the FEBS Journal

MINIREVIEW

Systems biology: parameter estimation for biochemical models

Maksat Ashyraliyev[1], Yves Fomekong-Nanfack[2], Jaap A. Kaandorp[2] and Joke G. Blom[1]

1 Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands
2 Section Computational Science, University of Amsterdam, The Netherlands

Keywords
a prioiri and *a posteriori* identifiability; local and global optimization; parameter estimation

Correspondence
J. G. Blom, Centrum voor Wiskunde en Informatica, Science Park 123, 1098 XG Amsterdam, The Netherlands
Fax: +31 20 5924199
Tel: +31 20 5924263
E-mail: joke.blom@cwi.nl

(Received 8 April 2008, revised 21 October 2008, accepted 28 November 2008)

doi:10.1111/j.1742-4658.2008.06844.x

Mathematical models of biological processes have various applications: to assist in understanding the functioning of a system, to simulate experiments before actually performing them, to study situations that cannot be dealt with experimentally, etc. Some parameters in the model can be directly obtained from experiments or from the literature. Others have to be inferred by comparing model results to experiments. In this minireview, we discuss the identifiability of models, both intrinsic to the model and taking into account the available data. Furthermore, we give an overview of the most frequently used approaches to search the parameter space.

Introduction

Parameter estimation in systems biology is usually part of an iterative process to develop data-driven models for biological systems that should have predictive value. In this minireview, we discuss how to obtain parameters for mathematical models by data fitting. We restrict ourselves to the case where a deterministic model in the form of a mathematical function-based model is available, such as a system of differential and algebraic equations. For example, in the case of a biochemical process, hypotheses based on the knowledge of the underlying network structure of a pathway are translated into a system of kinetic equations, parameters are obtained from literature or estimated from a data fit, and, with the resulting model, predictions are made that can be tested with further experiments. To compare model results with the experimental data, one first has to simulate the mathematical model to produce these results, the forward problem. The inverse problem is the problem at hand: the estimation of parameters in a mathematical model from measured observations. There are a number of difficulties involved [6]. The forward problem requires a fast and robust time integrator. Fast, because the model will be evaluated many times. Robust, because the whole parameter and state space will be visited, which most likely will result in a different character of the mathematical model (i.e. number and range of time scales involved). The inverse problem has even more pitfalls. The first question is whether the parameters for the mathematical model can be determined assuming that for all observables continuous and error-free data are available. This is the subject of *a priori* identifiability or structural identifiability analysis of the mathematical model. The actual parameter estimation or data fitting typically starts with a guess about parameter values and then changes those values to minimize the discrepancy between

Abbreviations
DAE, differential algebraic equations; EA, evolutionary algorithms; MLE, maximum likelihood estimator; SA, simulated annealing; SRES, stochastic ranking evolutionary strategy.

model and data using a particular metric. Kinetic models with nonlinear rate equations have in general multiple sets of parameters that lead to such minimizations, some of those minima may only be local. The value of parameters and model variables may range over many orders of magnitude, one can get stuck in a local minimum or one can wander around in a very flat part of the solution space. Given a particular set of experimental data, and one particular acceptable model parameterization obtained by a parameter estimation procedure, does not mean that all obtained parameters can be trusted. After the minimum has been found, an *a posteriori* or practical identifiability study can show how well the parameter vector has been determined given a data set that is possibly sparse and noisy. That this part of model fitting should not be underestimated is shown by Gutenkunst *et al.* [7]. For all 17 systems biology models that they considered, the obtained parameters are 'sloppy', meaning not well-defined. On the other hand, one could argue that often the precise value of a parameter is not required to draw biological conclusions [8].

In this minireview, we first discuss the identifiability of the model, both *a priori* and *a posteriori*, the latter by a small example. Next, we give a brief survey of the current methods used in parameter estimation with a focus on those that are implemented in toolboxes for systems biology. In the Discussion, we give some guidelines on the application of these methods in practice. Finally, in the supporting information (Doc. S1), an overview is given of the contents of some well-known toolboxes.

For further reading on identifiability, we refer to the classical textbook of Ljung [9] and the recent review paper on regression by Jaqaman and Danuser [4]. An overview on local and global parameter estimation methods applied to a systems biology benchmark set is given elsewhere [10,11]. We also recommend the easily readable books on this subject by Schittkowski [6] and by Aster *et al.* [12], which touch many of the subjects discussed in this minireview, with the exception of global search methods.

Problem definition

Deterministic models arising from kinetic equations are typically given by a system of differential algebraic equations (DAEs)[1] (i.e. ordinary differential equations coupled to algebraic equations) of the form:

$$\begin{cases} A\dfrac{\mathrm{d}\mathbf{x}(t,\mathbf{p})}{\mathrm{d}t} = \mathbf{f}(t,\mathbf{x}(t,\mathbf{p}),\mathbf{p},\mathbf{u}(t)), & t_0 < t \le t_e \\ \mathbf{x}(t_0,\mathbf{p}) = \mathbf{x}_0(\mathbf{p}) \end{cases} \qquad (1)$$

where t denotes time, the m-dimensional vector $\mathbf{p}$ contains all unknown parameters, $\mathbf{x}$ is an n-dimensional vector with the state variables (e.g. concentration values), $\mathbf{u}$ are the externally input signals, and $\mathbf{f}$ is a given vector function. When components of the initial state vector $\mathbf{x}_0$ are not known, they are considered as unknown parameters, so $\mathbf{x}_0$ may depend on $\mathbf{p}$. In most cases, A is a constant diagonal $n \times n$ matrix with 1 or 0 on the diagonal; 1 for an ODE and 0 for an algebraic equation.

In addition, a vector of observables is given:

$$\mathbf{g}(t,\mathbf{x}(t,\mathbf{p}),\mathbf{p},\mathbf{u}(t)) \qquad (2)$$

which are quantities in the model [in general (a combination of) state variables] that can be experimentally measured, and possibly a vector of (non)linear constraints:

$$\mathbf{c}(t,\mathbf{x}(t,\mathbf{p}),\mathbf{p},\mathbf{u}(t)) \ge 0 \qquad (3)$$

Let us assume that N measurements are available to find parameters of Eqns (1–3). Each measurement, which we denote by y_i, is specified by the time t_i when the ith component of the observable vector $\mathbf{g}$ is measured. The corresponding model value for a specific parameter vector $\hat{\mathbf{p}}$,which can be obtained sufficiently accurate by numerical integration of Eqn (1) and computing the observable function of Eqn (2), is denoted by $\hat{g}_i = g_i(t_i,\mathbf{x},\hat{\mathbf{p}},\mathbf{u})$. The vector of discrepancies between the model values and the experimental values is then given by $\mathbf{e}(\hat{\mathbf{p}}) = |\mathbf{g}(t,\mathbf{x}(t,\hat{\mathbf{p}}),\hat{\mathbf{p}},\mathbf{u}(t)) - \mathbf{y}|$. We assume that Eqn (1) is a sufficiently accurate mathematical description approximating reality. This means that all relevant knowledge about the biological processes is incorporated correctly in the vector function $\mathbf{f}$. Thus, the only uncertainty in Eqn (1) is the vector of unknown parameters $\mathbf{p}$. In this case, the difference $e_i(\mathbf{p}^*) = |g_i(t_i,\mathbf{x},\mathbf{p}^*,\mathbf{u}) - y_i|$ is solely due to experimental errors, where $\mathbf{p}^*$ is the true solution.

The m-dimensional optimization problem is given by the task to minimize some measure, $V(\mathbf{p})$, for the discrepancy $\mathbf{e}(\mathbf{p})$. By far the most used measure for the discrepancy is the Euclidean norm or the sum of the squares weighted with the error in the measurement:

$$V_{MLE}(\mathbf{p}) = \sum_{i=1}^{N} \frac{(g_i(t_i,\mathbf{x},\mathbf{p},\mathbf{u}) - y_i)^2}{\sigma_i^2} = \mathbf{e}^T(\mathbf{p})W\mathbf{e}(\mathbf{p}) \qquad (4)$$

see [13,14]. This measure results from the maximum likelihood estimator (MLE) theory. Under the assump-

[1] The content of this paper is also applicable to (discretized) systems of partial differential equations and delay differential equations. Fitting parameters of stochastic models requires a different approach [1–3].

tion that the experimental errors are independent and normally distributed with standard deviation σ_i, the least squares estimate $\hat{\mathbf{p}}$ of the parameters is the value of $\mathbf{p}$ that minimizes the sum of squares:

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} V_{MLE}(\mathbf{p}) \tag{5}$$

When these assumptions do not hold, other measures than $V_{MLE}(\mathbf{p})$ might be used like the sum of the absolute values. The MLE theory then does not apply so $\hat{\mathbf{p}}$ is not the least squares estimate and the statistical analysis in the section '*A Posteriori* identifiability' does not hold. Dependent on the optimization method or the mathematical discipline the function $V(\mathbf{p})$ is called objective function, cost function, goal function, energy function or fitness function.

Identifiability

Whether the inverse problem is solvable is dependent on (a) the mathematical model; (b) the significance of the data; and (c) the experimental errors. In the following, we assume that the model is properly scaled such that both the parameter values and the state variables are of the same order of magnitude. Otherwise, a proper scaling should be applied to the model.

