

Supplementary information - Efficient gradient-based parameter estimation for dynamic models using qualitative data

Leonard Schmiester^{1,2}, Daniel Weindl¹, and Jan Hasenauer^{1,2,3,*}

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

³ Faculty of Mathematics and Natural Sciences, University of Bonn, 53113 Bonn, Germany

* To whom correspondence should be addressed (jan.hasenauer@uni-bonn.de)

List of Figures

S1	Computation times for an objective function evaluation for 50 randomly sampled parameter vectors for all five application examples	8
S2	Number of converged starts per hour for different convergence thresholds . .	9
S3	Waterfall plots for all models using gradient-free and gradient-based optimization	10
S4	Model simulation and optimal surrogate data for the best found parameters for model M4	11
S5	Gradient landscape for the toy model T1.	12
S6	Parameter estimation results for model M2 with 16 and 8 categories	12

1 Derivation of gradient formulas

Here, we derive the algorithm for calculating the gradient of the optimal scaling objective function in detail for the case of one observable. First, we briefly recapitulate the necessary notation and prerequisites. We consider the general case of parameter and simulation dependent weights $W(\theta, \xi(\theta))$ and minimal interval and gap sizes which are collected in the vector

$d(\theta, \xi(\theta))$. The parameters of the inner optimization problem, \tilde{y} , l and u , are collected in the parameter vector $\tilde{\xi} = (\tilde{y}, l, u)^T$. The simulations are collected in $\xi(\theta) = (y(\theta), 0, 0)^T$. The optimization problem is given by

$$\min_{\theta} J(\theta, \xi(\theta), \tilde{\xi}^*(\theta)) \quad (1)$$

$$\begin{aligned} \text{s.t. } \tilde{\xi}^*(\theta) &= \arg \min_{\tilde{\xi}} J(\theta, \xi(\theta), \tilde{\xi}) \\ \text{s.t. } C\tilde{\xi} + d(\theta, \xi(\theta)) &\leq 0, \end{aligned} \quad (2)$$

with the objective function

$$J(\theta, \xi(\theta), \tilde{\xi}) = \left(\tilde{\xi} - \xi(\theta) \right)^T W(\theta, \xi(\theta)) \left(\tilde{\xi} - \xi(\theta) \right). \quad (3)$$

$\tilde{\xi}^*(\theta)$ are the optimal surrogate data and interval bounds, which solve the problem (2). Note that while $\tilde{\xi}^*(\theta)$ depends directly on the parameters θ , for $\tilde{\xi}$ this depends on whether the inner or outer problem is considered. In the inner problem $\tilde{\xi}$ does not depend on θ , yet, the optimal solution $\tilde{\xi}^*(\theta)$ of the inner problem (which is used in the outer problem) depends on θ . We are now interested in calculating the derivatives of J w.r.t. the outer parameters θ evaluated at $\tilde{\xi}^*(\theta)$. This can be calculated by

$$\frac{dJ}{d\theta} \Big|_{\theta, \xi(\theta), \tilde{\xi}^*(\theta)} = \frac{\partial J}{\partial \theta} \Big|_{\theta, \xi(\theta), \tilde{\xi}^*(\theta)} + \frac{\partial J}{\partial \tilde{\xi}} \Big|_{\theta, \xi(\theta), \tilde{\xi}^*(\theta)} \frac{\partial \xi(\theta)}{\partial \theta} + \frac{\partial J}{\partial \tilde{\xi}} \Big|_{\theta, \xi(\theta), \tilde{\xi}^*(\theta)} \frac{\partial \tilde{\xi}^*(\theta)}{\partial \theta}. \quad (4)$$

The partial derivatives of J are

$$\frac{\partial J}{\partial \theta} \Big|_{\theta, \xi(\theta), \tilde{\xi}^*(\theta)} = (\tilde{\xi}^*(\theta) - \xi(\theta))^T \frac{\partial W}{\partial \theta_j} \Big|_{\theta, \xi(\theta)} (\tilde{\xi}^*(\theta) - \xi(\theta)), \quad (5)$$

$$\frac{\partial J}{\partial \xi} \Big|_{\theta, \xi(\theta), \tilde{\xi}^*(\theta)} = -2(\tilde{\xi}^*(\theta) - \xi(\theta))^T W(\theta, \xi(\theta)) + (\tilde{\xi}^*(\theta) - \xi(\theta))^T \frac{\partial W}{\partial \xi} \Big|_{\theta, \xi(\theta)} (\tilde{\xi}^*(\theta) - \xi(\theta)), \quad (6)$$

$$\frac{\partial J}{\partial \tilde{\xi}} \Big|_{\theta, \xi(\theta), \tilde{\xi}^*(\theta)} = 2(\tilde{\xi}^*(\theta) - \xi(\theta))^T W(\theta, \xi(\theta)). \quad (7)$$

The evaluation of these partial derivatives requires the optimal surrogate data $\tilde{\xi}^*(\theta)$ and simulated observable $\xi(\theta)$. Depending on the structure of W , also the sensitivities of the simulated observable $\frac{\partial \xi(\theta)}{\partial \theta}$ is required. These components are accessible as solutions of an optimization problem, an ODE, or a forward sensitivity equation.

