Chapter 2

Analyzing and Constraining Signaling Networks: Parameter Estimation for the User

Florian Geier, Georgios Fengos, Federico Felizzi, and Dagmar Iber

Abstract

The behavior of most dynamical models not only depends on the wiring but also on the kind and strength of interactions which are reflected in the parameter values of the model. The predictive value of mathematical models therefore critically hinges on the quality of the parameter estimates. Constraining a dynamical model by an appropriate parameterization follows a 3-step process. In an initial step, it is important to evaluate the sensitivity of the parameters of the model with respect to the model output of interest. This analysis points at the identifiability of model parameters and can guide the design of experiments. In the second step, the actual fitting needs to be carried out. This step requires special care as, on the one hand, noisy as well as partial observations can corrupt the identification of system parameters. On the other hand, the solution of the dynamical system usually depends in a highly nonlinear fashion on its parameters and, as a consequence, parameter estimation procedures get easily trapped in local optima. Therefore any useful parameter estimation procedure has to be robust and efficient with respect to both challenges. In the final step, it is important to access the validity of the optimized model. A number of reviews have been published on the subject. A good, nontechnical overview is provided by Jaqaman and Danuser (Nat Rev Mol Cell Biol 7(11):813–819, 2006) and a classical introduction, focussing on the algorithmic side, is given in Press (Numerical recipes: The art of scientific computing, Cambridge University Press, 3rd edn., 2007, Chapters 10 and 15). We will focus on the practical issues related to parameter estimation and use a model of the TGFβ-signaling pathway as an educative example. Corresponding parameter estimation software and models based on MATLAB code can be downloaded from the authors's web page (http://www.bsse.ethz.ch/cobi).

Key words: Parameter estimation, Dynamics models, User guide

1. Pre-regression Diagnostics

Typically, modeling efforts are started after some experimental data has already been obtained, and based on these data a model is developed and parameterized. Initially, some (or most) parameter values will be unknown or can only be constrained to a biologically relevant range. Due to this incomplete information, a pre-regression analysis of the model is important to evaluate which and how

Xuedong Liu and Meredith D. Betterton (eds.), *Computational Modeling of Signaling Networks*, Methods in Molecular Biology, vol. 880, DOI 10.1007/978-1-61779-833-7_2,

model parameters can be estimated from future data. To estimate a parameter, changes in this parameter value need to affect the prediction of the model with regard to a state value for which there is data. In other words: the measured output must be sensitive to each parameter that we seek to estimate. Therefore, pre-regression analysis of the model is centered around sensitivity analysis, answering the question: "How does a system output depend of a certain parameter value?".

1.1. Sensitivity Analysis

A sensitivity analysis evaluates the dependence of a system output on a certain set of model parameters. Let $m = \{ m_1, m_2, \ldots, m_M\}$ be a set of M measurable output and $p = \{ p_1, p_2, \ldots, p_P\}$ the set of P unknown model parameters. The matrix of sensitivity coefficients ${S_p}^m$ (see Chapter 1) of the output with respect to the parameters is defined for each entry as

$$S_{p_j}^{m_i} = \frac{\partial m_i}{\partial p_j}, \quad i = 1, \ldots, M \quad j = 1, \ldots, P. \tag{1}$$

To optimally identify a model parameter, its corresponding sensitivities should be large and distinct from the sensitivities of all the other parameters. In other words, each column of the sensitivity matrix as defined by Eq. 1 should have at least one large entry and all columns must be linearly independent. If the latter does not hold, certain parameter changes can compensate for each other leaving the model output unchanged. As a result, parameter estimates will be correlated when inferred from the measurements. Since the sensitivities are state and time dependent, the way that the system is measured has a strong impact on the calculated sensitivities. Accordingly, sensitivity analysis can be used to plan new informative experiments with the aim to maximize the identifiability of the model parameters (2). Note that this type of analysis is local, i.e., applies only for a specific set of (a priori defined) parameter values. For a generally valid conclusion, the analysis may have to be carried out over a large set of biologically plausible parameter values.

2. Parameter Estimation

We now turn to the problem of parameter estimation. We highlight the major steps in the optimization of parameters with a special emphasis on gradient-based optimization methods.

2.1. The Model

Our dynamical system with N state variables can be described by a set of ordinary differential equations. If we write $x(t)$ for a vector with all state variables, k for a vector with all parameters, and x_0 for

the vector for all initial expressions, this set of differential equations can be expressed as

$$\frac{dx(t)}{dt} = f(x(t), t, k), \qquad x(t_0) = x_0. \tag{2}$$

Often, the state variables cannot be directly observed, and there are combinations of state variables, or relative quantities (or possibly even more complicated functions of the state variables) that are measured in experiments. We therefore specify an observation function $g : \mathbb{R}^N \to \mathbb{R}^M$ which maps the state variables x to a set of M observables,

$$y(t) = g(x(t), s) \tag{3}$$

We require both $f(\cdot)$ and $g(\cdot)$ to be continuously differentiable functions with respect to their parameters. The vector s comprises the parameters of the observation function. Note that our formulation includes measurement settings where we can only partially observe the system such that $M < N$. The set of problem-specific parameters p includes the initial conditions, the model parameters, and the parameters that are specific to the measurements, i.e., $p = \{x_0, k, s\}$. The initial values of the dynamical system are also parameters as they are usually unknown.

2.2. Measurement Data

We denote the measured data by y_{ij}. Generally, these measurements y_{ij} are subject to error, i.e., they are the sum of the observables $y_j(t_i)$ and a measurement error, ε_{ij}.

$$y_{ij} = y_j(t_i) + \epsilon_{ij}. \tag{4}$$

In the following, measurement errors are assumed to be independent across all observations and all time points and follow independent Gaussian distributions with zero mean and state and time-dependent variance ${\sigma_{ij}}^2$. Due to the law of large numbers these assumptions apply in many practical settings. However, other distributions, in particular log-normal distributions, are also encountered, for instance when protein concentrations are low. Therefore, independence and normality should be checked in the course of data pre-processing. Data should be transformed appropriately in case of deviations from normality as discussed in (3). The observation function $g(\cdot)$ can take account for this transformation.

