JOURNAL OF THE ROYAL SOCIETY Interface

J. R. Soc. Interface (2012) **9**, 1925–1933
doi:10.1098/rsif.2011.0767
Published online 18 January 2012

Squeeze-and-breathe evolutionary Monte Carlo optimization with local search acceleration and its application to parameter fitting

Mariano Beguerisse-Díaz[1,2,*], Baojun Wang[1], Radhika Desikan[2] and Mauricio Barahona[1,*]

[1]*Department of Mathematics, and* [2]*Department of Life Sciences, Imperial College London, London SW7 2AZ, UK*

Estimating parameters from data is a key stage of the modelling process, particularly in biological systems where many parameters need to be estimated from sparse and noisy datasets. Over the years, a variety of heuristics have been proposed to solve this complex optimization problem, with good results in some cases yet with limitations in the biological setting. In this work, we develop an algorithm for model parameter fitting that combines ideas from evolutionary algorithms, sequential Monte Carlo and direct search optimization. Our method performs well even when the order of magnitude and/or the range of the parameters is unknown. The method refines iteratively a sequence of parameter distributions through local optimization combined with partial resampling from a historical prior defined over the support of all previous iterations. We exemplify our method with biological models using both simulated and real experimental data and estimate the parameters efficiently even in the absence of *a priori* knowledge about the parameters.

Keywords: parameter fitting; optimization; evolutionary algorithms; ordinary differential equation models; Monte Carlo methods

1. INTRODUCTION

The increasing drive towards quantitative technologies in biology has brought with it a renewed interest in the modelling of biological systems. Models of biological systems and other complex phenomena are generally nonlinear with uncertain parameters, many of which are often unknown and/or unmeasurable [1,2]. Crucially, the values of the parameters dictate not only the quantitative but also the qualitative behaviour of such models [3,4]. A fundamental task in quantitative and systems biology is to use experimental data to infer parameter values that minimize the discrepancy between the behaviour of the model and experimental observations. The parameters thus obtained can then be cross-validated against unused data before employing the fitted model as a predictive tool [2]. Ideally, this process could help close the modelling experiment loop by: suggesting specific experimental measurements; identifying relevant parameters to be measured; or discriminating between alternative models [5–7].

The problem of parameter estimation and data fitting is classically posed as the minimization of a cost function (i.e. the error) [8]. In the case of overdetermined linear systems with quadratic error functions, this problem leads to least-square solutions, convex optimizations that can be solved efficiently and globally based on the singular value decomposition of the covariance matrix of the data [9]. However, data fitting in nonlinear systems with small amounts of data remains difficult, as it usually leads to non-convex optimizations with many local minima [10].

A classic case in biological modelling is the description of the time evolution of a system through ordinary differential equations (ODEs), usually based on mechanistic functional forms. Examples include models of biochemical reactions, infectious spread and neuronal dynamics [1,11]. Typically, optimal parameters of the nonlinear ODEs must be inferred from experimental time courses but the associated optimization is far from straightforward. Standard optimization techniques that require an explicit cost function are unsuitable for this problem because of the difficulty in obtaining full analytical solutions for nonlinear ODEs [4,12,13]. Spline-based methods, which approximate the solution through an implicit integration of the differential equation [10], require linearity in the parameters and are therefore not applicable to models with nonlinear parameter dependencies, e.g. Michaelis–Menten and Hill kinetics.

Implicit techniques, such as direct search methods [14], simulated annealing [15], evolutionary algorithms

Authors for correspondence (m.beguerisse-diaz08@imperial.ac.uk; m.barahona@imperial.ac.uk).

Electronic supplementary material is available at http://dx.doi.org/10.1098/rsif.2011.0767 or via http://rsif.royalsocietypublishing.org.

Received 8 November 2011
Accepted 16 December 2011

(EAs) [16,17] or sequential Monte Carlo (SMC) [18], do not require an explicit cost function. However, if as is usually the case, the cost function is a complicated (hyper)surface in parameter space with many local minima; gradient and direct search methods tend to get trapped in local minima because of their use of local information. Although still a local method, simulated annealing alleviates some of the problems related to local minima through the use of stochasticity. However, this comes at the cost of high computational overhead and slow convergence and, yet, with no guarantee of finding the global minimum.

Instead of an optimization based on local criteria, EAs produce an ensemble of possible answers and evolve them globally through random mutation and cross-over followed by ranking and culling of the worst solutions [16,17,19]. This heuristic has been shown to provide an efficient protocol for parameter fitting in the life sciences [20,21]. However, EA methods can be inefficient when the feasible region in parameter space is too large, a case typical of models with large uncertainty in the parameters.

Probabilistic methods, such as SMC [18], propose a different conceptual framework. Rather than finding a *unique* optimal parameter set, SMC maps a prior probability distribution of the parameters onto a posterior constructed from samples with low errors until reaching a converged posterior. Recently, SMC has been combined with approximate Bayesian computation (ABC) and applied to data fitting and model selection [22]. However, methods such as ABC–SMC are not only computationally expensive but also require that the starting prior include the *true* value of the parameters. This requirement dents its applicability to many biological models, in which not even the order of magnitude of the parameters is known. In that case, the support of the starting priors must be made overly large (leading to extremely slow convergence) in order to avoid the risk of excluding the true parameter value from the search space.

In this work, we present a novel optimization algorithm for data fitting that takes inspiration from EA, SMC and direct search optimization. Our method iterates and refines samples from a probability distribution of the parameters in a 'squeeze-and-breathe' sequence. At each iteration, the probability distribution is 'squeezed' by the consecutive application of local optimization followed by ranking and culling of the local optima. The parameter distribution is then allowed to 'breathe' through a random update from a historical prior that includes the union of all past supports of the solutions (figure 1). This iteration proceeds until convergence of the distribution of solutions and their average error. A key, distinctive feature of our algorithm is the accelerated step-to-step convergence through a combination of local optimization and of culling of local solutions. Importantly, the method can also find parameters that lie outside of the range of the initial prior, and can deal with parameter values that extend across several orders of magnitude. We now provide definitions and a full description of our algorithm and showcase its applicability to different biological models of interest.

2. ALGORITHM

2.1. Formulation of the problem

Let $\mathbf{X}(t) = [x_1(t), \ldots, x_d(t)]$ denote the state of a system with d variables at time t. The time evolution of the state is described by a system of (possibly nonlinear) ODEs:

$$\dot{\mathbf{X}} = f(\mathbf{X}, t; \boldsymbol{\theta}). \tag{2.1}$$

Here, $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_N]$ is the vector of N parameters of our model.

The experimental dataset is formed by M observations of some of the variables of the system:

$$\mathcal{D} = \{\tilde{\mathbf{X}}(t_i) | i = 1, \ldots, M\}, \tag{2.2}$$

where $\tilde{\mathbf{X}}(t_i)$ corresponds to the real value of the system plus observational error. Ideally, $M > 2N + 1$ as $2N + 1$ experiments are enough for unequivocal identification of an ODE model with N parameters when no measurement error is present [23].

