A protocol for dynamic model calibration

Alejandro F. Villaverde, Dilan Pathirana, Fabian Fröhlich, Jan Hasenauer and Julio R. Banga

Abstract

Ordinary differential equation models are nowadays widely used for the mechanistic description of biological processes and their temporal evolution. These models typically have many unknown and nonmeasurable parameters, which have to be determined by fitting the model to experimental data. In order to perform this task, known as parameter estimation or model calibration, the modeller faces challenges such as poor parameter identifiability, lack of sufficiently informative experimental data and the existence of local minima in the objective function landscape. These issues tend to worsen with larger model sizes, increasing the computational complexity and the number of unknown parameters. An incorrectly calibrated model is problematic because it may result in inaccurate predictions and misleading conclusions. For nonexpert users, there are a large number of potential pitfalls. Here, we provide a protocol that guides the user through all the steps involved in the calibration of dynamic models. We illustrate the methodology with two models and provide all the code required to reproduce the results and perform the same analysis on new models. Our protocol provides practitioners and researchers in biological modelling with a one-stop guide that is at the same time compact and sufficiently comprehensive to cover all aspects of the problem.

Key words: systems biology; dynamic modelling; parameter estimation; identification; identifiability; optimization

Introduction

The use of dynamic models has become common practice in the life sciences. Mathematical modelling provides a rigorous, compact way of encapsulating the available knowledge about a biological process. Perhaps more importantly, it is also a tool for understanding, analysing and predicting the behaviour of a complex system under conditions for which no experimental data are available. To these ends, it is particularly important that the model has been developed with that specific purpose in mind.

In bio-medicine, dynamic models are used for basic research as well as for medical applications. On one hand, dynamic models facilitate an understanding of biological processes, e.g., by identifying from a list of alternative mechanisms the most plausible one [1]. On the other hand, dynamic models with sufficient mechanistic detail can be used to make predictions, including the selection of drug targets [2], and the outcome of individual and combination treatments [3, 4]. In bio- and process engineering, dynamic models are used to design and optimize biotechnological processes. Here, models are, for instance, used...
to find the genetic and regulatory modifications that enhance the production of a target metabolite while enforcing constraints on certain metabolite levels [5–8]. In synthetic biology, dynamic models guide the design of artificial biological circuits where fine-tuned expression levels are necessary to ensure the correct functioning of regulatory elements [9–12]. Beyond these topics, there is a broad spectrum of additional research areas.

The choice of model type and complexity depends on which biological question(s) the model will be used to answer. Once this has been decided, the relevant biological knowledge is collected, e.g. from databases such as KEGG [13], STRING [14] and REACTOME [15], or from the literature. Furthermore, already available models can be used, e.g. from JWS Online [16] or Biomodels [17], and information about kinetic parameters can be extracted, e.g. from BRENDA [18] or Sabio-RK [19]. This information is then used to determine the biological species and biochemical reactions that are relevant to the process. In combination with assumptions about reaction kinetics – e.g. mass action or Michaelis-Menten—these elements allow the construction of a tailored mathematical model, which will usually have nonlinear dynamics and uncertainties associated to its structure and parameter values [20]. The model can be specified in a standard format such as SBML, to take advantage of the ecosystem of tools that already support a standard format [21].

The advent of high-throughput experimental techniques and the ever-growing availability of computational resources have led to the development of increasingly larger models. Common models possess tens of state variables and tens to a few hundreds of parameters [22, 23]. Large models can even possess thousands of state variables and parameters [3]. Dynamic models need to be calibrated, i.e. their unknown parameters have to be estimated from experimental data. In model calibration, the mismatch between simulated model output and experimental data is minimized to find the best parameter values [24–28]. Model calibration may be seen as part of a more general problem sometimes called reverse engineering [29] or (nonlinear) systems identification [30]. It is a process composed of a sequence of steps, which usually need to be iterated [31] until a satisfactory result is found. The definition of “satisfactory” depends on the ultimate goal of the model calibration procedure: it may focus on obtaining the most accurate parameter estimates or the most accurate predictions. While related, these two applications may lead to different outcomes, namely in regard to experimental design.

In this work, we consider the calibration of ordinary differential equation (ODE) models. ODE models are widely used to describe biological processes, and their calibration has been discussed in protocols for different classes of processes, including gene regulatory circuits [32], signalling networks [26], biocatalytic reactions [33], wastewater treatment [34, 35], food processing [36], biomolecular systems [37], and cardiac electrophysiology models [38]. Yet, these protocols focus on individual aspects of the calibration process (relevant for the subdiscipline) and/or lack illustration examples and codes that can be reused. The papers [34] and [35] focus on parameter subset selection via sensitivity and correlation analysis and on subsequent model optimization. The works of [32], [36] and [33] consider only low-dimensional models and do not provide in-depth discussion of scalability. The paper [26] neither covers structural identifiability (SI) analysis nor experimental design and describes a prediction uncertainty approach with limited applicability. The works of [33], [37], [38] discuss most aspects of the calibration process, but do not provide a step-by-step illustration with an example model and codes. The work of [39] is tailored to users of the MATLAB software toolbox Data2Dynamics [40].

The protocol presented here aims to provide a comprehensive description of the steps of the calibration process, which integrates recent advances. An outline of the procedure is depicted in Figure 1. The article is structured as follows. First we describe the requirements for running the calibration protocol. Then, we describe the individual steps of the protocol. The theoretical background for each step, along with a brief review of available methodologies, is provided in boxes. After some troubleshooting advice, we illustrate the application of the protocol for two case studies. For the sake of clarity, only a
difficult step to generalize to other model types is arguably Step 1. Box 1 mentions recent efforts in this direction.

**Data**: A set of time-resolved measurements of the model outputs. In the present work, data was taken from the aforementioned PEstb benchmark collection.

### Materials

This section describes the inputs and equipment required to run the protocol.

**Hardware**: A standard personal computer, or a computer cluster. For demonstrating the application of the protocol, in the present work we have performed Step 1 on a standard laptop with a 2.40 GHz processor and 8 GB RAM. Optimization, likelihood profiling and sampling were performed on a laptop with an Intel Core i7-10610U CPU (eight 1.80 GHz cores) and 32 GB RAM, with a total runtime of up to 2 days, per model.

**Software**: A software environment with numerical computation and visualization capabilities, along with specialized toolboxes that facilitate performing specific protocol steps. Table 1 lists the software resources used in this work.

**Model**: A dynamic model described by nonlinear ODEs of the following form:

\[
\dot{x} = f(x, \theta, t), \quad x(t_0) = x_0(\theta),
\]

\[
y = g(x, \theta, t),
\]

\[\text{with initial conditions } x_0(\theta), \text{ outputs } y(t), \text{ and parameters } \theta \text{, are possibly nonlinear functions and } x \text{ is the state vector at time } t \text{, for } t \in [t_0, t_f].\]

**Table 1**: Software resources for dynamic model calibration used in this work

| Name          | Type           | Steps | Reference | Website                                      | Environment     |
|---------------|----------------|-------|-----------|----------------------------------------------|-----------------|
| MATLAB        | Environment    | All   |           | http://www.mathworks.com                     | MATLAB, Python  |
| Python        | Environment    | All   |           | http://www.python.org                         | Python          |
| SBML          | Model format   | Input | [21]      | http://www.sbl.org                           | MATLAB, Python  |
| PEstb         | Data format    | Input | [43]      | https://github.com/PEtab-dev/PEtab           | Python          |
| STRIKE-GOLDD  | Tool (SI analysis) | 1   | [44]      | https://github.com/afvillaverde/strike-golddd | Python          |
| AMICI         | Tool (simulation) | 2   | [45]      | https://github.com/AMICI-dev/AMICI           | Python          |
| pyPESTO       | Tool (various steps) | 3, 5, 6 | [46] | https://github.com/ICB-DCM/pyPESTO | Python |
| Fides         | Tool (optimization) | 3, 5 | [47]      | https://github.com/fides-dev/fides           | Python          |
| SciPy         | Tool (various steps) | 3, 5 | [48]      | https://www.scipy.org                         | Python          |
| Data2Dynamics | Tool (various steps) | 3, 5, 6, (O) | [49] | http://www.data2dynamics.org | MATLAB |

The workflow is depicted in Figure 1 and described in the following paragraphs.