Definitions

The sensitivity matrix J of the model is given by the sensitivity coefficients of the observables with respect to the parameters:

$$J = \left(\frac{\partial g_i(\mathbf{p})}{\partial p_j}\right) \tag{6}$$

A parameter is globally identifiable if it can be uniquely determined given the input profile $\mathbf{u}(t)$ and assuming continuous and error-free data for the observables of the model. If there is a countable number of solutions the parameter is locally identifiable; it is unidentifiable if there exist uncountable many solutions. A model is structurally globally/locally identifiable if all its parameters are globally/locally identifiable[2].

Practical or *a posteriori* identifiability analysis studies whether the parameters can be globally or locally determined with the available, noisy, experimental data. In this case, locally means in the neighborhood of the obtained parameter.

A priori identifiability

There are several techniques to determine *a priori* global identifiability of the model, but for realistic situations (i.e. nonlinear models of a certain size), it is very difficult to obtain any results. Still, it is advisable to always perform an *a priori* analysis because parameter estimation methods can have problems with locally identifiable or unidentifiable systems. Symbolic algebra packages like MAPLE [15] and MATHEMATICA [16] can be of great help.

For linear models, the Laplace transform or transfer function approach can be applied. For nonlinear models, the oldest method and most simple to understand is the Taylor or power series expansion [17]. The observable function is expanded in a Taylor series at a particular time point. The time derivatives are evaluated in terms of the parameters, resulting in a system of nonlinear equations for the parameters. If this system has a unique solution, the model is structurally identifiable. For simple examples using the Laplace transform (linear model) and Taylor series (Michaelis–Menten kinetics), we refer to Godfrey and Fitch [18]. Another classic method is the similarity transformation approach [19–21]. These two methods have been compared without a decisive preference [22]. Recently, methods were developed that use differential algebra techniques [23]. Also, a publicly available software tool, DAISY [24], is available that can check the identifiability of a nonlinear system. DAISY is implemented in the symbolic language REDUCE [25].

A posteriori identifiability

The difficulty in estimating the parameters in a quantitative mathematical model is not so much how to compute them, but more how to assess the quality of the obtained parameters because this not only depends on how well the model describes the phenomenon studied and on the existence of a unique set of parameters, but also on whether the experimental data are sufficient in number, sufficiently significant and sufficiently accurate. With respect to the first two requirements, a sufficient and significant amount of data, it is clear that, whatever method one uses to fit a model with experimental data: to estimate m unknown parameters, one needs at least m experimental values. On the other hand, it is not necessary to have experimental data for all state variables involved in the model at all possible

[2] Note that these definitions are not always the same. Other definitions are: A model is structurally identifiable if its sensitivity matrix satisfies two conditions: each column has at least one large entry and the matrix has full rank [4]. A model is locally identifiable if it is globally identifiable in a neigborhood of the parameter [5].

time points, often only a few measurements for the right observable at significant times are needed. The last question, sufficiently accurate data, is related to the fact that measurement errors imply that we do not have precise data points to fit our model with, but that each point represents a whole cloud of possible data values, implying also that the inferred parameters are not point-values but are contained in a cloud. Depending on the model, the cloud of possible parameter values varies in size and shape and can be much larger than the original uncertainty in the data.

The most applied method [12,26] to study this uncertainty in the parameters is to compute the sensitivity matrix J of Eqn (6) evaluated for the given data points and the parameter vector $\hat{\mathbf{p}}$ obtained by the data fit. This can be done either by finite differencing or by solving the variational equations[3]. Note that this is a linear analysis, and local both with respect to $\hat{\mathbf{p}}$ and to the given data points.

In the following, we assume that the measurement errors are independent of each other and normally distributed with the same standard deviation σ[4]. Then $\hat{\mathbf{p}} - \mathbf{p}^*$ approximately has an m-dimensional multivariate normal distribution with mean zero and variance $\sigma^2(J^{\mathrm{T}}(\hat{\mathbf{p}})J(\hat{\mathbf{p}}))^{-1}$. How 'close' the estimate $\hat{\mathbf{p}}$ is to the true parameter vector $\mathbf{p}^*$ is expressed by the (1−α)-confidence region for $\mathbf{p}^*$, given by:

$$(\mathbf{p}^* - \hat{\mathbf{p}})^{\mathrm{T}}\left(J^{\mathrm{T}}(\hat{\mathbf{p}})J(\hat{\mathbf{p}})\right)(\mathbf{p}^* - \hat{\mathbf{p}}) \le C(\alpha) \tag{7}$$

with:

$$C(\alpha) = \frac{m}{N-m} V_{MLE}(\hat{\mathbf{p}}) F_\alpha(m, N-m) \tag{8}$$

where $F_\alpha(m,N{-}m)$ is the upper α part of Fisher's distribution with m and $N - m$ degrees of freedom. Note that $V_{MLE}(\hat{\mathbf{p}})/(N-m)$ is an unbiased estimator of the measurement variance σ^2. The (1−α)-confidence region implies that there is a probability of 1−α that the true parameter vector $\mathbf{p}^*$ lies in this ellipsoid that is centered at $\hat{\mathbf{p}}$ and has its principal axes directed along the eigenvectors of $J^{\mathrm{T}}(\hat{\mathbf{p}})J(\hat{\mathbf{p}})$. Using the singular value decomposition for $J(\hat{\mathbf{p}}) = \mathcal{U}\Sigma\mathcal{V}^{\mathrm{T}}$, we get $J^{\mathrm{T}}(\hat{\mathbf{p}})J(\hat{\mathbf{p}}) = \mathcal{V}(\hat{\mathbf{p}})\Sigma^2(\hat{\mathbf{p}})\mathcal{V}^{\mathrm{T}}(\hat{\mathbf{p}})$, where the eigenvectors of $J^{\mathrm{T}}(\hat{\mathbf{p}})J(\hat{\mathbf{p}})$ are the columns of the matrix $\mathcal{V}(\hat{\mathbf{p}})$. So, the principal axes of the ellipsoidal confidence region are given by the singular vectors, the column vectors of the matrix $\mathcal{V}(\hat{\mathbf{p}})$, and the length of the principal axes is proportional to the reciprocal of the corresponding singular values, the diagonal elements of $\Sigma(\hat{\mathbf{p}})$. Using the transformation (rotation):

$$\mathbf{z} = \mathcal{V}^{\mathrm{T}}(\hat{\mathbf{p}})(\mathbf{p}^* - \hat{\mathbf{p}}) \tag{9}$$

the equation for the ellipsoid (7) can be rewritten as:

$$\sum_{i=1}^{m} \sigma_i^2 z_i^2 = C(\alpha) \tag{10}$$

Note that $C(\alpha)$ is approximately proportional to the variance in the measurement errors. The precise definition of 'practical identifiable' depends on the level of accuracy, r_ε, one requires for the parameter estimates. This defines the sphere:

$$\sum_{i=1}^{m} z_i^2 = r_\epsilon^2 \tag{11}$$

To be able to determine z_i accurately enough, the radius along the ellipsoid's ith principal axis should not exceed the radius of the sphere, which leads to the following inequality:

$$\sigma_i \ge \frac{\sqrt{C(\alpha)}}{r_\epsilon} \tag{12}$$

A graphical representation of the ellipsoid and the sphere is given in Fig. 1 for the 2D case. Suppose that only the first k largest singular values satisfy (12), then only the first k entries of $\mathbf{z}$ are estimated with the required accuracy. If a principal axis of the ellipsoid makes a significant angle with the axis in parameter space (i.e. there exists more than one significant entry in the eigenvector), this corresponds to the presence of

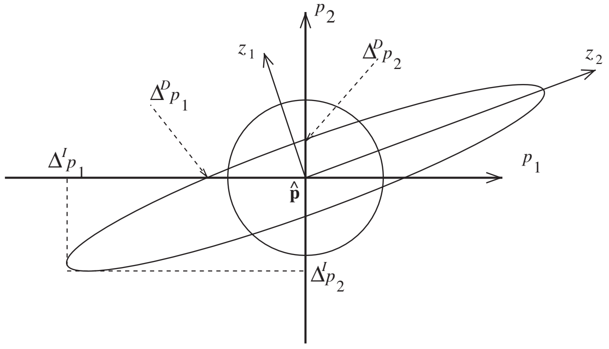

Fig. 1. Example of an ellipsoidal confidence region and an accuracy sphere in the 2D case; parameters p_1 and p_2 are correlated, the linear combination z_1 is well-determined, whereas z_2 is not. The dependent confidence interval, $\Delta^D p_i$, for a parameter is given by the intersection of the ellipsoid with the parameter axis; the independent confidence interval, $\Delta^I p_i$, by the projection onto the axis.

[3] Variational equations are obtained by taking the derivative of the DAE system (1) with respect to the parameters. This results in m DAE systems in the variables $\partial\mathbf{x}(t,\mathbf{p})/\partial p_i$, $i = 1,\ldots,m$.

[4] The assumption that all measurements have the same variance is not required but it makes the formulation easier.

correlation among parameters in $\hat{\mathbf{p}}$. In this case, only a combination of parameters can be determined.

To summarize, the level of noise in the data, in combination with the accuracy requirement for the parameter estimates, defines the threshold for significant singular values in the matrix Σ. The number of singular values exceeding this threshold determines the number of parameter relations that can be derived from the experiment. How these relations relate to the individual parameters is described by the corresponding columns in the matrix V.