2.3. Nonlinear Regression

Intuitively, an optimal model should minimize the deviation between model prediction and data and thus make the measurements most likely given the model. In other words, an optimal parameter set is obtained by maximizing the likelihood L of the

data y with respect to the parameter set p. The likelihood L takes the following form given our assumptions above:

$$L(y|p) = \prod_{i=1}^{T} \prod_{j=1}^{M} \frac{1}{\sigma_{ij}\sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{(y_{ij} - g_j(x(t_i, p), p)))^2}{\sigma_{ij}^2} \right). \quad (5)$$

Due to the asymptotic properties of the maximum likelihood principle, it occupies a central position in estimation theory. In the limit of infinitely many data, it yields an unbiased, normally distributed parameter estimate with a minimal variance (4). However, in many practical settings, these properties are not matched due to limited amounts of data. Still, maximum likelihood is the most commonly used estimation principle because it is rather easy to implement. In practical terms, to find the maximum of the likelihood function the negative log likelihood is minimized.

$$\begin{aligned} -\log[L(y|p)] &= \sum_{i=1}^{T} \sum_{j=1}^{M} \frac{1}{2} R_{ij}(p)^2 + c_{ij}, \\ R_{ij}(p) &= \frac{y_{ij} - g_j(x(t_i, p), p))}{\sigma_{ij}}, \qquad c_{ij} = \log\left[\sigma_{ij}\sqrt{2\pi}\right]. \end{aligned} \quad (6)$$

The term c_{ij} in Eq. 6 is independent of p, and can be left out of the minimization. The maximum likelihood estimator for the model parameters is thus given by

$$p^* = \arg\min_{p} \sum_{i=1}^{T} \sum_{j=1}^{M} \frac{1}{2} R_{ij}(p)^2. \quad (7)$$

Note that Eq. 7 is essentially a least squares minimization problem (5).

2.4. Gradient Calculation

Common methods to minimize Eq. 7 are gradient based such as the classical Gauss–Newton or Levenberg–Marquardt methods (5). Gradient-based methods follow an iterative procedure in order to minimize Eq. 7 where in each step the gradient of the residuals $R_{ij}(p)$ is used to calculate a parameter update.

$$\begin{aligned} \frac{\partial}{\partial p_l} R_{ij}(p) &= -\frac{1}{\sigma_{ij}} \frac{\partial}{\partial p_l} g_j(x(t_i, p_l), p_l)) \\ &= -\frac{1}{\sigma_{ij}} \left(\sum_{n=1}^{N} \frac{\partial g_j}{\partial x_n}\bigg|_{t_i} \frac{dx_n}{dp_l}\bigg|_{t_i} + \frac{\partial g_j}{\partial p_l}\bigg|_{t_i} \right) \end{aligned} \quad (8)$$

$\frac{\partial g_j}{\partial x_n}$ and $\frac{\partial g_j}{\partial p_l}$ in Eq. 8 are the Jacobians of the differential equation system with respect to the state variables and with respect to the parameters. $S_{p_l}^n = (dx_n)/(dp_l)$ in Eq. 8 are the so-called sensitivities of the state variables to changes in the parameter values that we discussed in the previous chapter. In general there is no analytic solution to the trajectories $x(t, p)$ and therefore the sensitivities

$S^n_{p_l} = (dx_n)/(dp_l)$ in Eq. 8 have to be calculated numerically. Naively, one may approximate them by finite differences, which is also the default in many optimization software packages. However, this approximation is numerically unstable and becomes computationally very expensive in case of a high-dimensional parameter space as it demands many integrations of the differential equations (Eq. 2) (6). Alternatively, the sensitivities can be computed by an integration of the sensitivity equations (as discussed in the previous chapter) in parallel with Eq. 2.

$$\frac{dS^n_{p_l}}{dt} = \frac{d}{dt}\frac{dx_n}{dp_l} = \frac{d}{dp_l}\frac{dx_n}{dt} = \frac{df(t,x(t),k)}{dp_l} = \sum_{q=1}^{N}\frac{\partial f_n}{\partial x_q}\frac{dx_q}{dp_l} + \frac{\partial f_n}{\partial p_l}$$
$$S^n_{p_l}(0) = \left(\frac{dx_q}{dp_l}\right)(0) = \begin{cases} 1: & p_1 \in \{x_0\} \\ 0: & p_1 \in \{s,k\} \end{cases} \tag{9}$$

Note that $\partial f/\partial x_0 = 0$ and $\partial g/\partial x_0 = 0$. Since $f(\cdot)$ and $g(\cdot)$ are specified beforehand, their derivatives with respect to parameters and state variables can also be computed beforehand, either manually or for more complex systems by applying symbolic computations. In case of using implicit ODE solvers, the Jacobian $\partial f / \partial x$ should also be provided to the solver as it speeds up the calculations considerably. In our experience, an efficient and reliable computation of the sensitivities and gradients of the residuals is absolutely crucial for the success of a gradient-based minimization in high-dimensional parameter space. The parallel solution of the sensitivity equation has also the advantage of getting the sensitivity of certain system properties, defined by an appropriate observation function, in parallel with the solution of Eq. 2.

