IET Systems Biology

Research Article

IET Journals
The Institution of
Engineering and Technology

Application of conditional robust calibration to ordinary differential equations models in computational systems biology: a comparison of two sampling strategies

ISSN 1751-8849
Received on 22nd September 2018
Revised 30th July 2019
Accepted on 15th November 2019
E-First on 12th February 2020
doi: 10.1049/iet-syb.2018.5091
www.ietdl.org

Fortunato Bianconi[1] ✉, Chiara Antonini[2], Lorenzo Tomassoni[2], Paolo Valigi[2]
[1]Independent Researcher, Belvedere 44, 06036 Montefalco, Perugia, Italy
[2]Department of Engineering, University of Perugia, G. Duranti 95, 06132 Perugia, Italy
✉ *E-mail: fortunato.bianconi@gmail.com*

Abstract: Mathematical modelling is a widely used technique for describing the temporal behaviour of biological systems. One of the most challenging topics in computational systems biology is the calibration of non-linear models; i.e. the estimation of their unknown parameters. The state-of-the-art methods in this field are the frequentist and Bayesian approaches. For both of them, the performance and accuracy of results greatly depend on the sampling technique employed. Here, the authors test a novel Bayesian procedure for parameter estimation, called conditional robust calibration (CRC), comparing two different sampling techniques: uniform and logarithmic Latin hypercube sampling. CRC is an iterative algorithm based on parameter space sampling and on the estimation of parameter density functions. They apply CRC with both sampling strategies to the three ordinary differential equations (ODEs) models of increasing complexity. They obtain a more precise and reliable solution through logarithmically spaced samples.

1 Introduction

In systems biology, a key issue is to understand the dynamic interactions occurring within and between cells, which determine their structure and basic functions [1, 2]. Since most aspects of a biological system are not directly accessible to experimental observation, mathematical modelling is widely employed to formalise the dynamic and temporal evolution of system variables and to make predictions about biological mechanisms [3, 4].

In this study, we consider biological systems modelled with deterministic ordinary differential equations (ODEs) [5]. These models have a set of unknown kinetic parameters which cannot be measured experimentally. Thus, one of the challenging tasks of model building is model calibration; i.e. the inverse problem of estimating those parameters from available experimental data [6].

The state-of-the-art methods for parameter estimation are classified in frequentist and Bayesian methodologies. Both approaches are based on the concept of the probability density of observing the data given certain parameter values [7]. However, the frequentist methods aim at maximising the likelihood function through an optimisation algorithm while the Bayesian compute the posterior distribution using sampling based techniques [8, 9]. In both classes of algorithms, sampling is a fundamental element for dealing with parameter uncertainty. In order to avoid local optima, the frequentist approach employs bootstrapping from the available experimental replicates and Latin hypercube sampling (LHS) in order to start the estimation from different and independent parameter samples [10]. On the other hand, Bayesian methods sample directly from a prior distribution and accept only those samples that are in a region of high probability. Thus, in this class of methods, the sampler is of primary importance to guarantee the convergence to the desired posterior distribution. Samples should be dense and properly distributed in order to ensure a wide and appropriate exploration of the parameter space [11]. Currently, different sampling techniques are available, including multivariate uniform or logarithmic distribution, LHS and its variations, Markov Chain Monte Carlo and sequential Monte Carlo methods [12–15].

In this paper, we aim to understand the performances of two sampling strategies, namely uniform and logarithmic LHS, in a new Bayesian procedure called conditional robust calibration (CRC), presented in [16, 17]. CRC is an iterative algorithm that samples the parameter space and returns in output the probability density functions (pdfs) of parameters. Each pdf is estimated on the region of the parameter space that best reproduces the experimental dataset. CRC is applied to calibrate three ODE models with different characteristics; i.e. the well-known Lotka Volterra model [12], the erythroprotein receptor (EpoR) model [18] and the multiple myeloma (MM) model [19]. The comparison of CRC results between logarithmic and uniform sampling show that, while the logarithmic LHS successfully estimated unknown parameter values, the uniform LHS returned less reliable and precise results in all studied models.

2 Methods

2.1 Mathematical model

Mathematically, ODE models are represented as follows:

$$\dot{\boldsymbol{x}}(\theta, t) = f(\boldsymbol{x}(\theta, t), \boldsymbol{u}(t), \theta), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{1}$$

$$\boldsymbol{y}(\gamma, t) = g(\boldsymbol{x}(t), \boldsymbol{u}(t), \gamma), \tag{2}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the vector of state variables with initial conditions $\boldsymbol{x}_0$, $\boldsymbol{u} \in \mathbb{R}^s$ is the vector of control variables and $\boldsymbol{y} \in \mathbb{R}^m$ is the vector of output variables; i.e. the concentrations of measured states or a function of them. Note that $\theta \in \Theta$, where Θ is a subset of the positive orthant, $\mathbb{R}^t_{>0}$ is the vector of dynamical model parameters. Vector $\gamma \in \mathbb{R}^h$ contains scaling and offset parameters. The function $f(\cdot): \mathbb{R}^{n \times s \times t} \to \mathbb{R}^n$ specifies the model while $g(\cdot): \mathbb{R}^{n \times s \times h} \to \mathbb{R}^m$ is the observation function for mapping the state variables to the observed quantities. By setting $p = \{\theta, \gamma, \boldsymbol{x}_0\}$, $\boldsymbol{P} \in \mathbb{R}^{q = t + h + n}$, all model parameters are known. Each parameter $p_i, i = 1, \ldots, q$, is usually characterised by an interval of validity, due to its physical or biological meaning. Let $p_{n,i}$ denote the nominal value of p_i. Then, the lower and upper boundaries of parameter p_i can be written, respectively, as $b_{\ell,i} = c_{\ell,i} \times p_{n,i}$ and $b_{u,i} = c_{u,i} \times p_{n,i}$, where $c_{\ell,i} > 0$ and $c_{u,i} > 0$ are multiplicative

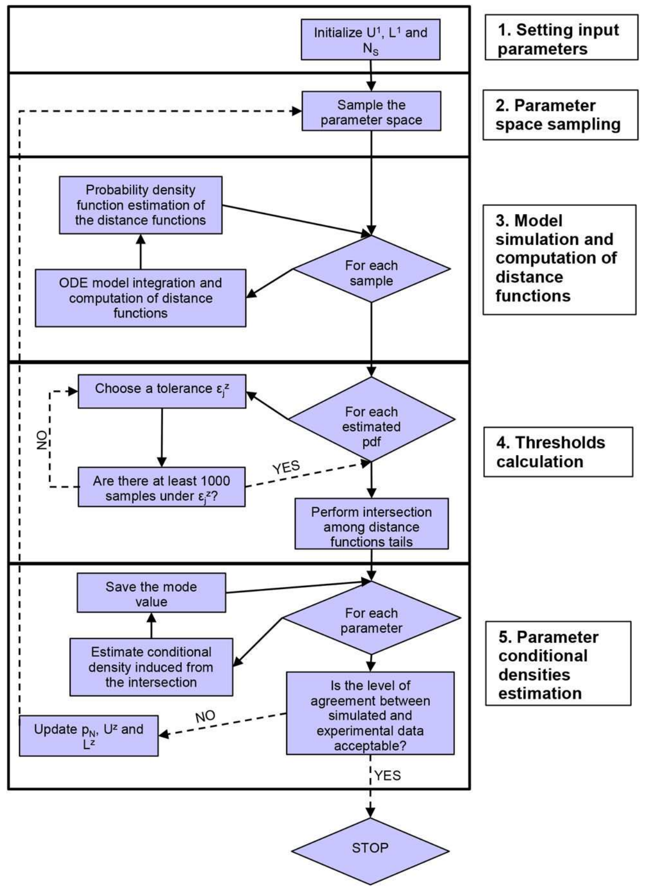

1. Setting input parameters
Initialize U¹, L¹ and N_S
2. Parameter space sampling
Sample the parameter space
3. Model simulation and computation of distance functions
Probability density function estimation of the distance functions
ODE model integration and computation of distance functions
For each sample
4. Thresholds calculation
Choose a tolerance ε_j^z
For each estimated pdf
NO
YES
Are there at least 1000 samples under ε_j^z?
Perform intersection among distance functions tails
5. Parameter conditional densities estimation
Save the mode value
For each parameter
Estimate conditional density induced from the intersection
Is the level of agreement between simulated and experimental data acceptable?
NO
Update p_N, U^z and L^z
YES
STOP

Fig. 1 *Main steps of CRC*

coefficients. Thus, the parameter space $\mathbb{P}$ results from the Cartesian product $\mathbb{P} := \prod_{i=1}^{q}[b_{\ell,i}, b_{u,i}]$.

