Computer Physics Communications 190 (2015) 15–22

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Reducing complexity: An iterative strategy for parameter determination in biological networks

Sebastian C. Binder[a], Esteban A. Hernandez-Vargas[b], Michael Meyer-Hermann[a,c,*]

[a] Department of Systems Immunology and Braunschweig Integrated Centre of Systems Biology, Helmholtz Centre for Infection Research, Inhoffenstr. 7, 38124 Braunschweig, Germany

[b] Systems Medicine of Infectious Diseases, Department of Systems Immunology and Braunschweig Integrated Centre of Systems Biology, Helmholtz Centre for Infection Research, Inhoffenstr. 7, 38124 Braunschweig, Germany

[c] Institute for Biochemistry, Biotechnology and Bioinformatics, Technische Universität Braunschweig, Braunschweig, Germany

ARTICLE INFO

Article history:
Received 26 May 2014
Received in revised form
10 December 2014
Accepted 5 January 2015
Available online 12 January 2015

Keywords:
Parameter estimation
Signalling network
Biology
Differential evolution

ABSTRACT

The dynamics of biological networks are fundamental to a variety of processes in many areas of biology and medicine. Understanding of such networks on a systemic level is facilitated by mathematical models describing these networks. However, since mathematical models of signalling networks commonly aim to describe several highly connected biological quantities and many model parameters cannot be measured directly, quantitative dynamic models often present challenges with respect to model calibration. Here, we propose an iterative fitting routine to decompose the problem of fitting a system of coupled ordinary differential equations describing a signalling network into smaller subproblems. Parameters for each differential equation are estimated separately using a Differential Evolution algorithm while all other dynamic quantities in the model are treated as input to the system. The performance of this algorithm is evaluated on artificial networks with known structure and known model parameters and compared to a conventional optimisation procedure for the same problem. Our analysis indicates that the procedure results in a significantly higher quality of fit and more efficient reconstruction of the true parameters than the conventional algorithm.

1. Introduction

A quantitative description of biological networks is critical to an understanding of biological processes at different scales. In particular, signalling networks regulate a plethora of biological systems [1] and play a key role in the control of immune functions, e.g. in intracellular signal transduction like T cell receptor signalling or intercellular communication by a network of cytokines [2]. While the interactions between the constituents of such biological networks are often unknown, there is an increasing amount of quantitative data on their dynamics. The complex behaviour that can emerge even from simple interactions [3] has motivated increasing interest in dynamic models of signalling networks (see [4–6] and references therein).

In the common case where quantitative knowledge about the underlying kinetic properties of modelled reactions is missing, either due to a lack of data or because there is no direct physical equivalent, model parameters have to be estimated by fitting the model output to suitable data. This problem usually involves the minimisation of a function measuring the disagreement between model output and data. Although this problem is well-studied [7], parameter estimation can be challenging, as biological data are often noisy, contain measurement errors and are incomplete. Furthermore, especially in the case of signalling networks, the systems under study can be too complex. Even parameter estimation problems in simplified network models can be high-dimensional and almost inevitably have multiple local minima.

Among the various optimisation methods, the most traditional are gradient descent based methods. While these are widely employed and very efficient in the case of local minimisation problems, they tend to converge prematurely in case of multiple minima or areas in the parameter space where the objective function is flat [8]. Hence, stochastic optimisation algorithms are widely used, like Genetic Algorithms and Simulated Annealing [9] or Particle Swarm Optimisation [10]. A particularly interesting

[*] Corresponding author at: Department of Systems Immunology and Braunschweig Integrated Centre of Systems Biology, Helmholtz Centre for Infection Research, Inhoffenstr. 7, 38124 Braunschweig, Germany.

E-mail addresses: sb@theoretical-biology.de (S.C. Binder), esteban.vargas@theoretical-biology.de (E.A. Hernandez-Vargas), mmh@theoretical-biology.de (M. Meyer-Hermann).

http://dx.doi.org/10.1016/j.cpc.2015.01.001

metaheuristic algorithm has been proposed by Storn and Price [11] who coined the term Differential Evolution (DE). The general idea of the algorithm is very simple: A number of random parameter vectors is generated and the parameter vectors are mutated in subsequent iterations, called "generations" in the DE terminology. For each parameter vector to be computed in a generation, three vectors from the previous generation are picked and the new parameter vector is created by adding the weighted difference between two parameters to the third one. Furthermore, a process recombining parameter vectors called "crossover" is introduced to further increase the diversity within the vectors. If the mutated target vector performs better in the minimisation of the objective function than the corresponding vector from the previous generation, it is retained for the next generation and otherwise discarded. While the general strategy outlined above is simple, the algorithm has been directly compared to six other global optimisation algorithms by Moles et al. [12], where DE outperformed the other algorithms in terms of finding the best solution.