2.5. Minimization Process

A brief workflow of the gradient-based optimization procedure is given in Table 1. The procedure starts with an initial set of parameter values. In each cycle of the iteration, the ODE systems (Eqs. 2 and 9) are solved and the residuals and gradient of the residuals (Eq. 8) are calculated. Based on the current parameter values and the gradient, new parameter values that minimize the sum of squared residuals (Eq. 7) are minimized. It is noteworthy that the minimization step itself is only a minor contributor to the total computation time. The main computational burden is created by the need to solve the system of ODEs for the state variables and for the sensitivities many times. There are three commonly used procedures to update the parameters. We will only briefly mention them and refers the more interested reader to the discussions and algorithmic implementations presented in (5). In the Gauss–Newton procedure, the update step is calculated by solving by a linear regression in the unknown parameter increments. The Levenberg–Marquardt procedure adds considerable robustness to

Table 1
General workflow of gradient-based minimization procedures. In each parameter update step, the system of ODEs and the sensitivity equations are integrated

Initialize model system and parameters

LOOP

Integrate ODE (Eq. 2.2) and sensitivity equations (Eq. 2.9) based on current parameter vector

Calculate residuals defined in Eq. 2.7 and Jacobian of residuals based on sensitivities and Eq. 2.8

Use gradient-based technique such as Trust-Region to calculate parameter increment utilizing

Jacobian of residuals

IF convergence criteria fulfilled

BREAK

ELSE

Update parameter vector

ENDIF

ENDLOOP

Calculate fit statistics, parameter variances and confidence limits

Gauss–Newton by an adaptive regularization of the linear regression problem to catch ill-conditioned cases, e.g., if some parameters are nonidentifiable. The third commonly used update scheme approximates the optimized function (Eq. 7) in a local region by a simpler, possibly lower dimensional, function. This "trust region" is chosen adaptively and the minimization is performed herein. MATLAB's lsqnonlin function, which was used in the example given below, implements all three procedures with a trust region-based method as the default. The iteration continues until a certain stopping criterion is matched, e.g., if the change in residual norm (or the relative parameter change) is smaller than a predefined value. Finally, postregression statistics such as the goodness of fit (GOF), parameter covariances and confidence limits are performed, as described below.

The success and robustness of any optimization procedure is intricately linked with the model dynamics and data complexity. For some data settings, e.g., oscillatory time series data, parameter estimating is particularly difficult due to many local minima in Eq. 7. In this case, a multiple shooting approach can greatly

increase the convergence radius of the global minimum (6). Since all optimizers have intrinsic parameters which influence the success of the minimization we generally recommend to test different optimization settings as well as different optimizers.

In Table 2, we compare the performance of different minimization algorithms from MATLAB (lsqnonlin) and the Systems Biology Toolbox (www.sbtoolbox2.org). To this end, we used a model of the TGFβ signalling pathways which is described in detail below. We simulated time series data from the model including 14 observations with 20 data points each and fitted 16 model parameters. The table shows a clear result: only the gradient-based Trust-Region method of MATLAB's lsqnonlin leads to accurate parameter estimates and performs efficient and robust. However, this excellent performance requires the specification of the gradient of the residuals (Eq. 8). If this information is not given, the gradient is approximated by finite differences, which leads to a considerable increase in computation time (ODE integrations) and the number of convergent fits is reduced. We also tested the Levenberg-Marquardt procedure of MATLAB's lsqnonlin function, which performed very poorly. As the implementation cannot handle bounds on parameter values, the algorithm frequently ran into negative values, especially for the unidentifiable parameters. As a result, ODE integration failed and the optimization was canceled. The Trust-Region methods can easily handle parameter constraints and the ODE integration never failed in all 30 trails. In contrast to the previous methods, the simplex and simulated annealing algorithms provided by the Systems Biology Toolbox don't use gradient information. Table 2 shows that both perform poorly. The simplex method converged only in 4% (1 of 30 trials) to the true minimum as measured by the GOF. Simulated annealing converged to the true minimum in 17% (5 out of 30 trials). This is in stark contrast to the Trust-Region method which found the true minimum in 100% of all convergent fits (which are 97%). Interestingly, the number of identifiable parameters as measured by the coefficient of variation is low in case of simulated annealing. Nevertheless, the smaller number of identifiable parameters were fitted with very good accuracy (see parameter norm, last column). Note, that we used the standard settings provided for both methods and only changed those that control the total computation time (e.g. the total number of function evaluations) in order to allow sufficient time for each minimization. In summary, the gradient based Trust-Region method greatly outperforms the two other gradient free methods in computation time, robustness and accuracy.

Table 2
Performance of different optimization algorithms. All optimizers were run on the same data set containing all but the TGFβ receptor and the I-Smad mRNA states (20 data points and 10% measurement error). In total, 16 parameters where fitted. Results are averaged over 30 different parameter starting conditions and are given as median and interquartile ranges (IQR). The IQR is a robust measure of spread and indicates the width of the range containing 50% of the data. Column 1 states the optimization algorithm. Simplex is a gradient free, local minimization method while simulated annealing is a stochastic, global minimization method. All methods were run with the default settings except for those that restrict the total number of function evaluation. These were increased to 10^5 in order to allow for sufficiently long minimization times. Column 2 indicates whether the Jacobian of the residuals (Eq. 8) was used. Column 3: percentage of true convergent fits, as evaluated by the GOF probability $Pr[\chi_{264}^2] \geq 0.1$. Column 4: average computation time in minutes. All minimizations were run on an Intel Xeon(R) CPU, 2.83 GHz. Column 5: optimal χ^2 value (bold font) and GOF probability (regular font). The χ^2 distribution has 264 degrees of freedom. Column 6: percent of identifiable parameters. Parameters are called identifiable if their coefficient of variation is smaller than 1. Column 7: norm of the relative parameter deviation defined as $\left\| \frac{p_{true} - p^*}{p_{true}} \right\|^2$. The norm is given for all (bold font) and only for the identifiable (regular font) parameter