The *cost function* (i.e. the error) to be minimized is:

$$E_{\mathcal{D}}(\boldsymbol{\theta}) = \sum_{i=1}^{M} \parallel \mathbf{X}(t_i; \boldsymbol{\theta}) - \tilde{\mathbf{X}}(t_i) \parallel, \tag{2.3}$$

where $\| \cdot \|$ is a relevant vector norm. A standard choice is the Euclidean norm (or 2-norm) that corresponds to the sum of squared errors:

$$E^{(2)}_{\mathcal{D}}(\boldsymbol{\theta}) = \sum_{i=1}^{M} \sum_{j=1}^{d'} (X_j(t_i; \boldsymbol{\theta}) - \tilde{X}_j(t_i))^2, \tag{2.4}$$

where we assume that d' variables are observed. The cost function $E_{\mathcal{D}}$: $\mathbb{R}^N \to \mathbb{R}_+$ maps an N-dimensional parameter vector onto its corresponding error, thus quantifying how far the data and the model predictions are for that particular parameter set.

The aim of the data-fitting procedure is to find the parameter vector $\boldsymbol{\theta}^{**}$ that minimizes the error globally subject to restrictions dictated by the problem of interest:

$$\boldsymbol{\theta}^{**} = \min_{\boldsymbol{\theta}} E_{\mathcal{D}}(\boldsymbol{\theta}), \quad \text{subject to constraints on } \boldsymbol{\theta}. \tag{2.5}$$

2.2. Definitions

— *Data set*: $\mathcal{D}$, a set of M observations, as defined in equation (2.2).
— *Parameter set*: $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_N] \in \mathbb{R}^N_+$. Owing to the nature of the models considered, $\theta_i \geq 0$, $\forall i$.
— *Objective function*: $E_{\mathcal{D}}(\boldsymbol{\theta})$, the error function to be minimized, as defined in equation (2.4).
— *Set of local minima of* $E_{\mathcal{D}}(\boldsymbol{\theta})$: $\mathbb{M} = \{\boldsymbol{\theta}^* | E_{\mathcal{D}}(\boldsymbol{\theta}^*) \leq E_{\mathcal{D}}(\boldsymbol{\theta}), \forall \boldsymbol{\theta} \in \mathcal{N}(\boldsymbol{\theta}^*)\}$, where $\mathcal{N}(\boldsymbol{\theta}^*)$ is a neighbourhood of $\boldsymbol{\theta}^*$.
— *Global minimum of* $E_{\mathcal{D}}(\boldsymbol{\theta})$: $\boldsymbol{\theta}^{**}$, a parameter set such that $E_{\mathcal{D}}(\boldsymbol{\theta}^{**}) \leq E_{\mathcal{D}}(\boldsymbol{\theta})$, $\forall \boldsymbol{\theta}$. Clearly, $\boldsymbol{\theta}^{**} \in \mathbb{M}$.
— *Local minimization mapping*: $L : \mathbb{R}^N_+ \to \mathbb{M}$. Local minimization maps $\boldsymbol{\theta}$ onto a local minimum: $L(\boldsymbol{\theta}) = \boldsymbol{\theta}^* \in \mathbb{M}$.
— *Ranking and culling of local minima*: $\{\boldsymbol{\theta}^\dagger\}_1^B = \mathcal{RC}_B(\{\boldsymbol{\theta}\}_1^J)$. This operation ranks J parameter sets and selects the B parameter sets with the lowest $E_{\mathcal{D}}$.