2.2 Experimental dataset

Since mathematical models represent an *in silico* explanation of experimental data, the observables predicted by a model have to be in agreement with the experimental dataset. Let denote with $\boldsymbol{Y}^* = [\boldsymbol{y}_1^*, \ldots, \boldsymbol{y}_m^*]$ the matrix of available data. Supposing that experimental data are collected at different time points $t_k \in [t_1, \ldots, t_f]$, each $\boldsymbol{y}_j^*, j = 1, \ldots, m$ is a time series vector of the form $\boldsymbol{y}_j^* = [y_{j,1}^*, \ldots, y_{j,k}^*, \ldots, y_{j,f}^*]$. The *j*th observable measured at the *k*th time point can be divided as $y_j^*(t_k) = \tilde{y_j}(t_k) + \nu(t)$, where $\tilde{y_j}(t_k)$ is the nominal unknown value and $\nu(t)$ is the corresponding measurement error, usually assumed to be normally distributed.

2.3 Calibration method

To compare the performances of uniform and logarithmic sampling when using CRC, we estimate the parameters of three different ODE models. Here, we briefly describe the main steps of CRC, whose details are fully explained in [16]. Fig. 1 shows the CRC workflow.

2.3.1 Initialisation of input parameters: First of all, it is necessary to initialise the input parameters of the procedure. They are U^1 and L^1 which are, respectively, the upper and lower boundaries of the parameter vector range and N_S which is the number of parameter samples to generate at each iteration.

2.3.2 Parameter space sampling: The parameter vector is considered a random variable, denoted as $\boldsymbol{P}$. The purpose of CRC is to identify a sufficient number of parameter realisations that approximate the posterior distribution $f_{\boldsymbol{P}|\boldsymbol{Y}^*}(\boldsymbol{p})$. Therefore, the parameter space $\mathbb{P}$ is sampled in order to generate N_S realisations of the parameter vector. Thus, the choice of N_S depends on the cardinality of $\mathbb{P}$. The higher the number of parameters to estimate, the higher N_S should be set in order to guarantee a wide exploration and coverage of the parameter space.

LHS is chosen as the sampling technique in order to spread the sample points more evenly across all possible values. To generate Latin hypercube samples in a q-dimensional space, a unit hypercube $C^q = [0, 1]^q$ is first divided into N_S intervals with an equal length of $1/N_S$ along each axis. This creates N_S equally probable intervals, $[0, 1/N_S), [(1/N_S), (2/N_S)), \ldots, [((N_S - 1)/N_S), 1)$, for each dimension. Thus, LHS can be represented as a $N_S - \text{by} - q$ sample matrix, which contains one sample in each row and in each column. Each column corresponds to a variable and it is a random permutation of the intervals while each row is a sample point [11]. Afterwards, the LHS matrix needs to be centred around the nominal parameter values and between the defined lower and upper boundaries.

In this study, we assume that, in one case, parameters are uniformly distributed among their intervals while, in another case, they are logarithmically distributed. Let denote with LHS(N_s, q) the matrix that contains the N_S samples generated. Under the hypothesis of uniform distribution, the matrix $\boldsymbol{P}_S$ of the generated parameter samples is computed as

$$\boldsymbol{P}_S = \boldsymbol{p}_n \times [L^z + (U^z - L^z) \times \text{LHS}(N_S, q)], \quad (3)$$

where $\boldsymbol{p}_n$ is the nominal parameter vector, U^z and L^z are the lower and upper boundaries of the sampling interval at the zth iteration. On the contrary, when using logarithmic sampling, $\boldsymbol{P}_S$ is calculated as

$$\boldsymbol{P}_S = \boldsymbol{p}_n \times e^{lb^z + (ub^z - lb^z) \times \text{LHS}(N_S, q)}, \quad (4)$$

where $lb^z = \log(L^z)$ and $ub^z = \log(U^z)$ are the lower and upper boundaries transformed to the logarithmic space. At each iteration, the sampling interval is shrinked according to the following formula:

$$U^z(L^z) = \begin{cases} U^1(L^1), & \text{if } z = 1 \\ \dfrac{U^{(z-1)}(L^{(z-1)}) + k_{U,1}(k_{L,1})}{k_{U,2}(k_{L,2})}, & \text{if } z > 1 \end{cases} \quad (5)$$

where $k_{U,1}, k_{L,1}, k_{U,2}, k_{L,2}$ are input parameters chosen by the user.

Since U^z and L^z are multiplied with the nominal parameter vector $\boldsymbol{p}_n$, they determine the percentage variation of $\boldsymbol{p}_n$ at each iteration. This means that the interval where the parameter samples are selected for each iteration can be defined as follows: $[\boldsymbol{p}_n \times L^z, \boldsymbol{p}_n \times U^z]$. Thus, given that each iteration gets closer to a solution of the parameter values, the tuning parameters $k_{U,1}$, $k_{L,1}$, $k_{U,2}$, $k_{L,2}$ should be chosen in order to shrink the percentage variation of $\boldsymbol{p}_n$ with respect to the previous iteration.

2.3.3 Model simulation and computation of distance functions (DFs): For each sample $\boldsymbol{p} \in P_S$, the ODE model is integrated in order to compute the *in silico* dataset $\boldsymbol{Y}$. Then $\boldsymbol{Y}$ is compared with $\boldsymbol{Y}^*$ by computing a DF for each output variable. The DF is defined in two ways

$$\text{NNDF}_j = \sum_{k=1}^{f} \left| y_j(t_k) - y_{jk}^* \right|, \quad j = 1, \ldots, m. \quad (6)$$

$$\text{NDF}_j = \frac{1}{k} \sum_{k=1}^{f} \frac{\left| y_j(t_k) - y_{jk}^* \right|}{y_{jk}^*}, \quad j = 1, \ldots, m. \quad (7)$$

As $\boldsymbol{P}$ is a random variable, each vector $\boldsymbol{y}_j$ can be seen as a transformation from the random variable $\boldsymbol{P}$ and, consequently, each DF_j is a transformation from the random variable $\boldsymbol{y}_j$. Let denote with df_j a realisation of the random variable DF_j and with $f_{\mathrm{DF}_j}(df_j)$ the pdf of DF_j. Each $f_{\mathrm{DF}_j}(df_j)$, $\forall j = 1, \ldots, m$ is estimated using a kernel density approach [20]. The pdfs of the DFs are good indicators to infer the ability of the sampling technique to find parameter values that correctly calibrate the mathematical model under investigation. The higher is the area under each $f_{\mathrm{DF}_j}(df_j)$ in a region close to zero and the higher is the percentage of parameter vectors identified by the LHS that are able to make the observables of the mathematical model close to the experimental data.

2.3.4 Thresholds calculation: In this step, a set of thresholds $\varepsilon_j^z \geq 0$, $\quad j = 1, \ldots, m$ is defined. They correspond to the minimum accepted level of agreement between simulated and experimental data, for each output variable. Thus, CRC selects only those DF values that belong to the low tail of the corresponding pdf. The rationale that the user has to follow to fix the threshold values is explained in Section 2.3.5. Then, all DF tails are intersected among each other obtaining the following set: $\Delta = \{\bigcap_{j=1}^{m}(df_j(y_j, y_j^*)) : df_j(y_j, y_j^*) \leq \epsilon_j^z,\ \forall\ j = 1, \ldots, m\}$. In the parameter space, the accepted DF values correspond to a set of parameter samples having as joint conditional distribution $f_{\boldsymbol{P}|\Delta}(\boldsymbol{p})$.

2.3.5 Parameter conditional densities estimation: For each parameter, the corresponding conditional density $f_{P_i|\Delta}(p_i)$ is estimated using a kernel density approach. Concretely, the input data for the estimation of $f_{P_i|\Delta}(p_i)$, $\forall\ j = 1, \ldots, m$ are the subset of the parameter vectors generated through the LHS for which $df_j(y_j, y_j^*) \leq \epsilon_j^z$, $\forall\ j = 1, \ldots, m$. As a consequence, the lower are the values of the thresholds and the fewer are the available parameter vectors to estimate the conditional densities of each parameter. In order to have a reliable and stable estimation of these pdfs, at least 1000 parameter samples are required [20]. For this reason, each threshold ϵ_j^z is chosen in order to guarantee at least 1000 samples in $df_j(y_j, y_j^*)$ under $\epsilon_j^z, \forall\ j = 1, \ldots, m$; i.e. $|\Delta| \geq 1000$. Finally, for each parameter the corresponding mode value $p_{\mathrm{mode}, i}$ is selected. Multiple sets of thresholds can meet the requirements, but the purpose is to find the lowest possible values because it means that the distance between simulated and experimental data is minimum. If it is necessary to perform another iteration of CRC, the nominal value of every parameter is set equal to the mode value of $f_{P_i|\Delta}(p_i)$; i.e. $p_{n,i} = p_{\mathrm{mode}, i}$, $\forall\ i = 1, \ldots, q$ and the sampling interval is updated by changing U^z and L^z according to (5). It is important to underline that at the next iteration the interval where P_S is selected can be wider or tighter than the previous one because the lower and upper boundaries are multiplied by the mode value of each parameter.

The procedure stops when the desired level of agreement between simulated and experimental data is met. For instance, in case of *in silico* data whose source of error is known, CRC is performed until each threshold is slightly over the difference between nominal and noisy data, in order to avoid noise fitting.