For the evaluation of the objective function gradient, we additionally need to calculate the derivative of the optimal surrogate data $\tilde{\xi}^*$ w.r.t. the parameters θ . This is the derivative of an optimal solution. To assess it, we follow the ideas from Fiacco (1976) and calculate the derivatives of the first order optimality conditions of the inner optimization problem (2) w.r.t. θ . With the Lagrangian function

$$\mathcal{L}(\tilde{\xi}, \mu) = J(\theta, \xi(\theta), \tilde{\xi}) + \mu^T (C\tilde{\xi} + d(\theta, \xi(\theta))), \quad (8)$$

and Lagrange multipliers $\mu \in \mathbb{R}^{n_c}$, the necessary first order optimality conditions of problem (2) are

$$\nabla_{\tilde{\xi}} \mathcal{L}(\tilde{\xi}^*(\theta), \mu) = 2(\tilde{\xi}^*(\theta) - \xi(\theta))^T W(\theta, \xi(\theta)) + \mu(\theta)^T C = 0 \quad (9)$$

$$\mu_i(\theta)(C_i \tilde{\xi}^*(\theta) + d_i(\theta, \xi(\theta))) = 0, \quad \text{for } i = 1, \dots, n_c \quad (10)$$

$$C_i \tilde{\xi}^*(\theta) + d_i(\theta, \xi(\theta)) \leq 0, \quad \text{for } i = 1, \dots, n_c \quad (11)$$

$$\mu_i(\theta) \geq 0, \quad \text{for } i = 1, \dots, n_c. \quad (12)$$

For a convex objective function, which we have in this case (Schmiester *et al.*, 2020), these conditions are necessary and sufficient for an optimum. To obtain the desired derivatives $\frac{\partial \tilde{\xi}^*}{\partial \theta}$, we calculate the derivatives of equations (9) and (10) w.r.t. θ_j :

$$2 \left(\frac{\partial \tilde{\xi}^*(\theta)}{\partial \theta_j} - \frac{\partial \xi(\theta)}{\partial \theta_j} \right)^T W(\theta, \xi(\theta)) + 2 \left(\tilde{\xi}^*(\theta) - \xi(\theta) \right)^T \left(\frac{\partial W(\theta, \xi(\theta))}{\partial \theta_j} + \frac{\partial W(\theta, \xi(\theta))}{\partial \xi} \frac{\partial \xi}{\partial \theta_j} \right) + \frac{\partial \mu(\theta)^T}{\partial \theta_j} C = 0 \quad (13)$$

$$\frac{\partial \mu_i(\theta)}{\partial \theta_j} (C_i \tilde{\xi}^*(\theta) + d_i(\theta, \xi(\theta))) + \mu_i(\theta) \left(C_i \frac{\partial \tilde{\xi}^*(\theta)}{\partial \theta_j} - \left(\frac{\partial d(\theta, \xi(\theta))}{\partial \theta_j} + \frac{\partial d(\theta, \xi(\theta))}{\partial \xi} \frac{\partial \xi(\theta)}{\partial \theta_j} \right) \right) = 0. \quad (14)$$

This yields the following linear system of equations that can be solved for $\frac{\partial \tilde{\xi}^*(\theta)}{\partial \theta}$ and $\frac{\partial \mu(\theta)}{\partial \theta}$ for each parameter θ_j :

$$\begin{pmatrix} 2W(\theta, \xi(\theta)) & C^T \\ \text{diag}(\mu(\theta))C & \text{diag}(C\tilde{\xi}^*(\theta) + d(\theta, \xi(\theta))) \end{pmatrix} \begin{pmatrix} \frac{\partial \tilde{\xi}^*(\theta)}{\partial \theta_j} \\ \frac{\partial \mu(\theta)}{\partial \theta_j} \end{pmatrix} = \begin{pmatrix} 2W(\theta, \xi(\theta)) \frac{\partial \xi(\theta)}{\partial \theta_j} - 2 \left(\frac{\partial W(\theta, \xi(\theta))}{\partial \theta_j} + \frac{\partial W(\theta, \xi(\theta))}{\partial \xi} \frac{\partial \xi}{\partial \theta_j} \right) (\tilde{\xi}^*(\theta) - \xi(\theta)) \\ -\text{diag}(\mu(\theta)) \left(\frac{\partial d(\theta, \xi(\theta))}{\partial \theta_j} + \frac{\partial d(\theta, \xi(\theta))}{\partial \xi} \frac{\partial \xi(\theta)}{\partial \theta_j} \right) \end{pmatrix}. \quad (15)$$

After solving this linear system, we can calculate the gradients of the objective function J w.r.t. the model parameters θ .

Special case of constant W and $d = 0$: The gradients for the special case of a weight matrix which is independent of the parameters and the model observables, and zero minimal interval and gap sizes, i.e. $d = 0$ and $W(\theta, \xi) = W$, can be easily derived from the general result stated above. The linear system determining the derivative of the optimal solution is given by

$$\begin{pmatrix} 2W & C^T \\ \text{diag}(\mu(\theta))C & \text{diag}(C\tilde{\xi}^*(\theta)) \end{pmatrix} \begin{pmatrix} \frac{\partial \tilde{\xi}^*(\theta)}{\partial \theta_j} \\ \frac{\partial \mu}{\partial \theta_j} \end{pmatrix} = \begin{pmatrix} 2W \frac{\partial \xi(\theta)}{\partial \theta_j} \\ 0 \end{pmatrix}. \quad (16)$$