Optimizer	Jacobian (Eq. 8) used	% True convergent fits	Computation time [min]	Goodness of fit: χ^2, $Pr[\chi^2]$	% Identifiable parameter	Deviation of parameter estimates: all, identifiable
MATLAB lsqnonlin: Trust-region	Yes	97	2.3 (1.3)	**214.55 (1.25)**, 0.99 (0)	75 (0)	**1,513 (11,662)**, 0.42 (0.15)
	No	73	39.5 (12.2)	**187.84 (0.18)**, 0.99 (0)	75 (0)	**1,124 (15,958)**, 0.27 (0.1)
SB toolbox: simplex	No	3	7.8 (7.8)	**3,705.16 (2,758.20)**, 0 (0)	50 (23)	**551 (7,819)**, 0.95 (0.4)
SB toolbox: annealing	No	16	23.9 (6.7)	**1,677.12 (2,519.10)**, 0 (0)	38 (16)	**62,521 (89,370)**, 0 (0)

Average values are given in the following format: median (IQR)

3. Post-regression Diagnostics

After parameter fitting the quality of the fit should be evaluated. We start with an evaluation of the goodness-of fit followed by an estimation of the confidence intervals for the estimated parameters. These can be used to calculate confidence intervals for future predictions generated with the model. Finally, we evaluate the correlations between estimated parameters. Because of such correlations we may not be able to determine a unique set of parameters as best fits, but instead we may obtain families of parameter solutions.

3.1. Goodness of Fit

Since the measurement error is Gaussian distributed, the weighted residuals are also Gaussian distributed with unit variance. Therefore, the sum of squared residuals follows a χ^2 distribution.

$$\sum_{i=1}^{T}\sum_{j=1}^{M} R_{ij}(p)^2 \sim \chi^2(p). \tag{10}$$

Intuitively we expect from a good fit that the deviations of the model from the data should be of the same order as the measurement error, i.e., $R_{ij} \approx 1$, which means that the sum in Eq. 10 should be centered around $T \cdot M$. A much larger χ^2 value than $T \cdot M$ indicates some variation in the data which is not accounted for by the model. This fact can be used to evaluate the quality of the fit in a GOF test which gives the probability of observing an as large or larger value than the value of $\chi^2(p^*)$ at the minimum (5). However, since the parameters were adjusted in order to minimize Eq. 7 the degrees of freedom of the χ^2 distribution are $dof = T \cdot M - P$, where P is the number of parameters. Usually a cut-off value such as $\Pr[\chi^2_{\mathrm{dof}}(p^*)] < 0.05$ is used to reject the fit. Note that an underestimation of the measurements errors or non-normality of the errors also results in an exceptionally large $\chi^2_{\mathrm{dof}}(p^*)$ value, i.e., a small probability $\Pr[\chi^2_{\mathrm{dof}}(p^*)]$. The GOF test is not powerful in detecting overfitting. Overfitting results if a model, which is too complex, would also fit the particular realization of the measurement error and thus have a much smaller value of $\chi^2_{\mathrm{dof}}(p^*)$ than the expected value which is equal to dof. A more appropriate way to detect overfitting is the comparison with a simpler model through a likelihood ratio test (2).

3.2. Confidence Intervals

The complexity of nonlinear optimization precludes a straightforward way of calculating confidence limits for parameter estimates. However, we can employ an approximate result which is valid in the limit of infinitely many data and complete parameter identifiability. Specifically, one can relate the variance in the parameters to the curvature of the $\chi^2_{\mathrm{dof}}(p^*)$ function at its minimum in order to derive parameter covariances and asymptotic confidence intervals (5).

Intuitively, the curvature determines how well the minimum is confined and therefore how well parameter estimates can be defined. The curvature of $\chi^2_{\mathrm{dof}}(p^*)$ at the minimum is determined by the $dof \times P$ matrix of second derivatives, the so-called Fisher information matrix (FIM).

$$FIM = (\partial^2 R_{ij}/\partial p_l^2) \approx J^T J. \tag{11}$$

The approximation in Eq. 11 neglects the second derivative terms but is computationally inexpensive as $J_{ij}^{p_l} = (\partial R_{ij}/\partial p_l)$ is the calculated gradient matrix of the residuals during minimization. The covariance of the parameters C is related to the inverse of the FIM as

$$C = 2(FIM)^{-1}. \tag{12}$$

Asymptotic confidence intervals can be calculated by taking into account the distribution of the χ^2 values, which are approximately Gaussian for large degrees of freedom. The 95% confidence intervals are given by

$$p^* \pm 1.96\sqrt{diag(C)}. \tag{13}$$

Symmetric confidence intervals are problematic if a parameter estimate is close to the boundary of the admissible parameter space. Moreover, the approximation Eq. 13 gives misleading results whenever the above-stated assumptions are heavily violated, a fact which can be difficult to evaluate beforehand (7).

3.3. Bootstrap

This method offers an alternative to the asymptotic approximation of parameter uncertainties. It is a heuristic but nevertheless exact way of determining parameter uncertainties. Bootstrap methods construct an empirical distribution of the parameter estimates by a repeated data resampling and consecutive parameter estimation. Parameter uncertainties can be inferred from the shape of the empirical parameter distribution (8). The elegance and simplicity of the bootstrap comes at a price. As it involves many parameter estimations, it is computationally more expensive.

3.4. Parameter Correlation and Identifiability

Frequently, the optimization procedure does not yield a unique optimal parameter set, because there is no unique optimal $\chi^2(p^*)$ value given the available data. In this case the value of some or all parameters is nonidentifiable. Nonidentifiability is the result of a nonunique χ^2 minimum, which can be caused, e.g., by a very flat χ^2 landscape. The later implies a functional relation between parameters along which the χ^2 value is unaltered. Parameter estimates appear highly correlated if this functional relationship is linear. Hengl et al. (9) suggest an approach based on mutual information and bootstrap to detect parameter dependencies taking many,

equally well-fitting, parameter estimates as an input. This heuristic approach is particularly easy to apply as it does not require any a priori knowledge on the model structure.