3 Results

In this section, we compare the performances of CRC with uniform and logarithmic sampling approach when calibrating three different ODE models. All computational analyses were run using Matlab (R2014), on a Intel Core i7-4700HQ CPU, 2.4 GHz, 16 GB Memory, Ubuntu 16.04 LTS (64bit).

3.1 Lotka-Volterra model (M1)

We apply CRC to the Lotka-Volterra ODE model. The model describes the interaction between the prey species x, and predator species y, through parameters a and b

$$\begin{aligned} \dot{x}(a, t) &= a \times x(t) - x(t) \times y(t), \quad x(0) = 1, \\ \dot{y}(b, t) &= b \times x(t) \times y(t) - y(t), \quad y(0) = 0.5, \\ \boldsymbol{w}(a, b, t) &= [x(t), y(t)]. \end{aligned} \tag{8}$$

Observables of the model are both x and y while the parameter space is $\mathbb{P} \in \mathbb{R}^2_{>0}$. Both nominal values of parameters are set to 1. *In silico* data are generated as in [12]; i.e. sampling eight values at the specified time points and adding Gaussian noise $\mathcal{N}(0, (0.5^2))$ (see Table 1).

We calculate the distance between the nominal and noisy dataset according to (6) and we obtain the following values: 3.09 for species x and 2.6 for species y. Accordingly, they represent the minimum accepted level of agreement between simulated and experimental data. Since the purpose of this paper is the comprehension of the role of the sampling approach in the performances and efficiency of CRC, the tuning parameters are set in the same way when both the techniques are applied. Therefore, we perform six iterations of the entire CRC procedure setting $U^1 = 10$, $L^1 = 0.1$ and shrinking U^z and L^z according to (5), setting $k_{U,1} = k_{L,1} = 1$ and $k_{U,2} = k_{L,2} = 2$. In this way, we progressively shrink the percentage variation of the parameter vector, approaching 0 at the limit. Table 2 shows the mode vector of the parameters obtained at the end of each iteration of CRC and also the interval where the parameter samples are selected. Each interval depends on the mode vector in output from the previous iteration and on U^z and L^z. We generate $N_S = 10^4$ parameter samples for each iteration. Table 3 summarises the resulting thresholds for both uniform and logarithmic sampling approach and it also shows the cardinality of the set Δ. As explained in Section 2.3.5, each threshold is chosen in order to guarantee at least 1000 samples in the set of intersection between DF tails, for a reliable estimation of the conditional parameter pdfs.

According to the workflow depicted in Fig. 1, at the end of each iteration we estimate the conditional densities $f_{P_i|\Delta}$ for both parameters a and b. The evolution of the domains of the two conditional densities $f_{a|\Delta}$ and $f_{b|\Delta}$ are shown in Fig. 2.

Thus, the scatter plots are the accepted parameter samples from the first until the final iteration of CRC. Fig. 3 shows the time behaviour of output variables generated with the parameter samples accepted in the last iteration. When using logarithmic LHS, the final mode values are 1.06 for parameter a and 1.06 for parameter b while CRC with uniform LHS estimates a equal to 2.07 and b equal to 6.51.

Table 1 Model M1: *in silico* noisy dataset used for calibration

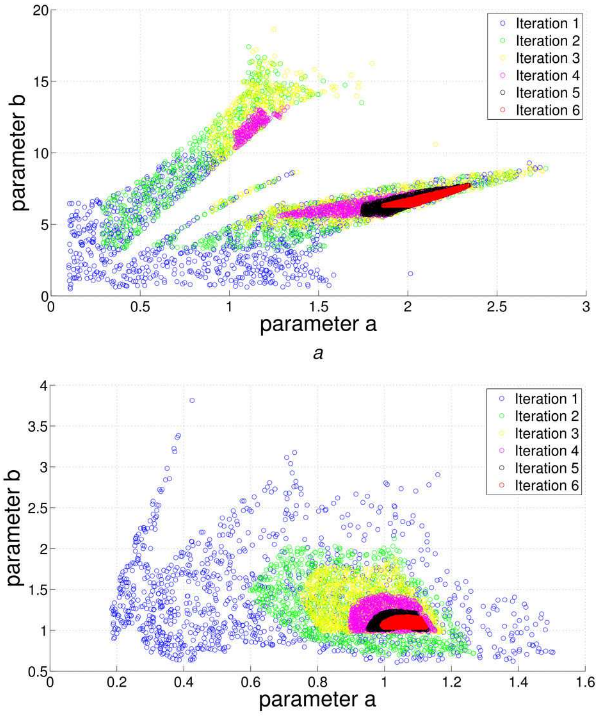
Time points, min	*x*	*y*
1.1	1.8823	0.5214
2.4	1.5510	1.2471
3.9	0.4007	0.9858
5.6	0.0454	0.2924
7.5	1.8881	0.6133
9.6	0.1839	1.3163
11.9	−0.1940	0.8171
14.4	1.6015	1.7212

Table 2 Model M1: mode vector obtained at the end of each CRC iteration, specified by *z* (first column of the table), and interval where the samples of P_S are selected

	Uniform		Logarithmic	
z	$\boldsymbol{p}_{\text{mode}}$	$[\boldsymbol{p}_{\text{mode}} \times L^z, \boldsymbol{p}_{\text{mode}} \times U^z]$	$\boldsymbol{p}_{\text{mode}}$	$[\boldsymbol{p}_{\text{mode}} \times L^z, \boldsymbol{p}_{\text{mode}} \times U^z]$
1	*a* = 0.52	[0.1,10]	*a* = 0.37	[0.1,10]
	b = 6.03	[0.1,10]	*b* = 0.96	[0.1,10]
2	*a* = 1.14	[0.63,6.3]	*a* = 0.81	[0.2,2.03]
	b = 6.13	[3.37,33.71]	*b* = 1.26	[0.53,5.28]
3	*a* = 1.16	[0.9,3.77]	*a* = 0.84	[0.63,2.63]
	b = 6.27	[4.86,20.38]	*b* = 1.11	[0.98,4.1]
4	*a* = 1.85	[1.64,3.93]	*a* = 0.98	[0.75,1.79]
	b = 5.99	[5.32,12.73]	*b* = 1.07	[0.95,2.27]
5	*a* = 1.90	[1.79,2.97]	*a* = 1.01	[0.95,1.58]
	b = 6.49	[6.12,10.14]	*b* = 1.07	[1,1.67]
6	*a* = 2.07	[2.01,2.65]	*a* = 1.06	[1.03,1.36]
	b = 6.51	[6.33,8.34]	*b* = 1.06	[1.03,1.36]

Table 3 Model M1: threshold schedule used at each iteration of CRC and the resulting cardinality of Δ

	Uniform			Logarithmic		
Iteration(*z*)	ϵ_x^z	ϵ_y^z	|Δ|	ϵ_x^z	ϵ_y^z	|Δ|
1	7.7	15	1010	7.3	6	1071
2	7.7	15	1008	5.8	4	1012
3	7.7	15	1038	5	3.4	1122
4	7.4	13.3	1002	4	3	1038
5	6.8	13.3	1022	3.4	2.7	1035
6	6.3	13.3	1170	3.1	2.7	1535