It is obvious that inspecting the ellipsoidal region is not possible for high-dimensional problems. But based on the sensitivity matrix J or rather on the Fisher information matrix $J^{\mathrm{T}}J$, there are a number of easy-to-compute indicators. Assuming that all other parameters are exact, a confidence interval for a specific parameter is the intersection of the ellipsoidal region with the parameter axis. This is the dependent confidence interval:

$$\Delta^D p_i = C(\alpha) / \sqrt{\left(J^{\mathrm{T}}(\hat{\mathbf{p}})J(\hat{\mathbf{p}})\right)_{ii}} \tag{13}$$

The independent confidence interval is given by the projection of the ellipsoidal region onto the parameter axis:

$$\Delta^I p_i = C(\alpha) \sqrt{\left(\left(J^{\mathrm{T}}(\hat{\mathbf{p}})J(\hat{\mathbf{p}})\right)^{-1}\right)_{ii}} \tag{14}$$

If dependent and independent confidence intervals are similar and small, $\hat{p_i}$ is well-determined. In case of a strong correlation between parameters, the dependent confidence intervals underestimate the confidence region, whereas the independent confidence intervals overestimate it. Another way to obtain information about the correlations between parameters is to look at the covariance matrix $\mathrm{cov} = (J^{\mathrm{T}}J)^{-1}$. The correlation coefficient of the *i*th and *j*th parameter is given by:

$$\mathrm{cor}_{ij} = \frac{\mathrm{cov}_{ij}}{\sqrt{\mathrm{cov}_{ii}\mathrm{cov}_{jj}}} \tag{15}$$

Finally, Eqn (10) indicates that having, for example, two times more accurate data so that the standard deviation σ is halved will decrease the radii along the ellipsoid's principal axes by a factor of 2. Therefore, in case of very small singular values σ_i (i.e. strongly elongated ellipsoids), more accurate data obtained by the experimentalist will not improve much the quality of the corresponding parameter estimates. In such a case, one certainly needs additional measurements of a different type (e.g. different components, different time points, or in the case of partial differential equations, different spatial points).

Other approaches

Hengl *et al.* [27] propose a nonlinear analysis: repeated fitting for different initial guesses of the parameter vector. The resulting parameter vector matrix is analyzed with Alternating Conditional Expectation [28], resulting in optimal transformations for the parameters to come to an identifiable model. This approach is implemented in MATLAB [29]/PottersWheel [30].

Finally, we want to mention an interesting idea described [31,32] regarding the cluster-based parameter estimation. This approach uses the sensitivity matrix to define subsets of state variables that depend on a subset of the parameters. The parameters are then split into two classes: global if state variables from more than one cluster depend on it and local otherwise. A hierarchical parameter estimation is performed reducing the dimensionality. On the high level, the global parameters are fitted by optimization of the clusters and, recursively, parameters in each cluster are estimated.

Example insignificant data

On the basis of a very simple artificial example [33,34], we show the influence of the experimental data on the parameter determinability.

Consider the simple enzymatic reaction:

$$\begin{aligned} \mathrm{E} + \mathrm{S} &\underset{k_2}{\overset{k_1}{\rightleftharpoons}} \mathrm{C} \\ \mathrm{C} &\xrightarrow{k_3} \mathrm{E} + \mathrm{P} \end{aligned} \tag{16}$$

with as state variables the concentrations of the substrate [S], the enzyme [E], and complex [C]. The product P is not part of the model but could easily be added. The mathematical model, a DAE-system, is then given by:

$$\begin{aligned} \frac{\mathrm{d}[S]}{\mathrm{d}t} &= -k_1[\mathrm{E}][\mathrm{S}] + k_2[\mathrm{C}] \\ \frac{\mathrm{d}[\mathrm{C}]}{\mathrm{d}t} &= k_1[\mathrm{E}][\mathrm{S}] - k_2[\mathrm{C}] - k_3[\mathrm{C}] \\ [\mathrm{E}] + [\mathrm{C}] &= [\mathrm{E}_0] + [\mathrm{C}_0] \end{aligned} \tag{17}$$

Suppose the initial concentration of the state variables, $[\mathrm{S}_0]$, $[\mathrm{E}_0]$ and $[\mathrm{C}_0]$ is known, and the concentration of [C] is measured rather precisely at regular time points $t = 1,\ldots,20$. For this example, the 'measurements' are generated artificially by adding an independent,

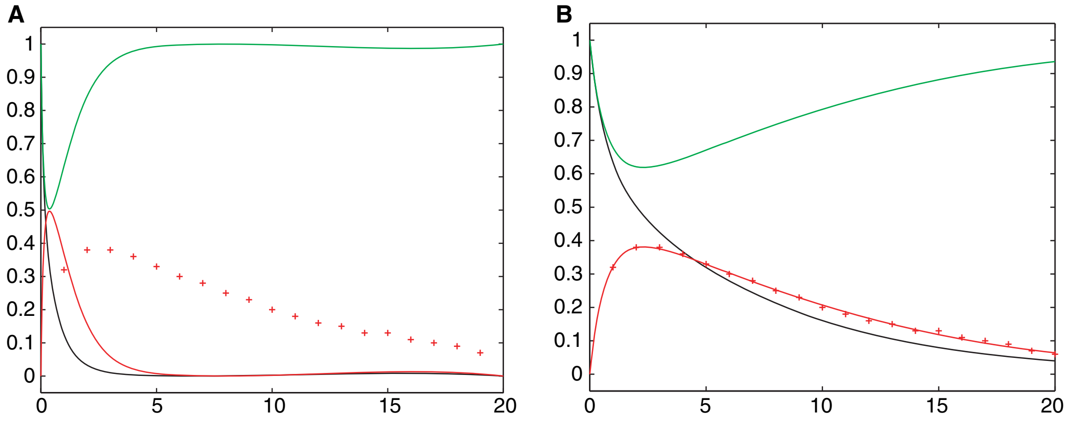

Fig. 2. Model results for initial (left) and final (right) parameter vector, black: [S], red: [C], green: [E]; and measurements of [C]: red +.

normally distributed perturbance with zero expectation and a fixed variance to the model results (red +-marks in Fig. 2). The initial parameter values are $\mathbf{p}_0 = (6,0.8,1.2)$. With these parameter values, the model results are given by the solid lines in the left plot in Fig. 2. Fitting the model to these measurements with the Levenberg–Marquardt method (see below) results in the parameter vector $\hat{\mathbf{p}} = (k_1, k_2, k_3) = (0.683, 0.312, 0.212)$ (for the model results, see Fig. 2, right).

We define the discrepancy of the model with respect to the data:

$$\mathbf{e}(\mathbf{p}) = (c_i(t_i, \mathbf{p}) - \tilde{c}_i)_{i=1,\ldots,N} \qquad (18)$$

the vector of the differences between the ith data value, $\tilde{c}_i$, which is the measured concentration of [C] at time t_i, and the corresponding value from the model, c_i. In the present example, the sensitivity matrix J is an $N \times 3$ matrix, with $N = 20$. For this simple three-parameter problem, one can easily visualize the confidence region (Eqn 7) and we can see from the left plot in Fig. 3 that the true parameter vector lies in a small disc around $\hat{\mathbf{p}}$, implying that we can estimate all three parameters with a reasonable accuracy by measuring only the complex (or any of the two other concentrations in this case). With 95% confidence, all parameters can be estimated with one digit accuracy and k_3 even with two digits. Using only three well-chosen time-points for measuring ($t = 1,2,20$), the axes-length of the ellipsoid increases with a factor of about 4, but still all parameters can be determined reasonably well.

Suppose now that it is not possible to measure before time $t = 6$ but that we take 20 samples of the

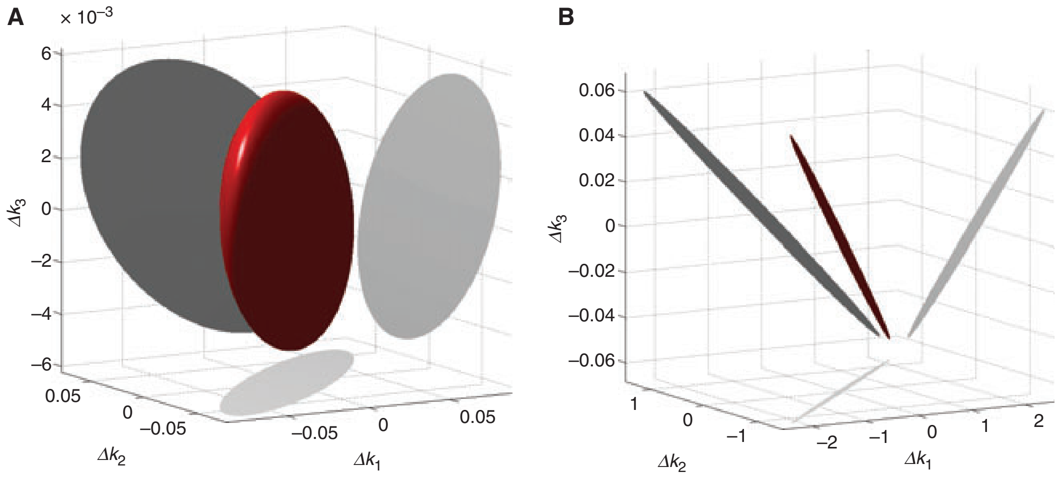

Fig. 3. Confidence region $\Delta\mathbf{k}$ (cf. Eqn 7) in parameter space around computed parameter vector (origin in the plots) and its projection on the parameter-planes. The region contains the true parameter vector with a 95% probability. Left: 20 measurements at $t = 1,\ldots,20$; right: 20 measurements at time points distributed uniformly over [6,20].

complex at regular times from $t = 6,\ldots,20$. Suppose also that the same parameter vector $\hat{\mathbf{p}}$ results from minimizing the least squares error $\mathbf{e}^{\mathrm{T}}\mathbf{e}$. In this case, the confidence region gives much more reason for distrusting the result. As can be seen in Fig. 3 (right), the true parameter vector now lies in a long elongated 'cigar' and especially for k_1 and k_2 we can not even trust the order of magnitude.