There are three common ways to deal with nonidentifiability. One solution is to fix some of the nonidentifiable parameter at educated values and only estimate the remaining parameters. These estimates are of course biased since their optimum is in a functional relation to the fixed parameters. Alternatively, subsequent analyses can be based on all admissible parameter sets and the parameter sets can then be clustered according to the predictions derived from them. A third approach might be to reduce the model such that it does not contain the nonidentifiable parameters, e.g., by phenomenological descriptions or timescale separation techniques (10). It is noteworthy that nonidentifiability of parameters does not imply a poor fit to the data, but that parameter values cannot be constrained to a unique value. The predictive power of the model will therefore be limited to model predictions that are not sensitive to nonidentifiable parameters.

4. An Application: TGFβ Signaling

We now apply the above-mentioned methods to a specific example, the TGFβ-signaling pathway. We will concentrate on a previously published TGFβ-signaling model (11) which is extended by a transcriptional negative feedback involving an inhibitory Smad. An outline of the TGFβ model is presented in Fig. 1. The model consists of 18 variables and 19 kinetic rate constants, most of which have been determined in previous studies (11; 12). A brief recapitulation of the system dynamics is given in the caption of Fig. 1. A practical guideline should offer means to easily recapitulate the main analysis steps. We therefore provide for the interested reader the model as well as all functions in MATLAB format to reiterate the results presented in this section by own computer simulations from the authors web page (http://www.bsse.ethz.ch/cobi). The experienced programmer can easily extend the code for own modeling projects.

4.1. Sensitivities

In order to better understand the functioning of the complex network, we will first turn our attention to the sensitivities of the model output with respect to the model parameters. Figure 2 summarizes our results. The accompanying MATLAB script script_sensitivity.m can be used to reproduce the results. Panel **a** shows time courses of the stimulation protocol and the output variable (nuclear Smad2*/Smad4 complex). We stimulate the model by an initial TGFβ pulse and apply an inhibitor of TGFβ-receptor auto-phosphorylation after 3h for a duration of one hour.