Even though the success of the DE algorithm in terms of finding good parameter estimates is convincing, high-dimensional parameter spaces of real biological networks still are a major challenge calling for innovative fitting procedures. Here, we define a generalised network structure with randomised interactions and arbitrary units in order to test the performance of a new fitting strategy combining an iterative parameter estimation procedure inspired by [13] with the DE algorithm. The idea of the iterative strategy is the decomposition of the parameter fitting into subproblems in which data for all but one quantity are used as fixed input to the fitting process. To show the advantages of the iterative fitting routine, we use it on noisy data generated from a model with known parameters and compare it to the results of a conventional application of the DE optimiser in terms of quality of the fit, variation within the results and distance of the estimated parameters from the true parameter vector.

2. A hypothetical signalling network

The proposed optimisation method has been developed in the context of biological networks. To motivate the development of this new method and analyse its performance, we consider systems of coupled ordinary differential equations (ODE) describing hypothetical biological networks of known structure and known true parameters. These models are used to introduce the methodology and to compare its performance to a conventional parameter estimation method that requires similar computational effort. To this end, synthetic data are generated by solving the ODE systems numerically and adding artificial noise. These generated data are then used in an inverse modelling approach, i.e. adjusting the model parameters to fit the generated data. The advantage of this procedure is (a) that the performance at estimating parameters is isolated from any uncertainties in the model structure, since the true network structure is known, and (b) that not only goodness of fit, but also the distance from the known true parameter values can be compared to assess the algorithm's performance. Many biological signalling networks consist of highly complex, densely connected structures and the true network structure is often not known. Even with good biological information about the network, it is difficult to find reliable parameter estimates as a reference point for the assessment of new estimation methods. Hence, only synthetic data were used to avoid any bias introduced by assumptions about the network structure and model parameters in real biological data.

2.1. Network structure

In many biological networks such as gene regulatory or neural networks, sigmoid and log-sigmoid functions are commonly used to describe regulatory interactions between nodes [14–16]. The network considered here consists of several nodes connected by a network of regulatory relationships similar to those described in [16]. The dynamics of the network can be described by a system of the general form

$$\dot{X} = f(\Theta, X) = \begin{bmatrix} \alpha_1 \cdot \prod_{j=1}^{k} \sigma_{1j}(X_j) - \gamma_1 \cdot X_1 \\ \vdots \\ \alpha_i \cdot \prod_{j=1}^{k} \sigma_{ij}(X_j) - \gamma_i \cdot X_i \\ \vdots \\ \alpha_k \cdot \prod_{j=1}^{k} \sigma_{kj}(X_j) - \gamma_k \cdot X_k \end{bmatrix}, \tag{1}$$

where X is an k-dimensional vector of quantities (nodes) in the network. Θ is a vector with model parameters, α_i and γ_i are production and degradation rates of the quantity X_i, respectively, and $\sigma_{ij}(X_j)$ is the regulation of node X_i as a function of all quantities X_j.

The sigmoid regulatory interaction functions denoted by σ_{ij} modulate the production of the quantity X_i. In general, function values >1 have an activating effect and values <1 are inhibiting. The sigmoid functions are defined by

$$\sigma_{ij}(X_j) = p_{ij} + (q_{ij} - p_{ij}) \cdot \frac{X_j^{n_{ij}}}{K_{ij}^{n_{ij}} + X_j^{n_{ij}}}. \tag{2}$$

Each sigmoidal is uniquely defined by the parameters p, q, K and n. Whether the sigmoidals can have an activating or inhibiting effect is determined by the parameters p and q. q is the maximum value of the function in case of an activating and the minimum function value in the case of an inhibiting interaction. The parameter p gives the minimum value in case of an activation and the maximum in case of an inhibition, accordingly. K denotes the half-maximal concentration of X_j, i.e. the value of X_j where the sigmoidal reaches $\frac{1}{2}$ of its maximal effect, and n is the Hill coefficient that regulates the width of the sigmoidal. Here, a known equilibrium concentration of 1 for all quantities is assumed and the sigmoid functions are defined to be exactly 1 under these equilibrium conditions. The parameter p is fixed according to this equilibrium concentration, effectively leading to three free parameters for each sigmoidal:

$$p_{ij} = \frac{1 + K_{ij}^{n_{ij}} - q_{ij}}{K_{ij}^{n_{ij}}}. \tag{3}$$

It follows from Eqs. (2) and (3), that the sigmoid function is activating (for $X_j > 1$) if $q > 1$ and inhibiting if $0 < q < 1$. Hence, the type of an interaction function, either activating, inhibiting or no regulation, can be defined by choosing q accordingly.