### Procedure

The protocol consists of six main steps, numbered 1–6, which consist of substeps. Furthermore, we describe two optional steps.

**STEP 1**: SI analysis

**SI** is analyzed to assess whether the values of all unknown parameters can be determined from perfect continuous-time and noise-free measurements of the observables under the given set of experimental conditions [66, 67]. Structural nonidentifiabilities imply that there are several model parameterizations, e.g. due to symmetries or redundancies in the model structure, which yield exactly the same observables. An overview of the available methodologies for SI analysis is provided in Box 1. Figure 2 illustrates possible sources of structural nonidentifiability and the related issues. The SI analysis can be complemented by observability analysis, which determines if the trajectory of the model state can be uniquely determined from the observables.

The first step in the protocol is thus:

**STEP 1.1**

Analyze the SI of the model with one of the methods described in Box 1.

If all parameters are structurally identifiable and all state variables are observable, we continue with Step 2.1. Otherwise, we recommend to determine the source of the structural nonidentifiability as an intermediate step (1.2). Ideally, the parametric form of the nonidentifiable manifold (i.e. the set of parameters that yield identical observables) is determined. Some tools offer this functionality or at least provide hints, e.g. COMBOS [58], STRIKE-GOLDD [69], ObservabilityTest [52] or the method in [70].
STEP 1.2
If parameters are structurally nonidentifiable or state variables unobservable, use knowledge about the structure of the non-identifiable manifold to

(i) Reformulate the model by merging the nonidentifiable parameters into identifiable combinations, OR
(ii) Fix the nonidentifiable parameters to reasonable values, taken e.g. from the literature or from publicly available biological knowledge databases.

In both cases, the information about the nonidentifiability needs to be retained to later perform a proper analysis of the prediction uncertainties. That is, since parametric uncertainty can propagate to prediction uncertainty, when calculating the confidence interval of a prediction (STEP 6) the fixed parameters must be varied along the range of values that was initially considered. If this point is not taken into account, the obtained confidence intervals are only valid for the reformulated model, but not for the original one—a fact that is often disregarded.

Model reformulation can be illustrated with the example in Figure 2. Using the AutoRepar function in STRIKE-GOLDD, a structurally identifiable reparameterization of the mRNA translation model is obtained. The new variables are $M = k \cdot s \cdot mRNA$, $P = s \cdot GFP$, $M_0 = k \cdot s \cdot mRNA_0$. The new equations are $\dot{M} = -\gamma \cdot M$, $\dot{P} = -\delta \cdot P + M$, $y = P$. Note that, while the resulting model is structurally identifiable and observable, its variables no longer have their original full mechanistic meaning. This is very often, but not always, the case [69]. The model user must decide if such a transformation is acceptable depending on the model purposes. It should also be noted that it is not always possible to find an identifiable reparameterization.

An alternative to the reformulation of the model or the fixing of parameters is to plan additional experiments, if possible. These can be experiments with new experimental conditions, new observables or both (keeping experimental constraints in mind). The additional information should be recorded such that more, ideally all, parameters are structurally identifiable.

STEP 2: Formulation of objective function
The objective function measuring the mismatch of simulated model observables and measurement data is defined. The choice of the objective function depends on the characteristics of the measurement technique and accounts for knowledge about its accuracy. Possible choices are discussed in Box 2.

STEP 2.1
Construct an objective function.

STEP 3: Parameter optimization
Parameter estimates are obtained by minimizing the objective function. To this end, numerical optimization methods suited for nonlinear problems with local minima should be employed. Available methodologies and practical tips for their application are discussed in Box 3, and key aspects are illustrated in Fig. 3.

STEP 3.1
Launch multiple runs of local, global or hybrid optimization algorithms. The number of runs required is model-dependent. For an initial optimization we recommend at least 50 runs with purely local searches or at least 10 runs with global or hybrid searches.

Accurate gradient computation is required for gradient-based optimization. Before optimization, check that the gradients appear correct by evaluating the gradient at a point, and then compare this with forward, backward and central finite difference approximations of the gradient that are evaluated with different step sizes. Such a gradient check is a common, possibly optional, feature of tools that provide gradient-based optimization.
A protocol for dynamic model calibration

STEP 3.2
Evaluate the reproducibility of the fitting results by comparing the optimal objective function values achieved by different runs. The optimal objective function values should be robustly reproducible, meaning that a substantial number of runs (rule-of-thumb: 5) should find it. If this is not the case, repeat Step 3.1 with a larger number of runs. Note that the difference between runs that is considered negligible should be statistically motivated. For the use of log-likelihood and log-posterior this corresponds to an absolute difference, not a relative one [23].

STEP 4: Goodness of fit
The quality of the fitted model should be assessed by visual inspection or use quantitative metrics. Details are provided in Box 4.

STEP 4.1
Assess the goodness of the fit achieved by the parameter optimization procedure.
If the fit is not good, further action is required. Proceed to STEP 4.2.

STEP 4.2
If the fit is not good enough, check convergence of the optimization methods.

1. If there are hints that searches were stopped prematurely (e.g. error messages that indicate that local optimizations did not converge), go back to STEP 3: modify the settings of the optimization algorithms (e.g. increase maximum allowed time and/or number of evaluations) and run the optimizations again.

2. If there are no signs of a premature stop, the problem may be that the optimal solution lies outside the initially chosen parameter bounds → go back to STEP 3: set larger parameter bounds and run the optimizations again. In fact, this action is advisable whenever there are parameter estimates that hit the bounds, even if the fit is good. The exceptions are parameters with hard bounds, originated by physical or mathematical constraints, which should not be enlarged beyond the meaningful limit.

3. If the actions above do not solve the issue, it may be because the optimization method is not well suited for the problem → go back to STEP 3: choose a different method and run the optimizations again.

Figure 2. SI analysis. (A) Diagram of a simplified model of mRNA translation considering only the process in the cytosol. The model captures the translation of mRNA and the degradation of mRNA and protein. (B) Mathematical formulation (ODEs) of mRNA translation dynamics [68] involving two states, mRNA and green fluorescent protein (GFP). (C) The model output is the fluorescence intensity, which is proportional to the GFP level. The model has five unknown parameters: the initial condition of the unmeasured state (mRNA0), three kinetic parameters (γ, δ, k) and an output scaling parameter(s). Given its simplicity, it is possible to calculate the output time-course analytically (here shown for γ ≠ δ). The resulting function contains the product of three parameters (s · k · mRNA0), which is shown in orange, and an expression involving δ and γ, which are shown in green. The latter expression is symmetrical with respect to δ and γ: their values can be exchanged without changing the result. Thus, these two parameters are not structurally globally identifiable, but only locally identifiable with two possible solutions. Furthermore, the product (s · k · mRNA0) allows for an infinite number of parameter combinations; the three involved parameters are structurally nonidentifiable. (D) Illustration of structural nonidentifiability: the time-course of the model output is identical for an infinite number of parameter vectors. (E) Illustration of unobservability caused by nonidentifiability. For illustration purposes, three different parameter vectors are shown, all of which produce the same model output. Each of them yields a different simulation of the mRNA time-course; thus, this state cannot be determined. (F) Illustration of the correlations between the nonidentifiable parameters. The line indicates parameter combinations for which the time-dependent output is identical.
If the new optimizations performed in STEP 4.2 do not yet yield a good fit, there may be a problem with the choice of objective function. Proceed to STEP 4.3.