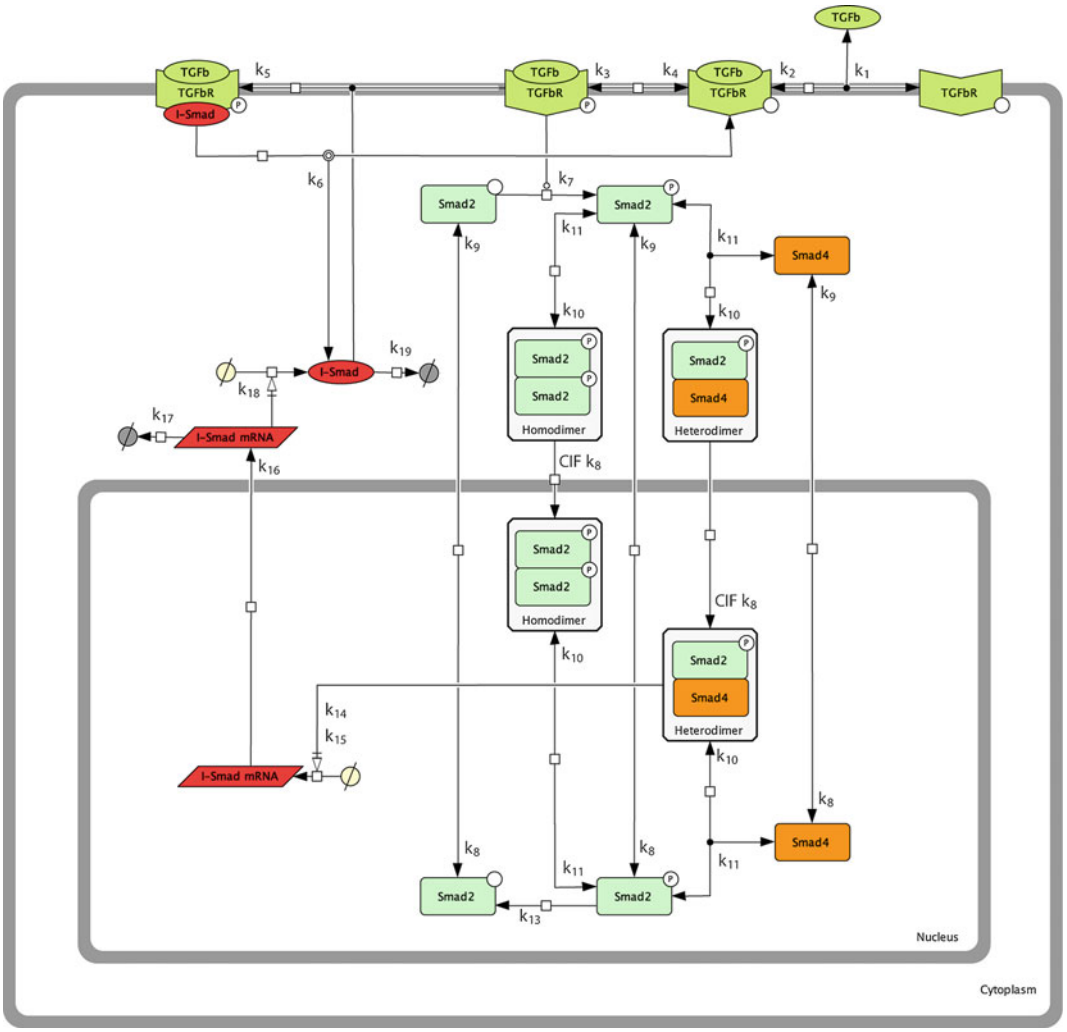

Fig. 1. Outline of the TGFβ model adopted from (11) including an inhibitory Smad. TGFβ binds with strong affinity to the receptor. The receptor complex gets autophosphorylated and signals by enhancing the phosphorylation of Smad2. The active form of Smad2 can either form Smad2 dimers or heterodimerize with Smad4. The Smads and their complexes can shuttle between the cytoplasm and the nucleus. The effect of a phosphatase (dephosphorylation of Smad2) is only considered in the nuclear compartment. The Smad2–Smad4 heterodimer serves as a transcription factor for the production of I-Smad mRNA. The mRNA needs to transfer to the cytoplasm to be translated into I-Smad protein. Finally the I-Smad can bind to the active ligand–receptor complex and decrease the total amount of the initial signal, thereby acting s as an inhibitor. The respective rate constants of each reaction are indicated as $k_1 - k_{19}$ and the complex import factor (CIF).

This stimulation protocol results in a complex response of the model. Note that the initial transient dynamics after stimulation is very fast due to the high affinity of TGFβ to its receptor. Panel **b** shows the time-resolved control coefficients of the five, most influential parameters controlling the dynamics of the nuclear Smad2*/Smad4 complex which is our model output of interest.

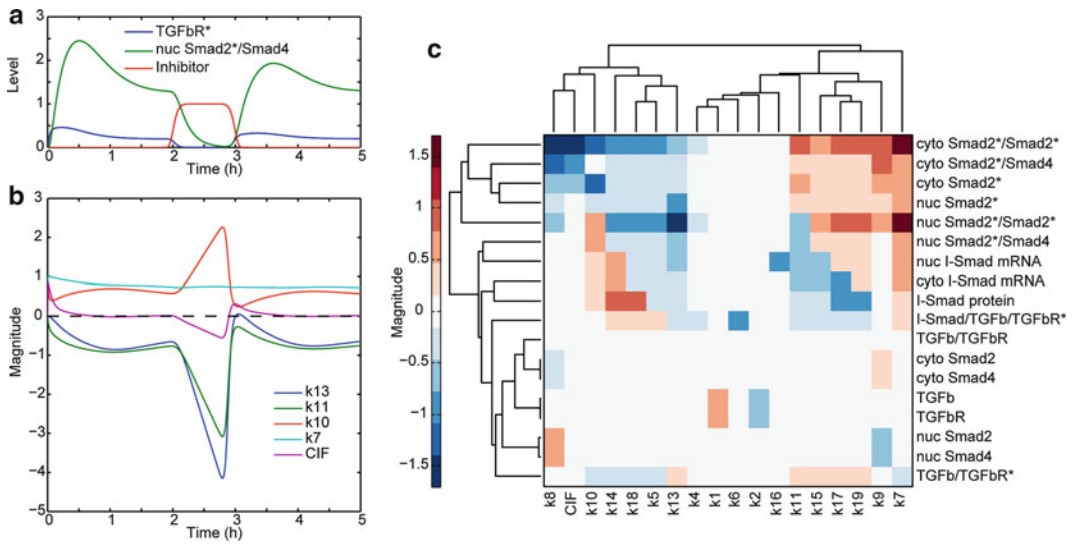

Fig. 2. Sensitivity analysis of the TGF-signaling model. (**a**) Time course of the stimulation protocol and the system output, nuclear Smad2 * /Smad4 complex. (**b**) Time-resolved control coefficients of parameters controlling the dynamics of the nuclear Smad2 * /Smad4 complex, i.e., the output signal of the signaling network. For details, see main text. (**c**) Clustergram of the steady-state control coefficients.

The time-dependent control coefficients are normalized sensitivities defined as

$$C^{m_i}_{p_j}(t) = \frac{p_j}{[m_i](t)} S^{m_i}_{p_j}(t), \tag{14}$$

where $[m_i](t)$ denotes the time-dependent concentration of the ith model output. The control coefficients of parameter k_7 (Smad2 phosphorylation rate) and k_{10} (Smad compex formation rate) are positive during the whole time course. Increasing these parameter will always lead to an elevation in the level of nuclear Smad2* /Smad4. On the contrary, k_{11} (complex dissociation rate) and k_{13} (Smad2 dephosphorylation rate) have mostly a negative effect with the strongest impact during the application of the TGFβ phosphorylation inhibitor. Note that the impact of a change in the complex import factor (CIF) can have a positive or a negative effect on the nuclear Smad2* /Smad4 concentration depending on the time point of a change. The remaining model parameters have a lower, but nevertheless nonzero impact on the nuclear Smad2* /Smad4 levels. In general, we expect parameters with small overall control coefficients to be difficult to identify. Parameters with strong correlations in the temporal profile of their control coefficients point to a strong underlying functional link and will also be correlated when estimated from data.

Next, we focus on the sensitivity of the steady-state levels with respect to all model parameters, again in terms of control coefficients. Figure 2c shows a clustergram of the steady-state control

coefficients in which rows represent observables and columns represent parameters. Rows and columns of the clustergram are sorted such as to maximize the similarity between neighboring row- and column vectors. In this way, observables and parameters with similar control coefficients (and thus similar function) across their respective dimension are grouped closely to each other. Figure 2c highlights two groups of control coefficients which are all related to the phosphorylated observables. They show either a decrease (upper-left, blue cluster) or an increase (upper-right, red cluster) in the concentration of the phosphoforms. The latter set of parameters include the Smad phosporylation rate and the I-Smad degradation rate (k_{19}) while the former include the dephosphorylation rate of nuclear Smad and the I-Smad association rate to the active receptor (k_5). Some parameters (k_1, k_6, and k_{16}) have very low overall control coefficients, and we anticipate problems in estimating these parameters from steady-state data. In conclusion, sensitivity analysis can provide a first glimpse to the functioning of a complex model and allows to group reactions and parameters with similar impact on the system output into functional groups.