A hypothetical biological network following this structure was constructed to produce the synthetic data in this study. The network consists of five quantities, $a, \ldots, e$. For each of these network nodes, a connection with any other node can be either activating, inhibiting, or absent and each connection was assigned by picking randomly from these choices with equal probabilities. The resulting network is depicted in Fig. 1 and the network structure was used throughout this project if not stated otherwise.

2.2. Synthetic data

A parameter search space was defined and model parameters were chosen randomly from this search space (Table 1). The whole model is defined in arbitrary dimensions. The coupled ODE system

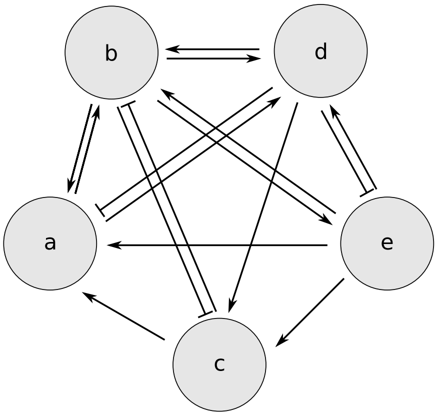

Fig. 1. The structure of the hypothetical network considered here (system 01 in Table 2). Dynamic quantities (network nodes) are depicted as circles, arrows depict an activating and crossbars an inhibiting relationship between two nodes. These interactions are sigmoid functions of the type in Eq. (2). In addition, each node has a base production rate α and degradation rate γ (not shown). Combined with the interactions, the dynamics of each network node are given by Eq. (1).

Table 1
Lower and upper boundaries defining the parameter search space (arbitrary units). The maximum effect q of the sigmoid function was chosen from [1, 100] in the case of an activating interaction and from [0.01, 1] in the case of an inhibiting interaction in order to ensure the right type of sigmoidal as given by the network in Fig. 1.

	q^{act}	q^{inh}	K	n	α	γ
Minimum	1	0.01	0.5	1	10^{-5}	10^{-5}
Maximum	100	1	100	3	1	1

defined by the network in Fig. 1 was solved numerically in the time interval [0, 100]. Data were generated for $n_t = 100$ time points. For each time point, 5 values were generated from the model output by adding 10% noise. These data are shown in Fig. 2.

2.3. Other network topologies

To assess the influence of different network sizes and topologies, networks with 2, 3, 5, 7, and 10 variables were generated, parameters randomly chosen and data generated by simulations as described above. In these networks, interactions were only considered between two different nodes in the network.

In order to test the influence of autofeedback, i.e. the regulation of one node by itself, an additional network with five nodes was generated and autofeedback was introduced with the same probability as all other regulatory relationships.

To test the influence of the interaction function used to describe the connections in the network, a logistic function given by

$$g_{ij}(X_j) = p_{ij} + (q_{ij} - p_{ij}) \cdot \frac{1}{1 + e^{-\beta x}} \tag{4}$$

was used instead of the sigmoidal function for the network in Fig. 1; simulation and data generation were performed as before.

Table 2 shows a summary of all networks and their properties as used in our simulations. For all networks, the CPU time until no significant improvement of the RSS value could be detected anymore was recorded in order to estimate the computational efficiency. The threshold for termination was set to an improvement of 0.005% over 100 generations.

3. Iterative parameter estimation strategy

3.1. Parameter estimation

We consider the case that only the synthetic data (Fig. 2) and the network structure (Fig. 1) are known while the model parameters Θ are unknown. Parameters were estimated by minimising the error between the model output and the synthetic data for each