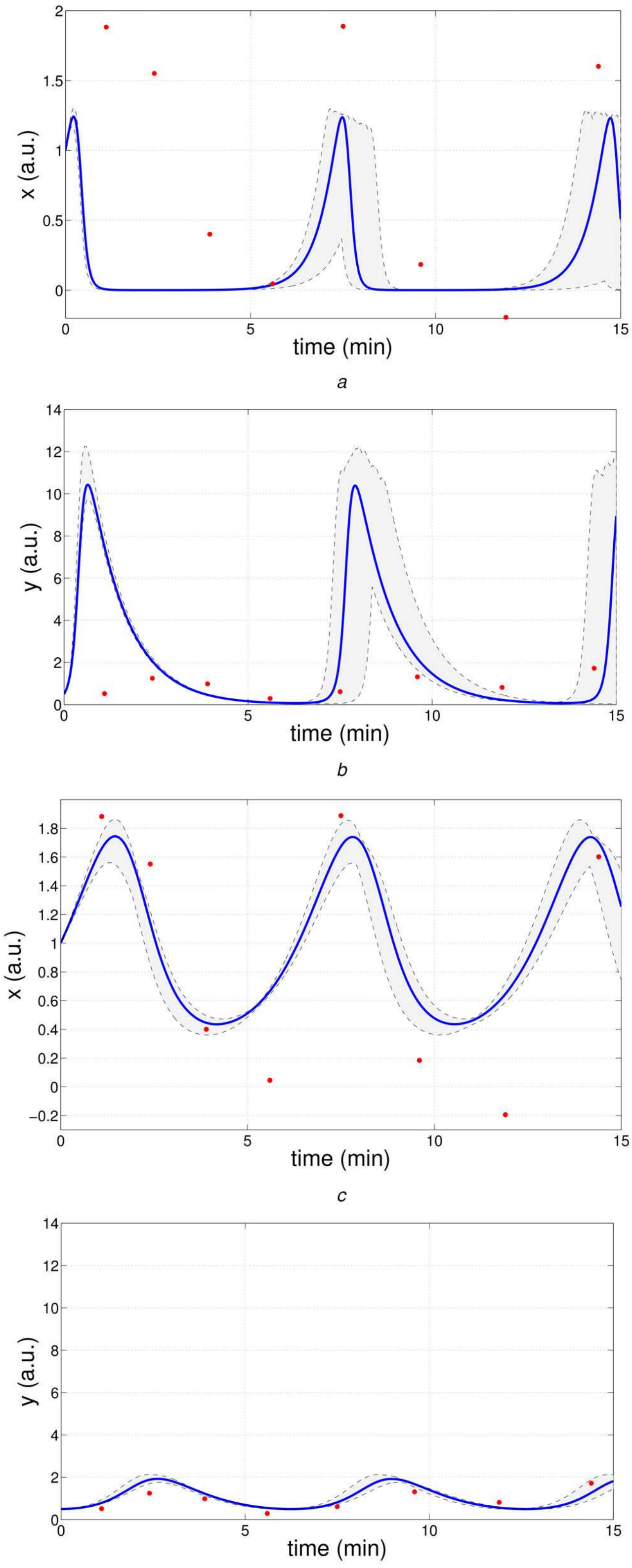
a

b

Fig. 2 *Model M1: scatter plots of both model parameters*
(a) Uniform sampling,
(b) Logarithmic sampling

3.2 EpoR model (M2)

We apply CRC to the EpoR model. In this model, an external ligand L activates, via two steps, an enzyme E that catalyses a substrate S. These reactions generate a product P that generally cannot be measured directly. For this reason the dynamical behaviour of P is the purpose of model prediction. Since the concentration over time of P is unknown, it is supposed to have available experimental data for the substrate S and the inactive form of enzyme E. The following equations, (9), (10) and (11), model the kinetics, the initial conditions and the observables of the EpoR model, respectively. The star symbol upon a variable distinguishes its active form from the inactive one



a

b

c

d

Fig. 3 *Model M1: blue lines are the temporal behaviour of the output variables when the parameters a and b are set equal to the mode values of the last iteration of CRC; red dots are the in silico data; grey regions are the confidence bandwidths when parameters vary from the 2.5th and the 97.5th percentile of their corresponding final conditional densities*

(a, b) Uniform sampling,

(c, d) Logarithmic sampling

$$
\begin{aligned}
\dot{E} &= -k_1 \times E \times L \\
\dot{E}^* &= k_1 \times E \times L - k_2 \times E^* \\
\dot{E}^{**} &= k_2 \times E^* \\
\dot{S} &= -k_3 \times E^{**} \times S \\
\dot{P} &= k_3 \times E^{**} \times S
\end{aligned}
\tag{9}
$$

$$
\begin{aligned}
E(0) &= \text{init}_E \\
E^*(0) &= 0 \\
E^{**}(0) &= 0 \\
S(0) &= \text{init}_S \\
P(0) &= 0
\end{aligned}
\tag{10}
$$

$$
\begin{aligned}
y_1(t_i) &= \text{scale}_E \times E(t_i) \\
y_2(t_i) &= \text{scale}_S \times S(t_i)
\end{aligned}
\tag{11}
$$

The EpoR model has three kinetic parameters that appear in (9), named k_1, k_2 and k_3. Moreover, two other parameters of the model are the initial conditions of the inactive enzyme E and of the substrate S, denoted as init_E and init_S, respectively. Finally, the last two model parameters are scale factors of the output variables y_1 and y_2 called scale_E and scale_S, respectively. Differently from the previous example (M1 model), in the EpoR model the purpose is to estimate not only the kinetic parameters but also the initial conditions and the scale factors of the output variables. Indeed for this model, the initial conditions of the two state variables are unknown as well as the scale factors of the output variables. Let us denote the overall parameter vector to estimate with $\boldsymbol{p} = [k_1, k_2, k_3, \text{init}_E, \text{init}_S, \text{scale}_E, \text{scale}_S]$, $\boldsymbol{p} \in \mathbb{R}^7_{>0}$. The nominal parameter vector is $\boldsymbol{p}_n = [0.1, 0.1, 0.1, 10, 5, 4, 2]$ and the noisy and noiseless *in silico* dataset used for model calibration are reported in Table 4.

We apply CRC using two different sampling approaches: uniformly and logarithmically spaced samples. As explained in Section 2, the application of CRC implies the setting of different tuning parameters. First, we fix the number of generated samples at each iteration equal to $N_S = 10^5$. Then, the boundaries of the sampling interval, L^z and U^z, evolve according to the values reported in Table 5 as the iteration number passes from 1 to 9. These values are obtained by setting $k_{U,1} = k_{L,1} = 0$, $k_{U,2} = 2$ and $k_{L,2} = 0.5$ for the first seven iterations and $k_{U,1} = k_{L,1} = 1$, $k_{U,2} = k_{L,2} = 2$ for the eighth and ninth iterations. As already explained in Section 2.3.2, the values of L^z and U^z in Table 5 are the percentage variation of the mode vector $\boldsymbol{p}_{\text{mode}}$ and they are used to compute the sampling interval at each iteration. Using (6), we calculate the error between noisy and noiseless data points for both the output variables. These errors, 12.78 for y_1 and 5.6 for y_2, represent the objective values for the thresholds of CRC in order to assert its success. From Table 5 it is possible to see that after nine iterations of CRC, ϵ_1^9 and ϵ_2^9 are very similar to the target values reported above, only when logarithmically spaced sampling is applied. On the other hand, ϵ_1^9 and ϵ_2^9 are too far from the target ones when uniform sampling is used. This demonstrates that the application of CRC combined with logarithmic sampling estimates a parameter vector that guarantees the desired level of agreement between simulated and experimental data.

Table 6 reports the mean value, the variance and the mode of the conditional densities of each parameter estimated at the end of each iteration of CRC when uniform and logarithmic sampling are applied, respectively. Thus, the mode parameter vectors resulting in output from the last iteration of CRC are $\boldsymbol{p}_{\text{mode}} = [0.1143, 0.0191, 0.0758, 34.9329, 2.9954, 1.1224, 3.3674]$ and $\boldsymbol{p}_{\text{mode}} = [534420, 563730, 678010, 6.6177, 3.1323, 6.2581, 3.3459]$ for the logarithmic and uniform sampling, respectively. Figs. 4–6 show the scatter plots of the accepted parameter values at the end of each iteration of CRC, once the threshold values are set.

Table 4 Model M2: noisy and noiseless dataset used for parameter estimation through CRC

Time, min	Noisy dataset		Noiseless dataset	
	y_1	y_2	y_1	y_2
0	39.24	10.24	40	10
3.33	—	10.61	—	9.4909
6.66	—	7.23	—	6.9968
10	21.89	5.47	14.7152	3.5448
13.33	—	1.43	—	1.2052
16.66	—	0.53	—	0.2768
20	3.08	1.08	5.4134	0.0446
23.33	—	0.38	—	0.0054
26.66	—	0.11	—	5.0042×10^{-4}
30	0.12	0.08	1.9915	3.8284×10^{-5}
40	1.37	—	0.732	—

Table 5 Model M2: CRC tuning parameters

			Uniform		Logarithmic	
Iteration(z)	L^z	U^z	ϵ_1^z	ϵ_2^z	ϵ_1^z	ϵ_2^z
1	0.01	100	200	300	51.0	31.0
2	0.02	50	400	500	50.0	31.0
3	0.04	25	130	180	46.0	30.9
4	0.08	12.5	45.0	53.0	40.0	30.3
5	0.16	6.25	33.2	28.7	33.5	29.1
6	0.32	3.125	31.5	28.1	25.0	27.5
7	0.64	1.5625	30.1	28.0	15.5	22.0
8	0.82	1.2813	27.7	27.3	13.4	10.0
9	0.91	1.1406	27.7	27.3	13.0	5.75

The first column lists the iteration number z, the second and the third ones the boundaries of the sampling intervals and the rest of the columns report the resulting thresholds for both the output variables when uniformly and logarithmically spaced sampling is applied, respectively.