**STEP 4.3**
If the fit is not good enough, go back to STEP 2 and select a different objective function.

If the new optimization results are still inappropriate, the problem might be the model structure. Proceed to STEP 4.4.

**STEP 4.4**
If the fit is not good enough, go back to the model equations and perform a model refinement.

**STEP 5: Practical identifiability analysis**

The task of quantifying the uncertainty in parameter estimates is known as practical (or numerical) identifiability analysis. It involves calculating univariate confidence intervals or multivariate confidence regions for the parameter values. Key concepts and tools for practical identifiability analysis are listed in Box 5. Practical identifiability issues are illustrated in Figures 5D and 6D.

**STEP 5.1**
Perform practical identifiability analysis with one of the methods described in Box 5. If this analysis reveals uncertainties in parameter estimates that are too large for the intended application of the model, then proceed to STEP 5.2.

**STEP 5.2**
If there are large uncertainties, then:

1. If it is possible to perform new experiments → add more experimental data. In this case, the experiment should be optimally designed in order to yield maximally informative data. This is described in the following section.
2. If it is not possible to perform new experiments → assess the possibility of simplifying the model parameterization without losing biological interpretability.
3. If neither (1) nor (2) are possible → include prior knowledge about parameter values. Such information (either about the value of a parameter or about its bounds) can sometimes be found in publicly available databases.

After performing one of the above actions, go back to STEP 3.

**OPTIONAL STEP**: Alternative experimental design for parameter estimation

If practical identifiability analysis concludes that there are large uncertainties in the parameter estimates, a solution may be to collect new data. Ideally, it should be obtained by designing and performing new experiments in an optimal way. Optimal experiment design (OED) seeks to maximize the information content of the new experiments. It can be performed using optimization techniques that minimize an objective function that represents some measure of the uncertainty in the parameters. It is also possible to perform OED for other goals, such as model discrimination or decreasing prediction uncertainty. OED techniques are discussed in Box (O).
A protocol for dynamic model calibration

Box 2. Theory for STEP 2: Formulation of objective function

The objective function encodes the characteristics of the measurement process and potential prior knowledge. It can be composed of up to two parts:

- the likelihood function $p(D|\theta)$ provides the likelihood of measuring the dataset $D$ given the model parameters $\theta$, and
- the prior distribution $p(\theta)$ encodes additional belief.

Frequentist approaches only use the likelihood function, and the common choice of the objective function is the negative log-likelihood function, $J(\theta) = -\log p(D|\theta)$. Bayesian approaches use the posterior instead of only the likelihood, hence also consider the prior distribution, yielding the unnormalised negative log-posterior $J(\theta) = -\log p(D|\theta) - \log p(\theta)$ (which disregards the marginal probability). In contrast to the likelihood function and the posterior distribution, the logarithmic transformations are numerically easier to evaluate and allow for faster optimiser convergence [23].

Under the assumption of independent measurements and normally distributed noise, the negative log-likelihood function is

$$J(\theta) = \frac{1}{2} \sum_{j=1}^{n_j} \sum_{k=1}^{n_k} \left[ \log(2\pi\sigma_j^2(t_k)) + \left( \frac{y_j^m(t_k) - y_j(t_k, \theta)}{\sigma_j(t_k)} \right)^2 \right],$$

in which $y_j^m(t_k)$ is the measured value and $y_j(t_k, \theta)$ is the simulation result of the $j$-th observable, at time point $t_k$. The corresponding standard deviation of the measurement noise is denoted by $\sigma_j(t_k)$, and can be known (e.g. determined from multiple measurement replicates) or a function of the parameters. In many applications the measurements are repeated to obtain biological or at least technical replicates. In this case, one can either (i) use all replicates as measurements and set the noise level to the standard deviation, or (ii) use the mean of the replicates as measurement and set the noise level to the standard error of mean. In the case of known standard deviations, the summation $\log(2\pi\sigma_j^2(t_k))$ are independent of the parameter vector, and can be disregarded for parameter optimisation and uncertainty analysis. This yields the weighted least squares objective function:

$$J(\theta) = \sum_{j=1}^{n_j} \sum_{k=1}^{n_k} w_j(t_k) \left( \frac{y_j^m(t_k) - y_j(t_k, \theta)}{\sigma_j(t_k)} \right)^2$$

with $w_j(t_k) = \sigma_j^{-2}(t_k)$. Sometimes this objective function is also applied without proper statistical motivation, e.g. without linking the weights to the noise levels, which does not yield a proper frequentist formulation of the calibration problem.

The likelihood function is based on the assumed or observed probability distribution of the experimental error. Normal and log-normal distributions are common choices [71], but in a recent study Laplace distributions have also been used to achieve robustness against outliers [72]. For count measurement, distributions such as the binomial and negative binomial distributions appear to be suited. Furthermore, the consideration of the correlation of measurement errors might be necessary.

In addition to quantitative data, also qualitative and ordinal data might be considered for parameter estimation. Such data are obtained using experimental techniques with difficult to quantify readouts, such as FRET experiment. In contrast to quantitative data, the model simulation cannot be directly compared with the measurements, but only ordering constraints need to be satisfied. These can either be implemented directly [73], captured using statistically motivated objective functions [74] or using an optimal scaling approach [75, 76]. The resulting objective functions have more complex structures and the statistical interpretability is limited.

In addition to the statistically motivated likelihood functions, also statistically motivated prior distributions $p(\theta)$ are sometimes replaced by more mathematically motivated regularisation functions. In general, regularisation is used to tackle the problem that models are often over-parameterised. In this case, the calibration problem is ill-posed and small perturbations in the data can result in very different parameter estimates [77]. Regularisation techniques address this problem by penalizing undesirable parameter choices. Tikhonov regularisation penalises large parameter values, and pushes estimates towards zero. Mathematically, it is identical to a normally distributed prior $p(\theta)$ with mean zero. Alternative regularisation schemes include [78] parameter subset estimation, truncated singular value decomposition, principal component analysis, and Bregman regularisation.

The objective functions encountered in model calibration are mostly nonlinear and non-convex, and possess multiple optima. Furthermore, there are frequently flat regions (in particular in the presence of non-identifiabilities), and rims (e.g. at bifurcation points).

---

STEP 0.1

Define the constraints of the new experimental setup and, in case of optimal design, the criterion to optimize.

STEP 0.2

Obtain a new set of experiments, either by optimization or from an educated guess.

STEP 0.3

Perform experiments and collect data.

STEP 0.4

Include the new data in the objective function and repeat STEPS 2-5.
STEP 6: Prediction uncertainty quantification

If the calibrated model is used for making predictions, for example about the time course of its states, it is useful to assess the prediction uncertainty. This assessment is nontrivial because uncertainty in parameters does not directly translate to uncertainty in predictions. Hence, it is pertinent to quantify to which extent the uncertainty in model parameters leads to uncertainty in the predictions of state trajectories. Note that, if some parameters were fixed in STEP 1 to achieve SI, in this step several values within their plausible range should be considered, in order to obtain realistic confidence intervals of the state predictions. The available methods for prediction uncertainty quantification are reviewed in Box 6. Their application to case studies is shown in Figures 5E and 6E.