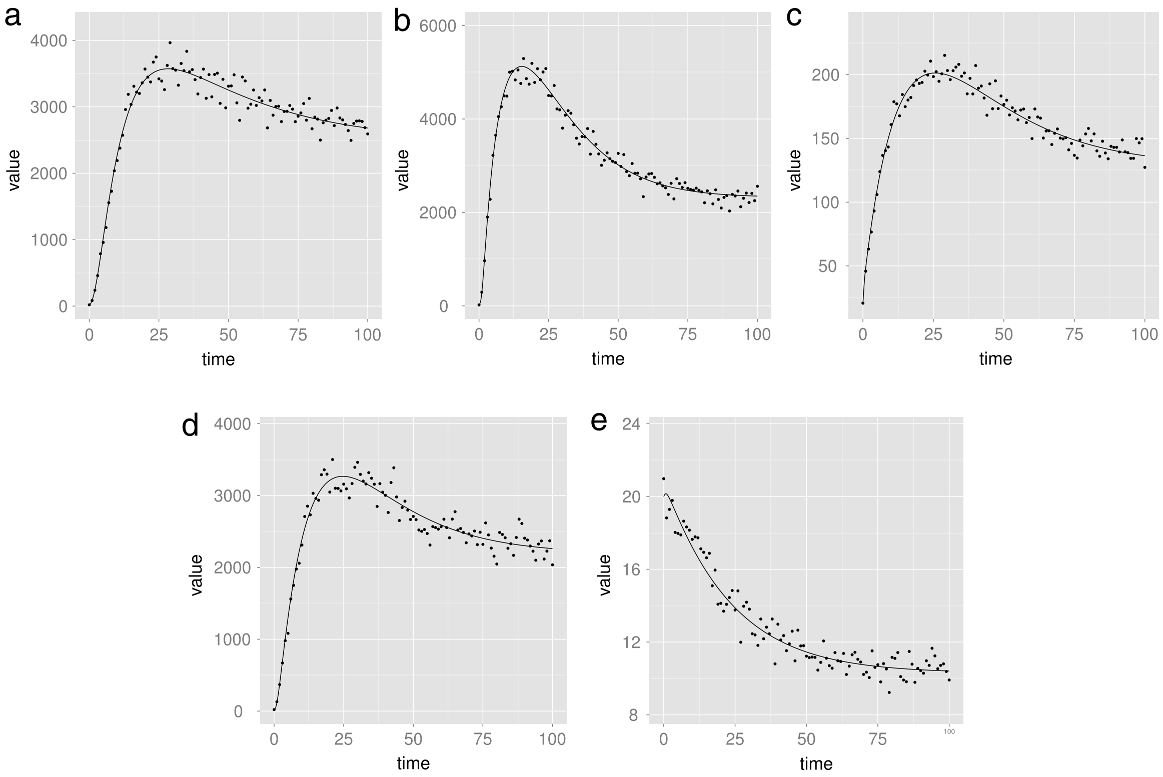

Fig. 2. Mean values of the synthetic data for the quantities a–e (network nodes in Fig. 1, system 01 in Table 2) generated as described in the text (black dots). The curve depicts the error-free values generated by numerically solving the ODE system.

Table 2
Different networks used to test the influence of network size and structure.

System	Variables	Interaction functions	Parameters	Remarks
01	5	16	53	See Fig. 1
02	2	8	2	
03	3	12	3	
04	7	13	46	
05	10	23	79	
06	5	10	35	Including autofeedback
07	5	16	53	Same structure as 01, but logistic interaction

time point as measured by the sum of all squared (weighted) residuals:

$$RSS = \sum_{j=1}^{k} \sum_{i=1}^{n_t} \left(\frac{x_{ij} - \overline{x_{ij}}}{s_{ij}} \right)^2 , \tag{5}$$

where k is the number of nodes, n_t the number of data time points, x_{ij} is the mean of the synthetic data point i for the jth node, s_{ij} is the standard deviation at data point i and $\overline{x_{ij}}$ is the current model output.

The Differential Evolution (DE) algorithm [11] with the classical DE/rand/1/bin search strategy and parameters $F = 0.8$ and $CR = 0.5$ was used for minimisation of the error function. A 10-fold of the number of parameters to estimate was used as the population size, i.e. as the number of random vectors considered in each generation of the algorithm.

3.2. Decomposing into smaller sub-problems

To decompose the high-dimensional parameter estimation task into several sub-problems, each of the coupled differential equations is analysed separately. The dynamics of all regulating quantities X_j are considered as fixed input to the system. Since the ODE integration routine requires dense data for all input nodes at different times depending on the adaptive step size used, data between two points are approximated by piecewise linear interpolation. Hence, the dynamics of one quantity X_i can be given by a single ODE:

$$\dot{X}_i = \alpha_i \cdot \prod_{j=1}^{k} \sigma_{ij}(\hat{X}_j(t)) - \gamma_i \cdot X_i , \tag{6}$$

where $\hat{X}_j(t)$ are linear piecewise functions interpolating the given synthetic data (Fig. 2) for the node X_j.

The objective function for parameter estimation (the weighted sum of squared residuals as defined in Eq. (5)) is then minimised using the DE algorithm. This procedure is repeated for all nodes in the system, each step resulting in an optimised subset of the parameter vector Θ (see Eq. (1)). These subsets are then combined and the resulting parameter vector is used as an initial guess for a final optimisation step with the whole ODE system. In this final optimisation step, parameter variation is restricted to a factor of $\delta = 10\%$ from the initial guess.