Additionally, $\frac{\partial J}{\partial \theta} \Big|_{\theta, \xi(\theta), \tilde{\xi}^*(\theta)}$ vanishes and $\frac{\partial J}{\partial \xi} \Big|_{\theta, \xi(\theta), \tilde{\xi}^*(\theta)}$ reduces to

$$\frac{\partial J}{\partial \xi} \Big|_{\theta, \xi(\theta), \tilde{\xi}^*(\theta)} = -2(\tilde{\xi}^*(\theta) - \xi(\theta))^T W. \quad (17)$$

2 Optimal scaling approach for multiple observables

In this section, we extend the case of a single observable presented in the main manuscript to the more general case of n_y observables, i.e. $y(t, \theta) \in \mathbb{R}^{n_y}$. Quantitative data is then linked to observables via

$$\bar{y}_{m,i} = y_m(t_i, \theta) + \varepsilon_{m,i}, \quad m = 1, \dots, n_y, \quad i = 1, \dots, n_t. \quad (18)$$

For qualitative data, we assume that the ordering of datapoints within one observable is known, but no information on the relation of datapoints across different observables is available. We denote this by introducing a group of n_{k^m} categories for each observable y_m , i.e. \mathcal{C}_{k^m} , $k^m = 1, \dots, n_{k^m}$. The respective intervals for the categories are denoted by $[l_{k^m}, u_{k^m}]$. We assume that the categories are ordered as $\mathcal{C}_1 \prec \dots, \mathcal{C}_{n_{k^m}}$ and that no relation between categories $\mathcal{C}_{k^m}, \mathcal{C}_{k^{m'}}$ are known for $m' \neq m$. The optimal scaling approach then consists of n_y inner subproblems, which need to be solved:

$$\begin{aligned} \min_{\tilde{\xi}_m} & \left\{ J_m := \left(\tilde{\xi}_m - \xi_m(\theta) \right)^T W_m \left(\tilde{\xi}_m - \xi_m(\theta) \right) \right\} \\ \text{s.t. } & C_m \tilde{\xi}_m \leq 0 \end{aligned} \quad (19)$$

for $m = 1, \dots, n_y$. Here, $\tilde{\xi}_m$ is the vector of the surrogate data \tilde{y}_m and category bounds l_m, u_m , and $\xi_m(\theta)$ the vector of simulations $y_m(t, \theta)$, belonging to the observable with index m . C_m contains the constraints of the categories of this observable. The overall objective function value can be calculated by summing over the n_y values obtained from solving (19). Therefore, the optimization problem is given by

$$\min_{\theta} \sum_{m=1}^{n_y} J_m(\theta, \tilde{\xi}_m^*(\theta)) \quad (20)$$

$$\text{s.t. } \begin{cases} \tilde{\xi}_m^*(\theta) = \arg \min_{\tilde{\xi}_m} J_m(\theta, \tilde{\xi}_m) \\ \text{s.t. } C_m \tilde{\xi}_m \leq 0 \end{cases} \quad \forall m = 1, \dots, n_y \quad (21)$$

Similarly, the gradient can be calculated via

$$\sum_{m=1}^{n_y} \frac{dJ_m}{d\theta} \Big|_{\theta, \xi_m(\theta), \tilde{\xi}_m(\theta)} \quad (22)$$

with

$$\left. \frac{dJ_m}{d\theta} \right|_{\theta, \xi_m(\theta), \tilde{\xi}_m(\theta)} = \left. \frac{\partial J_m}{\partial \theta} \right|_{\theta, \xi_m(\theta), \tilde{\xi}_m^*(\theta)} + \left. \frac{\partial J_m}{\partial \xi_m} \right|_{\theta, \xi_m(\theta), \tilde{\xi}_m^*(\theta)} \frac{\partial \xi_m(\theta)}{\partial \theta} + \left. \frac{\partial J_m}{\partial \tilde{\xi}_m} \right|_{\theta, \xi_m(\theta), \tilde{\xi}_m^*(\theta)} \frac{\partial \tilde{\xi}_m^*(\theta)}{\partial \theta} \quad (23)$$

3 Reformulation example

To illustrate the reformulation of the optimal scaling problem in matrix-vector notation, we consider a minimal example of two readouts from two categories $z_1 \in \mathcal{C}_1, z_2 \in \mathcal{C}_2$ at timepoints t_1, t_2 , with $\mathcal{C}_1 \prec \mathcal{C}_2$. The inner optimization problem of the optimal scaling method for this example is

$$\min_{\tilde{y}_1, \tilde{y}_2, l_1, l_2, u_1, u_2} w_1(\tilde{y}_1 - y(t_1, \theta))^2 + w_2(\tilde{y}_2 - y(t_2, \theta))^2 \quad (24)$$

$$\text{s.t. } l_1 \leq \tilde{y}_1 \leq u_1 \quad (25)$$

$$l_2 \leq \tilde{y}_2 \leq u_2 \quad (26)$$

$$u_1 \leq l_2. \quad (27)$$

We can reformulate the objective function in the form (3) using

$$\tilde{\xi} = \begin{pmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ l_1 \\ l_2 \\ u_1 \\ u_2 \end{pmatrix}, \xi(\theta) = \begin{pmatrix} y(t_1, \theta) \\ y(t_2, \theta) \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, W = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & & \\ \vdots & & 0 & \\ & & & \ddots \\ 0 & & & & 0 \end{pmatrix} \in \mathbb{R}^{6 \times 6}. \quad (28)$$