Looking at the eigensystem, we see that, for the left ellipsoid in Fig. 3, the matrices V_6 and Σ are given by:

$$\mathcal{V} = \begin{pmatrix} -0.01 & 0.66 & -0.75 \\ 0.05 & -0.75 & -0.66 \\ 0.99 & 0.04 & 0.02 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 3.5 & 0 & 0 \\ 0 & 0.75 & 0 \\ 0 & 0 & 0.17 \end{pmatrix} \tag{19}$$

where rows one to three correspond to k_1, k_2 and k_3, respectively. From Σ, we can learn that the principal axis corresponding to the first column of V is the shortest and because this column is almost the unit vector for k_3, the shortest principal axis almost coincides with the k_3-axis. The second principal axis is approximately five times longer; moreover, the second column of V shows that this axis corresponds to a combination of k_1 and k_2, so we can determine the combination of k_1 and k_2 approximately five times worse than k_3. Individually, k_1 and k_2 lie inside an ellips for which the other axis is approximately 20 times longer than the k_3-axis. An upperbound for the error in k_1 and k_2 is then given by the projection of this ellips on the corresponding axis. (In general, one has to project the ellipsoid.)

The matrices V and Σ corresponding to the right-hand plot in Fig. 3 are given by:

$$\mathcal{V}_6 = \begin{pmatrix} -0.03 & 0.49 & 0.87 \\ -0.001 & -0.87 & 0.49 \\ -0.99 & -0.01 & -0.02 \end{pmatrix} \quad \Sigma_6 = \begin{pmatrix} 3.89 & 0 & 0 \\ 0 & 0.41 & 0 \\ 0 & 0 & 0.006 \end{pmatrix} \tag{20}$$

Comparing these matrices, corresponding to 20 measurements uniformly distributed in the time interval [6,20], with the matrices in Eqn (19), which correspond to 20 measurements at $t = 1,2,\ldots,20$, it is clear that k_3 still can be determined with good accuracy, and even the combination of k_1 and k_2 can be determined reasonably well, but the third principal axis of the ellipsoidal confidence region has increased almost by a factor of 30! This implies that it is no longer possible to determine k_1 and k_2 individually.

From the discussion above, it is clear that it is not easy to *a priori* give an indication whether experimental data are sufficient in number and sufficiently significant. With three 'lucky' data points, one can estimate three parameters, but 20 data points in a region where 'nothing happens' are not sufficient.

Next, we examine the influence of experimental noise, (i.e. whether the experimental data are sufficiently accurate). Because $C(\alpha)$ is proportional to the variance of the measurement error distribution, the principal axes of the ellipsoidal confidence region are proportional to the standard deviation. Roughly speaking: reducing the (standard deviation of the) error by a factor of two, implies that a parameter, or combination of parameters, can be determined more accurately by a factor of two. This means that to shrink the ellipsoidal confidence region for the $t > 6$ experiment (Fig. 3, right) such that it fits into an 'accuracy'-sphere that is equal to the experiment with measurements between 1 and 20, one has to reduce the variance of the experimental error beyond reasonable experimental accuracy.

Finally, if we just look at the computable information from the Fisher matrix we get for the confidence intervals:

Exp.	$\Delta^D(k_1)$	$\Delta^D(k_2)$	$\Delta^D(k_3)$	$\Delta^I(k_1)$	$\Delta^I(k_2)$	$\Delta^I(k_3)$
[1,20]	0.033	0.028	0.005	0.076	0.067	0.005
[6,20]	0.074	0.047	0.004	2.217	1.267	0.060

The correlation matrices for the two cases are:

$$R_{20} = \begin{pmatrix} 1 & 0.9 & -0.37 \\ -0.9 & 1 & -0.45 \\ -0.37 & -0.45 & 1 \end{pmatrix}$$

$$R_6 = \begin{pmatrix} 1 & 0.999 & -0.997 \\ 0.999 & 1 & -0.996 \\ -0.997 & -0.996 & 1 \end{pmatrix} \tag{21}$$

Also, this simple to compute information shows that, for the second case, the parameters are strongly correlated and the model is not identifiable.

Parameter estimation methods

To find the minimum of the objective function optimization methods are used. We describe here two classes: local and global. Local search methods typically converge fast to a minimum, but, as the name suggests, this might be a local minimum and the method has no possibility to escape from this minimum to find the true or global minimum. For local search methods, there is in general a theoretical proof of convergence (and of convergence speed) to the minimum if the initial guess is sufficiently close to that minimum.

Global optimization searches all over the parameter space to find smaller and smaller values for the objective function, but in general there is no proof for convergence to the minimum (with exception of the simulated annealing algorithm).

Various numerical algorithms exist for global and local optimization. A number of global and local methods have been applied to a benchmark of biochemical pathway [10,11]. Below, we describe briefly the methods that are frequently used when estimating model parameters of biological problems and the methods that are available in general toolboxes used in systems biology.

Some definitions and theorems

$\hat{\mathbf{p}}$ is a global minimizer of the objective function V if it gives the lowest obtainable objective function value from an arbitrary starting point:

$$\hat{\mathbf{p}}_{\text{global}} = \arg\min_{\mathbf{p}} V(\mathbf{p}) \quad \forall \mathbf{p} \text{ in the parameter space} \quad (22)$$

$\hat{\mathbf{p}}$ is a local minimizer of the objective function V if it gives the lowest obtainable objective function value in the neighborhood of the starting point:

$$\hat{\mathbf{p}}_{\text{local}} = \arg\min_{\mathbf{p}} V(\mathbf{p}) \qquad \forall ||\mathbf{p} - \hat{\mathbf{p}}_{\text{start}}|| < \delta, \quad \delta > 0 \quad (23)$$

A stationary point x^* of a function f is a point for which the gradient is zero:

$$\nabla f(x^*) = 0 \quad (24)$$

The following theorems hold for unconstrained optimization and a sufficiently differentiable objective function V. In this case, V can be extended into a Taylor series around $\hat{\mathbf{p}}$:

$$V(\hat{\mathbf{p}} + \delta\mathbf{p}) = V(\hat{\mathbf{p}}) + \delta\mathbf{p}^{\mathrm{T}}\nabla V(\hat{\mathbf{p}}) + \tfrac{1}{2}\delta\mathbf{p}^{\mathrm{T}}\nabla^2 V(\hat{\mathbf{p}})\delta\mathbf{p} + \cdots \quad (25)$$

with the gradient:

$$\nabla V(\hat{\mathbf{p}}) = \left[\frac{\partial V}{\partial \mathbf{p}}(\hat{\mathbf{p}})\right] \quad (26)$$

and the Hessian or second derivative:

$$\nabla^2 V(\hat{\mathbf{p}}) = \left[\frac{\partial V}{\partial \mathbf{p}_i\,\partial \mathbf{p}_j}(\hat{\mathbf{p}})\right] \quad (27)$$

A necessary condition for a parameter vector $\hat{\mathbf{p}}$ to be a local minimizer of V is that $\hat{\mathbf{p}}$ is a stationary point of V:

$$\nabla V(\hat{\mathbf{p}}) = 0$$

A sufficient condition for a local minimizer is that $\hat{\mathbf{p}}$ is a stationary point of V and the Hessian of V is positive definite:

$$\nabla V(\hat{\mathbf{p}}) = 0; \quad \mathbf{p}^{\mathrm{T}}\nabla^2 V(\hat{\mathbf{p}})\mathbf{p} > 0 \quad \forall \mathbf{p} \neq 0$$

Global optimization

Most global optimization methods are stochastic of nature to prevent the search process being trapped in a local minimum. Moles *et al.* [11] have performed a comparison of a number of global optimization methods on parameter estimation problems for biochemical pathways.

Simulated annealing

Simulated Annealing (SA) is a stochastic optimization algorithm proposed by Kirkpatrick *et al.* [37] in 1983. The term annealing comes from physics. It is the process of heating up a solid until it melts, followed by a slow cooling down until the molecules are aligned in a crystalline structure corresponding to the minimum energy state. The cooling must occur at a sufficiently slow rate, otherwise the system will end up in an amorphous or polycrystalline state and thus the system will not be at its minimum energy state. In optimization, the SA algorithm attempts to mathematically capture the process of controlled cooling associated with physical processes; the analogy to the minimum energy state is the minimum value for the objective function.

SA is based on the Metropolis algorithm [38] which is a Monte Carlo method to sample a thermodynamic system. Rephrased for the parameter estimation problem, it samples for a fixed 'temperature' the parameter space according to the Boltzmann–Gibbs probability distribution:

$$P(\mathbf{p}) = C\exp\left(-\frac{V(\mathbf{p})}{k_{\mathrm{B}}T}\right) \quad (28)$$

where C is a normalization constant, k_{B} the Boltzmann constant, and T the temperature. Starting from an initial (random) parameter vector, in each step, a random new state (parameter vector) is generated based on the previous one. This new state is accepted with a certain probability (see below under Transition probability). If it is rejected, a new state is generated based on the same parameter vector as before. In this way, a Markov chain is obtained which, if it is sufficiently long, describes the required probability distribution. The macroscopic observable, the minimizing parameter vector, is the average over all states in the Markov chain. In SA, the Metropolis algorithm is applied with a slowly decreasing T. SA starts with a high 'temperature' implying that all states, or parameter vectors, are equally probable. The original algorithm (i.e. the homogeneous Markov chain method) computes for a constant temperature a complete Markov chain (i.e. the required probability distribution is obtained). Then the temperature is slowly decreased and the next distribution is sampled. By contrast, the inhomogeneous Markov chain method decreases the temperature every time a new state has been found. Devising the cooling schedule (i.e. initial temperature, method of lowering the temperature, and the stop criterion) is the art of simulated annealing. Under certain conditions (ergodicity, cooling schedule), it has been proven that SA converges to the global minimum [39].