3.3. Assessment of parameter estimation success

The stochastic nature of the DE optimisation algorithm requires statistical evaluation of the properties of the tested fitting procedure. Thus, both procedures, the simple simultaneous estimation of all parameters and the iterative procedure, were repeated 50 times and the results after the final optimisation step were compared. In case of the global optimisation, the DE algorithm was terminated after 600 generations regardless of the *RSS* value and relative convergence, as the success should be assessed with a comparable computational effort in both fitting approaches. Since the iterative fitting strategy consists of six distinct steps, a maximum of 100 generations was allowed for each of these steps.

The goodness of fit of the synthetic data generated with the network (Fig. 1) is evaluated using Eq. (5). Since the true parameter values are known, the success of the parameter estimation procedure can also be assessed based on the difference between the estimated and the true parameter vector. The concordance correlation coefficient (CCC) [17] was used as a measure of the agreement between the true parameters and the estimated vectors. Similar to other correlation coefficients, CCC can assume values between −1 and 1, where 1 indicates perfect agreement between two measurements. For each parameter, a vector of 50 estimated values from the 50 optimisation runs exists. In addition to the network in Fig. 1, this procedure was repeated with all other systems in Table 2.

Furthermore, the DE algorithm itself has two central parameters that control its behaviour, the crossover probability *CR* and the differential weight *F*. Both parameter estimation methods were tested with varying *CR* and *F* to detect any differences between the two strategies in terms of the optimal DE parameters and to exclude that the results are biased by the choice of DE parameters. Due to the high computational effort of screening the whole parameter space of *F* and *CR* by repeating both fitting procedures, this screening was only repeated three times. To compare the results from the iterative fitting strategy outlined below to the results of a conventional fitting procedure using the well-known DE algorithm, the best fit was first determined by fitting all free parameters in one step. The procedure was repeated with optimisation by Simulated Annealing in place of the DE algorithm in order to compare DE to another global optimisation algorithm.

3.4. Software and libraries

The R statistics software was used for synthetic data generation and handling, statistical tests, plotting and implementation of the iterative fitting routine. The Differential Evolution algorithm was used in the excellent implementation by Mullen et al. in the DEoptim package [18] that offers a variety of search strategies implemented in C as library for R. For performance reasons, the model was implemented in C++ and numerically solved using a 5th order Dormand–Prince algorithm. Rcpp [19] was used to integrate the compiled library into the fitting routines in R. Plots were generated with the help of the ggplot2 package [20]. Generalised Simulated Annealing was performed using the GenSA package [21] using the default settings for complex fitting problems.

4. Results

4.1. Quality of the fit and reconstruction of true parameter values

When attempting to fit all parameters simultaneously (called *global* in the following), the mean of the best *RSS* values in 50 runs was 689.3. The achieved *RSS* values in each generation of the optimiser is depicted in Fig. 3. After 300–400 generations, the

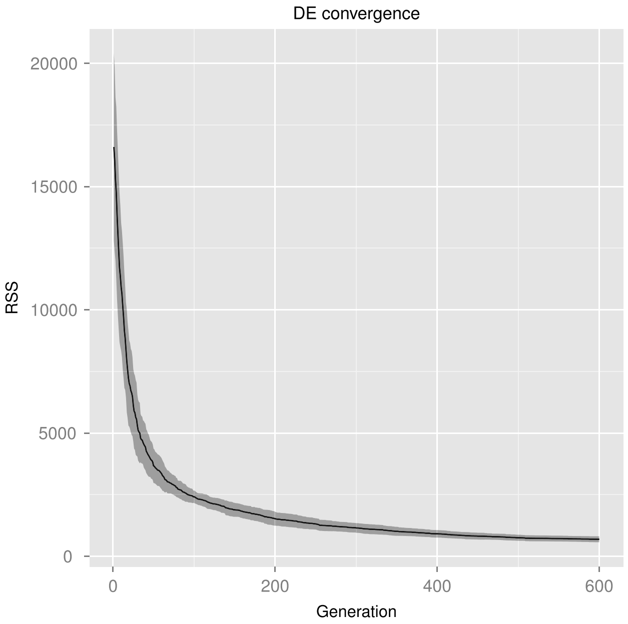

Fig. 3. Best *RSS* values in each generation of the optimiser when fitting all parameters of the model in Fig. 1 (system 01 in Table 2) simultaneously (*global* fit). The curve shows the mean of 50 runs, the shaded area represents the standard deviation.

decrease in *RSS* values is rather small, indicating that running the algorithm for more generations is unlikely to further improve the model fit to the data. When the optimisation was stopped at 600 generations, the achieved *RSS* value can thus be considered to be close to the best *RSS* value achievable in the fitting procedure.