To obtain the matrix of constraints C , we first rewrite the constraints to

$$l_1 - \tilde{y}_1 \leq 0 \quad (29)$$

$$\tilde{y}_1 - u_1 \leq 0 \quad (30)$$

$$l_2 - \tilde{y}_2 \leq 0 \quad (31)$$

$$\tilde{y}_2 - u_2 \leq 0 \quad (32)$$

$$u_1 - l_2 \leq 0, \quad (33)$$

which is equivalent to

$$\tilde{\xi}_3 - \tilde{\xi}_1 \leq 0 \quad (34)$$

$$\tilde{\xi}_1 - \tilde{\xi}_5 \leq 0 \quad (35)$$

$$\tilde{\xi}_4 - \tilde{\xi}_2 \leq 0 \quad (36)$$

$$\tilde{\xi}_2 - \tilde{\xi}_6 \leq 0 \quad (37)$$

$$\tilde{\xi}_5 - \tilde{\xi}_4 \leq 0. \quad (38)$$

These inequalities can be formulated to $C\tilde{\xi} \leq 0$ with

$$C = \begin{pmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 1 & 0 \end{pmatrix}. \quad (39)$$

4 Implementation

We implemented the optimal scaling objective function into the pyPESTO framework. This function requires a parameter vector as input and returns the value and (optionally) the gradient of the objective function. This implementation allows to use the optimal scaling approach also within other Python-based toolboxes, such as Tellurium (Choi *et al.*, 2018) or pySB (Lopez *et al.*, 2013). We provide a Jupyter notebook illustrating how to use the pyPESTO implementation in the GitHub repository and as Supplementary Information to this manuscript.

Parameter estimations were run on a Intel(R) Xeon(R) Gold 6126 @ 2.60GHz processor with 384 GB RAM. For all models and methods, each of the local optimizations were run on a single core with a wall-time limit of 2h for model M4 and 1h for the other models. Profiles were run on a Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz with 4 GB RAM. The gradient-based optimization was performed using the L-BFGS-B algorithm from SciPy (Jones *et al.*, 2001) with default options except for `fatol` = 10^{-8} and `gtol` = 10^{-8} . The gradient-free optimization was run with SciPy's Powell algorithm with default options except for `fatol` = 10^{-8} .

For profile calculation, we used the implemented routines in pyPESTO. We adapted some hyperparameters, namely `min_step_size` = 0.0001, `delta_ratio_max` = 0.00005, `default_step_size` = 0.0005, `reg_order` = 4 and `reg_points` = 50. Additionally, we adapted the optimizer tolerances to `fatol` = 10^{-10} and `gtol` = 10^{-10} .

For the inner optimization, we used the reparameterized and reduced formulation proposed by Schmiester *et al.* (2020). As optimizer, we chose SciPy's L-BFGS-B with `maxiter` = 2000

and $\text{ftol} = 10^{-10}$. The linear system of equations for gradient computation was solved using the sparse solver `spsolve` from `scipy.sparse.linalg`.

As weights, we used

$$w = \sum_i y(t_i, \theta) + \varepsilon, \quad \varepsilon = 10^{-8}. \quad (40)$$

As minimal values for the interval and gap sizes, we used

$$\begin{aligned} g &= \frac{\max(y(\theta))}{4(n_k - 1) + 1} + \gamma \\ s &= \frac{\max(y(\theta))}{2n_k + 1} \end{aligned} \quad (41)$$

with $\gamma = 10^{-1}$ for models M2 and M4, and $\gamma = 10^{-2}$ for models M1, M3 and M5.

5 Qualitative data generation

The models M1–M5 were taken from a collection of benchmark problems for parameter estimation Hass *et al.* (2019). Specifically, we employed the reimplementation in the PETab format (Schmiester *et al.*, 2021).

For all models quantitative data are available. These quantitative data are used to construct the qualitative data. For this, we used the ordering of the quantitative values. Datapoints with an equal quantitative value were assumed to belong to the same category. Otherwise the datapoints were assigned to different categories. This lead to 1–4 datapoints per category.

The construction of the categories affects the computational resources required to solve the inner optimization problem of the optimal scaling approach. The dimensionality of the inner problem increases linearly with the number of categories. Additionally, also the dimension of the linear system to calculate the gradient scales with the number of categories. In our previous work, we observed an approximately linear scaling of computation time with respect to the number of categories, when evaluating the objective function without gradients (Schmiester *et al.*, 2020). To analyse the influence of categorization further, we performed parameter estimation for the model M2 with 8 instead of 16 categories per observable (Figure S6). This revealed that the computation time per local optimization decreased for the problem with less categories, while the number of function evaluations stayed almost constant.