Cooling schedules

Many have attempted to derive theoretical or experimental proofs of an efficient cooling schedule scheme [40]. Among the most popular ones, three different theoretical concepts are used.

Logarithmic: Introduced by Geman and Geman [41], this has special theoretical importance. The temperature is decreased according to: $t_i = \gamma/\log(i + d)$ with i being the iteration count and d is usually set to one. Although it has been proven that for $\gamma \geq E_{\max}$, the true global minima can be found (in the limit of infinite time), with $E_{\max}$ being the maximum energy barrier (problem dependent and *a priori* unknown), this method is very slow and impractical because of its asymptotically slow temperature decrease [42].

Geometric: The original cooling schedule proposed by Kirkpatrick *et al.* [37] and still widely used with major or minor variants. The temperature is updated by: $t_i = \alpha t_{i-1}$. The cooling factor α is assumed to be a constant smaller than one. Examples of usage and a good explanation of the underlying mechanisms are given by Johnson *et al.* [43].

Adaptive: The previous cooling schedules always apply the same cooling factor irrespective the state of the system. It is known that, at high temperature, almost all new parameter vectors are accepted, although some of them are bad solutions. It is obvious that using an appropriate cooling schedule depending on the state of the system can lead to large improvements. A variety of adaptive temperature annealing strategies have been proposed. The main techniques are presented by Boese [40]. The most important ones are: (a) Lam [44,45]: the temperature is updated aiming to maintain the system in thermodynamical equilibrium; and (b) Ingber [46,47]: a very popular cooling schedule. The strength of this algorithm is that it takes into account the sensitivity of the cost function for each parameter. The goal is to extend the insensitive parameter's search range relative to the range over the more sensitive parameters. Each parameter has its own temperature, equally initialized at the beginning. After every N_{acc} accepted steps, the sensitivity for the best solution parameters is computed and, after every N_{gen} generation steps, the temperatures are re-annealed scaled by the sensitivities. A very limited number of method parameters has to be assigned by the user: the rate control parameter C, N_{acc}, and N_{gen}. The other method parameters are automatically set and updated by the algorithm. The optimal values of the three parameters are problem dependent [48], but the performance of the algorithm is not critically influenced for choices of C in the range 1–10, $N_{\mathrm{acc}}O(10\text{–}100)$ and $N_{\mathrm{gen}}O(200\text{–}1000)$.

Transition probability

If the objective function of the new parameter vector $\mathbf{p}'$ is smaller than the previous one then the new parameter vector is accepted. However, to prevent getting stuck in a local minimum, the new parameter vector is also accepted with a probability according to the sampled distribution:

$$P(\Delta V, T) = \exp\left(-\frac{\Delta V}{k_{\mathrm{B}}T}\right) \quad \text{with } \Delta V = V(\mathbf{p}') - V(\mathbf{p}) \tag{29}$$

Equation (29) is known as the Metropolis Criterion. For $T \to 0$ and $\delta V > 0$, the probability $P(\delta V, T) \to 0$. Therefore, for sufficiently small values of T, the process will more and more go 'downhill': new accepted parameter vectors tend to have lower objective function values.

Evolutionary algorithms

Evolutionary algorithms (EA) are inspired by biological evolution. Potential solutions (parameter vectors) are the individuals of a population. To get new solutions (a next generation) the individuals in the population are replaced using mechanisms as reproduction, natural selection, mutation, recombination, and survival of the fittest.

Initially, a population of random individuals (possible parameter vectors) is created. Next, the corresponding objective functions are computed that define the fitness of an individual (the higher the fitness, the better the solution). The selection process is mimicked by assigning probabilities to individuals related to their fitness to indicate the chance of being selected for the next generation. Individuals with a high fitness are assigned high probabilities. New individuals are created by two operators: recombination (or cross-over) and mutation. Recombination consists of selecting some parents (at least two) and results in one or more children (new candidates). Mutation acts on one candidate and results in a new candidate. These operators create the offspring (a set of new candidates). These new candidates compete with old candidates for their place in the next generation (survival of the fittest). This process can be repeated until a candidate with sufficient quality (a solution) is found or a predefined computational limit is reached. There are many different ways of writing these operators and one can find exhaustive literature focussing on this aspect of EAs [49].

EA operators

The selection operator is responsible for convergence to the minimum, the recombination operator for exploring the parameter space and the mutation operator gives nearby solutions a chance to survive.

Fitness

A commonly used objective-to-fitness transformation results in a fitness value of max(0,$C_{\max}-V(\mathbf{p})$) with $C_{\max}$ either being a user-defined constant or the maximum V-value thus far. To prevent almost equal selection probabilities in later stages of the algorithm, the fitness values should be scaled accordingly [49]. Another transformation is simply rank-based, where the population is sorted according to their objective values and fitness assignment depends only on the position [50,51].

Selection

This determines which individuals are chosen for mating (recombination) and how many offspring each selected individual produces. The first step is fitness assignment. Next, the actual selection is performed. Parents are selected according to their fitness by means of one of the following algorithms [49]:

Truncation: the only deterministic selection: select the m best individuals and reproduce them until the pool is filled;

Roulette-wheel: selection with size of wheel part proportional to fitness [52];

Stochastic remainder: sampling. First **entier** $(f_i/\bar{f})$[5] times individual i are selected with f_i the individual and $\bar{f}$ the average fitness. Next, the pool is filled using a weighted toss [52];

Tournament: N 'tournaments' will be held with K randomly picked individuals as competitors for a place in the pool. Winner is the one with highest fitness [53].

The selection process is an extremely important part of the convergence of the algorithm: if the selection pressure is high (as with roulette-wheel) then the convergence time is fast, but the solution can be a local one. If the selection pressure is low (as with tournament with small K) it is the other way around.

Recombination or cross-over

This produces new individuals by combining the information contained in the parents (parents: mating population). In the case of real-valued variables, the algorithms all choose a point on the line connecting the two parents, either deterministically [line recombination (interpolation with a fixed constant)] or stochastically. In the latter case, one distinguishes intermediate recombination in which a point is chosen in an interval slightly larger than the connecting line segment and extended line recombination where the complete line is used but the probability decreases with the distance from a parent.

Mutation

This consists of randomly altering an individual. The mutation step (usually very small) is the probability of mutating a variable, and the mutation rate is the effective mutation applied to that variable. Although, in general, the mutation step is inversely proportional to the dimension of the problem, the mutation rate does not depend on the problem.

Reinsertion (survival of the fittest)

After producing offspring, they must be inserted into the population. This is especially important if the number of offspring does not equal the size of the original population. To guarantee that the best individual(s) survive, the elitist strategy [49] can be used.

Note that evolutionary algorithms lack a proper theory. Choosing the right (combination of) operators and devising a good stop criterion is the art of implementing and using evolutionary algorithms.

[5] **entier**(x) is the largest integer value not exceeding x.

Covering methods

Covering methods are deterministic global optimization algorithms that guarantee that a solution with a given accuracy is obtained. The price paid for this guarantee, however, is that some *a priori* information of the function must be available.

Branch and bound

This requires that the search space is finite (parameters are constrained) and can be divided to create smaller subspaces [54,55]. To apply branch and bound, one must have a means of computing upper and lower estimated bounds of the objective function to be minimized.

The method starts by considering the original problem with the complete search space (i.e. the root problem). The lower-bounding and upper-bounding procedures are applied to the root problem. If the bounds match, then an optimal solution has been found and the procedure terminates. Otherwise, the search space is partitioned into two or more regions. These subproblems become children of the root search node. The algorithm is applied recursively to the subproblems, generating a tree of subproblems. If an optimal solution is found to a subproblem, it is a feasible solution to the full problem, but not necessarily globally optimal. Because it is feasible, it can be used to prune the rest of the tree: if the lower bound for a node exceeds the best known feasible solution, no globally optimal solution can exist in the subspace of the feasible region represented by the node. Therefore, the node can be removed from consideration. The search proceeds until all nodes have been solved or pruned, or until some specified threshold is met between the best solution found and the lower bounds on all unsolved subproblems.

Although this method is widely used in engineering, the technique is not that popular among the biologists and computational biology community.

Overview

Simulated annealing and branch and bound have a proper convergence theory. The disadvantage of branch and bound is that it can only be applied if it is possible to compute lower and upper bounds for the objective function. SA is generally applicable, but the theoretical convergence is in practice not much worth because it is critically dependent on the cooling-down schedule. At each temperature the inner-loop (Metropolis) needs to be iterated long enough to explore the regions of search space. However, the balance between the maximum step size and the number of Monte Carlo steps is often difficult to achieve, and depends very much on the characteristics of the search space or energy landscape. SA is computationally very expensive and is not easily paralellizable.