Table 6 Model M2: mean values, variances and modes for each one of the conditional estimated densities of the seven parameters

		Uniform			Logarithmic		
Parameter	Iteration	Mean value	Variance	Mode	Mean value	Variance	Mode
k_1	1	49.7290	28.9609	47.2136	11.6207	19.9535	0.1676
	2	1175.8	684.1150	1195.1	1.4223	1.9153	0.1057
	3	1489.7	8648	13,687	0.5564	0.6338	0.0955
	4	861,500	49,062	787,600	0.3016	0.2861	0.0928
	5	252,280	139,190	226,530	0.1828	0.1280	0.0925
	6	391,760	183,120	476,180	0.1254	0.0490	0.0976
	7	524,440	125,540	515,750	0.1058	0.0183	0.1024
	8	540,940	688,040	521,970	0.1096	0.0122	0.1042
	9	535,160	345,400	534,420	0.1092	0.0062	0.1143
k_2	1	50.1379	28.8361	50.0661	11.3579	18.8793	0.1031
	2	1250	725.1931	1344.1	0.7662	1.1162	0.0248
	3	16,804	9721	18,070	0.1360	0.1512	0.0128
	4	113,370	64,590	103,060	0.0480	0.0425	0.0095
	5	329,270	181,910	252,480	0.0237	0.0161	0.0086
	6	436,220	204,810	480,790	0.0145	0.0065	0.0106
	7	529,100	127,830	520,070	0.0124	0.0025	0.0143
	8	546,420	688,560	546,070	0.0156	0.0017	0.0171
	9	560,090	36,606	563,730	0.0183	0.0009	0.0191
k_3	1	50.2296	28.8879	52.4291	8.7008	17.8986	0.1031
	2	1324.9	758.2055	1423.3	0.8074	1.1006	0.0422
	3	18,017	10,287	19,339	0.2615	0.2676	0.0327
	4	121,480	69,621	105,810	0.1316	0.1108	0.0292
	5	337,600	186,520	303,990	0.0758	0.0497	0.0296
	6	523,450	245,120	537,010	0.0514	0.0223	0.0415
	7	592,050	143,260	598,680	0.0491	0.0098	0.0569
	8	628,680	79,859	642,390	0.0623	0.0068	0.0678
	9	659,140	42,881	678,010	0.0728	0.0033	0.0758
init_E	1	22.6627	24.8233	2.7915	16.6944	22.0296	1.3032
	2	321	34.6861	0.0039	15.6962	16.4395	1.9840
	3	22	24	3	15.8885	13.4175	3.8127
	4	10.3727	9.1662	2.6501	18.3333	12.9594	6.6465
	5	7.3006	4.0041	3.9394	19.8313	10.6221	10.9966
	6	6.7829	2.5373	4.6579	21.4556	7.0448	20.2614
	7	6.3909	0.5098	6.5185	25.1709	3.8984	28.8037
	8	6.5604	0.7176	5.8516	30.1755	2.9611	32.2857
	9	6.5416	0.0890	6.6177	33.9041	1.5291	34.9329

This is an alternative and effective way to visualise the distributions of parameters, whose main features are summarised in Table 6. Fig. 7 shows the time behaviour of the two output variables when both sampling strategies are applied and the parameter vector of the model is set equal to the corresponding mode vector reported above. Moreover, we also plot the confidence bands (grey regions) of the observables y_1 and y_2 when parameter values vary between the 2.5th and 97.5th percentile of their corresponding conditional pdfs. In order to verify if this calibration process is robust against independent realisations, we perform ten realisations of the entire procedure explained above.

3.3 Multiple myeloma model (M3)

In this section, we describe the calibration of the ODE model presented in [19], which investigates the mechanisms of the p38 MAPK isoforms in Multiple Myeloma (MM). The model has 40 ODEs and 53 kinetic parameters while the experimental data used for parameter estimation are reverse phase protein array (RPPA) data [21]. RPPA was used to analyse MM cell lines, measuring in total 153 proteins in 80 samples at six different time points (0, 5, 10, 30, 60, 90 min). The 16 proteins both present in the model and in the dataset represent model observables. Since RPPA data are normalised with respect to the initial concentration value, initial conditions of proteins in the ODEs are all set to 1. For this reason, parameters to estimate are only the kinetic ones; i.e. $\mathbb{P} \in \mathbb{R}^{53}_{>0}$. Equations of the model and the corresponding RPPA dataset are reported in Supplementary Materials.

Tuning parameters of CRC are set in the same way in both cases of uniform and logarithmic LHS. Sixteen normalised DFs are computed according to (7) and six iterations of CRC are run. We define $U^1 = 10$ and $L^1 = 0.1$ and then we shrink the sampling interval as in model M1. We initialise the other parameters as follows: $N_S = 10^6$ and $p_{n,i} = 1, \quad \forall i = 1, \ldots, 53$. We perform ten independent realisations of the procedure in order to test the stability and reliability of results.

Table 7 summarises the thresholds used in both sampling techniques at the different iterations. Fig. 8 compares the evolution of the conditional density of parameter k_2, chosen as an example, in the two cases.

In Supplementary Materials means, modes and variances of all parameters during the iterations are reported. Fig. 9 shows the time behaviour of output variables generated with the parameter samples accepted in the last iteration, respectively, with logarithmic and uniform sampling.

4 Discussion and conclusion

In [16], the authors presented CRC, a novel Bayesian procedure for parameter estimation of ODE models. In Section 2, the main steps

		Uniform			Logarithmic		
Parameter	Iteration	Mean value	Variance	Mode	Mean value	Variance	Mode
init_S	1	22.5516	24.7948	2.7552	15.6737	22.0031	0.5592
	2	31.6295	34.3431	4.0123	6.1501	6.8975	0.7217
	3	22.8575	24.6608	3.0272	4.5241	4.5694	0.7536
	4	8.5900	9.0537	1.2937	3.0397	2.4375	0.9278
	5	3.5322	1.9527	1.8958	2.3863	1.4694	1.1437
	6	3.2603	1.2380	2.1938	1.9944	0.8117	1.6620
	7	3.0610	0.2181	3.1696	2.0976	0.3166	2.3654
	8	3.1393	0.3199	2.8206	2.7230	0.1935	2.8951
	9	3.0815	0.0817	3.1323	2.9790	0.1756	2.9554
s_1	1	22.5673	24.5414	2.7896	19.6923	23.2733	1.7029
	2	31.5443	34.7847	3.8333	14.2438	18.3015	1.4727
	3	21.3923	23.4380	2.7350	6.9718	7.9647	1.3845
	4	10.0299	8.8134	2.5722	4.0516	3.8055	1.3002
	5	6.9566	3.8456	3.7233	2.6284	1.7600	1.3185
	6	6.3775	2.3999	4.3986	2.0039	0.7726	1.4515
	7	6.0312	0.4802	6.1260	1.5910	0.2682	1.3950
	8	6.1679	0.6745	5.5367	1.3388	0.1368	1.2086
	9	6.1897	0.0834	6.2581	1.1705	0.0535	1.1224
s_2	1	22.8189	25.0087	2.8391	16.3713	22.4893	0.6056
	2	32.2900	35.2887	4.1628	6.7946	7.4791	0.8120
	3	23.9120	25.9322	3.2004	5.0803	5.1099	0.8483
	4	9.0155	9.5794	1.3909	3.4347	2.7596	1.0551
	5	3.7739	2.0848	2.0165	2.7124	1.6798	1.3281
	6	3.4703	1.3127	2.3428	2.3114	0.9483	1.9099
	7	3.2667	0.2354	3.3859	2.4076	0.3647	2.6942
	8	3.3562	0.3414	3.0139	3.0965	0.2210	3.2922
	9	3.2900	0.0878	3.3459	3.3867	0.2018	3.3674

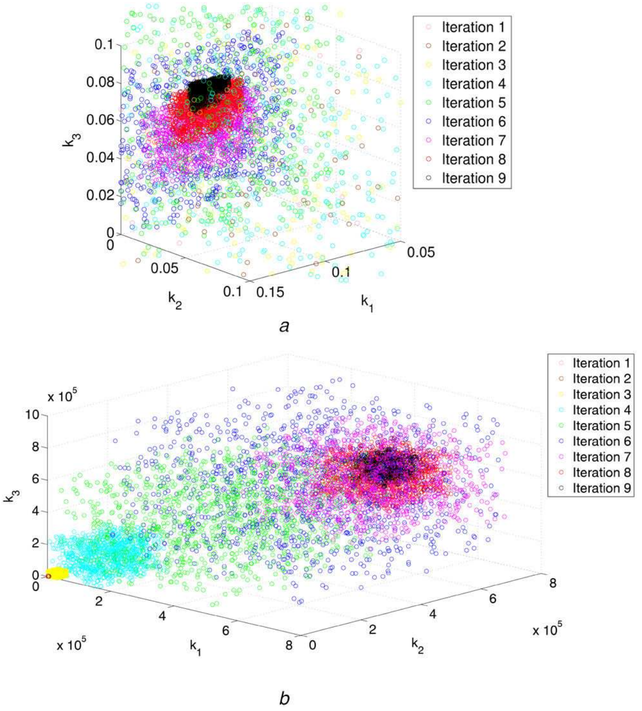


Fig. 4 *Model M2: scatter plots of the three kinetic model parameters k_1, k_2 and k_3. Circles in the graphs represent clouds of accepted parameter values at the end of each iteration of CRC*

(a) Application of CRC with logarithmically spaced sampling,

(b) Application of CRC with uniformly spaced sampling

of CRC are briefly summarised with specific attention to the introduced innovations. One of the key points of CRC is the sampling of the parameter space which is completely overturned compared to other Bayesian algorithms. Thus, in CRC the number of samples is an input parameter which remains unchanged in all the iterations. Other substantial innovations are the definition of as many DFs as the number of observables of the model and the corresponding intersection of all the DFs tails.