In contrast, the iterative strategy (called *iterative* in the following) led to better results with a mean *RSS* value of 417.95. Directly comparing the distributions of the best *RSS* values achieved in both procedures indicates a statistically significant difference in the quality of the fits (Fig. 4(a)). The distribution of *RSS* values after the *global* fit is comparatively wide (Fig. 4(b), blue), indicating a strong variation in the quality of the fit, whereas the *RSS* values achieved by the *iterative* fit are closer to each other (Fig. 4(b), green).

In a final optimisation step which starts from the result of the *iterative* fit, simultaneous optimisation of all parameters is allowed in a small region of the parameter space with a maximum of 10% variation for each parameter (called *iterative* + *global* in the following). Interestingly, this final optimisation step leads to a further improvement of the quality of fit to an average *RSS* value of 266.5 (Figs. 4(a) and (b), red), indicating that the failure of the simultaneous parameter estimation to produce better fits might result from local minima or flat regions of the objective function in regions of the parameter space that were excluded by constraining the search to a reasonable range.

It is noteworthy that the average *RSS* value achieved by the *iterative+global* fit is below the *RSS* value of 273.22 achieved by comparing the simulation generated with the true parameters to the synthetic (noisy) data generated by the same parameters,

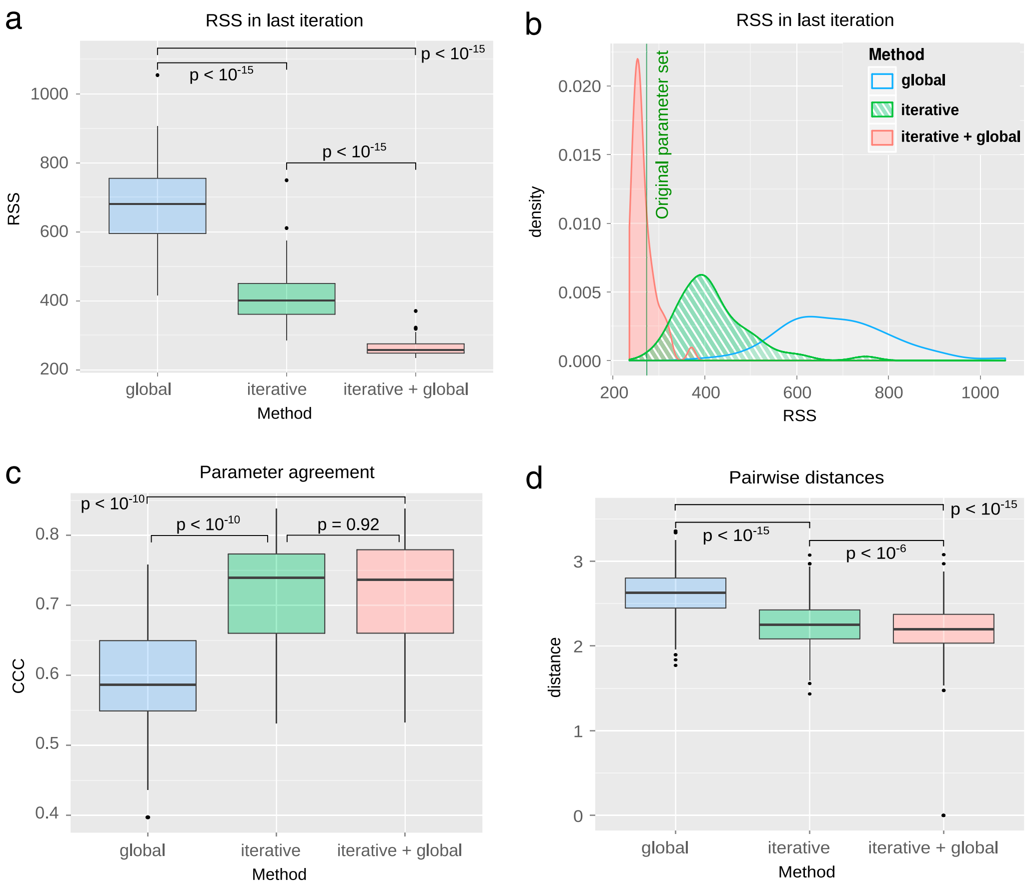

Fig. 4. Comparison of the best fits achieved in 50 attempts to estimate parameters of the model in Fig. 1 (system 01 in Table 2) using the *global* (blue), the *iterative* (green), and the *iterative* + *global* (red) fitting procedure. (a) Boxplots of the RSS values, (b) Density distribution (kernel density estimate) of the *RSS* values from 50 runs, (c) Concordance Correlation Coefficient as a measure of agreement between estimates and true parameters, (d) Euclidean Distances between all 50 estimated parameter sets are calculated by comparing each vector against each other and indicate variation within the results. Statistical significance was tested using the Welch two-sample *t*-test. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

indicated by the vertical line in Fig. 4(b). This does not represent a limitation of the fitting procedure but more generally illustrates the limits of estimating parameters from fits to data with higher amounts of noise, since the parameter vector minimising the objective function obviously differs from the true parameter vector used to generate the data. Although this certainly limits the information that can be gained from these data, time series data from biological systems frequently consist of sparse measurements with high variation and it is often desirable to have a reasonably close estimate of the parameters, even if the true parameters cannot be estimated from the data. Hence, an algorithm that estimates parameters *closer* to the true parameters can be helpful in the case of sparse or noisy data.