EAs consistently perform well for all types of problems and are well-suited to solve problems with a truly large search space. The critical factor to escape local minima is the cross-over operator that allows each individual to explore other possibilities by means of information transfer [56]. The critical factor for fast convergence is the selection operator. Premature convergence occurs if an individual that is more fit than most of its competitors emerges too early, it may reproduce so abundantly that it drives down the population's diversity too soon. This will lead the algorithm to converge to the local optimum of that specific individual rather than searching the fitness landscape thoroughly enough to find the global optimum [57]. For a proper behavior, the population size should be sufficiently large, which means that the method is expensive if the computation of the objective function is not extremely cheap. Fortunately, EA is intrinsically parallel. Multiple individuals can explore the search space in different directions. By contrast to SA, EA can be implemented as a self-tuning method, the most successful example is the stochastic ranking evolutionary strategy (SRES) [58,59].

Local optimization

If the gradient of the objective function can be computed one can solve the minimization problem by finding the point where the gradient vanishes using gradient-based methods. Direct-search methods try to find the minimizing point of the objective function without explicitly using derivatives. As for the global search methods, these methods only require an order relation ($V(\mathbf{p}_1) < V(\mathbf{p}_2)$) for all points in parameter space.

Direct-search methods

The term direct-search method has first been used in 1961 in the classical paper of Hooke and Jeeves [60] that describes their pattern search method, but it is more generally used for all methods that find a local minimum without the use of a derivative. Direct-search methods select a finite (i.e. generally not large) number of possibilities each step and check whether one of these is better than the current one. Reviews on direct-search or derivative-free methods are available elsewhere [61–63]. Here, we discuss the two most used methods: the classical Hooke–Jeeves method [60] and the Nelder–Mead or Downhill Simplex method [64].

Hooke–Jeeves method

The pattern search method of Hooke and Jeeves [60] consists of two steps. In the first, a series of exploratory changes of the current parameter vector are made, typically a positive and negative perturbation of one parameter at a time. The exploratory step then has formed a basis for the parameter space with information in which directions the objective function decreases. In the next step, the pattern move, the information obtained is used to find the best direction for the minimization process. The original method is a special case of generalized pattern search methods for which it is shown that the search directions span the parameter space [65]. For a good discussion on this type of direct-search methods, the broad class of generating set search methods, including convergence results, some history and references to other ideas, we refer to the extensive review paper of Kolda *et al.* [63]. They show, amongst other things, that these methods have the same type of convergence guarantee as gradient-based methods.

Nelder–Mead simplex algorithm

The Nelder–Mead method [64,66] is based on the idea of an adaptive simplex: the simplest polytope of $m + 1$ vertices in m dimensions (2D, triangle; 3D, tetrahedron). The objective function is evaluated in all vertices (**p**'s) and the vertices are ordered according to the value. The next step tries to replace the 'worst' vertex by a better one. A line search is performed along the line through this vertex and the centroid of the remaining vertices: $\mathbf{p}_{\text{new}} = \bar{\mathbf{p}} + \alpha\mathbf{p}_{\text{worst}}$. For $\alpha = 1, 2, \frac{1}{2}, -\frac{1}{2}$, it is tested whether the new objective value is better than the old one. If this is the case, the simplex is adapted by replacing the old vertex by the new one. If not, a shrink procedure is performed: the 'best' vertex stays in the simplex, all other ones are replaced by a vertex half-way along the line from the best vertex. If the line search is successful, the method uses just 1–4 function evaluations per step and the aim is that the simplex adapts itself to the minimizing function. But, in contrast to the Hooke–Jeeves method, it improves the objective function value along the sequence of worst vertices.

Gradient-based methods

By constrast to all other methods this class of methods described above, not only requires the value of the objective function, but also of its first derivative with respect to the parameters. These type of methods are not so straightforward to implement as the direct-search methods, but, if it is possible to use them, it is in general preferrable to do so. Often in implementations, approximations of the gradient and/or the Hessian (second derivative) are used (e.g. by finite differences). However, with the current automatic differentation tools such as ADIFOR [67], symbolic algebra packages such as MAPLE [15] and MATHEMATICA [16], and modeling languages with automatic computation of derivatives such as AMPL [68] and GAMS [69], it is doable and preferrable to use the exact derivative.

Because these methods are more mathematical based, we discuss them more rigorously. For a general treatment of this subject, we refer to Nocedal and Wright [70].

Remember that a requirement for a local minimizer $\mathbf{p}^*$ is that the gradient $\nabla V(\mathbf{p}^*) = 0$ (stationary point). A sufficient condition requires that the Hessian is positive definite. Note that none of the methods below guarantees the latter requirement!

Gradient-based methods are all descent methods. These methods first find a descent direction $d\mathbf{p}$ and then take a step $\alpha d\mathbf{p}$ in that direction, with α such that it results in a 'good' decrease of the objective function:

$$\mathbf{p}_{\text{new}} = \mathbf{p} + \alpha d\mathbf{p}, \quad V(\mathbf{p}_{\text{new}}) < V(\mathbf{p}) \tag{30}$$

The largest gain is obviously obtained when α is determined by a line-search, (i.e. by finding the minimum value of $V(\mathbf{p} + \alpha d\mathbf{p})$ for all $\alpha > 0$).

Note that a simple decrease in the objective function ($f(x_{k+1}) < f(x_k)$) is not sufficient to converge to a stationary point of f. (Counterexample: $V(x) = x^2$ and $x_i = 1 + 2^{-i}$; [71])

Steepest descent or gradient method

In this method, the search direction is defined by the gradient:

$$d\mathbf{p} = -\nabla V(\mathbf{p}) \tag{31}$$

In the final stage, however, this method has a slow convergence. In fact, if combined with exact line search, it can even fail.

Newton's method

Newton's method iteratively solves the equation for a stationary point $\nabla V(\mathbf{p}^*) = 0$ by linearization. The search direction for the line-search method is in this case:

$$d\mathbf{p} = -\nabla^{-2} V(\mathbf{p}) \nabla V(\mathbf{p}) \tag{32}$$

In quasi-Newton methods, the Hessian is approximated. If the starting point is sufficiently close to the solution, Newton's method has a quadratic order of convergence.

Trust region method [72]
The objective function $V(\mathbf{p})$ is approximated by a simpler function, which mimicks the behaviour of V in a neighbourhood of $\mathbf{p}$. This function is then minimized over this neighbourhood, the trust region, and if the objective function decreases the new value is accepted. Otherwise, the trust region is decreased. Originally, the approximation consisted of the first two terms of the Taylor expansion of V at $\mathbf{p}$ but, for high-dimensional problems, this is still too expensive. In this case, the trust region is restricted to two dimensions [35]. This subspace is spanned by the gradient vector ∇V (Eqn. 31) and a direction of negative curvature given by $\mathrm{d}\mathbf{p}^{\mathrm{T}}\nabla^2 V(\mathbf{p})\mathrm{d}\mathbf{p} < 0$ or the Newton direction (Eqn. 32). The aim of the first combination is global convergence and of the second fast local convergence.

Gradient-based methods for least-squares

Gauss–Newton
If the function to be minimized is a sum of squares (as is the case when solving a least-squares problem), Newton's method is often replaced by a modification: the Gauss–Newton algorithm, in which the Hessian is not used. The gradient of $V_{\mathrm{MLE}}(\mathbf{p}) = \mathbf{e}^{\mathrm{T}}\mathbf{e}$ is given by $\nabla V_{\mathrm{MLE}} = J^{\mathrm{T}}\mathbf{e}$, where the Jacobian $J(\mathbf{p}) = \frac{\partial \mathbf{e}}{\partial \mathbf{p}}(\mathbf{p})$ is the so-called 'sensitivity' matrix of size $N \times m$ (cf. Eqn 6).

To solve for the stationary point, again linearization is used which results in the task to solve the normal equations:

$$J^{\mathrm{T}}(\mathbf{p})J(\mathbf{p})\delta\mathbf{p} = -J^{\mathrm{T}}(\mathbf{p})\mathbf{e}(\mathbf{p}) \tag{33}$$

Note that $\delta\mathbf{p}$ is a descent direction because $\delta\mathbf{p}^{\mathrm{T}}\nabla V_{MLE} = \delta\mathbf{p}^{\mathrm{T}}J^{\mathrm{T}}\mathbf{e} = -\delta\mathbf{p}^{\mathrm{T}}J^{\mathrm{T}}J\delta\mathbf{p} < 0$. As in Newton, this is an iterative process.

Levenberg–Marquardt method
This can be seen as Gauss–Newton with damping or as a combination of Gauss–Newton with steepest descent [73]. The search direction is defined by:

$$\left(J^{\mathrm{T}}(\mathbf{p})J(\mathbf{p}) + \lambda I_m\right)\delta\mathbf{p} = -J^{\mathrm{T}}(\mathbf{p})\mathbf{e}(\mathbf{p}) \tag{34}$$

where $\lambda \geq 0$ is some constant and I_m the identity matrix of size m. $\delta\mathbf{p}$ is a descent direction for all $\lambda > 0$; for λ large Eqn (34) results in the steepest descent method and for λ small in the Gauss–Newton process. The first is a good strategy in the initial stage of the process, the latter in the final stages. The art of the Levenberg–Marquardt method is the design of the damping factor λ [74,75].

Overview

Direct-search methods are generally applicable, but they are less efficient especially for high-dimensional problems. If possible (i.e. if the problem is smooth), we recommend to use Newton or trusted region and, for a least-squares fit, Levenberg-Marquardt. In non-smooth problems, the objective function is discontinuous or has a discontinuous derivative (e.g. because the mathematical model contains step-functions, absolute values, if-then-else constructions, etc.). In this case, gradient-based methods can not be applied. The Hooke–Jeeves method or, more generally, the generating set search methods are reliable but slow. The Nelder–Mead simplex method is in most cases efficient, but it can fail unpredictably [76].