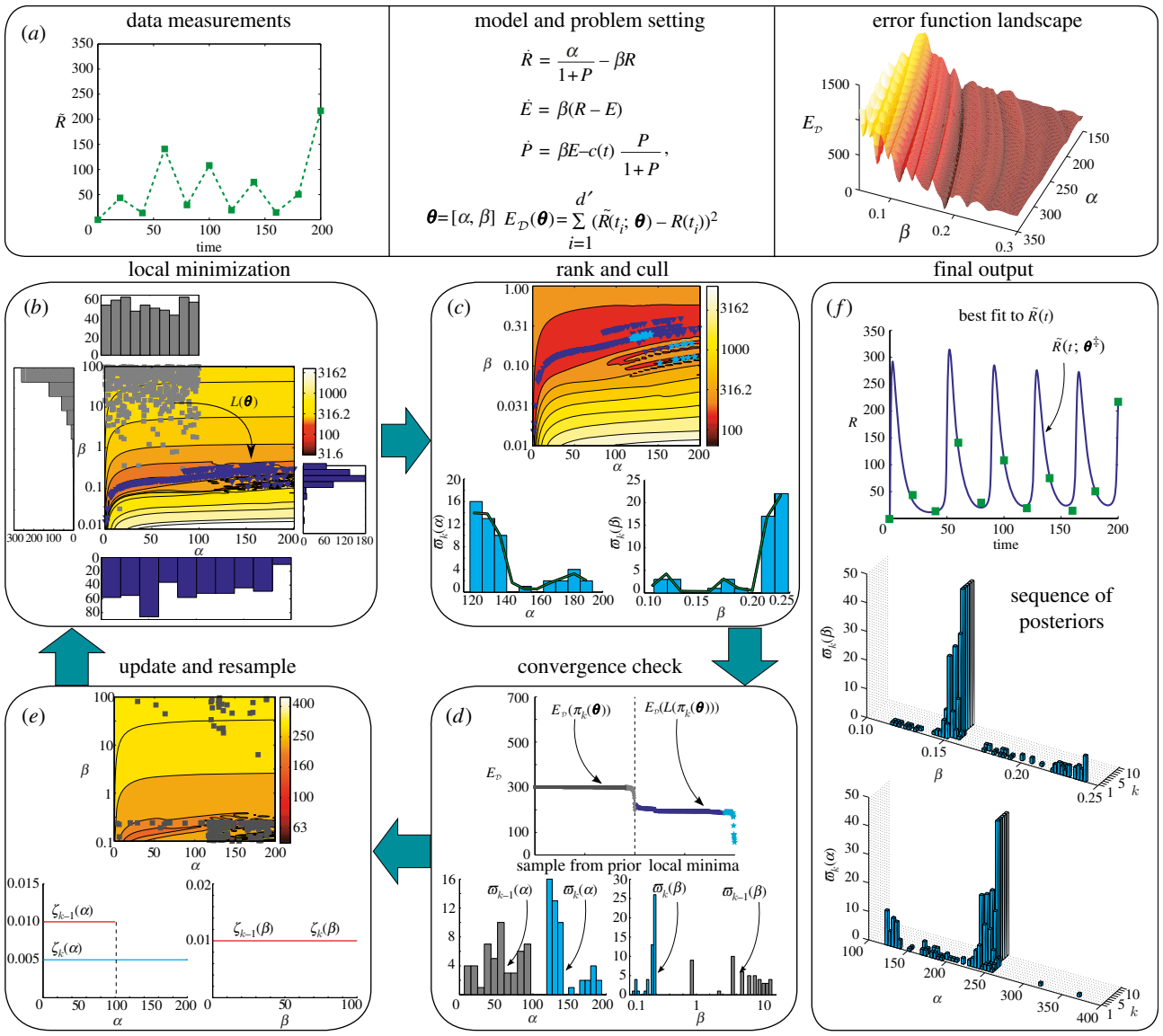

Figure 1. Steps of algorithm 1 exemplified through the BPM model (3.1). (*a*) The problem is defined by the dataset, the model and the error function to be minimized. Note the rugged landscape of the error function in the parameter plane (α, β), with many local minima. (*b*) In the first iteration, we simulate J points in parameter space from the uniform initial prior $\boldsymbol{\pi}_0(\boldsymbol{\theta})$ (squares in plot, top and left histograms) which are then minimized locally with a Nelder–Mead algorithm $L(\boldsymbol{\theta})$ (triangles in plot, bottom and right histograms). The local optimization aligns the parameter sets onto the level curves of $E_{\mathcal{D}}$. (*c*) The B best local minima (top, light stars) are selected and considered to be samples from the posterior distribution (bottom, light histograms). (*d*) Convergence of the error of the samples (top plot on the right, B lowest minima are the light stars) and of the posterior distributions (bottom, lighter histograms) are checked against the errors of the sample (top plot on the left) and the priors (bottom, darker histograms). (*e*) If convergence is not achieved, the historical prior is updated (previous historical prior in bold, updated in light) and a new set of J points are simulated from the posterior with probability p_m and from the historical prior with probability $1 - p_m$ (squares in plot). This new sample is fed back to the local minimization step (*b*). (*f*) The algorithm stops when convergence is reached (after nine iterations, in this case) providing an optimal parameter set $\boldsymbol{\theta}^{\ddagger}$ and a time course (top) and the sequence of optimized posteriors at each iteration (bottom). (Online version in colour.)

— *Joint probability distributions of the parameters at iteration k*: $\boldsymbol{\pi}_k(\boldsymbol{\theta})$ (prior) and $\boldsymbol{\varpi}_k(\boldsymbol{\theta})$ (posterior).
— *Marginal probability distribution of the ith component of* $\boldsymbol{\theta}$: for instance, $\pi(\theta_i) = \int \boldsymbol{\pi}(\boldsymbol{\theta}) \prod_{r \neq i} \mathrm{d}\theta_r$.
— *Historical prior at iteration k*: $\boldsymbol{\zeta}_k(\boldsymbol{\theta}) = \prod_{i=1}^{N} \zeta_k(\theta_i)$ where

$$\zeta_k(\theta_i) \sim U(\min(\Im_k(\theta_i)), \max(\Im_k(\theta_i))). \tag{2.6}$$

Here $U(a, b)$ is a uniform distribution with support in $[a, b]$ and $\Im_k(\theta_i) = \zeta_{k-1}^{-1} \cup \boldsymbol{\varpi}_k^{-1}$ is the union of the supports of $\boldsymbol{\varpi}_k(\theta_i)$ and $\zeta_{k-1}(\theta_i)$.

— *Update of the prior at iteration k*: $\boldsymbol{\pi}_k(\boldsymbol{\theta}) = \prod_{i=1}^{N} \boldsymbol{\pi}_k(\theta_i)$ with

$$\boldsymbol{\pi}_k(\theta_i) \sim p_m \boldsymbol{\varpi}_k(\theta_i) + (1 - p_m)\zeta_k(\theta_i), \tag{2.7}$$

that is, a convex mixture of the posterior and the historical prior with weight p_m, from which a new population is sampled in iteration $k + 1$.
— *Re-population*: obtain population of J random points simulated from the prior $\boldsymbol{\pi}_{k-1}(\boldsymbol{\theta})$.
— *Convergence criterion for the error*: the difference between the means of the errors of the posteriors

in consecutive iterations is smaller than the predetermined tolerance:

$$\phi_k = \overline{E_{\mathcal{D}}(\varpi_{k-1}(\boldsymbol{\theta}))} - \overline{E_{\mathcal{D}}(\varpi_k(\boldsymbol{\theta}))} < Tol. \qquad (2.8)$$

— *Convergence criterion for the empirical distributions*: the samples of the posteriors in consecutive iterations are indistinguishable at the 5 per cent significance level according to the non-parametric Mann–Whitney rank sum test:

$$\mathcal{MW}(\varpi_k(\boldsymbol{\theta}), \varpi_{k-1}(\boldsymbol{\theta})) = 0. \qquad (2.9)$$

2.3. Description of the algorithm

Algorithm 1 presents the pseudo-code for our method using the definitions above. The iterations produce progressively more refined distributions of the parameter vector. At each iteration k, a population simulated from the prior distribution $\pi_{k-1}(\boldsymbol{\theta})$ is locally minimized followed by ranking and culling of the local minima to create a posterior distribution $\varpi_k(\boldsymbol{\theta})$ (squeeze step). This distribution is then combined with an encompassing historical prior to generate the updated prior $\pi_k(\boldsymbol{\theta})$ (breathe step). The iteration loop terminates when the difference between the mean errors of consecutive posteriors is smaller than the tolerance and the samples of the posteriors are indistinguishable. We now explain these steps in detail (figure 1) through the Bliss–Painter–Marr (BPM) model (see 3.1).

— *Formulation of the optimization*: the dataset $\mathcal{D}$ and the model equations parametrized by $\boldsymbol{\theta}$ allow us to define an error function $E_{\mathcal{D}}(\boldsymbol{\theta})$ whose global minimum corresponds to the best model.

In our illustrative example, the BPM model (3.1) has the parameter vector $\boldsymbol{\theta} = [\alpha, \beta]$ and the error function is depicted in figure 1*a*. The global optimization on the rugged landscape of this function is computationally hard.

— *Initialization:*

(i) Set the running parameters of the algorithm: the size of the simulated population, J; the size of the surviving population after culling, B; the update probability, p_m; and the tolerance, *Tol*. In this example, $J = 500$, $B = 50$, $p_m = 0.95$ and $Tol = 10^{-5}$.

(ii) Choose $\pi_0(\boldsymbol{\theta})$, the initial prior distribution of the parameter vector. In this case, we take α and β to be independent and uniformly distributed: $\pi_0(\boldsymbol{\theta}) \sim U(0, 100) \times U(0, 100)$.

(iii) Initialize $\zeta_0(\boldsymbol{\theta}) = \pi_0(\boldsymbol{\theta})$, the historical prior of the parameters.