STEP 6.1

Calculate confidence intervals for the time courses of the predicted quantities of interest using one of the methods in Box 6.

(Optional STEP): Model selection

The protocol presented so far assumes that the model structure is known, except for the specific values of the parameters. Sometimes the form of the dynamic equations that define the model—and not only the parameter values—is not completely known a priori, and a family of candidate models may be considered. Model selection techniques choose the best model from the set of possible ones, aiming at a balance between model complexity and goodness of fit. They are discussed in Box (MS).
A protocol for dynamic model calibration

Troubleshooting

Troubleshooting advice can be found in Table 2.

Examples

Here, we demonstrate the protocol by describing its application to two examples. The results described here can be reproduced with Matlab live scripts and Jupyter notebooks, which are provided as supplementary material. Additionally, PDF documents that show the scripts and the output generated by them are also included.

Carotenoid pathway model

Our first case study is the carotenoid pathway model by Bruno et al. [41], with 7 states, 13 parameters and no inputs. The model output differs among the experimental conditions: in each of the six experimental conditions for which data is available, only one of the 7 state variables is measured (one is measured in two experiments, and two states are never measured).

The application of the protocol is summarized in the following paragraphs, and the main results are shown in Figure 5.

Box 4. Methods for STEP 4: Assessment of the goodness of fit.

If the noise level is known, the root-mean-square error (RMSE) between simulated and measured observables, i.e. the square root of the mean squared error, provides a quantitative metric of the goodness of fit. The normalised root mean squared error (NRMSSE) is obtained by dividing the RMSE by the range of the measurements, and it is usually more useful since it enables a direct comparison between different observables and/or estimation methods. The NRMSSE has the additional advantage of being independent of the noise in measurements and the number of data points used for the fitting. These and other metrics are further discussed in [84].

Complementary to this, the achieved objective function value can be compared with the expected objective function value. The distribution of expected objective function values for the data generating model with the true parameters can be constructed from the knowledge of the model and the measurement setup. For normally distributed measurement noise with known standard deviation, the sum of squared residuals follows a chi-squared distribution with $n_D$ (number of data points) degrees of freedom. However, while the distribution for the true parameter is analytically tractable, for the estimated parameter this is not the case. To assess whether the achieved objective function value is plausible, approximations can be employed. For linear regression problems it is known that the sum of squared residuals at the optimal parameters follows a chi-squared distribution with $n_D - n_s$ degrees of freedom, where $n_s$ is the number of estimated parameters. This is often also used as an approximation for ODE models [26], but for certain models (e.g. those with oscillatory dynamics) this can be off. An alternative is the use of bootstrapping procedures, in which a problem-specific distribution is constructed [86].

If the achieved objective function value is much larger than most values expected under the distribution, this can indicate underfitting. This implies that either the optimisation was not successful or that the model is inappropriate. Depending on what is accepted for the specific application, the optimisation algorithm might be changed/tuned or the model might be refined. If the achieved objective function value is much smaller than expected, this is a sign of overfitting, meaning that not only the signal in the measurement data but the noise is described. Under- and overfitting are possible causes of wrong model predictions and should be avoided.

A way of controlling for overfitting is to perform cross-validation. To this end, a subset of the data must be left out of the optimisation in STEP 3. There are many ways of partitioning the original data into an estimation and a validation data set [86]. It is common to save e.g. 30% or 40% of the data for the validation phase, but the choice is largely problem-dependent. Afterwards, the calibrated model is used for predicting this data subset. Overfitting appears if the model achieves a good fit in the optimisation, but then fails to generalise to observables or experimental conditions that it was not trained with.

The appropriateness of the noise model in combination with the model topology can also be assessed by checking the distribution of the residuals. In the case of normally distributed measurement noise, the residuals should also be normally distributed. This can be visually assessed using histograms and quantile-quantile (Q-Q) plots or quantified using normality tests.

The different methods provide statistically interpretable results, yet – as with other statistical methods –, in the end the used has to decide about an acceptable error/threshold. These values depend on the purpose of the model and the specific application. To avoid the construction of overly complex models, we suggest for instance to use relative low significance values (0.001 or lower) for the refinement of the model based on indication by the distribution of the expected objective function.

STEP 1.1: SI analysis

We first assess SI and observability for each individual experimental condition, obtaining a different subset of identifiable parameters for each one. Next, we repeat the analysis after combining the information from all experiments, obtaining that all parameters are structurally identifiable. However, the two state variables that are not measured in any experiment ($\beta$-io and OH-$\beta$-io) are not observable. If the initial conditions of these two states were considered as unknown parameters, they would be nonidentifiable.

STEP 1.2: Address structural nonidentifiabilities

We are not interested in the two unobservable states. Hence, we omit this step and proceed with the original model.

STEP 2.1: Objective function

We use the negative log-likelihood objective function described in Equation 2, which is the common choice in frequentist approaches.

STEP 3.1 and 3.2: Parameter optimization

We estimate model parameters using the multi-start local optimization method L-BFGS-B implemented in the Python package...
With 100 starting points we achieve convergence to the maximum likelihood estimate (MLE), as indicated in the waterfall plot (Figure 5). The parameters plot shows that the parameter vector is similar amongst the best starts, indicating that the parameters are well determined by the optimization problem and the optimizer.

**STEP 4.1: Assess goodness of fit**

Visual inspection indicates a good quality of the fit, with simulations closely matching measurements.

**STEP 4.2: Address fit issues**

As the fit is good, this step is skipped.

**STEP 5.1: Practical identifiability analysis**

We analyze practical identifiability using PLs and MCMC sampling. PLs suggest that all parameters are practically identifiable, as the confidence intervals span relatively small regions of the parameter space. The profiles peak at the MLE, suggesting that optimization was successful. MCMC sampling yields...
A protocol for dynamic model calibration

Box (O). Optimal experimental design.
Alternative experimental designs may increase the information content of the data, thereby improving the parameter estimates. If the best possible experiment is found via optimisation, the procedure is called optimal experimental design (OED). OED is formulated as a dynamic optimisation problem, in which the variables that can be changed are the experimental conditions (allowed perturbations, measured quantities, number of experiments, experiment duration, number and location of sampling times), and the objective to maximise is some measure of the information content of the data. The optimisation constraints are the system dynamics and the experimental limitations. The optimisation problem can be solved by control vector parameterization [99].

OED can be performed with several purposes: decreasing the uncertainty in parameter estimates [100, 89, 101, 102, 103], decreasing the uncertainty in model predictions [104, 105], obtaining the best approximate model for control purposes [106], or discriminating between model alternatives [107, 108]. The objective function to optimise depends on this final goal. For the purpose of decreasing parameter uncertainty it is common to use objective functions based on the FIM (4). Typical choices are the D-optimum and E-optimum criteria, which maximise the determinant of the FIM and its minimum eigenvalue, respectively. The D criterion minimises the geometric mean of the errors in the parameters, while the E-criterion minimises the largest error.

STEP 6.1: Prediction uncertainty analysis
We calculate credibility intervals using ensembles of parameters from sampling. In this model, there is a one-to-one correspondence between states and observables; hence, the plots are the same. The prediction uncertainties are reasonably low, suggesting that the model has been successfully calibrated and might be used to predict new behaviour.

Akt pathway model
The second example is an AKT pathway model [42] with 22 unknown parameters, 3 of which are unknown initial conditions, 9 state variables, 3 outputs and 1 input. There are six experimental conditions, each of them with a different input EGF concentration.