4.2. Parameter Estimation

We now turn to the inverse problem: the estimation of model parameters from experimental data. The interested reader can reiterate our results with the MATLAB script script_fit.m provided on the authors web site. To this end, we simulate data, including measurement errors, in order to estimate the original parameter values used for model simulation. This allows us to evaluate the performance of the fitting procedure as we know the true parameter values underlying our data. We apply the same stimulation protocol as in Fig. 2a and estimating all 19 model parameters given that all model species except the two mRNA species of I-Smad are observed. The simulated data includes 20 data points per species and 10% measurement error. Figure 3a shows a fit of the active receptor and nuclear Smad2* /Smad4 trajectories to a simulated data set ($\chi^2_{342} = 365.2$, $\Pr[X \geq \chi^2_{342}] = 0.19$). The color-shaded area underlying each trajectory is the approximated uncertainty of a trajectory based on the uncertainty in the parameter estimates. It is calculated by error propagation as

$$\mathrm{Cov}[y(t)] = S_p^y(t) C S_p^y(t)^T, \tag{15}$$

where S_p^y is the matrix of the sensitivities of the observations defined by Eq. 1 and C is the parameter covariance matrix determined by Eq. 12. Correlations between the parameter estimates are shown in panel **b**. It is apparent that all parameter related with nuclear import/export and complex formation/dissociation are highly correlated. Additionally, parameters related to the TGFβ ligand/receptor interaction show a large positive correlation. Since this interaction happens on a fast timescale due to a high TGFβ receptor affinity the parameters cannot be well identified with the given temporal resolution of the data. Additionally, all

parameters related to the I-Smad expression are highly correlated. This is in part expected as the I-Smad mRNA states are not observed and therefore parameters related to the mRNA dynamics cannot be inferred from the data. Note that a strong correlation does not necessarily imply a bad identification as the coefficient of variation for the single parameters can still be very small. We will focus on these intrinsic variations in the next paragraph.

Instead of determining correlations by a single fit as in Fig. 3a, we next investigated parameter uncertainties and correlations arising from many fits to data sets that only differ in the noise