References

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

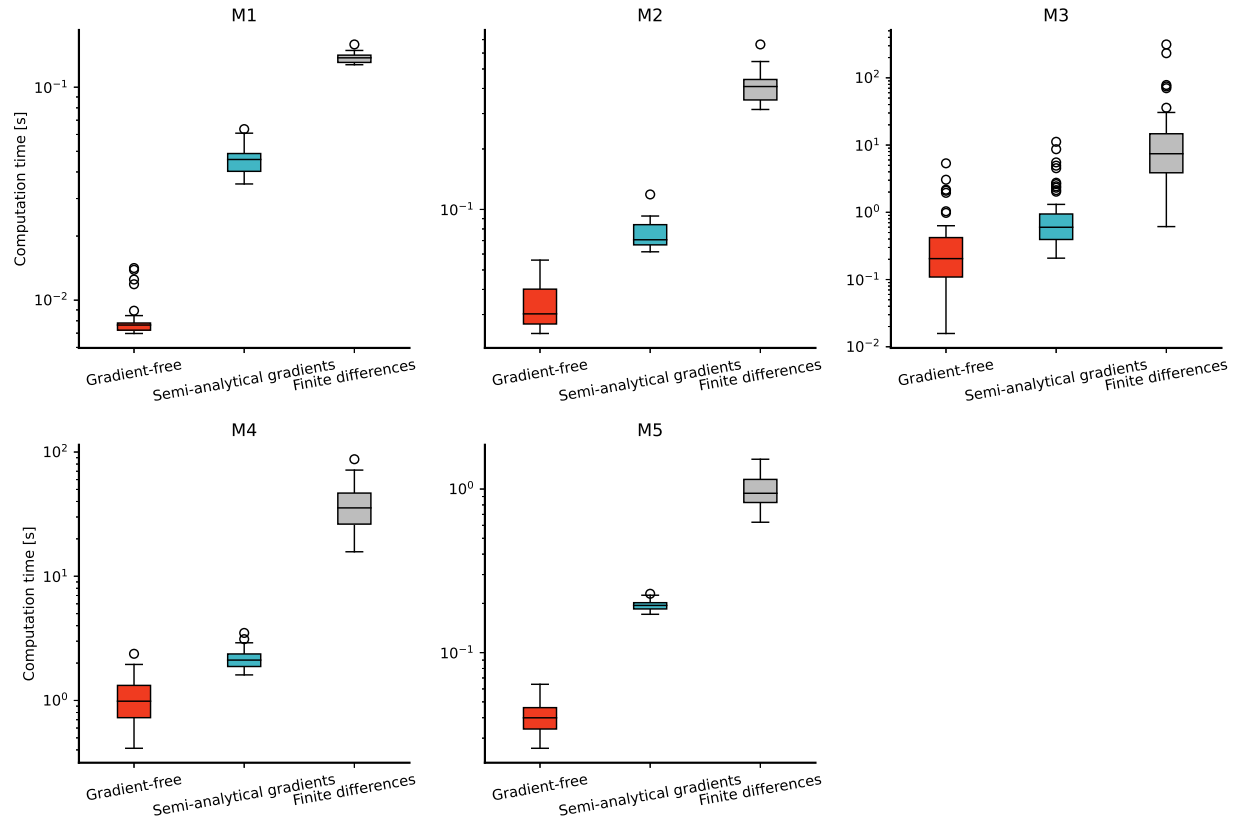


Figure S1: Computation times for an objective function evaluation for 50 randomly sampled parameter vectors for all five application examples without gradients, and with gradients using the semi-analytical approach and central finite differences.