Figure 4. MCMC sampling and PL. (A) Upper: traces of MCMC chains through parameter space. The initial sample of a chain is indicated with •. Parameters are labelled as θ with an index as the subscript. The initial sample of the black chain is the MLE from an optimization (at approximately θ₁ = θ₂ = 70). Colour is used in all subfigures to indicate results corresponding to the same MCMC chain. Middle: the marginal distribution (solid line) and 95% credibility interval (shaded region, which corresponds to the shaded region in the upper plot) for a parameter, given the black MCMC chain without burn-in (the set of samples in the chain before the chain converges). Lower: the PL for the global optimum after optimization (see Fig. 3) (dotted line, which corresponds to the dotted line in the upper plot). The 95% likelihood cutoff is indicated with a horizontal line. The corresponding confidence interval is delimited by vertical lines, which are also shown in the upper plot. (B) Traces of the objective function value across the MCMC chains, including burn-in (indicated with vertical grey lines) as detected by the Geweke test. The bottom plot is a zoom-in of the second-to-bottom plot.

similar results; parameter marginal distributions span a similar distance of parameter space compared with PLs, and credibility intervals are also similar.
Results are summarized in the following paragraphs and in Figure 6.

**STEP 1.1: SI analysis**

We consider the following scenarios:

1. For a single experiment with constant EGF, 11 parameters are structurally nonidentifiable, and 3 states are unobservable.
2. For a single experiment with time-varying EGF, the model becomes structurally identifiable and observable.
3. For multiple experiments (at least two) with constant EGF, the model is structurally identifiable and observable.

The experimental data available correspond to the scenario (3) above. The scenario (2) yields an identifiable and observable model, but it requires a continuously varying value of EGF, which is not practical. It is also interesting to note the role of initial conditions in this case study. The results summarized above...
Table 2. Troubleshooting table. Common problems that may appear at different stages of the procedure, their causes and solutions

| Step | Problem | Possible reason | Solution |
|------|---------|-----------------|----------|
| 1    | It is not feasible to analyze SI due to computational limitations | The model is too large and/or too complex | (A) Reduce the model complexity by fixing several parameters (conservative approach) (B) Use a numerical method (e.g. PL) to analyze practical identifiability as a proxy of SI |
| 3    | Parameter optimization takes very long | The size of the model makes this step computationally very expensive | Use parallel optimization approaches to decrease computation times, or try a different optimizer |
| 4    | Parameter optimization does not result in a good fit | (A) The optimizer was stuck in a local minimum (B) The parameter bounds are too small (C) The model is not an adequate representation of the system | (A) Use a global method and allow for enough time to reach the global optimum (B) Set larger bounds (C) Modify the model structure |
| 4    | Parameter optimization resulted in overfitting | Fitting the noise rather than the signal: very good calibration result that however generalizes poorly | In general: use hierarchical optimization if applicable Use cross-validation to detect overfitting. If present: (A) Use regularization in the calibration; (B) Simplify overparameterized models |
| 5    | The confidence intervals of the parameters are too large for the intended application of the model | The data are not sufficiently informative to constrain the values of the parameters sufficiently | (A) Add prior knowledge about parameter values and repeat the optimization (B) Obtain new experimental data (ideally through OED) and repeat the optimization (Same as the above solution) |
| 6    | The confidence intervals of the predictions are too large for the intended application of the model | The data are not sufficiently informative to constrain the values of the predictions sufficiently |

are obtained with generic (nonzero) initial conditions. However, in the available experimental datasets, there are several initial conditions equal to zero. Introducing this assumption in the analyses of the scenarios (2) and (3) leads to a loss of identifiability and observability: four parameters become nonidentifiable and one state becomes unobservable.

STEP 1.2: Address structural nonidentifiabilities

We assume a realistic scenario corresponding to the available experimental data: several experimental conditions with a constant input, EGF and certain initial conditions equal to zero. In this case the model has four nonidentifiable parameters and one unobservable state. To make the model fully observable and structurally identifiable, it is necessary and sufficient to fix the value of two of the nonidentifiable parameters. Thus, we fix two of these parameters and proceed with the next steps.

For comparison, we also performed the remaining steps without fixing the nonidentifiable parameters. We found that fixing the nonidentifiability issues resulted in slightly faster and more robustly convergent optimizations, as well as in better practical identifiability and reduced state uncertainty.

STEP 2.1: Objective function

We choose the negative log-likelihood objective function described in Equation 2.

STEP 3.1 and 3.2: Parameter optimization

Similarly to the other case study, we initially use the multi-start local optimization method ‘L-BFGS-B’.

STEP 4.1: Assess goodness of fit

Visual inspection (i.e. comparison of the simulations produced by the MLE with the measurements) reveals a poor fit to the data (not shown). This result is obtained even with the best result obtained from thousands of optimization runs from different starting points.

STEP 4.2: Address fit issues

First we try to improve the fit by tuning the settings of the optimization method, L-BFGS-B, without success. Then, we try a different method, Fides, which has a higher computational cost but achieves higher quality steps during optimization. With Fides we find an MLE that produces a fit comparable to the one reported in the original publication. The high number of starts (in the order of 10^3) required to find this fit reproducibly indicates that this is a difficult parameter optimization problem.

STEP 5.1: Practical identifiability analysis

Credibility intervals obtained from MCMC sampling indicate that several parameters are practically nonidentifiable. This result is not significantly improved by fixing parameters as suggested in
Figure 5. Calibration of the carotenoid pathway model. (A) Schematic of the model pathway. (B) Visualization of the fit. The plot shows the trajectories of the model observables, as well as the means (points) and standard errors of the means (error bars) of the measurements. (C) Upper: A waterfall plot, showing the number of starts that converged to the MLE. Here and in the remaining subfigures, green indicates results that correspond to the MLE. Lower: A parameters plot, showing variability of parameters among starts that converged to the possible global optimum (green). Vertical dotted lines indicate parameter bounds. (D) Plots related to parameter uncertainty analysis. Upper: a trace of the function values of samples from a MCMC chain. The vertical dotted line indicates burn-in. Middle: marginal density distributions of two parameters, using samples from the converged chain. The plots show a kernel density estimate, histogram and rug plot. Lower: profile likelihood of two parameters. (E) Plots related to prediction uncertainty analysis, computed as percentiles from predictions of samples. Upper: prediction uncertainties of two states. Lower: prediction uncertainties of two observables. Note that in this model, observables are states without transformation, hence, the observables and states have the same uncertainties.
Figure 6. Calibration of the Akt pathway model. (A) Schematic of the model pathway. (B) Visualization of the fit. The plot shows the trajectories of the model observables, as well as the means (points) and standard errors of the means (error bars) of the measurements. (C) Upper: A waterfall plot, showing the number of starts that converged to the MLE. Here and in the remaining subfigures, green indicates results that correspond to the MLE. Lower: A parameters plot, showing variability of parameters among starts that converged to the possible global optimum (green). Vertical dotted lines indicate parameter bounds. (D) Plots related to parameter uncertainty analysis. Upper: a trace of the function values of samples from an MCMC chain. The vertical dotted line indicates burn-in. Middle: marginal density distributions for two parameters, using samples from the converged chain. The plots show a kernel density estimate, histogram and rug plot. Lower: profile likelihood of two parameters. The dotted vertical line indicates a parameter bound. (E) Plots related to prediction uncertainty analysis, computed as percentiles from predictions of samples. Upper: prediction uncertainties of two states under one experimental condition. Lower: prediction uncertainties of two observables under one experimental condition.
STEP 1.2. Improving the practical identifiability of these parameters would require repeating the calibration with additional experimental data.