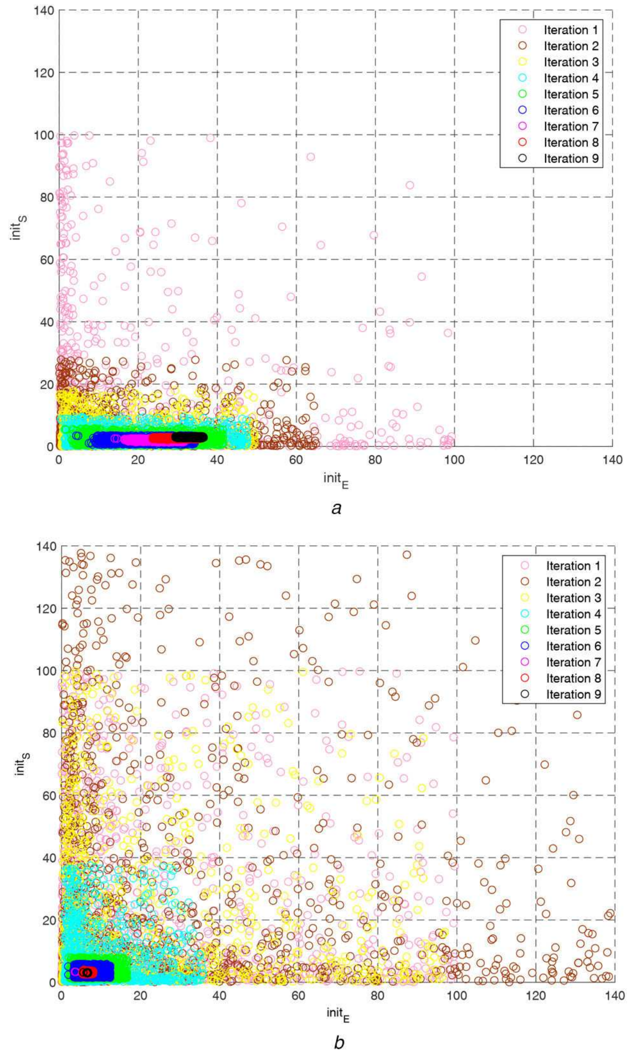


Fig. 5 *Model M2: scatter plots of the two initial condition parameters $init_E$ and $init_S$. Circles in the graphs represent clouds of accepted parameter values at the end of each iteration of CRC*

(a) Application of CRC with logarithmically spaced sampling,

(b) Application of CRC with uniformly spaced sampling

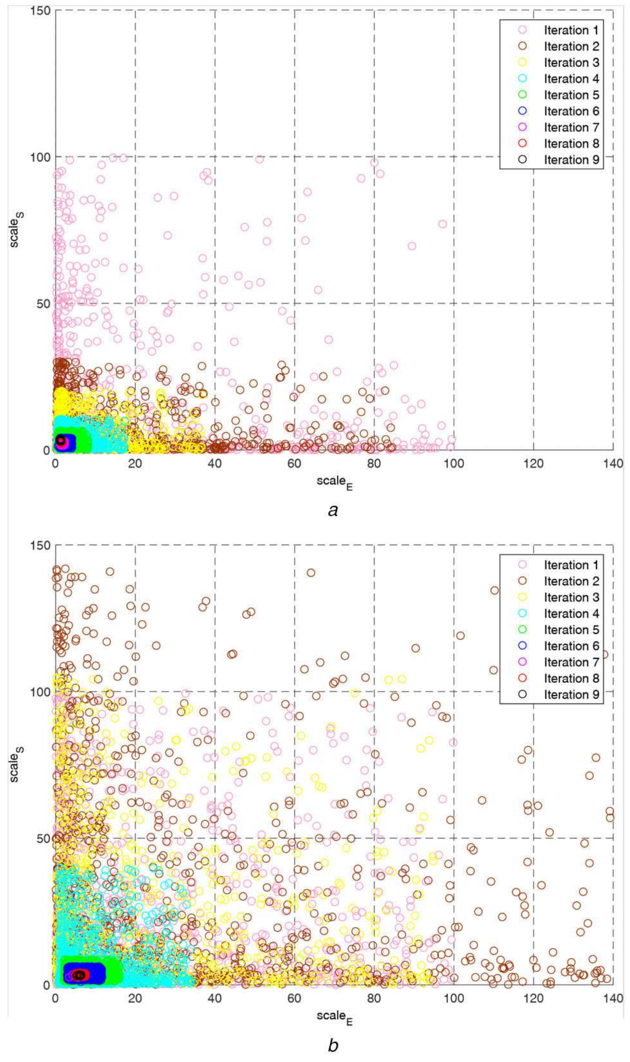


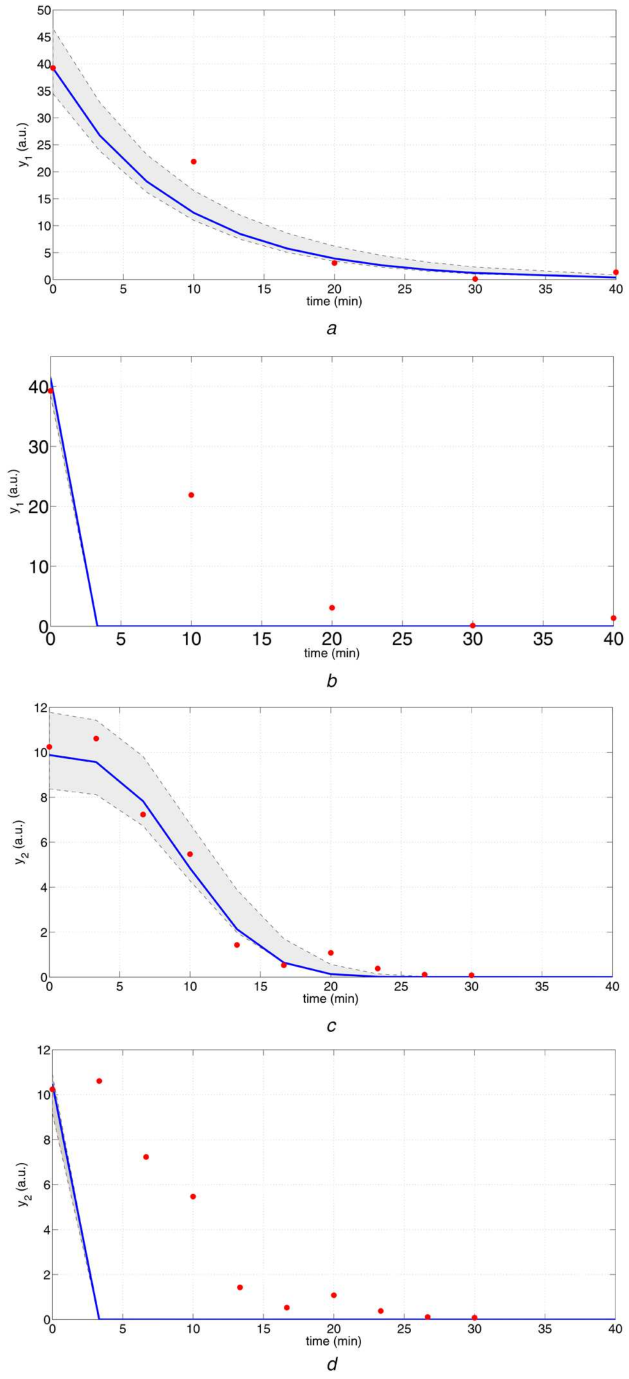
Fig. 6 *Model M2: scatter plots of the two scale factors parameters $scale_E$ and $scale_S$. Circles in the graphs represent clouds of accepted parameter values at the end of each iteration of CRC*

(a) Application of CRC with logarithmically spaced sampling,

(b) Application of CRC with uniformly spaced sampling

The purpose of this study is to apply CRC with two different sampling approaches in order to test which performs better and to understand the influence of the sampling process in this algorithm. We apply CRC using LHS with uniformly and logarithmically spaced samples. Even if the practice of applying different sampling strategies is common in the field of the Bayesian algorithms, the comparison we provide in this paper has never been performed before since CRC is a novel algorithm. We choose three ODE models with a parameter space and the corresponding space of state variables of increasing dimension. They are the Lotka-Volterra model, the EpoR model and the MM model.