(iv) Simulate J points from $\pi_0(\boldsymbol{\theta})$ to generate the initial sample $\{\hat{\boldsymbol{\theta}}_0\}_1^J$.

— *Iteration* (*step k*): repeated until termination criterion is satisfied. Figure 1 shows the first iteration of our method applied to the BPM example.

(i) *Local minimization*: apply local minimization to the simulated parameters from the 'prior' $\{\hat{\boldsymbol{\theta}}_{k-1}\}_1^J$ and map them onto local minima of $E_{\mathcal{D}}(\boldsymbol{\theta})$ to generate $\{L(\hat{\boldsymbol{\theta}}_{k-1})\}_1^J \in \mathbb{M}$.

Here, we use the Nelder–Mead simplex method [24], though others can be used. Figure 1*b* shows the simulated points from $\pi_0(\boldsymbol{\theta})$ (squares in plot) and its corresponding histograms (top and left). After local minimization, this sample is mapped onto the dark triangles in figure 1*b* (dark histograms bottom and right). Note how the local minima align with the level curves of $E_{\mathcal{D}}$ with a markedly different distribution to the uniform prior. Note also that many of the optimized values of α lie outside the range of the prior (0, 100) and are now distributed over the interval (0, 200). On the other hand, the values of β have collapsed inside (0, 1).

Algorithm 1. Squeeze-and-breathe optimization.

Set running parameters of algorithm:
B, $J \in \mathbb{N}$, $p_m \in [0, 1]$, *Tol*
Choose initial priors $\pi_0(\boldsymbol{\theta})$ and $\zeta_0(\boldsymbol{\theta})$.
Set $\mathcal{H}_0 = \emptyset$ and $k \leftarrow 1$.
repeat
  Let $\mathcal{H}_k = \mathcal{H}_{k-1}$.
  Simulate J points from $\pi_{k-1}(\boldsymbol{\theta})$ through re-population.
  for $\ell = 1 \rightarrow J$ **do**
    Obtain local minimum $\hat{\boldsymbol{\theta}}_\ell^* = L(\boldsymbol{\theta}_\ell)$.
    Store the pair $[\hat{\boldsymbol{\theta}}_\ell^*, E_{\mathcal{D}}(\hat{\boldsymbol{\theta}}_\ell^*)]$ in $\mathcal{H}_k$.
  end for
  Rank and cull the set of local minima:
  $\mathcal{H}_k = \mathcal{RC}_B(\mathcal{H}_k)$
  Define the posterior $\varpi_k(\boldsymbol{\theta})$ from the sample $\mathcal{H}_k$.
  Update $\zeta_k(\boldsymbol{\theta})$ from $\zeta_{k-1}(\boldsymbol{\theta})$ and $\varpi_k(\boldsymbol{\theta})$.
  Update the prior:
  $\pi_k(\boldsymbol{\theta}) \sim p_m \varpi_k(\boldsymbol{\theta}) + (1 - p_m)\zeta_k(\boldsymbol{\theta})$.
  $k \leftarrow k + 1$.
until $\phi_k < Tol$ and $\mathcal{MW}(\varpi_k(\boldsymbol{\theta}), \varpi_{k-1}(\boldsymbol{\theta})) = 0$

(ii) *Ranking and culling*: rank the $J + B$ local minima from the $k - 1$ and k iterations, select the B points with the lowest $E_{\mathcal{D}}$ and cull (discard) the rest:

$$\mathcal{RC}_B(\{L(\hat{\boldsymbol{\theta}}_{k-1})\}_1^J \cup \{\hat{\boldsymbol{\theta}}_{k-1}^\dagger\}_1^B) = \{\hat{\boldsymbol{\theta}}_k^\dagger\}_1^B.$$

We consider $\{\hat{\boldsymbol{\theta}}_k^\dagger\}_1^B$ to be a sample from the optimized ('posterior') distribution, $\varpi_k(\boldsymbol{\theta})$ and we denote the best parameter vector of this set as

$$\hat{\boldsymbol{\theta}}_k^\ddagger = \min_{E_{\mathcal{D}}}(\{\hat{\boldsymbol{\theta}}_k^\dagger\}_1^B).$$

The $B = 50$ best parameter sets are shown (light stars in plot) in figure 1*c* (bottom histograms).

(iii) *Termination criterion*: check that the difference between the mean errors of the consecutive optimized samples is smaller than the tolerance: $\phi_k \leq Tol$. We also gauge the 'convergence' of the posteriors through the Mann–Whitney (MW) test to determine if the samples from consecutive posteriors are distinguishable:

$$\mathcal{MW}(\varpi_{k-1}(\boldsymbol{\theta}), \varpi_k(\boldsymbol{\theta})) \equiv \mathcal{MW}(\{\hat{\boldsymbol{\theta}}_{k-1}^\dagger\}_1^B, \{\hat{\boldsymbol{\theta}}_k^\dagger\}_1^B),$$

where $\mathcal{MW}$ is a 0-1 flag. The MW test gives additional information about the change of the

optimized posteriors from one iteration to the next.

Figure 1*d* shows the convergence check for the first iteration of the BPM model: (i) top—errors of the sampled prior (left) with errors of the local minima (right) and the B surviving points (light stars); (ii) bottom—histograms of the prior and the posterior. Clearly, in this iteration neither the error nor the distributions have converged and so the algorithm does not stop.

(iv) *Update of historical prior and generation of new sample*: if convergence is not achieved, update the historical prior $\zeta_k(\boldsymbol{\theta})$ as a uniform distribution over the union of the supports of the existing historical prior and the calculated posterior (2.6). Equivalently, the support of the historical prior extends over the union of the sequence of all historical priors $\{\zeta_0(\boldsymbol{\theta}), \ldots, \zeta_{k-1}(\boldsymbol{\theta})\}$ and of all posteriors $\{\boldsymbol{\varpi}_1(\boldsymbol{\theta}), \ldots, \boldsymbol{\varpi}_k(\boldsymbol{\theta})\}$.

As shown in figure 1*e* for the BPM example, the marginal of the historical prior for α is expanded to $U(0, 200)$, as the optimized parameter sets have reached values as high as 200. Meanwhile, the β marginal of the historical prior remains unchanged as $U(0, 100)$ because there has been no expansion of the support.

The historical prior is used to mutate the updated prior before the next iteration by constructing a weighted mixture of the posterior and the historical prior with weight p_m, as shown in (2.7). We re-populate from this updated prior by simulating from the posterior with probability $p_m = 0.95$ and from the historical prior with probability $(1 - p_m)$ to generate the new sample $\{\hat{\boldsymbol{\theta}}_k\}_1^J$ and iterate back.

Figure 1*e* shows the sample of J points simulated from the new prior. The α-components of most points are between 100 and 200 and the β-components are between 0.1 and 1, but there are a few that lie outside the support of the posterior. The process in figure 1(*b*–*e*) is iterated for this new set of points.