Normally, the methods described here are used as single shooting methods, meaning that the integration path leading to the observable function value in the objective function is determined by the initial conditions for the state variables. Especially, if these initial conditions depend on parameters, this can lead to the wrong minimum. To avoid this, one can use the multiple shooting approach [77] where the time interval is partitioned and new initial conditions are used at the start of each part of the interval. To connect the integration paths smoothly, an augmented system has to be solved.

Constraints

For all optimization methods described above, it holds that it is the implementation that counts, where one version of an optimization method with different method parameters and strategy can result in a much better and faster convergence behaviour (for some problems) than the next. This holds even more for the implementation of constraints. Contraints can be implemented as penalties added to the objective function. This is often done in global and in direct–search methods. It implies that the constraints are not strictly obeyed, at least during the search. In direct–search methods, linear constraints restrict the search directions (i.e. the parameter space becomes a cone) and thus the chance of failure increases (the search directions no longer span the search space). For nonlinear constraints, a number of approaches exists; for an overview of methods used in generalized set search methods, see Kolda *et al.* [63]. If the constraints are differentiable, this direction can be used when computing the new search direction. For generalized set search and gradient-based methods, one can also solve an augmented nonlinear system where a Lagrange multiplier with the constraint is added and possibly other penalty terms [33,63].

Hybrid methods

Global methods in generally work well to explore the parameter space but are slow in finding the minimum of the objective function precisely [36]. By contrast, local methods are much faster in finding a minimum once in the neighborhood. Sequential application of both approaches combines the best of the two. Such hybrid methods use a global search method to identify promising regions of the search space that are further explored by a local optimizer.

Katare *et al.* [78,79] employ a particle swarm optimization [80,81] combined with Levenberg–Marquardt. However, their method appears to be sensitive to the 'swarm topology' that defines the information transfer between the parameter vectors. Combinations of local search with the SRES [58] seem to be more promising. Rodriguez-Fernandez *et al.* [5] apply, with good results, SRES + DN2GB (Gauss–Newton + trust region for stabilization) on the three-step pathway benchmark problem [11]. A challenging reaction-diffusion system has also been considered describing the early *Drosophila* development [8,36]. This results in a model with 348 state variables and a 66-dimensional optimization problem with (non)linear constraints. Jaeger *et al.* [82] obtained previously the parameters for that model with parallel simulated annealing. Fomekong-Nanfack *et al.* [36] show that the hybrid method SRES + Nelder-Mead is approximately 50 times as fast. The same problem was solved with SRES + Levenberg–Marquardt [8] with a comparable speed up, but a better approximation of the local minima.

Another interesting approach is an intrinsic global-local method such as the scatter-search method [83,84], an evolutionary algorithm with a local search method after (each) recombination step. Because this method is expensive for costly objective funtion evaluations SSKm (Scatter-search-Kriging) has been developed [85]. Here, the number of 'local-search' points is reduced by predicting the possibility that a new parameter vector will result in a lower minimum without evaluation of the objective function, based on the assumption that V has a Gaussian distribution (Kriging).

Discussion

The aim of this minireview was to give a comprehensive survey of parameter estimation (i.e. to discuss both the methods to fit the parameters of a mathematical model to experimental data and to analyze the results). A recent review paper of van Riel [86] discusses these subjects more from the perspective of systems biology but less extensively.

Unfortunately, we cannot recommend one or the other algorithm as the definitive method to search for parameters. An optimal use of the methods, especially of the global ones, is problem-dependent and, in practice, convergence to the minimum is not guaranteed. Global methods are often used with a computational time limit to prevent an endless search and local methods can get stuck in a local minimum. In general, a good initial guess (e.g. from experiments) will not be available for all parameters, ruling out the option of using only local search methods. A good strategy is often to use global search methods to find various 'promising' areas in the parameter space. Once in these areas, local search methods converge much faster to the minimum [5,8,36]). Because global methods explore the complete 'fitness landscape', it is also possible to find multiple parameter vectors that satisfy the experimental data.

In the overview, we compared the algorithms for global search. For most problems, an evolutionary algorithm, such as the SRES, is robust and easy to use. The local search methods were also evaluated. Here, the optimal method choice is dependent on the objective funtion and on the DAE system. For a least-squares fit and smooth problems, we recommend Levenberg–Marquardt. If the (derivative of) the objective funtion is discontinuous, a direct method such as Nelder–Mead should be used. If the initial conditions of the DAEs depend also on the parameters and the solution of the DAE system depends strongly on the initial conditions, the multiple shooting strategy could be advantageous. A promising, but not yet fully tested strategy is the intrinsic global-local approach implemented in SSKm. Most importantly, for all optimization algorithms, it is the implementation that counts, especially if the parameter space is restricted by constraints.

Finally, finding a parameter vector is only half the job. It is important to study how robust against perturbations the parameters are. If the objective function is the MLE (Eqn 4), the analysis method described in the section '*A posteriori* identifiability' can be applied. Otherwise, one can use a repeated fitting strategy [27] to study the fitness landscape.

Acknowledgements

We acknowledge financial support from the NWO program 'Computational Life Science', projectnr. 635.100.010. J. G. B. acknowledges also the Dutch BSIK/BRICKS program. We would like to thank Jan Stolze for help with the pictures and proofreading, and Piet Hemker and Jan Stolze for many helpful discussions.

References

1 Golightly A & Wilkinson DJ (2005) Bayesian inference for stochastic kinetic models using a diffusion approximation. *Biometrics* **61**, 781–788.

2 Timmer J (2000) Parameter estimation in nonlinear stochastic differential equations. *Chaos Solitons Fractals* **11**, 2571–2578.

3 Reinker S, Altman RM & Timmer J (2006) Parameter estimation in stochastic biochemical reactions. *IEE Proc-Syst Biol* **153**, 168–178.

4 Jaqaman K & Danuser G (2006) Linking data to models: data regression. *Nat Rev Mol Cell Biol* **7**, 813–819.

5 Rodriguez-Fernandez M, Mendes P & Banga JR (2006) A hybrid approach for efficient and robust parameter estimation in biochemical pathways. *BioSystems* **83**, 248–265.

6 Schittkowski K (2002) *Numerical Data Fitting in Dynamical Systems – A Practical Introduction with Applications and Software*. Kluwer Academic Publishers, Dordrecht.

7 Gutenkunst RN, Waterfall JJ, Casey FP, Brown KS, Myers CR & Sethna JP (2007) Universally sloppy parameter sensitivities in systems biology models. *PLOS Comp Biol* **3**, 1871–1878.

8 Ashyraliyev M, Jaeger J, & Blom JG (2008) Parameter estimation and determinability analysis applied to *Drosophila* gap gene circuits. *BMC Systems Biology* **2**, 83, doi: 10.1186/1752-0509-2-83.

9 Ljung L (1999) *System Identification – Theory For the User*. Prentice Hall, Upper Saddle River, NJ.

10 Mendes P & Kell DB (1998) Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics* **14**, 869–883.

11 Moles CG, Mendes P & Banga JR (2003) Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome Res* **13**, 2467–2474.

12 Aster RC, Borchers B & Thurber CH (2005) *Parameter Estimation and Inverse Problems*. Elsevier Academic Press, Burlington, MA.

13 Seber GAF & Wild CJ (1988) *Nonlinear Regression*. John Wiley & Sons, Inc, New York, NY.

14 Draper NR & Smith H (1988) *Applied Regression Analysis*. John Wiley & Sons, Inc, New York, NY.

15 Maple. Available at: http://www.maplesoft.com/

16 Mathematica. Available at: http://www.wolfram.com/

17 Pohjanpalo J (1978) System identifiability based on the power series expansion of the solution. *Math Biosci* **41**, 21–33.

18 Godfrey KR & Fitch WR (1984) The deterministic identifiability of nonlinear pharmacokinetic models. *J Pharmacokinet Biopharm* **12**, 177–191.

19 Vajda S, Godfrey KR & Rabitz H (1989) Similarity transformation approach to structural identifiability of nonlinear models. *Math Biosci* **93**, 217–248.

20 Evans ND, Chapman MJ, Chappell MJ & Godfrey KR (2002) Identifiability of uncontrolled nonlinear rational systems. *Automatica* **38**, 1799–1805.

21 Peeters RLM & Hanzon B (2005) Identifiability of homogeneous systems using the state isomorphism approach. *Automatica* **41**, 513–529.

22 Chappel MJ, Godfrey KR & Vajda S (1990). Global identifiability of the parameters of nonlinear systems with specified inputs: a comparison of methods. *Math Biosci* **102**, 41–73.

23 Audoly S, Bellu G, D' Angiò L, Saccomani MP & Cobelli C (2001) Global identifiability of nonlinear models of biological systems. *IEEE Trans Biomed Eng* **48**, 55–65.

24 Bellu G, Saccomani MP, Audoly S & D' Angiò L (2007) DAISY: a new software tool to test global identifiability of biological and physiological systems. *Comput Methods Programs Biomed* **88**, 52–61.

25 REDUCE. Available at: http://www.reduce-algebra.com/

26 Hidalgo ME & Ayesa E (2001) Numerical and graphical description of the information matrix in calibration experiments for state-space models. *Wat Res* **35**, 3206–3214.

27 Heng S, Kreutz C, Timmer J & Maiwald T (2007) Data-based identifiability analysis of non-linear dynamical models. *Bioinformatics* **23**, 2612–2618.