Tables 3, 5 and 7 prove that the logarithmic sampling is able to reach lower thresholds than the uniform ones, almost for all the output variables in all the tested models. Indeed, for the Lotka-Volterra model, the thresholds obtained at the sixth iteration with logarithmic sampling are as close as possible to the target ones, meaning that through logarithmic sampling it is possible to calibrate the ODE model also preventing overfitting. Logarithmic sampling generates more dense samples in the interval (0; 1]. Similarly, in the EpoR model, after nine iterations of CRC with logarithmic sampling, the resulting thresholds get very close to the target ones. As regards the MM model, since the nominal parameter vector is not known from [19], we cannot calculate the target values of the error. However, after six iterations of CRC with logarithmic sampling the maximum error between simulated and experimental data is only of the 16% (see Table 7). On the other hand, the application of CRC combined with uniform sampling leads to threshold values that are on average higher than those obtained through the logarithmic ones. In the Lotka-Volterra model, thresholds after six iterations are too distant from 3.09 and 2.6. Moreover, scatter plots in Fig. 2 confirm results of Table 3: it is clear how with the uniform sampling the accepted samples at each iteration evolve towards a region of the parameter space that does not correspond to the nominal values of kinetic parameters *a* and *b*. Conversely, the scatter plots of logarithmic sampling in Fig. 2*b* show how the red dots lie in a region that is exactly in the neighbourhood of the nominal point (1;1). Additionally, as shown in Fig. 3 only the logarithmic ones adequately fit the noisy data, following the typical oscillating behaviour of the prey-predator model. On the other hand, at many time points, the behaviour depicted in Fig. 3*a* and *b* is too far from *in silico* data. Similar observations are valid also for the EpoR model. Fig. 7 clearly shows that, since the threshold values obtained with uniform sampling are too far from the target ones, the time behaviour of output variables does not fit the experimental data. On the contrary, logarithmic sampling is able to adequately reproduce the correct time behaviour of y_1 and y_2, even if the resulting estimated parameter vector is an alternative parametrisation of the model. As regards the MM model instead, both sampling strategies produce acceptable time behaviour of output variables, as is shown in Fig. 9, since the maximum percentage error with the uniform sampling is of the 22%. However, the confidence bands in Fig. 9*b* are rather higher than those in Fig. 9*a*. This is due to the fact that, for most of the kinetic parameters, the corresponding conditional



a

b

c

d

Fig. 7 *Model M2: blue lines are the time behaviour of output variables y_1 and y_2 when the parameters are set equal to the mode in output from the last iteration of CRC; red dots are the noisy in silico data used to calibrate the model; grey regions are the temporal behaviour bandwidths when the model parameters vary between the 2.5th and 97.5th percentile of their corresponding conditional pdfs*

(a) Time behaviour of y_1 when CRC is applied with logarithmically spaced sampling,

(b) Time behaviour of y_1 when CRC is applied with uniformly spaced sampling,

(c) Time behaviour of y_2 when CRC is applied with logarithmically spaced sampling,

(d) Time behaviour of y_2 when CRC is applied with uniformly spaced sampling

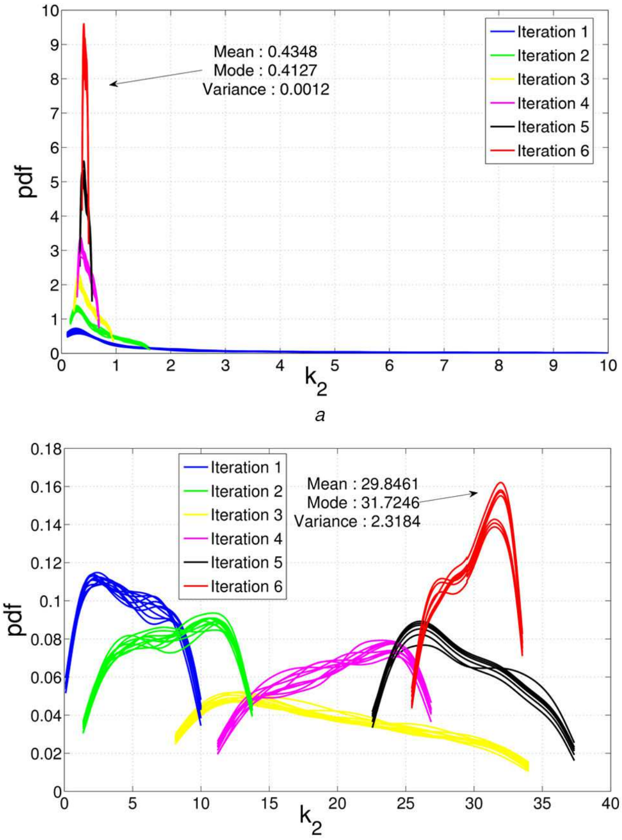
pdf, estimated in the last iteration of CRC, is tighter when logarithmic sampling is performed, as shown in Fig. 8. Moreover, in order to make a comparison of the computational cost of the two sampling techniques, we measure the time to perform each iteration of CRC, as reported in Table 8. In all examples, the logarithmic sampling has a minor computational burden compared to the uniform sampling, whose simulation time also increases with the model dimension. In conclusion, it is possible to state that logarithmic sampling performs better and has a reduced computational cost when calibrating a model using CRC. It is the best sampling technique when it is necessary to perturb the parameter space over different orders of magnitude. This is due to the intrinsic property of the logarithmic distribution that generates the same number of samples in different orders of magnitude [22]. The logarithmic sampling is able to generate more dense samples around the mode vector returned by the previous iteration of CRC, which is the most likely point in the parameter space at a given iteration.

5 Acknowledgments

Chiara Antonini and Lorenzo Tomassoni contributed equally to this study. This work was supported by the Italian Association for Cancer Research (AIRC) grant (15713/2014).

Table 7 Model M3: threshold schedule when both uniform and logarithmic LHS sampling are applied

	Iteration											
	1		2		3		4		5		6	
	Uniform	Logarithmic	Uniform	Logarithmic	Uniform	Logarithmic	Uniform	Logarithmic	Uniform	Logarithmic	Uniform	Logarithmic
x_4	0.45	0.6	0.3	0.3	0.3	0.2	0.2	0.1	0.14	0.05	0.08	0.05
x_6	0.5	0.7	0.4	0.4	0.4	0.3	0.4	0.2	0.2	0.1	0.15	0.08
x_{10}	0.5	0.7	0.4	0.4	0.4	0.3	0.4	0.2	0.2	0.1	0.18	0.06
x_{14}	0.4	0.7	0.4	0.4	0.35	0.3	0.15	0.15	0.12	0.13	0.1	0.09
x_{16}	0.5	0.8	0.4	0.5	0.4	0.3	0.4	0.2	0.2	0.1	0.2	0.06
x_{18}	0.25	0.3	0.2	0.2	0.2	0.15	0.2	0.12	0.1	0.095	0.1	0.088
x_{20}	0.35	0.5	0.25	0.3	0.2	0.2	0.2	0.13	0.1	0.1	0.1	0.088
x_{22}	0.45	0.7	0.3	0.4	0.25	0.2	0.15	0.15	0.1	0.1	0.08	0.08
x_{24}	0.4	0.7	0.4	0.4	0.35	0.3	0.3	0.2	0.15	0.1	0.15	0.08
x_{26}	0.35	0.6	0.35	0.4	0.35	0.2	0.25	0.1	0.17	0.05	0.08	0.04
x_{28}	0.35	0.5	0.25	0.3	0.2	0.2	0.15	0.15	0.12	0.1	0.07	0.08
x_{32}	0.45	0.6	0.3	0.3	0.2	0.2	0.25	0.15	0.2	0.12	0.1	0.12
x_{33}	0.8	1	0.7	0.7	0.45	0.5	0.45	0.3	0.4	0.2	0.22	0.15
x_{34}	0.9	0.8	0.7	0.6	0.55	0.5	0.55	0.35	0.2	0.2	0.18	0.15
x_{38}	0.4	0.6	0.3	0.4	0.2	0.25	0.2	0.2	0.14	0.18	0.14	0.16
x_{40}	0.5	0.7	0.35	0.4	0.3	0.3	0.35	0.2	0.2	0.15	0.1	0.12



a

b

Fig. 8 *Model M3: evolution of the conditional pdfs of parameter k_2 in all iterations, when using, respectively, logarithmic and uniform LHS*