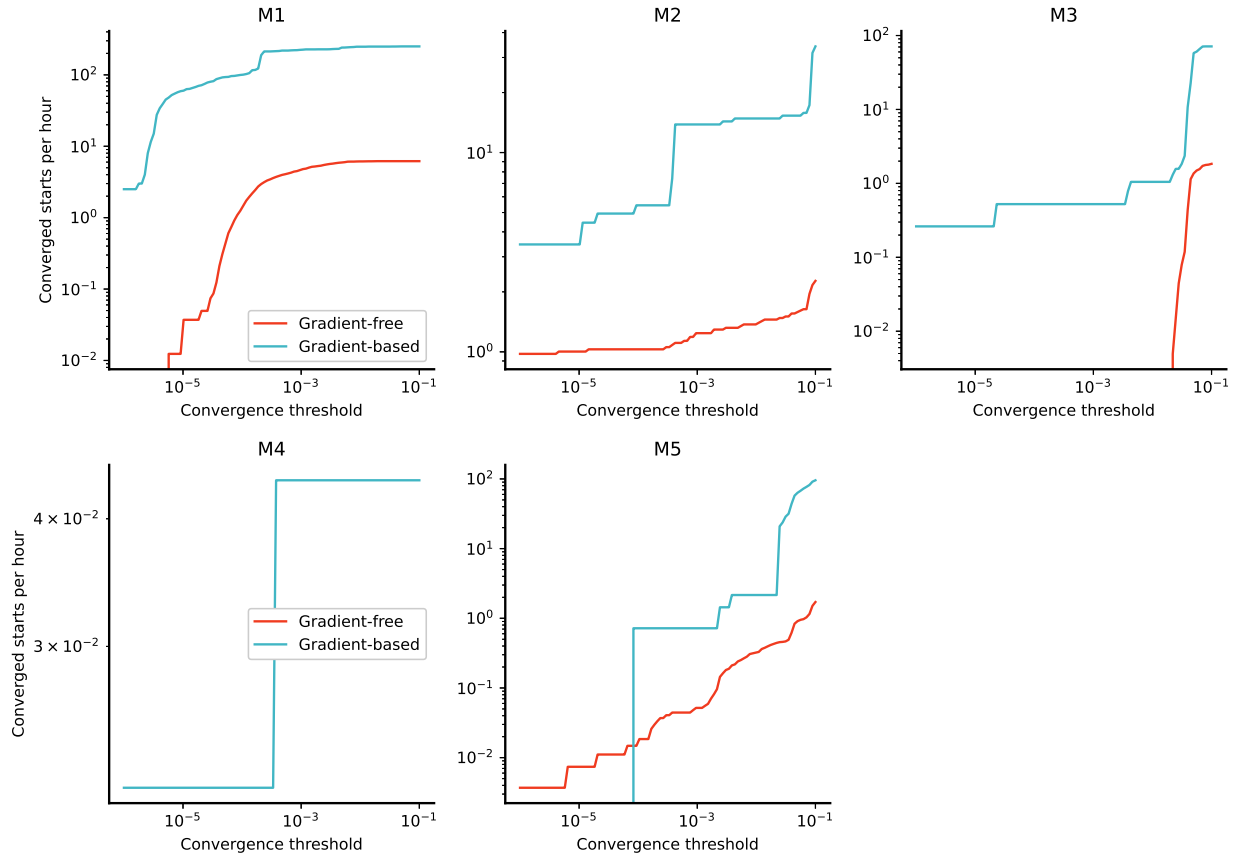


Figure S2: Number of converged starts per hour for different convergence thresholds, i.e. maximal absolute differences to the overall best objective function value.

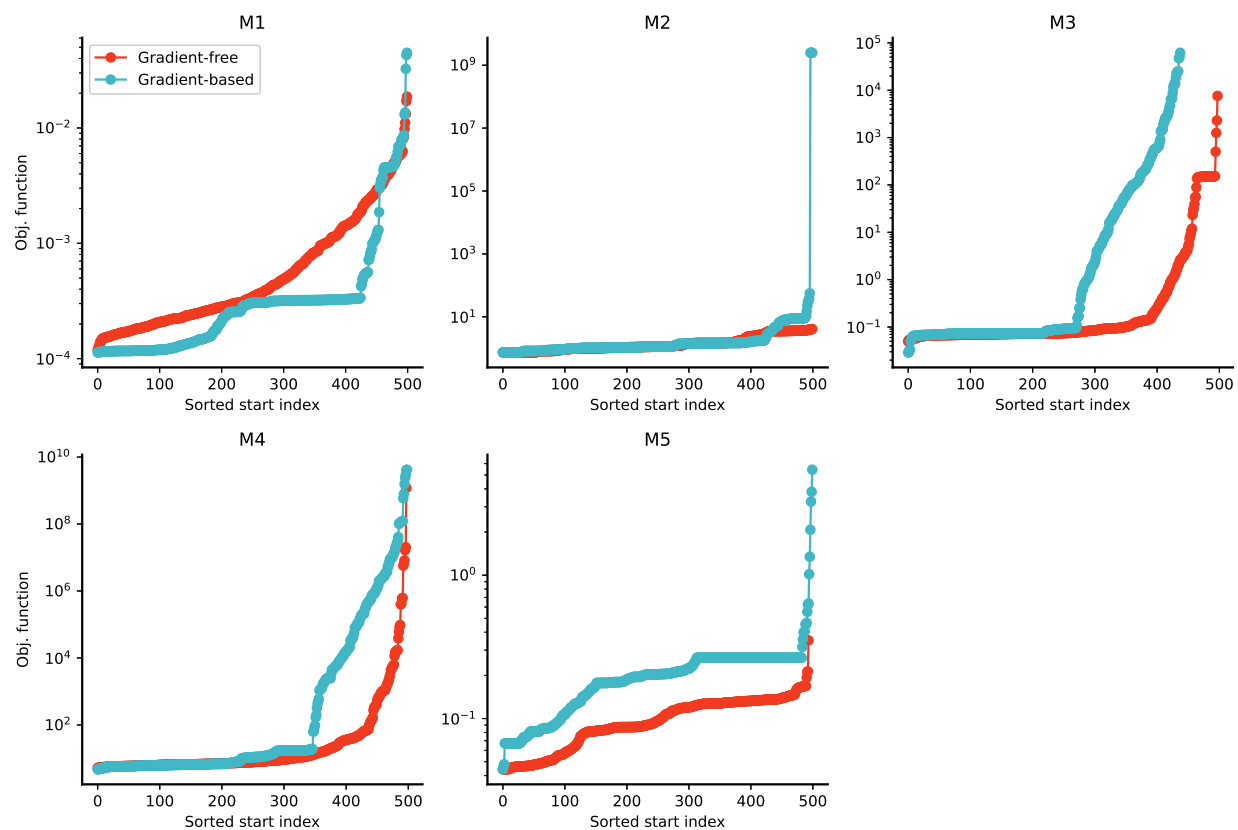
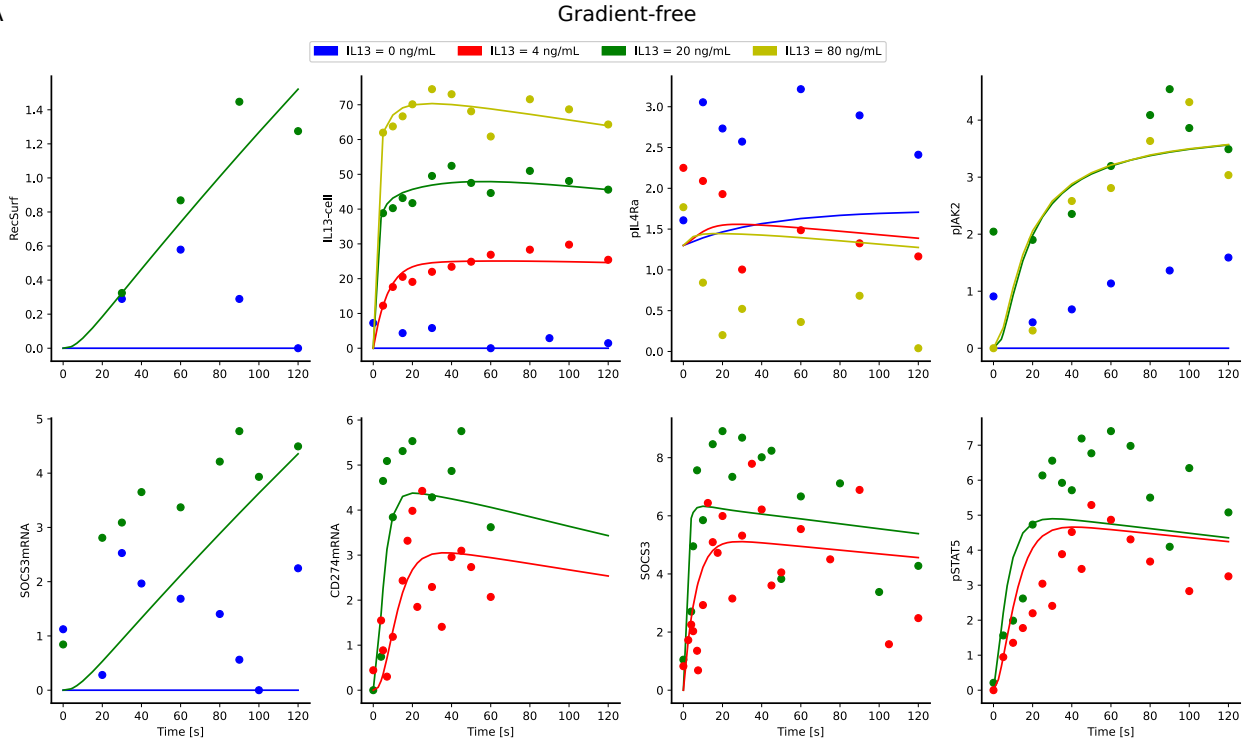