STEP 6.1: Prediction uncertainty analysis

Credibility intervals obtained from MCMC sampling indicate that the uncertainties in the observable trajectories are reasonably low. However, the state trajectories have larger uncertainties, which make this calibrated model unsuitable for predictions involving these states. The quality of the predictions can be improved by reducing practical nonidentifiabilities in the model, as mentioned in the previous step.

Discussion and conclusion

In this paper, we have proposed a pipeline of methods and resources for calibrating ODE models in the context of biological applications. Its end goal is to obtain a model that is capable of making predictions about quantities of interest with quantifiable uncertainty.

The pipeline consists of a series of steps, each of which represents a task that should be fulfilled before proceeding to the next one to ensure a successful calibration. Performing these tasks entails applying computational methods of different types, symbolic and numerical. The analyses and calculations can be computationally challenging in practice. While the protocol is not dependent on a particular choice of software, we have recommended a number of state-of-the-art tools that implement the methods.

To facilitate the application of the protocol by novices as well as by experienced modellers, we have described in detail how to perform each of the protocol steps. We have also provided the theoretical background required for understanding the underlying problems. Furthermore, we have illustrated its use with two case studies: a carotenoid pathway model in Arabidopsis thaliana and an EGF-dependent Akt pathway of the PC12 cell line. Finally, we have highlighted some of the most common pitfalls in biological modelling, showing how to avoid them.

Key Points

• The correct calibration of dynamic models is essential for obtaining correct predictions and insights.
• While a wide range of tools and resources are currently available, there are also many potential pitfalls, even for the expert.
• Here we propose a model calibration protocol that covers all aspects of the problem.
• The present paper guides the user through all the steps of the pipeline, providing a one-stop guide that is at the same time compact and comprehensive.
• We provide all the code required to reproduce the results and perform the same analysis on new models, so that the biological modelling community can benefit from this pipeline.

Supplementary data

All data, scripts and examples presented in this paper can be downloaded from https://github.com/ICB-DCM/model_calibration_protocol. Supplementary data are also available online at https://academic.oup.com/bib.

Funding

European Union’s Horizon 2020 Research and Innovation Programme (grant no. 686282) (‘CANPATHPRO’); Spanish MINECO/FEDER Project SYNBIOCONTROL (DPI2017-82896-C2-2-R to J.R.B.); Ramón y Cajal Fellowship (RYC-2019-027537-1 to A.F.V.) from the Ministerio de Ciencia e Innovación, Spain; Consellería de Cultura, Educación e Ordenación Universitaria, Xunta de Galicia (ED431F 2021/003 to A.F.V.); Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy (EXC 2151 - 390873048 to J.H.), (EXC-2047/1 - 390685813 to D.P.); German Federal Ministry of Economic Affairs and Energy (grant no. 16KN074236 to D.P.). Ministerio de Ciencia e Innovación, Spain (grant PID2020-117271RB-C22, ‘BIODYNAMICS’, to J.R.B.; Funding for open access charge: Universidade de Vigo/CIGUS).

References

1. Kuepf L, Peter M, Sauer U, et al. Ensemble modeling for analysis of cell signaling dynamics. Nat Biotechnol 2007;25(9):1001–6.
2. Sachs K, Itani S, Fitzgerald J, et al. Learning cyclic signaling pathway structures while minimizing data requirements. In: Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing. NIH Public Access, Kohala Coast, Hawaii, USA, 2009, 63.
3. Fröhlich F, Kessler T, Weindl D, et al. Efficient parameter estimation enables the prediction of drug response using a mechanistic pan-cancer pathway model. Cell Syst 2018;7(6):567–79.
4. Henriques D, Villaverde AF, Rocha M, et al. Data-driven reverse engineering of signaling pathways using ensembles of dynamic models. PLoS Comput Biol 2017;13(2):e1005379.
5. Song H-S, DeVilbiss F, Ramkrishna D. Modeling metabolic systems: the need for dynamics. Curr Opin Chem Eng 2013;2(4):373–82.
6. Almqist J, Cvijovic M, Hatzimanikatis V, et al. Kinetic models in industrial biotechnology–improving cell factory performance. Metab Eng 2014;24:38–60.
7. Villaverde AF, Bongard S, Mauch K, et al. Metabolic engineering with multi-objective optimization of kinetic models. J Biotechnol 2016;222:1–8.
8. Briat C, Khammash M. Perfect adaptation and optimal equilibrium productivity in a simple microbial biofuel metabolic pathway using dynamic integral control. ACS Synth Biol 2018;7(2):419–31.
9. Karamasioti E, Lormeau C, Stelling J. Computational design of biological circuits: putting parts into context. Mol Syst Design Eng 2017;2(4):410–21.
10. Hsiao V, Swaminathan A, Murray RM. Control theory for synthetic biology: recent advances in system characterization, control design, and controller implementation for synthetic biology. IEEE Control Syst 2018;38(3):32–62.
11. Steele H, Papachristodoulou A. Design constraints for biological systems that achieve adaptation and disturbance rejection. IEEE Trans Control Netw Syst 2018;5(2):807–17.
12. Tomazou M, Barahona M, Polizzi KM, et al. Computational re-design of synthetic genetic oscillators for independent amplitude and frequency modulation. Cell Syst 2018;6(4):508–20.
13. Kanehisa M, Goto S. Kegg: kyoto encyclopedia of genes and genomes. Nucleic Acids Res 2000;28(1):27–30.

14. Szklarczyk D, Morris JH, Cook H, et al. The string database in 2017: quality-controlled protein–protein association networks, made broadly accessible. Nucleic Acids Res 2016;45:gkw937.

15. Fabregat A, Jupe S, Matthews L, et al. The reactome pathway knowledgebase. Nucleic Acids Res 2017;46(D1):D649–55.

16. Olivier BG, Snoep JL. Web-based kinetic modelling using JWS online. Bioinformatics 2004;20(13):2143–4.

17. Le Novère N, Bornstein B, Broicher A, et al. BioModels database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. Nucleic Acids Res Jan 2006;34(database issue):D689–91.

18. Chang A, Schomburg I, Placzek S, et al. Brenda in 2015: exciting developments in its 25th year of existence. Nucleic Acids Res 2014;43:gku1068.

19. Witte G, Kania R, Colebiński M, et al. Sabio-rk database for biochemical reaction kinetics. Nucleic Acids Res 2012;40(D1):D790–6.

20. van Riel NAW. Dynamic modelling and analysis of biochemical networks: mechanism-based models and model-based experiments. Brief Bioinform 2006;7(4):364–74.

21. Hucka M, Finney A, Sauro HM, et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 2003;19:524–31.

22. Villaverde AF, Fröhlich F, Weindl D, et al. Benchmarking optimization methods for parameter estimation in large kinetic models. Bioinformatics 2018;35(5):830–8.

23. Hass H, Loos C, Raimúndez-Álvarez E, et al. Benchmark problems for dynamic modeling of intracellular processes. Bioinformatics 2019;35(17):3073–82.

24. Jaqaman K, Danuser G. Linking data to models: data regression. Nat Rev Mol Cell Bio 2006;7(11):813–9.

25. Ashyraliyev M, Fomekong-Nanfack Y, Kaandorp JA, et al. Systems biology: parameter estimation for biochemical models. FEBS J 2009;276(4):886–902.

26. Geier F, Fengos G, Felizzi F, et al. Analyzing and constraining signaling networks: parameter estimation for the user. In: Liu X, Betterton MD (eds). Computational Modeling of Signaling Networks, Volume 880 of Methods in Molecular Biology. Totowa, NJ: Humana Press, 2012, 23–40.

27. Raue A, Schilling M, Bachmann J, et al. Lessons learned from quantitative dynamical modelling in systems biology. PLoS One Jan 2013;8(9):e74335.