(a) Logarithmic sampling,

(b) Uniform sampling

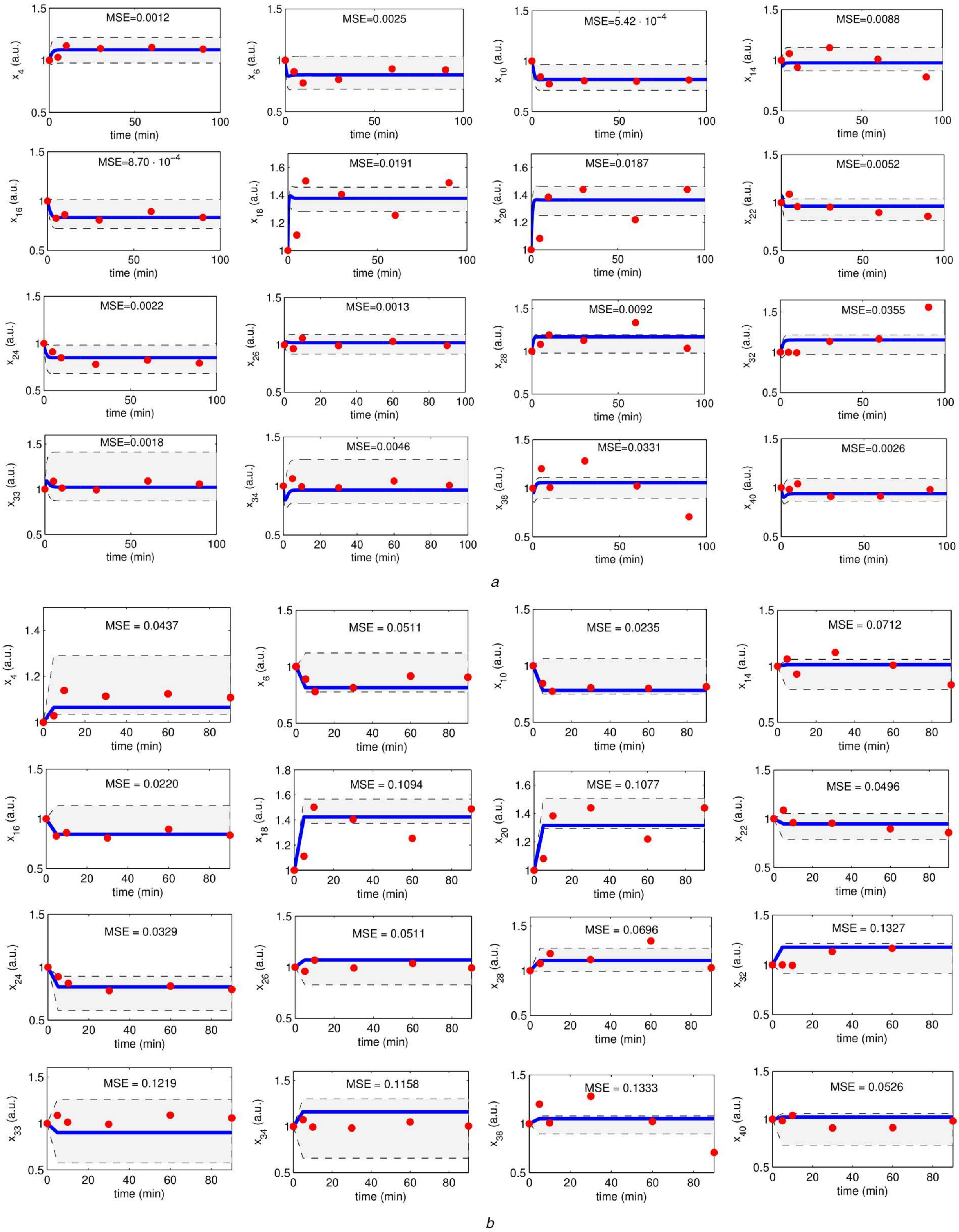


Fig. 9 *Model M3: results of CRC with logarithmic and uniform LHS; red dots are the RPPA data; blue lines are the time behaviour of output variables when parameter values are equal to mode values of the last iteration (see Supplementary Materials); grey area covers the possible temporal behaviours of observables when parameters vary between the 2.5th and 97.5th percentile of their corresponding conditional pdfs. MSE is the mean squared error between the simulation of the model and the data*

(a) Logarithmic sampling,

(b) Uniform sampling

Table 8 Time to perform each iteration of CRC in the three models

	M1		M2		M3	
Iteration(*z*)	Uniform	Logarithmic	Uniform	Logarithmic	Uniform	Logarithmic
1	431 s	288 s	798 s	449 s	360 min	110 min
2	323 s	213 s	835 s	305 s	720 min	80 min
3	300 s	251 s	856 s	245 s	1 d	70 min
4	268 s	225 s	750 s	223 s	2 d	60 min
5	260 s	216 s	674 s	210 s	3 d	60 min
6	245 s	202 s	693 s	195 s	3 d	60 min
7	—	—	685 s	217 s	—	—
8	—	—	650 s	310 s	—	—
9	—	—	645 s	360 s	—	—

Time is expressed in seconds (s), minutes (min) or days (d).

6 References

[1] Wolkenhauer, O., Kitano, H., Cho, K.H.: 'Systems biology', *IEEE Control Syst.*, 2003, **23**, (4), pp. 38–48

[2] Degasperi, A., Fey, D., Kholodenko, B.N.: 'Performance of objective functions and optimisation procedures for parameter estimation in system biology models', *NPJ Syst. Biol. Appl.*, 2017, **3**, (1), p. 20

[3] Liepe, J., Kirk, P., Filippi, S.*, et al.*: 'A framework for parameter estimation and model selection from experimental data in systems biology using approximate Bayesian computation', *Nat. Protoc.*, 2014, **9**, (2), p. 439

[4] Zhan, C., Situ, W., Yeung, L.F.*, et al.*: 'A parameter estimation method for biological systems modelled by ode/dde models using spline approximation and differential evolution algorithm', *IEEE/ACM Trans. Comput. Biol. Bioinform. (TCBB)*, 2014, **11**, (6), pp. 1066–1076

[5] Bartocci, E., Lió, P.: 'Computational modeling, formal analysis, and tools for systems biology', *PLoS Comput. Biol.*, 2016, **12**, (1), p. e1004591

[6] Rodriguez Fernandez, M., Mendes, P., Banga, J.R.: 'A hybrid approach for efficient and robust parameter estimation in biochemical pathways', *Biosystems*, 2006, **83**, (2–3), pp. 248–265

[7] Klipp, E., Liebermeister, W., Wierling, C.*, et al.*: '*Systems biology: a textbook*' (John Wiley & Sons, Hoboken, New jersey, USA, 2016)

[8] Kreutz, C., Raue, A., Timmer, J.: 'Likelihood based observability analysis and confidence intervals for predictions of dynamic models', *BMC Syst. Biol.*, 2012, **6**, (1), p. 120

[9] Vanlier, J., Tiemann, C., Hilbers, P.*, et al.*: 'Parameter uncertainty in biochemical models described by ordinary differential equations', *Math. Biosci.*, 2013, **246**, (2), pp. 305–314

[10] Mukhopadhyay, C.: 'Maximum likelihood analysis of masked series system lifetime data', *J. Stat. Plan. Inference*, 2006, **136**, (3), pp. 803–838

[11] Sheikholeslami, R., Razavi, S.: 'Progressive latin hypercube sampling: an efficient approach for robust sampling-based analysis of environmental models', *Environ. Model. Softw.*, 2017, **93**, pp. 109–126

[12] Toni, T., Welch, D., Strelkowa, N.*, et al.*: 'Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems', *J. R. Soc., Interface*, 2009, **6**, (31), pp. 187–202

[13] McKay, M.D., Beckman, R.J., Conover, W.J.: 'A comparison of three methods for selecting values of input variables in the analysis of output from a computer code', *Technometrics.*, 2000, **42**, (1), pp. 55–61

[14] Stein, M.: 'Large sample properties of simulations using latin hypercube sampling', *Technometrics.*, 1987, **29**, (2), pp. 143–151

[15] Wyss, G.D., Jorgensen, K.H.: 'A user's guide to lhs: Sandia's latin hypercube sampling software'. SAND98-0210, Sandia National Laboratories, Albuquerque, NM, 1998

[16] Bianconi, F., Antonini, C., Tomassoni, L.*, et al.*: 'Robust calibration of high dimension nonlinear dynamical models for omics data: an application in cancer systems biology', *IEEE Trans. Control Syst. Technol.*, 2018, **PP**, (99), pp. 1–12

[17] Bianconi, F., Tomassoni, L., Antonini, C.*, et al.*: 'Conditional robust calibration (crc): a new computational Bayesian methodology for model parameters estimation and identifiability analysis', bioRxiv, 2017, p. 197400

[18] Raue, A., Kreutz, C., Maiwald, T.*, et al.*: 'Addressing parameter identifiability by model-based experimentation', *IET Syst. Biol.*, 2011, **5**, (2), pp. 120–130

[19] Peng, H., Peng, T., Wen, J.*, et al.*: 'Characterization of p38 mapk isoforms for drug resistance study using systems biology approach', *Bioinformatics*, 2014, **30**, (13), pp. 1899–1907

[20] Bianconi, F., Baldelli, E., Luovini, V.*, et al.*: 'Conditional robustness analysis for fragility discovery and target identification in biochemical networks and in cancer systems biology', *BMC Syst. Biol.*, 2015, **9**, (1), p. 70

[21] Baldelli, E., Calvert, V., Hodge, A.*, et al.*: 'Reverse phase protein microarrays', in Wiley Online Library (Ed.): '*Molecular profiling*' (Springer, New York City, New York, USA, 2017) pp. 149–169

[22] Rodriguez Fernandez, M., Banga, J.R., Doyle, IIIF.J.: 'Novel global sensitivity analysis methodology accounting for the crucial role of the distribution of input parameters: application to systems biology models', *Int. J. Robust Nonlinear Control*, 2012, **22**, (10), pp. 1082–1102