— *Output of the algorithm*: when the convergence criteria have been met, the iteration stops at iteration k^* and the minimum of this last iteration, $\boldsymbol{\theta}^{\ddagger}_{k^*}$, is presented as the optimal parameter set for the model (i.e. the estimation of the global minimum $\boldsymbol{\theta}^{**}$ provided by the algorithm). We can also examine the sequence of optimized parameter distributions $\{\boldsymbol{\varpi}_1(\boldsymbol{\theta}), \ldots, \boldsymbol{\varpi}_{k^*}(\boldsymbol{\theta})\}$ obtained for all iterations (figure 1*f*).

3. APPLICATION TO BIOLOGICAL EXAMPLES

We apply our algorithm to three biological examples of interest. The first two correspond to simulated data from models in the literature, while in the third example, we apply our algorithm to unpublished experimental data of the dynamical response of an inducible genetic promoter constructed for an application in synthetic biology.

3.1. Bliss–Painter–Marr model of gene–product regulation

The BPM model [25] describes the behaviour of a gene–enzyme–product control unit with a negative feedback loop:

$$\left.\begin{aligned} \dot{R} &= \frac{\alpha}{1+P} - \beta R, \\ \dot{E} &= \beta(R - E) \\ \text{and} \qquad \dot{P} &= \beta E - c(t)\frac{P}{1+P}. \end{aligned}\right\} \tag{3.1}$$

Here, R, E and P are the concentrations (in arbitrary units) of mRNA, enzyme and product, respectively. The degradation rate of the product has an explicit time dependence, which in this case has the form of a ramp saturation:

$$c(t) = \begin{cases} 5 + 0.2t & 0 \le t < 50, \\ 15 & t \ge 50. \end{cases}$$

The model represents a gene that codes for an enzyme which in turn catalyses a product that inhibits the transcription of the gene. This self-inhibition can lead to oscillations, which have been shown to occur in the tryptophan operon in *Escherichia coli* [25].

We construct a dataset from simulations of this model with $\boldsymbol{\theta}_{\text{real}} = [\alpha, \beta] = [240, 0.15]$ and initial conditions $R(0) = E(0) = P(0) = 0$. The dataset $\mathcal{D}$ consists of 10 measurements of $R(t)$ at particular times with added Gaussian noise drawn from $\mathcal{N}(0, 15^2)$ (table 1 in the electronic supplementary material). The error function $E_{\mathcal{D}}(\boldsymbol{\theta})$ (2.4) corresponds to a non-convex optimization landscape:[1] a complex rugged surface with many local minima making global optimization hard (figure 1*a*).

We use algorithm 1 to estimate the 'unknown' parameter values from the 'measurements' of R, as illustrated in §2*c* and figure 1. Feigning ignorance of the true values, we choose a uniform prior distribution with range [0, 100] for both parameters: $\boldsymbol{\pi}_0(\boldsymbol{\theta}) \sim [U(0, 100), U(0, 100)]$. The rest of the parameters are set to: $J = 500$, $B = 50$, $p_m = 0.95$ and $Tol = 10^{-5}$. Note that the *true* value of α falls outside of the assumed range of our initial prior, while the range of β in our initial prior is two orders of magnitude larger than its true value. This level of uncertainty about parameter values is typical in data fitting for biological models.

Figure 1 highlights a key aspect of our algorithm: the local minimization can lead to local minima outside of the range of the initial prior. Furthermore, our definition of the historical prior ensures that successive iterations can find solutions within the largest hypercube of optimized solutions in parameter space. In this example, the algorithm moves away from the $U(0, 100)$ prior for α and finds a distribution around 240 (the true value) after three iterations, while in the case of β, the distribution collapses to values around 0.15 after one iteration. Although the algorithm finds the minimum $\boldsymbol{\theta}^{\ddagger}$ after five iterations, the algorithm is terminated after nine iterations, when the posterior distributions are

[1]We thank Markus Owen of the University of Nottingham for suggesting this example.

Table 1. Results of the fitting of the BPM model with algorithm 1: smallest error of iteration k; the best values $\boldsymbol{\alpha}_k^{\ddagger}$ and $\boldsymbol{\beta}_k^{\ddagger}$; whether the distributions have converged; and the difference of the mean errors of the optimized population.

k	min. error	$\boldsymbol{\alpha}_k^{\ddagger}$	$\boldsymbol{\beta}_k^{\ddagger}$	conv. $\boldsymbol{\varpi}_k(\boldsymbol{\alpha})$	conv. $\boldsymbol{\varpi}_k(\boldsymbol{\beta})$	$\boldsymbol{\phi}_k$
1	56.0941	193.7447	0.1304	—	—	—
2	28.2735	246.7510	0.1528	no	no	133.9020
3	27.2083	248.7557	0.1532	no	no	6.8542
4	26.9838	250.3593	0.1536	no	no	0.6532
5	26.6504	251.7189	0.1538	no	no	0.3281
6	26.6504	251.7189	0.1538	no	no	0.1963
7	26.6504	251.7189	0.1538	yes	yes	0.0118
8	26.6504	251.7189	0.1538	no	no	0.0131
9	26.6504	251.7189	0.1538	yes	yes	1.414×10^{-6}

Table 2. Parameter values obtained from *gfp*-30 and *gfp*-34 data. In the study of Wang [26], only the steady state solution was used. Hence, only the ratio of k_1 and d can be estimated.

	Wang [26]		algorithm 1	
parameter	*gfp*-30	*gfp*-34	*gfp*-30	*gfp*-34
$\boldsymbol{\alpha}^{\ddagger}$	0.0012 ± 0.027	1.4720×10^{-9}	0.0043	0.0024
$k_1^{\ddagger}$	n.a.	n.a.	76.1354	63.6650
$n_1^{\ddagger}$	1.3700 ± 0.270	1.3690 ± 0.021	1.4832	1.3879
$K_1^{\ddagger}$	0.2280 ± 0.039	0.2590 ± 0.021	0.2467	0.2641
$d^{\ddagger}$	n.a.	n.a.	0.0069	0.0052
$k_1^{\ddagger}/d^{\ddagger}$	9456 ± 487	7648 ± 152	10983.34	12163.04

similar (according to the MW test) and the mean errors have also converged (table 1). The estimated parameters for this noisy dataset are $\boldsymbol{\theta}_{k^*}^{\ddagger} = [251.7189, 0.1530]$. In fact, the error of the estimated parameter set is lower than that of the real parameters: $E_{\mathcal{D}}(\boldsymbol{\theta}^{\ddagger}) = 26.65 < E_{\mathcal{D}}(\boldsymbol{\theta}_{\text{real}}) = 28.26$, because of the noise introduced in the data. When a dataset without noise is used, the algorithm finds the true value of the parameters to nine significant digits (not shown).