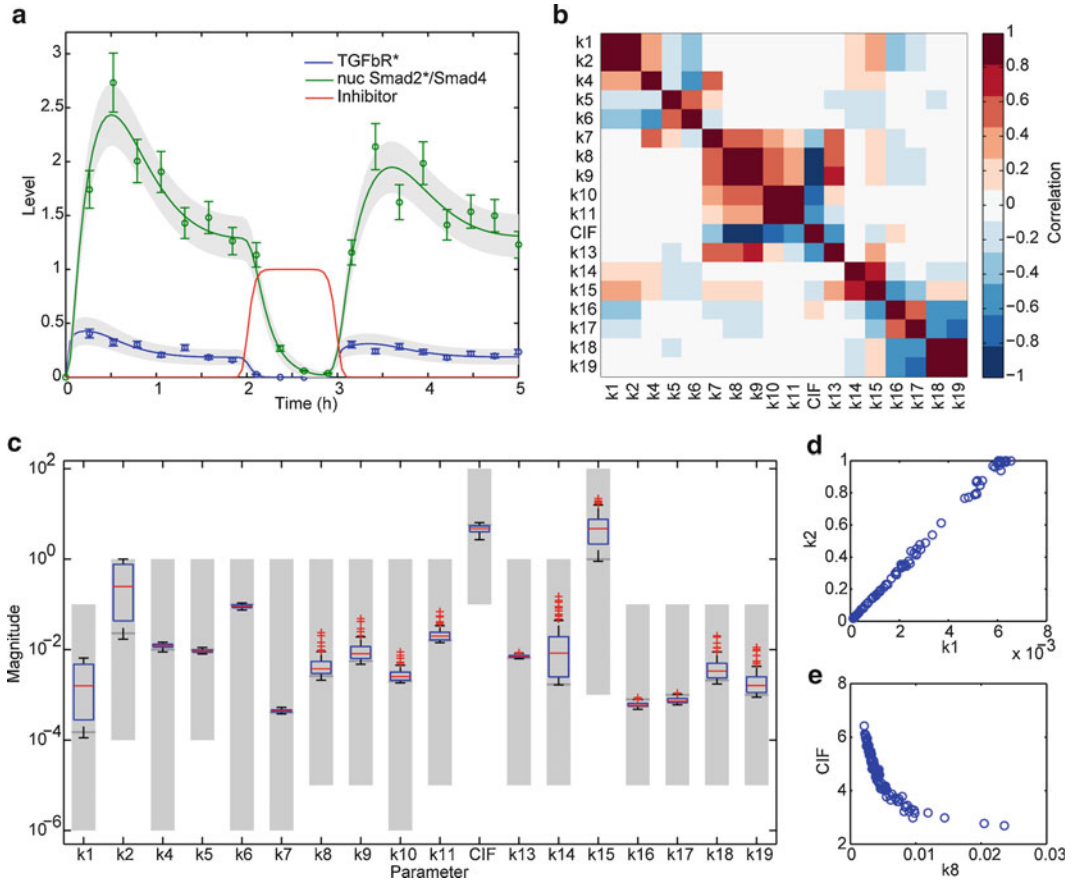

Fig. 3. Parameter estimation in the TGFβ model. The model was fitted to 16 observations (all model species except the two mRNA species of I-Smad; 20 data points each, 10% measurement error) optimizing 19 parameters. (**a**–**b**) Fit to one representative data set. (**a**) Data and time courses of phosphorylated TGF receptor complex and nuclear, phosphorylated Smad2/Smad4. Error bars denote data mean and standard deviation. *Gray-shaded areas* highlight error intervals for the trajectories calculated by error propagation (Eq. 15). (**b**) Correlation matrix of parameter estimates from fit in (**a**). (**c**–**e**) Evaluation of the expected variance in parameter estimates. Parameters were fitted to 100 data sets, each with a different realization of the measurement noise. (**c**) Box plots of estimated parameter sets (median=*red line*, 25% and 75% quantiles=*blue lines*, 5% and 95% quantiles=*black lines*, *red asterisk*=outliers). Parameter-fitting ranges are indicated in *light gray*, true parameter values are given in *dark gray*. (**d**,**e**) Scatter plots revealing a functional relation between the estimates for k_1 and k_2 (**d**) and k_8 and CIF (**e**).

realization. The corresponding MATLAB script is script_multi_fit.m. This strategy is similar to a bootstrap in case of a sufficiently high number of replicates per measurement point and reflects more accurately the expected parameter uncertainties than a single fit does. It also allows to clarify the expected accuracy in parameter estimates given the experimental design and measurement error. Figure 3c shows boxplots of the optimized parameter estimates from 100 data sets. Parameter limits used for the estimation are indicated as light gray bars. Some parameter estimates have a small variance and nicely fit the true parameter values indicated as dark gray bars. Some parameters can be sufficiently well identified despite having considerable correlations, e.g., k_7 and k_{13}. However, most parameter estimates have a large variance and do not match the true parameter on average. In fact, a closer look at the parameter distributions (e.g., by means of (9)) reveals a strong functional dependence with other parameters, meaning that they cannot be identified under the given experimental setup. Figure 3d, e shows two representative scatter plots of pairs of parameter estimates which are functionally related. The hyperbolic relation between the estimates of k_8 and CIF is apparent from the model formulation, as CIF is only a scaling factor for the import rate of the complexes. For other functional relations, the underlying mechanism is less clear. Most parameters related to I-Smad expression, which is only observed on the protein level, cannot be well identified. Generally, unobserved processes will corrupt the identifiablity of the model parameters. It should be noted that nonidentifiability due to a lack of sensitivity, which is sometimes interpreted as systems robustness, does not imply a lack in functionality of the respective parameters or processes. Sensitivities are always dependent on the particular way of defining and measuring a system output. A focussed, and necessarily limited, investigation cannot in general assess functionality in a larger context. This highlights the need for appropriate experiments and mathematical models which are designed for a particular question.

5. Conclusion

Parameter estimation from experimental data is a central part of modeling and analyzing biological signaling networks. It is, in general, an iterative process, and robust and efficient algorithms are key to obtain good estimates with reasonable computational effort. This chapter summarizes the main steps to attain these goals. The optimization problem is formulated in terms of a likelihood function and a gradient-based minimization algorithm is suggested to determine the parameter set that maximizes the likelihood. It is

important to subsequently analyze the GOF and to derive confidence intervals for the parameter estimates. Approximate symmetric confidence intervals can be formulated in terms of the variance in the data and parameter sensitivities. More accurate estimates can be obtained with computationally more demanding Bootstrap methods. Before and after the regression the identifiability of parameters should be analyzed and the number of parameters in the model should either be reduced accordingly or parameters should be excluded from the estimation. The further analysis of the model should, in any case, consider all admissible parameter sets and cluster these according to the predictions of the model.

Acknowledgments

We thank Andreas Raue and Martin Peifer for discussions and members of the Iber group for the critical reading of the manuscript. The work was partially funded through a SystemsX iPhD scholarship to G.F. and a SystemsX grant as part of the RTD InfectX.

References

1. Jaqaman K, Danuser G (2006) Linking data to models: data regression. Nat Rev Mol Cell Biol 7(11):813–819
2. Kreutz C, Timmer J (2009) Systems biology: experimental design. FEBS J 276(4):923–942
3. Kreutz C, Bartolome Rodriguez MM, Maiwald T, Seidl M, Blum HE, Mohr L, Timmer J (2007) An error model for protein quantification. Bioinformatics 23(20):2747–2753
4. Silvey SD (1970) Statistical inference. Penguin Books, Baltimore
5. Press WH (2007) Numerical recipes: The art of scientific computing, 3rd edn. Cambridge University Press, New York
6. Peifer M, Timmer J (2007) Parameter estimation in ordinary differential equations for biochemical processes using the method of multiple shooting. IET Syst Biol 1(2):78–88
7. Raue A, Kreutz C, Maiwald T, Bachmann J, Schilling M, Klingmuller U, Timmer J (2009) Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. Bioinformatics 25(15):1923–1929
8. Efron B, Tibshirani R (1986) Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. Statist Sci 1(1):54–75
9. Hengl S, Kreutz C, Timmer J, and Maiwald T (2007) Data-based identifiability analysis of non-linear dynamical models. Bioinformatics 23(19):2612–2618
10. Murray JD (2002) Mathematical biology: I. an introduction, 3rd edn. Springer, Berlin
11. Schmierer B, Tournier AL, Bates PA, Hill CS (2008) Mathematical modeling identifies smad nucleocytoplasmic shuttling as a dynamic signal-interpreting system. Proc Natl Acad Sci USA 105(18):6608–6613
12. Clarke DC, Liu X (2008) Decoding the quantitative nature of tgf-beta/smad signaling. Trends Cell Biol 18(9):430–442

http://www.springer.com/978-1-61779-832-0

Computational Modeling of Signaling Networks
(Eds.)X. Liu; M.D. Betterton
2012, XIV, 327 p. 75 illus., 10 in color., Hardcover
ISBN: 978-1-61779-832-0
A product of Humana Press