28 Breiman L & Friedman J (1985) Estimating optimal transformations for multiple regression and correlation. *J Am Stat Assoc* **80**, 580–598.

29 MATLAB. Available at: http://www.mathworks.com/

30 PottersWheel. Available at: http://www.potterswheel.de/

31 Bentele M, Lavrik I, Ulrich M, Stößer S, Heermann DW, Kalthoff H, Krammer PH & Eils R (2004) Mathematical modeling reveals threshold mechanism in CD95-induced apoptosis. *J Cell Biol* **166**, 839–851.

32 Bentele M (2004) *Stochastic simulation and system identification of large signal transduction networks in cells*. PhD thesis, University of Heidelberg, Germany.

33 Stortelder WJH (1998) *Parameter estimation in nonlinear dynamical systems*. PhD Thesis, University of Amsterdam, the Netherlands.

34 Kutalik Z, Cho KH & Wolkenhauer O (2004) Optimal sampling time selection for parameter estimation in dynamic pathway modelling. *Biosystems* **75**, 43–55.

35 Byrd RH, Schnabel RB & Shultz GA (1988) Approximate solution of the trust region problem by minimization over two-dimensional subspaces. *Math Programming* **40**, 247–263.

36 Fomekong-Nanfack Y, Kaandorp JA & Blom J (2007) Efficient parameter estimation for spatio-temporal models of pattern formation: case study of *Drosophila melanogaster*. *Bioinformatics* **23**, 3356–3363.

37 Kirkpatrick S, Gelatt CD & Vecchi MP (1983) Optimization by simulated annealing. *Science* **220**, 671–680.

38 Metropolis N, Rosenbluth AW, Rosenbluth MN & Teller AH (1953) Equation of state calculations by fast computing machines. *J Chem Phys* **21**, 1087–1092.

39 van Laarhoven PJM & Aarts EHL (1987) *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers, Dordrecht.

40 Boese KD (1996) *Models for iterative global optimization*. PhD thesis, University of California at Los Angeles, Los Angeles, CA.

41 Geman S & Geman D (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans Pattern Anal Mach Intell* **6**, 721.

42 Hajek B (1988) Cooling schedules for optimal annealing. *Math Oper Res* **13**:311–329.

43 Johnson DS, Aragon CR, McGeoch LA & Schevon C (1989) Optimization by simulated annealing: an experimental evaluation; part 1, graph partitioning. *Oper Res* **37**, 865–892.

44 Lam J & Delosme J-M (1988) *An Efficient Simulated Annealing Schedule: Derivation*. Technical Report 8816, Electrical Engineering Department, Yale, New Haven, CT.

45 Lam J & Delosme J-M (1988) *An Efficient Simulated Annealing Schedule: Implementation and Evaluation*. Technical Report 8817, Electrical Engineering Department, New Haven, CT.

46 Ingber L (1989) Very fast simulated reannealing. *Math Comput Modelling* **12**, 967.

47 Ingber, L & Rosen B (1992) Genetic algorithms and very fast simulated annealing – a comparison. *Math Comput Modeling* **16**, 87–100.

48 Ingber L (1996) Adaptive simulated annealing (asa): lessons learned. *Control Cybern* **25**, 33.

49 Goldberg DE (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, New York, NY.

50 Bäck T & Hoffmeister F (1991) Extended Selection Mechanisms in Genetic Algorithms. In *Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA-4)* (Belew RK & Booker LB, eds), pp. 92–99. Morgan Kaufmann, San Mateo, CA.

51 Whitley D (1989) The genitor algorithm and selection pressure: why rank-based allocation of reproductive trials is best. In *Proceedings of the Third International Conference on Genetic Algorithms* (Schaffer JD, ed.), pp. 116–121, Morgan Kaufmann Publishers Inc., San Francisco, CA.

52 Baker JE (1987) Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms and their Application*, (Grefenstette JJ, ed.), pp. 14–21. Lawrence Erlbaum Associates, Hillsdale, NJ.

53 Miller BL (1997) *Noise, sampling, and efficient genetic algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, Champaign, IL.

54 Lawler EL & Wood DE (1966) Branch-and-bound methods: a survey. *Oper Res* **14**, 699–719.

55 Mitten LG (1970) Branch-and-bound methods: general formulation and properties. *Oper Res* **18**:24–34.

56 Koza JR, Andre D, Bennett FH & Keane MA (1999) *Genetic Programming III: Darwinian Invention & Problem Solving*. Morgan Kaufmann Publishers Inc., San Francisco, CA.

57 Forrest S (1993) Genetic algorithms: principles of natural selection applied to computation. *Science* **261**,872–878.

58 Runarsson TP & Yao X (2000) Stochastic ranking for constrained evolutionary optimization. *IEEE Trans Evol Comput* **4**, 284–294.

59 Zi Z & Klipp E (2006) SBML-PET: a systems biology markup language based parameter estimation tool. *Bioinformatics* **22**, 2704–2705.

60 Hooke R & Jeeves TA (1961) Direct search solution of numerical and statistical problems. *J Assoc Comput Mach* **8**, 212–229.

61 Wright MH (1995) Direct search methods: once scorned, now respectable. In *Proceedings of the 1995 Biennial Dundee Conference on Numerical Analysis* (Griffiths DF & Watson GA, eds), pp. 191–208, Addison Wesley Longman, Harlow, UK.

62 Powell MJD (1998) Direct search algorithms for optimization calculations. *Acta Numerica* **7**, 287–336.

63 Kolda TG, Lewis RM & Torczon V (2003) Optimization by direct search: new perspectives on some classical and modern methods. *SIAM Rev* **45**, 385–482.

64 Nelder, JA & Mead R (1965) A simplex method for function minimization. *Comput J* **7**, 308-313.

65 Torczon V (1997) On the convergence of pattern search algorithms. *SIAM J Optim* **7**, 1–25.

66 Lagarias JC, Reeds JA, Wright MH & Wright PE (1998) Convergence properties of the Nelder-Mead

simplex method in low dimensions. *SIAM J Optim* **9**, 112–147.

67 Adifor. Available at: http://www-unix.mcs.anl.gov/autodiff/ADIFOR/

68 Ampl. Available at: http://www.ampl.com/

69 Gams. Available at: http://www.gams.com/

70 Nocedal J & Wright SJ (1999) *Numerical Optimization*. Springer, New York, NY.

71 Dennis JE Jr & Schnabel RB (1983) *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice Hall, Englewood Cliffs, NJ.

72 Conn AR, Gould IM & Toint PL (2000) *Trust-Region Methods*. Number 1 in MPS/SIAM Ser. Optim. SIAM, Philadelphia, PA.

73 Marquardt DW (1963) An algorithm for least-squares estimation of nonlinear parameters. *SIAM J Appl Math* **11**, 431–441.

74 Bus JCP, van Domselaar B & Kok J (1975) *Nonlinear Least Squares Estimation*. Report NW 17/75, Stichting Mathematisch Centrum, Amsterdam.

75 Madsen K, Nielsen HB & Tingleff O (2004) *Methods for Non-Linear Least Squares Problems*. IMM, DTU, Denmark.

76 McKinnon KIM (1998) Convergence of the Nelder-Mead simplex method to a nonstationary point. *SIAM J Optim* **9**, 148–158.

77 Timmer J (1998) Modeling noisy time series: physiological tremor. *Int J Bifurcation Chaos* **8**, 1505–1516.

78 Katare S, Kalos A & West D (2004) A hybrid swarm optimizer for efficient parameter estimation. In *Proceedings of the 2004 IEEE Congress on Evolutionary Computation* (Greenwood GW, ed.), pp. 309–315, 20–23 June, IEEE Press, Portland, OR.

79 Katare S, Bhan A, Caruthers JM, Delgass WN & Venkatasubramanian V (2004) A hybrid genetic algorithm for efficient parameter estimation of large kinetic models. *Comput Chem Eng* **28**, 2569–2581.

80 Kennedy J (1998) The behavior of particles. In *EP '98: Proceedings of the 7th International Conference on Evolutionary Programming VII* (Porto VW, Saravanan N, Waagen D & Eiben AE, eds.), pp. 581–589, Springer-Verlag, London, UK.

81 Kennedy J & Eberhart RC (2001) *Swarm Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA.

82 Jaeger J, Surkova S, Blagov M, Janssens H, Kosman D, Kozlov KN, Myasnikova E, Vanario-Alonso CE, Samsonova M, Sharp DH, & Reinitz J (2004) Dynamic control of positional information in the early *Drosophila* embryo. *Nature* **430**, 368–371

83 Laguna M & Marti R (2005) Experimental testing of advanced scatter search design for global optimization of multimodal functions. *J. Global Optim* **33**, 235–255.

84 Rodriguez-Fernandez M, Egea JA & Banga JR (2006) Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems. *BMC Bioinformatics* **7**, 483.

85 Egea JA, Rodríguez-Fernández M, Banga JR & Martí R (2007) Scatter search for chemical and bio-process optimization. *J Glob Optim* **37**, 481–503.

86 van Riel NAW (2006) Dynamic modelling and analysis of biochemical networks: mechanism-based models and model-based experiments. *Brief Bioinform* **7**, 364–374.

Supporting information

The following supplementary material is available:
Doc. S1. Systems biology: parameter estimation for biochemical models.

This supplementary material can be found in the online version of this article.

Please note: Wiley-Blackwell is not responsible for the content or functionality of any supplementary materials supplied by the authors. Any queries (other than missing material) should be directed to the corresponding author for the article.