calibration of large-scale mechanistic models using heterogeneous datasets.

Funding

This work was supported by the German Research Foundation [grant no. HA7376/1-1 to Y.S.], the German Federal Ministry of Education and Research [SYS-Stomach; grant no. 01ZX1310B to J.H.] and the European Union's Horizon 2020 research and innovation program [CanPathPro; grant no. 686282 to F.F., J.H. and D.W.]. Computer resources for this project have been provided by the Gauss Centre for Supercomputing/Leibniz Supercomputing Centre under grant pr62li.

Author contributions

Y.S. and J.H. derived the theoretical foundation; D.W., F.F., L.S. and Y.S. wrote the implementations; D.W. and L.S. performed the case study. All authors discussed the results and conclusions and jointly wrote and approved the final manuscript.

Conflict of Interest: none declared.

References

Baker,R. *et al.* (2018) Mechanistic models versus machine learning, a fight worth fighting for the biological community? *Biol. Lett.*, **14**, 20170660.

Barretina,J. *et al.* (2012) The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, **483**, 603–607.

Bouhaddou,M. *et al.* (2018) A mechanistic pan-cancer pathway model informed by multi-omics data interprets stochastic cell fate responses to drugs and mitogens. *PLoS Comput. Biol.*, **14**, e1005985.

Boyd,S. and Vandenberghe,L. (2004). *Convex Optimisation*. Cambridge University Press, UK.

Degasperi,A. *et al.* (2017) Performance of objective functions and optimisation procedures for parameter estimation in system biology models. *NPJ Syst. Biol. Appl.*, **3**, 20.

Eduati,F. *et al.* (2017) Drug resistance mechanisms in colorectal cancer dissected with cell type-specific dynamic logic models. *Cancer Res.*, **77**, 3364–3375.

Fey,D. *et al.* (2015) Signaling pathway models as biomarkers: patient-specific simulations of JNK activity predict the survival of neuroblastoma patients. *Sci. Signal*, **8**, ra130.

Frejno,M. *et al.* (2017) Pharmacoproteomic characterisation of human colon and rectal cancer. *Mol. Syst. Biol.*, **13**, 951.

Fröhlich,F. *et al.* (2017a) Parameter estimation for dynamical systems with discrete events and logical operations. *Bioinformatics*, **33**, 1049–1056.

Fröhlich,F. *et al.* (2017b) Scalable parameter estimation for genome-scale biochemical reaction networks. *PLoS Comput. Biol.*, **13**, e1005331.

Fröhlich,F. *et al.* (2018) Efficient parameter estimation enables the prediction of drug response using a mechanistic pan-cancer pathway model. *Cell Syst.*, **7**, 567–579.e6.

Fujarewicz,K. *et al.* (2005) On fitting of mathematical models of cell signaling pathways using adjoint systems. *Math Biosci. Eng.*, **2**, 527–534.

Gay,D.M. (1983) Algorithm 611: subroutines for unconstrained minimization using a model/trust-region approach. *ACM Trans. Math. Softw.*, **9**, 503–524.

Gholami,A.M. *et al.* (2013) Global proteome analysis of the nci-60 cell line panel. *Cell Rep.*, **4**, 609–620.

Hass,H. *et al.* (2017) Predicting ligand-dependent tumors from multi-dimensional signaling features. *NPJ Syst. Biol. Appl.*, **3**, 27.

Hass,H. *et al.* (2019) Benchmark problems for dynamic modeling of intracellular processes. *Bioinformatics*, **35**, 3073–3082.

Hucka,M. *et al.* (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**, 524–531.

Klipp,E. *et al.* (2005) Integrative model of the response of yeast to osmotic shock. *Nat. Biotechnol.*, **23**, 975–982.

Kühn,A. and Lehrach,H. (2012) The virtual patient system: modeling cancer using deep sequencing technologies for personalized cancer treatment. *J. Verbrauch. Lebensm.*, **7**, 55–62.

Li,J. *et al.* (2013) TCPA: a resource for cancer functional proteomics data. *Nat. Methods*, **10**, 1046–1047.

Li,J. *et al*. (2017) Characterization of human cancer cell lines by reverse-phase protein arrays. *Cancer Cell*, **31**, 225–239.

Loos,C. *et al*. (2018) Hierarchical optimization for the efficient parametrization of ODE models. *Bioinformatics*, **34**, 4266–4273.

Lu,J. *et al*. (2013) Inverse problems from biomedicine: inference of putative disease mechanisms and robust therapeutic strategies. *J. Math. Biol.*, **67**, 143–168.

Maier,C. *et al*. (2017) Robust parameter estimation for dynamical systems from outlier-corrupted data. *Bioinformatics*, **33**, 718–725.

Ogilvie,L.A. *et al*. (2015) Predictive modeling of drug treatment in the area of personalized medicine. *Cancer Inform.*, **14**, 95–103.

Penas,D.R. *et al*. (2015) Parallel metaheuristics in computational biology: an asynchronous cooperative enhanced scatter search method. *Procedia Comput. Sci.*, **51**, 630–639.

Raue,A. *et al*. (2013) Lessons learned from quantitative dynamical modeling in systems biology. *PLoS ONE*, **8**, e74335.

Schälte,Y. *et al*. (2018) Evaluation of derivative-free optimizers for parameter estimation in systems biology. *IFAC PapersOnLine*, **51**, 98–101.

Villaverde,A.F. *et al*. (2018) Benchmarking optimization methods for parameter estimation in large kinetic models. *Bioinformatics*, **35**, 830–838.

Wächter,A. and Biegler,L.T. (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.*, **106**, 25–57.

Weber,P. *et al*. (2011) Parameter estimation and identifiability of biological networks using relative data. In: Bittanti,S. *et al*. (eds) *Proc. of the 18th IFAC World Congress*. Vol. **18**. Elsevier, Milano, Italy, pp. 11648–11653.