Whether or not the improved quality of fit seen with the iterative strategy also corresponds to better parameter estimates is assessed by calculation of their distances from the true parameter vector. While all three procedures did not reconstruct the exact true parameter values, the average CCC of the true parameters with the parameters estimated by the iterative strategy was significantly higher than that of the parameters estimated by the *global* fit (Fig. 4(c)). Furthermore, the stochastic nature of the optimisation algorithm leads to some degree of variation in the results. As shown in Fig. 4(d), the distances between parameter sets generated by simply repeating the fitting procedure are significantly lower when using the iterative strategy, indicating a reduced impact of randomness on the parameter estimates and a lower sensitivity to local minima and flat regions in the objective function.

The advantage of the *iterative* + *global* fit in comparison to the *iterative* fit seen in the quality index (Figs. 4(a) and (b)) is not reflected in the average distance from the true parameter vector (Fig. 4(c)). While there is a strong difference between the *iterative* and the *global* fit, no significant improvements by a final optimisation step could be detected. The quality of fit can be improved by optimising within a local region defined by the *iterative* fit, but this does not result in better estimates of the true parameters. It follows that after reasonably close parameter estimates are found, further optimisation is of limited relevance for the reconstruction of true parameter values. Furthermore, while there is a statistically significant difference in the variation between the estimates found in individual runs of the algorithm when a final optimisation step is applied (Fig. 4(d)), the improvement is relatively minor. We conclude that the main improvement is achieved by the *iterative* fit alone and the global final optimisation step is not required.

The number of generations was intentionally chosen very conservatively with respect to the computational effort in the iterative strategy by interrupting the fitting after the same number of generations in total as in the global fitting of all parameters (100 generations in six steps including the final optimisation vs. 600 generations in one iteration). The number of parameter vectors per generation was set to 10 times the number of parameters to fit in all cases, which means that the average computational effort in terms of function evaluation and CPU cycles per generation can be assumed to be significantly lower in the iterative strategy. The results are likely to underestimate the performance of the iterative strategy. For practical applications and exploration of the limitations of the procedure, the algorithm should be terminated by a convergence criterion instead, i.e. it should terminate if it fails to produce a significant difference in the best value of each generation after a certain number of generations.

4.2. Quality of fit and reproduction of true parameter values in other network topologies

To assess the quality of parameter estimates in different network topologies, simulations were repeated in seven different systems ranging in complexity, number of parameters and structural details (see Table 2 for details). The *iterative* strategy produced significantly better parameter estimates and quality of fit than the *global* strategy in all cases (Fig. 5). The advantage for both agreement between parameter estimates and true parameters (Fig. 5(c)) and quality of fit (Fig. 5(d)) was consistent in all tested networks and slightly increased with the numbers of variables.

Although the quality of fit was significantly improved by adding a final optimisation step in four of the tested networks (Fig. 5(a)), this did not correspond to an improvement in the quality of the parameter estimates (Fig. 5(b)). In the most complex of the systems, system 05 (see Table 2), the parameter estimates were even worse with the *iterative* + *global* strategy than with *iterative* alone although the quality of fit was slightly, albeit not significantly, improved by the final optimisation step. The reason for this behaviour is the same as discussed for system 01: the quality of fit may be improved by further optimisation, but the data are not sufficient to lead to a corresponding improvement in parameter estimates.

Neither the presence of autofeedback (system 06) nor the choice of a logistic function as interaction function instead of the sigmoidals (system 07) resulted in qualitatively different results, indicating that the observed behaviour is robust in the tested networks.

The variation δ (see Section 3.2) for the final optimisation step in the *iterative* + *global* strategy was kept at a low level in order to make sure that parameters are only optimised in a local neighbourhood of the vector estimated in the preceding *iterative* strategy. The final optimisation step can generally not be recommended, as an increase in quality of fit did not result in better parameter estimates and larger variations are, hence, unlikely to improve parameter estimates.