Figure S3: Waterfall plots for all models using gradient-free and gradient-based optimization. All 500 starts are shown.

A



B

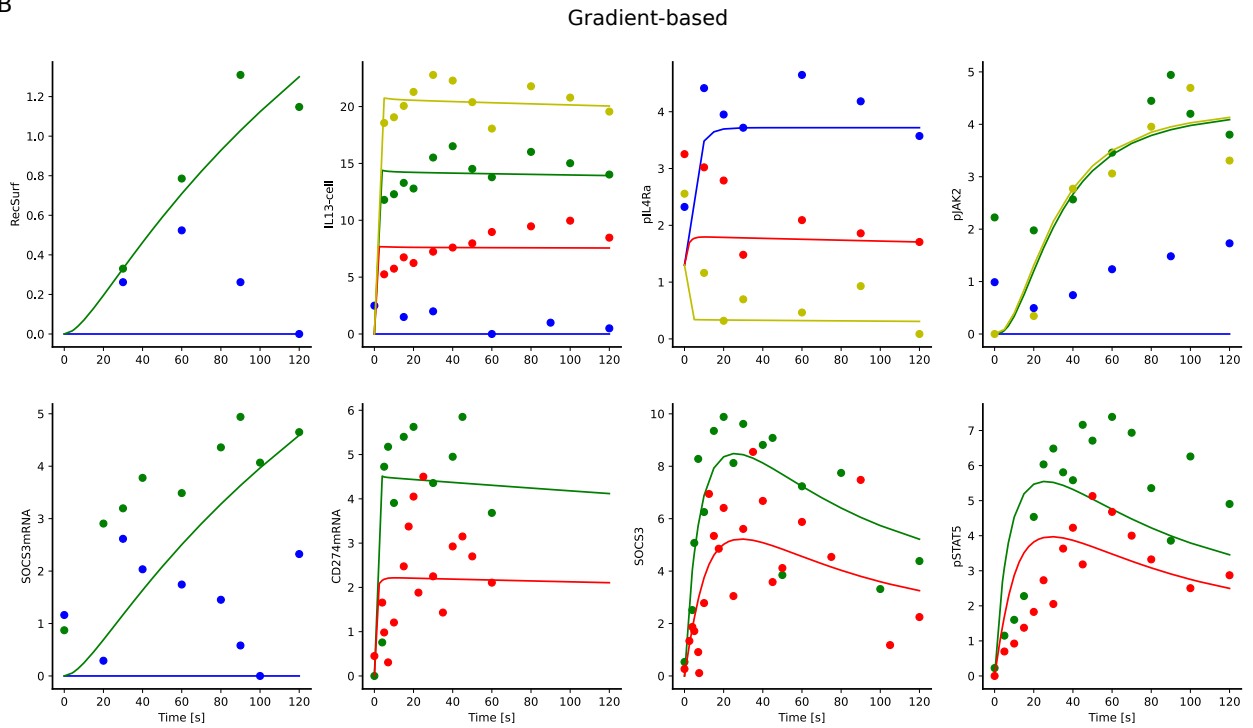


Figure S4: Model simulation and optimal surrogate data for the best found parameters of the gradient-free (A) and gradient-based (B) optimization for model M4. For this model all datapoints were distinguishable, leading to one datapoint per category.