3.2. Susceptible–Infected–Recovered epidemics model

Susceptible–Infected–Recovered (SIR) models are widely used in epidemiology to describe the evolution of an infection in a population [11]. In its simplest form, the SIR model has three variables: the susceptible population S, the infected population I and the recovered population R:

$$\left.\begin{aligned} \dot{S} &= \alpha - (\gamma I + d)S, \\ \dot{I} &= (\gamma S - v - d)I \\ \text{and} \qquad \dot{R} &= vI - R. \end{aligned}\right\} \tag{3.2}$$

The first equation describes the change in the susceptible population, growing with birth rate α and decreasing by the rate of infection γIS and the rate of death dS. The infected population grows by the rate of infection γIS and decreases by the rate of recovery vI and the rate of death dI. The recovered population grows by the rate of recovery vI and decreases by the death rate dR. Here, we use the same form of the equations as performed by Toni *et al.* [22].

The data generated from model (3.2) (table 2 in the electronic supplementary material) were obtained directly from the study of Toni *et al.* [22]. Hence, the original parameter values were not known to us and further we assumed the initial conditions also to be unknown and fitted them as parameters. We used algorithm 1 to estimate α, γ, v and d, and initial conditions S_0, I_0 and R_0. The prior marginal distributions for all parameters were set as $U(0, 100)$. The other parameters were set to: $J = 1000$, $B = 50$, $p_m = 0.95$ and $Tol = 10^{-5}$. The algorithm converged after six iterations. Figure 2*a* shows the prediction of the model (3.2) with the best parameters estimated by our algorithm. The fit is good with little difference between the curves obtained using the real initial conditions and those estimated by our method.

The posterior distributions after six iterations of the algorithm are shown in figure 2*b*. The errors obtained after each local minimization in a decreasing order on each iteration are shown on a semi-logarithmic scale in figure 2*c*. We can observe how the errors decrease by several orders of magnitude over the first three iterations and converge steadily during the last three iterations until $\phi_k \leq Tol$.

3.3. An inducible genetic switch from synthetic biology

The use of inducible genetic switches is widespread in synthetic biology and bioengineering as building blocks for more complicated gene circuit architectures.

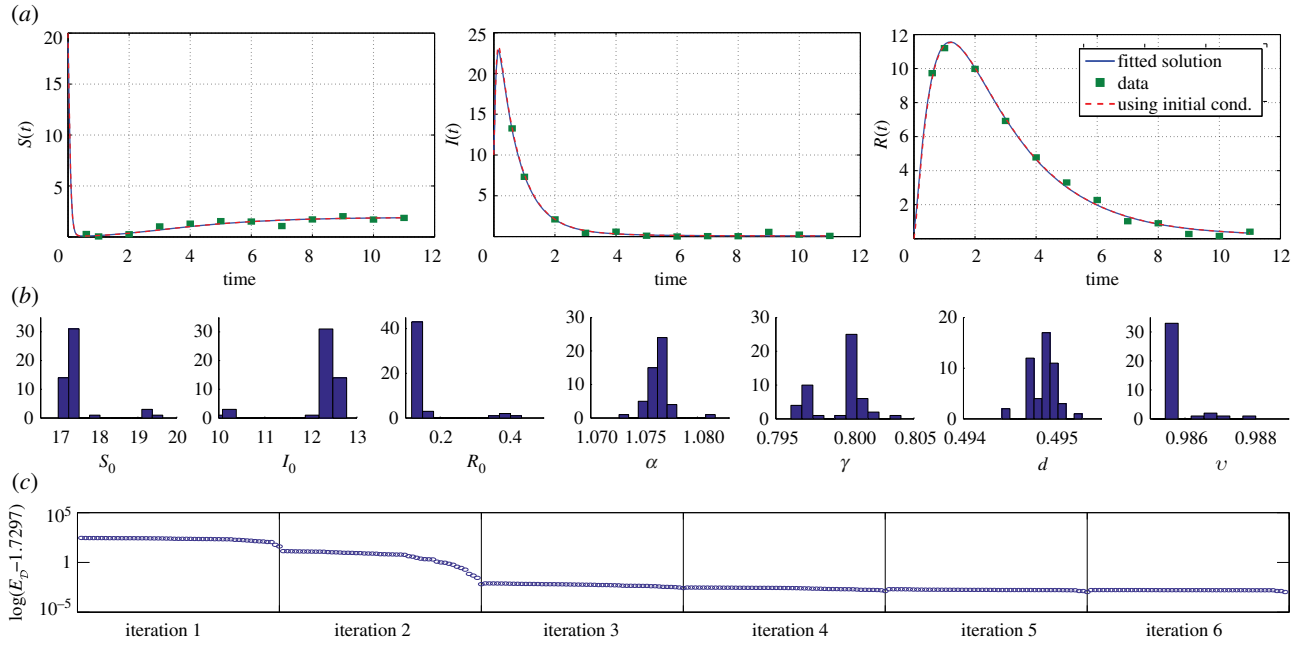

Figure 2. (*a*) Time courses of the SIR model (3.2). Squares are simulated 'data' points (table 2 and electronic supplementary material) and bold lines are the model fit with the best parameters $\alpha^{\ddagger} = 1.0726$, $\gamma^{\ddagger} = 0.7964$, $d^{\ddagger} = 0.4945$ and $\upsilon^{\ddagger} = 0.9863$ and the best-fit initial conditions $S_0^{\ddagger} = 19.1591$, $I_0^{\ddagger} = 10.3016$ and $R_0^{\ddagger} = 0.3861$. Dashed lines use the best-fit parameters and the real initial conditions. The minimum error is $E_{\mathcal{D}}(\boldsymbol{\theta}^{\ddagger}) = 1.7297$. Squares indicate data. (*b*) Histogram of the values of the 50 best parameters and initial conditions of the model obtained after convergence at six iterations. (*c*) Convergence of the error of the optimized samples at every iteration relative to the final error. (Online version in colour.)

An example is shown schematically in the inset of figure 3*a*. This environment-responsive switch is used to control the expression of a target gene G (usually tagged with green fluorescent protein or *gfp*) through the addition of an exogenous small molecule I_1 (e.g. isopropyl thiogalactopyranoside or IPTG). The input–output behaviour of this system can be described by the following ODE [2,27]:

$$\dot{G} = \alpha k_1 + \frac{k_1 I_1^{n_1}}{K_1^{n_1} + I_1^{n_1}} - dG. \tag{3.3}$$

Here, αk_1 is the basal activity of the promoter P_1 and dG is the linear degradation term. The second term is a Hill function that models the cooperative transcription activation in response to the inducer I_1 with maximum expression rate k_1, constant K_1 and Hill coefficient n_1.

The *lacI*–P_{lac} switch has been characterized experimentally with response to different doses of IPTG in different studies [26,28]. Equation (3.3) can be solved explicitly and one can use nonlinear least squares and the analytical solution to fit data at stationarity (i.e. at long times) and estimate α, n_1, K_1 and the ratio k_1/d. These estimates have been obtained, from an earlier study [28], assuming equilibrium ($\dot{G} = 0$) and initial condition $G(0) = 0$ (table 2).