28. Kapil G, Kirouac DC, Mager DE, et al. A six-stage workflow for robust application of systems pharmacology. CPT Pharmacometrics Syst Pharmacol 2016;5(5):235–49.

29. Villaverde AF, Banga JR. Reverse engineering and identification in systems biology: strategies, perspectives and challenges. J R Soc Interface 2014;11(91):20130505.

30. Schoukens J, Ljung L. Nonlinear system identification: a user-oriented road map. IEEE Control Syst Mag 2019;39(6):28–99.

31. Balsa-Canto E, Alonso AA, Banga JR. An iterative identification procedure for dynamic modeling of biochemical networks. BMC Syst Biol 2010;4:11.

32. Seaton DD. ODE-Based Modeling of Complex Regulatory Circuits. New York, NY: Springer New York, 2017, 317–30.

33. Eisenkolb I, Jensch A, Eisenkolb K, et al. Modeling of biocatalytic reactions: a workflow for model calibration, selection and validation using Bayesian statistics. AIChE Jl 2019;66(4):e16866.

34. Mannina G, Cosenza A, Vanrolleghem PA, et al. A practical protocol for calibration of nutrient removal wastewater treatment models. J Hydroinf 2011;13(4):575–95.

35. Zhu A, Guo J, Ni B-J, et al. A novel protocol for model calibration in biological wastewater treatment. Sci Rep 2015;5:8493.

36. Vilas C, Arias-Méndez A, García MR, et al. Toward predictive food process models: a protocol for parameter estimation. Crit Rev Food Sci Nutr 2018;58(3):436–49.

37. Tuza Z, Bandiera L, Gomez-Cabeza D, et al. A systematic framework for biomolecular system identification. In: Proceedings of the 58th IEEE Conference on Decision and Control, 2019. IEEE, Nice, France.

38. Whitaker DG, Clerx M, Lei CL, et al. Calibration of ionic and cellular cardiac electrophysiology models. Wiley Interdiscip Rev Syst Biol Med 2020;12(4):e1482.

39. Steiert B, Kreutz C, Raue A, et al. Recipes for analysis of molecular networks using the Data2Dynamics modeling environment. In: Modeling Biomolecular Site Dynamics. Springer, Cham, Switzerland, 2019, 341–62.

40. Raue A, Steiert B, Schelker M, et al. Data2dynamics: a modeling environment tailored to parameter estimation in dynamical systems. Bioinformatics 2015;31(21):3558–60.

41. Bruno M, Koschmieder J, Wuest F, et al. Enzymatic study on atccd4 and atccd7 and their potential to form acrylic regulatory metabolites. J Exp Bot 2016;67(21):5993–6005.

42. Fujita KA, Toyoshima Y, Uda S, et al. Decoupling of receptor and downstream signals in the Akt pathway by its low-pass filter characteristics. Sci Signal 2010;3(132):ra56–6.

43. Schmiester L, Schäle T, Bergmann FT, et al. Petab-interoperable specification of parameter estimation problems in systems biology. PLoS Comput Biol 2021;17(1):e1008646.

44. Villaverde AF, Tsiantis N, Banga JR. Full observability and estimation of unknown inputs, states and parameters of nonlinear biological models. J R Soc Interface 2019;16(156):20190043.

45. Fröhlich F, Kaltenbacher B, Theis FJ, et al. Scalable parameter estimation for genome-scale biochemical reaction networks. PLoS Comput Biol 2017;13(1):e1005331.

46. Stapor P, Weindl D, Ballnus B, et al. Pesto: parameter estimation toolbox. Bioinformatics 2017;34(4):705–7.

47. Froehlich F, Sörger PK. Fides: Reliable trust-region-optimization for parameter estimation of ordinary differential equation models. bioRxiv 2021; 2021.05.20.445065.

48. Virtanen P, Gommers R, Oliphant TE, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods 2020;17:261–72.

49. Miao H, Xia X, Perelson AS, et al. On identifiability of nonlinear ode models and applications in viral dynamics. SIAM Rev Soc Ind Appl Math 2011;53(1):3–39.

50. Chis O, Banga JR, Balsa-Canto E. Structural identifiability of systems biology models: a critical comparison of methods. PLoS One 2011;6(11):e27755.

51. Villaverde AF. Observability and structural identifiability of nonlinear biological systems. Complexity 2019;2019:8497093.

52. Sedoglavic A. A probabilistic algorithm to test local algebraic observability in polynomial time. J Symbolic Comput 2002;33(5):735–55.

53. Karlésson J, Anguelova M, Jirstrand M. An efficient method for structural identifiability analysis of large dynamic
systems. In: 16th IFAC Symposium on System Identification, IFAC, Brussels, Belgium. Vol. 16, 2012, 941–6.

54. Ohtsuka T. Model structure simplification of nonlinear systems via immersion. IEEE Trans Automatic Control 2005;50(5):607–18.

55. Chatzis MN, Chatzi EN, Smyth AW. On the observability and identifiability of nonlinear structural and mechanical systems. Struct Control Health Monit 2015;22(3):574–93.

56. Ligon TS, Fröhlich F, Chiş OT, et al. Genss 2.0: multi-experiment structural identifiability analysis of sbml models. Bioinformatics 2017;34(8):1421–3.

57. Hong H, Ovchinnikov A, Pogudin G, et al. Sian: software for structural identifiability analysis of ode models. Bioinformatics 2019;35(16):2873–4.

58. Meshkat N, Kuo CE, DiStefano JIII. On finding and using identifiable parameter combinations in nonlinear dynamic systems biology models and combos: a novel web implementation. PLoS One 2014;9(10):e110261.

59. Saccomani MP, Bellu G, Audoly S, et al. A new version of daisy to test structural identifiability of biological models. In: International Conference on Computational Methods in Systems Biology. Springer, Cham, Switzerland, 2019, 329–34.

60. Stitger JD, Molenaar J. A fast algorithm to assess local structural identifiability. Automatica 2015;58:118–24.

61. Alkhoury Z, Petreczky M, Mercère G. Identifiability of affine linear parameter-varying models. Automatica 2017;80:62–74.

62. Anstett F, Bloch G, Millérioux G, et al. Identifiability of discrete-time nonlinear systems: the local state isomorphism approach. Automatica 2008;44(11):2884–9.

63. Nömm S, Moog CH. Further results on identifiability of discrete-time nonlinear systems. Automatica 2016;68:69–74.

64. Browning AP, Warne DJ, Burrage K, et al. Parameter identifiability analysis for stochastic differential equation models in systems biology. J R Soc Interface 2020;17(173):20200652.

65. Renardy M, Kirschner D, Eisenberg M. Structural identifiability analysis of pdes: a case study in continuous age-structured epidemic models. arXiv preprint arXiv:2102.06178. 2021.

66. Walter E, Pronzato L. Identification of Parametric Models from Experimental Data. Masson: Springer, 1997.

67. DiStefano JIII. Dynamic Systems Biology Modeling and Simulation. Academic Press, Cambridge, Massachusetts, USA, 2015.

68. Ballnus B, Schaper S, Theis FJ, et al. Bayesian parameter estimation for biochemical reaction networks using region-based adaptive parallel tempering. Bioinformatics 2018;34(13):i494–501.

69. Massonis G, Banga JR, Villaverde AF. Repairing dynamic models: a method to obtain identifiable and observable reparameterizations with mechanistic insights. arXiv preprint arXiv:2012.09826. 2020.

70. Merkt B, Timmer J, Kaschek D. Higher-order lie symmetries in identifiability and predictability analysis of dynamic models. Phy Rev E 2015;92(1):012920.