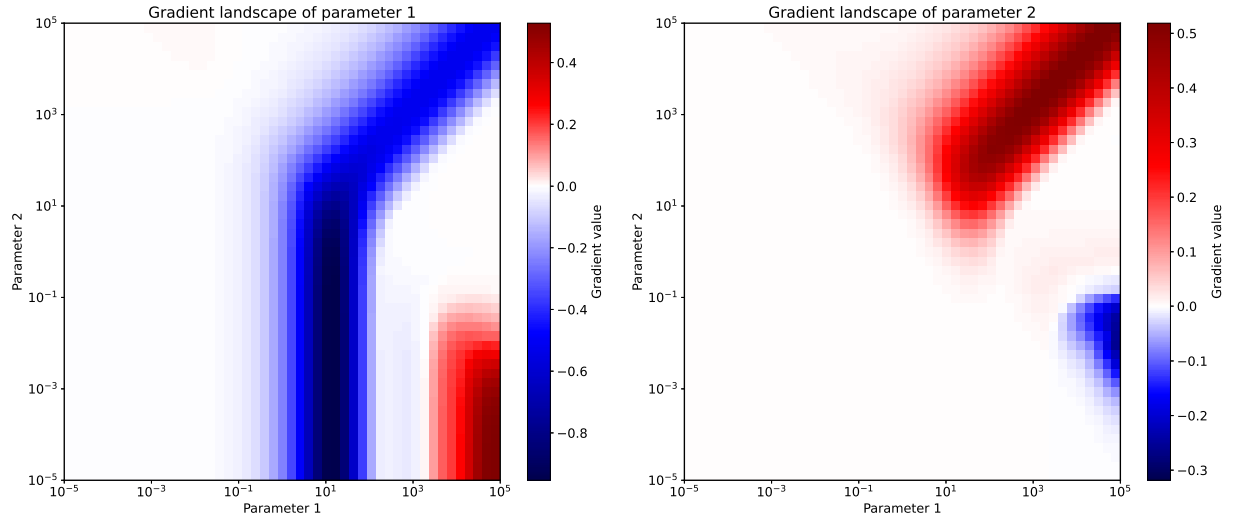


Figure S5: Landscapes of the gradients for parameter 1 and 2 for the model T1. The gradient is evaluated on a 50×50 grid using the semi-analytical algorithm.

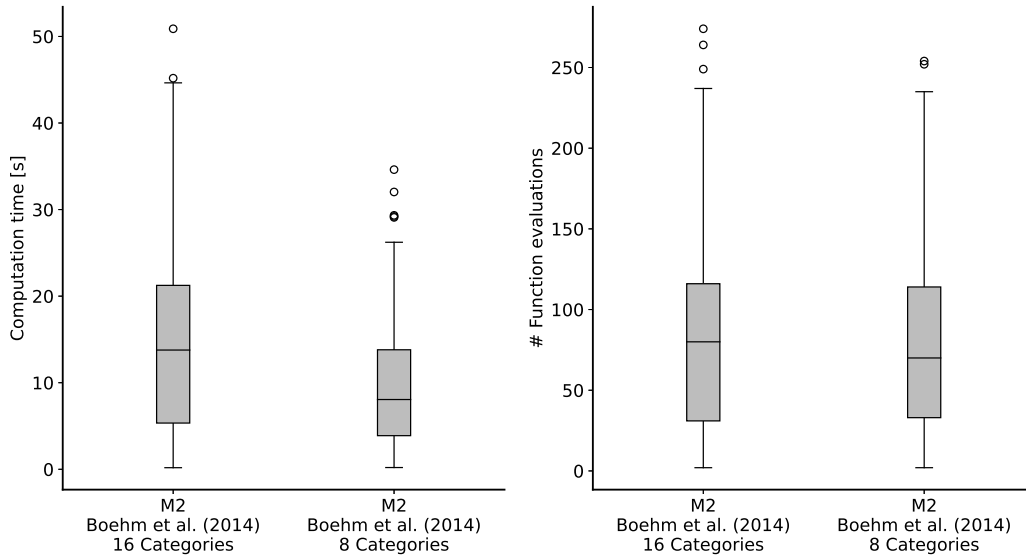


Figure S6: Parameter estimation results for the model M2 with datapoints discretized into 16 and 8 categories per observable. Boxplots show computation times and number of function evaluations per local optimization.

- Schmiester, L., Weindl, D., & Hasenauer, J. (2020). Parameterization of mechanistic models from qualitative data using an efficient optimal scaling approach. *J. Math. Biol.*, 81(2), 603–623.
- Choi, K., Medley, J. K., König, M., Stocking, K., Smith, L., Gu, S., & Sauro, H. M. (2018). Tellurium: An extensible python-based modeling environment for systems and synthetic biology. *Bio Systems*, 171, 74–79.
- Lopez, C. F., Muhlich, J. L., Bachman, J. A., & Sorger, P. K. (2013). Programming biological models in python using pysb. *Mol. Syst. Biol.*, 9, 646.
- Jones, E., Oliphant, T., Peterson, P., *et al.* (2001). SciPy: Open source scientific tools for Python.
- Hass, H., Loos, C., Raimúndez-Álvarez, E., Timmer, J., Hasenauer, J., & Kreutz, C. (2019). Benchmark problems for dynamic modeling of intracellular processes. *Bioinformatics*, 35(17), 3073–3082.
- Schmiester, L., Schälte, Y., Bergmann, F. T., 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. L., Städter, P., Wang, D., Wieland, F.-G., Banga, J. R., Timmer, J., Villaverde, A. F., Sahle, S., Kreutz, C., Hasenauer, J., & Weindl, D. (2021). PETab—interoperable specification of parameter estimation problems in systems biology. *PLOS Computational Biology*, 17(1), 1–10.