In fact, the experiments measured time series of the expression of G every 20 min from $t = 140$ to 360 min for different doses of inducer $I_1 = 0$, 3.9×10^{-4}, 1.6×10^{-3}, 6.3×10^{-3}, 2.5×10^{-2}, 0.1, 0.4, 1.6, 6.4, 12.8 mM, with two different reporters (*gfp*-30 and *gfp*-34; see tables 3 and 4 in the electronic supplementary material). Instead of assuming equilibrium and using only the data for $t > 300$ min as done previously [28], we apply algorithm 1 to all the data with the full dynamical equation (3.3) to estimate $\boldsymbol{\theta} = [\alpha,\ k_1,\ n_1,\ K_1,\ d]$. In this case, we used initial priors $U(0,\ 1)$ for α and n_1; and $U(0,\ 20)$ for k_1, K_1 and d. The other parameters were set to: $J = 1000$, $B = 50$, $p_m = 0.95$ and $Tol = 10^{-5}$.

Our algorithm converged after five iterations to the parameter values in table 2. The parameter estimates provide good fits to both the time courses (figure 3*b*) and to the dose–response data (figure 3*a*). The values of $K_1^{\ddagger}$ and $n_1^{\ddagger}$ obtained here are similar to those obtained from the study of Wang [26] by using only stationary data. This is reassuring as these parameters are related to the dose threshold to half-maximal response and to the steepness of the sigmoidal response, both static properties. On the other hand, the values of α and the ratio k_1/d differ to some extent owing to the (imperfect) assumption by Wang [26] that steady state had been reached at $t = 300$ min. As figure 3*b* shows, G is not at steady state then. Hence, the parameter values obtained with our method should give a more faithful representation of the true dynamical response of the switch.

4. DISCUSSION

In this work, we have presented an optimization algorithm that brings together ingredients from EAs, local optimization and SMC. The method is particularly useful for determining parameters of ODE models from data. Our approach can also be used in other contexts where an optimization problem has to be solved on complex landscapes, or when the objective function cannot be written explicitly. The algorithm proceeds by generating

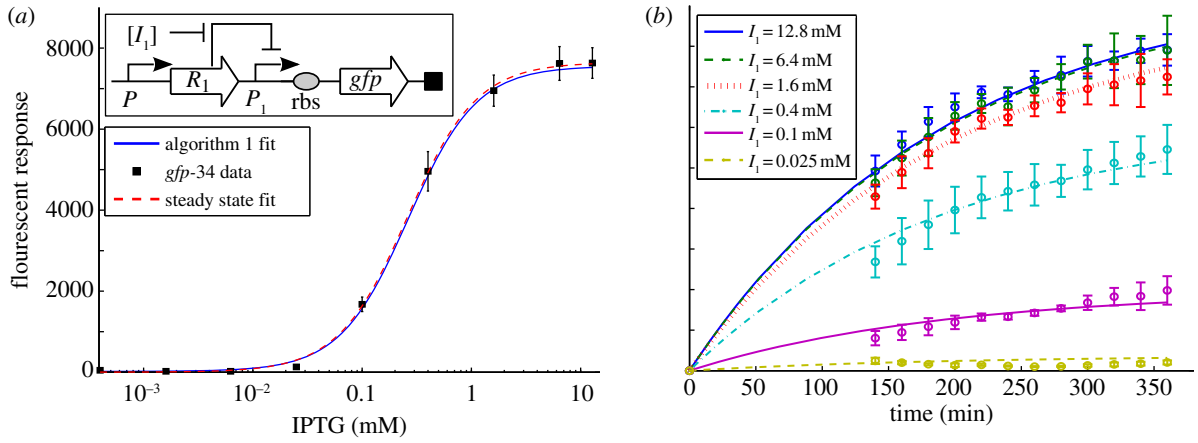

Figure 3. (*a*) Inset: an inducible genetic switch consisting of P_1, a negatively regulated environment-responsive promoter. The repressor R_1 promoted by P regulates P_1. The switch is responsive to an exogenous inducer I_1, which binds to R_1 to relieve its repression on P_1 and to turn on the transcription of the downstream target gene, such as a *gfp*. The ribosome-binding site (rbs) is used to tune the translation efficiency of the downstream gene. Plot: fluorescent response (arbitrary units) of the switch with *gfp*-34 to different doses of IPTG (squares). Solution of equation (3.3) using the parameters obtained with algorithm 1 (solid line), and stationary solution (dashed line). (*b*) Time course of the fluorescent response (arbitrary units) of the switch with *gfp*-34 to several doses of IPTG (circles) and time-dependent solutions of equation (3.3) using the parameters obtained with algorithm 1 (solid lines). Similarly good fits were obtained for responses to $I_1 = 0.0063$, 0.0016, 0.0004 and 0 mM (not shown). (Online version in colour.)

a population of solutions through Monte Carlo sampling from a prior distribution and refining those solutions through a combination of local optimization and culling. A new prior is then created as a mixture of a historical prior (which records the broadest possible range of solutions found) and the distribution of the optimized population. This iterative process induces a strong concentration of the Monte Carlo sampling through local optimization which accelerates convergence and increases precision, while at the same time the presence of the historical prior allows the possibility that solutions can be found outside of the initial presumed ranges for the parameter values.

We have illustrated the application of the algorithm to ODE models of biological interest and have found it to perform efficiently. The algorithm also works well when applied to larger problems with tens of parameters in a signal transduction model (paper in preparation). The efficiency of the algorithm hinges on selecting appropriate running parameters and priors. For instance, the number of samples from the prior J should be large enough to allow for significant sampling of the parameter space while small enough to limit the computational cost. We have found that simulating $J = 350$–500 points in models of up to 10 parameters and keeping the best 15 per cent of the local minima leads to termination within fewer than 20 iterations. In our implementation, the Nelder–Mead minimization is capped at 300 evaluations. These guidelines would result in up to 300 000 evaluations of the objective function per iteration. Therefore, our method can become computationally costly if the objective function is expensive to evaluate, e.g. in stiff models that are difficult to solve numerically. In essence, our algorithm proposes a trade-off: fewer but more costly iterations. It is important to remark that, as with any other optimization heuristic for non-convex problems, there are no strict guarantees of convergence to the global minimum. Therefore, it is always advisable to run the method with different starting points and different settings with enough sampling points in parameter space to check for consistency of the solutions obtained.