71. Hengl S, Kreutz D, Timmer J, et al. Data-based identifiability analysis of non-linear dynamical models. Bioinformatics 2007;23(19):2612–8.

72. Mayer A, Westphal S, Geimer T, et al. Fast pose verification for high-speed radiation therapy. In: Bildverarbeitung für die Medizin 2017. Springer, Cham, Switzerland, 2017, 104–9.

73. Mitra ED, Dias R, Posner RG, et al. Using both qualitative and quantitative data in parameter identification for systems biology models. Nat Commun 2018;9(1):1–8.

74. Mitra ED, Hlavacek WS. Bayesian inference using qualitative observations of underlying continuous variables. Bioinformatics 2020;36(10):3177–84.

75. Schmiester L, Weinidl D, Hasenauer J. Parameterization of mechanistic models from qualitative data using an efficient optimal scaling approach. J Math Biol 2020;81(2):603–23.

76. Schmiester L, Weinidl D, Hasenauer J. Efficient gradient-based parameter estimation for dynamic models using qualitative data. Bioinformatics 2021;tabab512.

77. Hadamard J. Sur les problèmes aux dérivées partielles et leur signification physique. Princeton Univ Bull 1902;13:49–52.

78. Lopez D, Barz T, Körkel S, et al. Nonlinear ill-posed problem analysis in model-based parameter estimation and experimental design. Comput Chem Eng 2015;77:24–42.

79. Hörs H, Hasenauer J. Analysis of CFSE time-series data using division-, age-and label-structured population models. Bioinformatics 2016;32(15):2321–9.

80. Kreutz C. New concepts for evaluating the performance of computational methods. IFAC-Papers OnLine 2016;49(26):63–70.

81. Loos C, Krause S, Hasenauer J. Hierarchical optimization for the efficient parametrization of ode models. Bioinformatics 2018;34(24):4266–73.

82. Schmiester L, Schältle Y, Fröhlich F, et al. Efficient parameterization of large-scale dynamic models based on relative measurements. Bioinformatics 2020;36(2):594–602.

83. Penas DR, González P, Egea JA, et al. Parameter estimation in large-scale systems biology models: a parallel and self-adaptive cooperative strategy. BMC Bioinformatics 2017;18(1):52.

84. Li J. Assessing the accuracy of predictive models for numerical data: Not r nor r2, why not? then what? PLoS One Aug 2017;12(8):e0183250.

85. Efron B, Tibshirani R. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. Stat Sci 1986;1(1):54–75.

86. Pillonetto G, Dinuzzo F, Chen T, et al. Kernel methods in system identification, machine learning and function estimation: a survey. Automatica 2014;50(3):657–82.

87. Cramér H. Mathematical Methods of Statistics (PMS-9), Vol. 9. Princeton university press, Princeton, New Jersey, USA, 2016.

88. Wieland F-G, Hauber AL, Rosenblatt M, et al. On structural and practical identifiability. Curr Opin Syst Biol 2021;25:60–9.

89. Banga JR, Balsa-Canto E. Parameter estimation and optimal experimental design. Essays Biochem 2008;45:195–210.

90. Joshi M, Seidel-Morgenstern A, Kremling A. Exploiting the bootstrap method for quantifying parameter confidence intervals in dynamical systems. Metab Eng 2006;8:447–55.

91. Fröhlich F, Theis FJ, Hasenauer J. Uncertainty analysis for non-identifiable dynamical systems: profile likelihoods, bootstrapping and more. In: International Conference on Computational Methods in Systems Biology. Springer, Cham, Switzerland, 2014, 61–72.

92. Tukey JW. Bias and confidence in not-quite large samples. Ann Math Statist 1958;29:614.

93. Efрон B, Stein C. The Jackknife estimate of variance. Ann Stat 1981;9(3):586–96.

94. Toni T, Welch D, Strelkowa N, et al. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. J R Soc Interface 2009;6(31):187–202.
95. Liepe J, Kirk P, Filippi S, et al. A framework for parameter estimation and model selection from experimental data in systems biology using approximate Bayesian computation. Nat Protoc 2014;9(2):439–56.

96. Hug S, Raue A, Hasenauer J, et al. High-dimensional Bayesian parameter estimation: case study for a model of jak2/stat5 signaling. Math Biosci 2013;246(2):293–304.

97. Vanlier J, Tiemann CA, Hilbers PAJ, et al. An integrated strategy for prediction uncertainty analysis. Bioinformatics 2012;28(8):1130–5.

98. Raue A, Kreutz C, Maiwald T, et al. Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. Bioinformatics Aug 2009;25(15):1923–9.

99. Balsa-Canto E, Alonso AA, Banga JR. Computational procedures for optimal experimental design in biological systems. IET Syst Biol 2008;2(4):163–72.

100. Steiert B, Raue A, Timmer J, et al. Experimental design for parameter estimation of gene regulatory networks. PLoS One 2012;7(7):e40052.

101. Franceschini G, Macchietto S. Model-based design of experiments for parameter precision: state of the art. Chem Eng Sci 2008;63(19):4846–72.

102. Pronzato L. Optimal experimental design and some related control problems. Automatica 2008;44(2):303–25.

103. Kreutz C, Raue A, Kaschek D, et al. Profile likelihood in systems biology. FEBS J 2013;280(11):2564–71.

104. Hagen DR, White JK, Tidor B. Convergence in parameters and predictions using computational experimental design. Interface Focus 2013;3(4):20130008.

105. Gevers M. Identification for control: from the early achievements to the revival of experiment design. Eur J Control 2005;11(4–5):335–52.

106. Casey FP, Baird D, Feng Q, et al. Optimal experimental design in an epidermal growth factor receptor signalling and down-regulation model. IET Syst Biol 2007;1(3):190–202.

107. Waldron C, Pankajakshan A, Quaglio M, et al. Closed-loop model-based design of experiments for kinetic model discrimination and parameter estimation: benzoic acid esterification on a heterogeneous catalyst. Ind Eng Chem Res 2019;58(49):22165–22177.

108. Villaverde AF, Raimúndez E, Hasenauer J, et al. A comparison of methods for quantifying prediction uncertainty in systems biology. IFAC-Papers OnLine 2019;52(26):45–51.

109. Shahmohammadi A, McAuley KB. Sequential model-based a-optimal design of experiments when the fisher information matrix is noninvertible. Ind Eng Chem Res 2019;58(3):1244–61.

110. Hass H, Kreutz C, Timmer J. Likelihood based observability analysis and confidence intervals for predictions of dynamic models. BMC Syst Biol 2012;6(1):120.

111. Bock HG, Körkel S, Schlöder JP. Parameter estimation and optimum experimental design for differential equation models. In: Model Based Parameter Estimation. Springer, Berlin, Heidelberg, 2013, 1–30.

112. Brown KS, Hill CC, Calero GA, et al. The statistical mechanics of complex signaling networks: nerve growth factor signaling. Phys Biol 2004;1(3):184.

113. Villaverde AF, Bongard S, Mauch K, et al. A consensus approach for estimating the predictive accuracy of dynamic models in biology. Comput Methods Programs Biomed 2015;119(1):17–28.

114. Vyshevinsky V, Girolami MA. Bayesian ranking of biochemical system models. Bioinformatics 2008;24(6):833–9.

115. Tibshirani R. Regression shrinkage and selection via the LASSO. J R Stat Soc B Methodol 1996;58(1):267–88.

116. Steiert B, Timmer J, Kreutz C. L 1 regularization facilitates detection of cell type-specific parameters in dynamical systems. Bioinformatics 2016;32(17):i718–26.