The generation of iterative samples of the parameters draws inspiration from Monte Carlo methods [6,18,22] but without pursuing the strict guarantees that the nested structure of the distributions provides in ABC–SMC. Our evolutionary approach adopts a highly focused Monte Carlo sampling driven by a sharp local search with culling. Hence, our iterative procedure generates samples that only reflect properties of the set of local minima (up to numerical cutoffs) without any focus on the global convergence of the distributions. As noted from the study of Toni and co-workers [22], the distributions of the parameters (both their sequence and the final distributions) give information about the sensitivity of the parameters: parameters with narrow support will be more sensitive than those with wider support. Future developments of the method will focus on establishing a suitable theoretical framework that facilitates its use in model selection. Broadening the choice of historical priors may be a way of establishing such framework. Currently, we make no assumptions about the parameter space, hence we use uniform distributions over the support of all the posteriors. However, other distributions (e.g. exponential or log-normal) may be considered as a way to bias the historical prior towards regions of particular interest. Other work will consider the possibility of incorporating a stochastic ranking strategy in the selection of solutions, similar to the one present in the SRES algorithm [17], in order to solve more general optimization problems with complex feasible regions.

The authors would like to thank C. Barnes, T. Ellis, E. Garduño, H. Harrington, M. Owen, M. Stumpf and S. Yaliraki for their comments and suggestions. M.B.D. is supported by a BBSRC-Microsoft Research Dorothy Hodgkin Postgraduate Award.

This work was partly supported by the US Office of Naval Research and by BBSRC through LoLa grant BB/G020434/1 (M.B.) and SABR grant BB/F005210/2 (M.B.), and by EPSRC through grant EP/I017267/1 under the *Mathematics underpinning the Digital Economy* programme (M.B.).

REFERENCES

1 Edelstein-Keshet, L. 1988 *Mathematical models in biology*. Classics in Applied Mathematics. Philadelphia, USA: SIAM.
2 Alon, U. 2007 *An introduction to systems biology: design principles of biological circuits*. London, UK: Chapman and Hall/CRC Mathematical and Computational Biology Series, Chapman & Hall/CRC.
3 Strogatz, S. H. 1994 *Nonlinear dynamics and chaos. With applications to physics, biology, chemistry, and engineering*. Studies in Nonlinearity. NY, USA: Perseus Books Group.
4 Brown, K. S. & Sethna, J. P. 2003 Statistical mechanical approaches to models with many poorly known parameters. *Phys. Rev. E* **68**, 021904.
5 Gutenkunst, R. N., Waterfall, J. J., Casey, F. P., Brown, K. S., Myers, C. R. & Sethna, J. P. 2007 Universally sloppy parameter sensitivities in systems biology models. *PLoS Comput. Biol.* **3**, e189. (doi:10.1371/journal.pcbi.0030189)
6 Toni, T. & Stumpf, M. P. H. 2010 Simulation-based model selection for dynamical systems in systems and population biology. *Bioinformatics* **26**, 104–110. (doi:10.1093/bioinformatics/btp619)
7 Yates, A., Chan, C. C. W., Callard, R. E., George, A. J. T. & Stark, J. 2001 An approach to modelling in immunology. *Brief Bioinform.* **2**, 245–257. (doi:10.1093/bib/2.3.245)
8 Gershenfeld, N. A. 1999 *The nature of mathematical modeling*. Cambridge, UK: Cambridge University Press.
9 Lawson, C. L. & Hanson, R. J. 1995 *Solving least squares problems*. Classics in Applied Mathematics. Philadelphia, USA: SIAM.
10 Brewer, D., Barenco, M., Callard, R., Hubank, M. & Stark, J. 2008 Fitting ordinary differential equations to short time course data. *Phil. Trans. R. Soc. A* **366**, 519–544. (doi:10.1098/rsta.2007.2108)
11 Anderson, R. M. & May, R. M. 1992 *Infectious diseases of humans dynamics and control*. Oxford, UK: Oxford University Press.
12 Chen, W. W., Niepel, M. & Sorger, P. K. 2010 Classic and contemporary approaches to modeling biochemical reactions. *Genes Dev.* **24**, 1861–1875. (doi:10.1101/gad.1945410)
13 Papachristodoulou, A. & Recht, B. 2007 Determining interconnections in chemical reaction networks. In *American Control Conference, New York, NY, USA, 9–13 July 2007*, pp. 4872–4877. Institute of Electrical and Electronics Engineers. (doi:10.1109/ACC.2007.4283084)
14 Powell, M. J. D. 1998 Direct search algorithms for optimization calculations. *Acta Numer.* **7**, 287–336. (doi:10.1017/S0962492900002841)
15 Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. 1983 Optimization by simulated annealing. *Science* **220**, 671–680. (doi:10.1126/science.220.4598.671)
16 Mitchell, T. M. 1997 *Machine learning*. New York, NY: McGraw-Hill.
17 Runarsson, T. P. & Yao, X. 2000 Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. Evol. Comput.* **4**, 284–294. (doi:10.1109/4235.873238)
18 Sisson, S. A., Fan, Y. & Tanaka, M. M. 2007 Sequential Monte Carlo without likelihoods. *Proc. Natl Acad. Sci. USA* **104**, 1760–1765. (doi:10.1073/pnas.0607208104)
19 Schwefel, H. P. 1995 *Evolution and optimum seeking*. Sixth-Generation Computer Technology Series. NY, USA: Wiley.
20 Moles, C. G., Mendes, P. & Banga, J. R. 2003 Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome Res.* **13**, 2467–2474. (doi:10.1101/gr.1262503)
21 Zi, Z. & Klipp, E. 2006 SBML-PET: a Systems Biology Markup Language-based parameter estimation tool. *Bioinformatics* **22**, 2704–2705. (doi:10.1093/bioinformatics/btl443)
22 Toni, T., Welch, D., Strelkowa, N., Ipsen, A. & Stumpf, M. P. H. 2009 Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *J. R. Soc. Interface* **6**, 187–202. (doi:10.1098/rsif.2008.0172)
23 Sontag, E. 2002 For differential equations with r parameters, $2r+1$ experiments are enough for identification. *J. Nonlinear Sci.* **12**, 553–583. (doi:10.1007/s00332-002-0506-0)
24 Nelder, J. A. & Mead, R. 1965 A simplex method for function minimization. *Comp. J.* **7**, 308–313.
25 Bliss, R. D., Painter, P. R. & Marr, A. G. 1982 Role of feedback inhibition in stabilizing the classical operon. *J. Theoret. Biol.* **97**, 177–193. (doi:10.1016/0022-5193(82)90098-4)
26 Wang, B. 2010 *Design and functional assembly of synthetic biological parts and devices*. London, UK: Imperial College.
27 Szallasi, Z., Stelling, J. & Periwal, V. 2006 *System modeling in cell biology: from concepts to nuts and bolts*. Cambridge, MA: Bradford Book, MIT Press.
28 Wang, B., Kitney, R. I., Joly, N. & Buck, M. 2011 Engineering modular and orthogonal genetic logic gates for robust digital-like synthetic biology. *Nat. Commun.* **2**, 508. (doi:10.1038/ncomms1516)