4.3. Robustness of results under different optimiser parameters

Comparing the quality of the fits produced by the two strategies, it is obvious that the *RSS* is consistently lower with the iterative fitting strategy in a wide region of DE parameters (Fig. 6). The performance of the iterative fitting routine is stable over most of the scanned parameter space, although large crossover probabilities *CR* seem to slightly decrease the quality of the results. Although these results show that the advantage of the iterative routine is independent of the DE parameters, optimisation of DE parameters and DE search strategies other than the classical DE strategy with fixed parameters might lead to further improvements. The integration of adaptive meta-optimisation of the DE parameters as proposed in [22] or [23] might significantly improve convergence and performance of the algorithm.

4.4. Results using generalised simulated annealing

The advantage of the *iterative* strategy is maintained even using a completely different algorithm for each optimisation step Fig. 7. However, the differences were less pronounced and the variation was higher than with Differential Evolution. Overall, the quality of the parameter estimates was worse with Generalised Simulated Annealing than with the DE algorithm. Since DE has been shown to produce parameter estimates of better quality than other optimisation algorithms for large nonconvex problems [12], its general advantage for the problem class in this study is expected. It should be noted that the CCC values achieved by Simulated Annealing indicate a poor fit between parameter estimates and true parameters. Furthermore, the computation time required was on average more than ten times higher with Simulated Annealing compared to Differential Evolution.

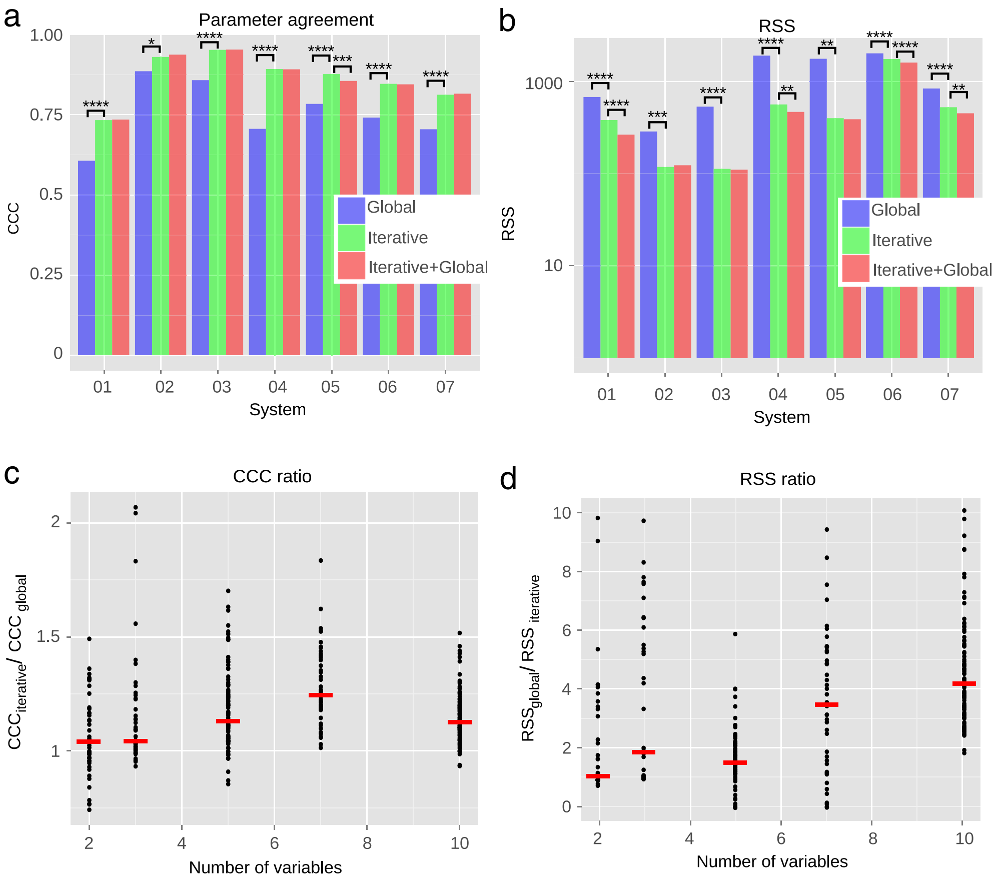

Fig. 5. Comparison of the best fits achieved in 50 attempts to estimate model parameters using the *global* (blue/left bar), the *iterative* (green/centre), and the *iterative* + *global* (red/right bar) fitting procedure in different network topologies (see Table 2). (a) Concordance correlation coefficient, (b) RSS values, (c) Ratio of concordance correlation coefficients of the iterative to the global method, (d) Ratio of RSS values of the global to the iterative method. Asterisks indicate statistical significance between methods; ****: $p < 0.0001$, ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$ as determined by the Welch two-sample t-test. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

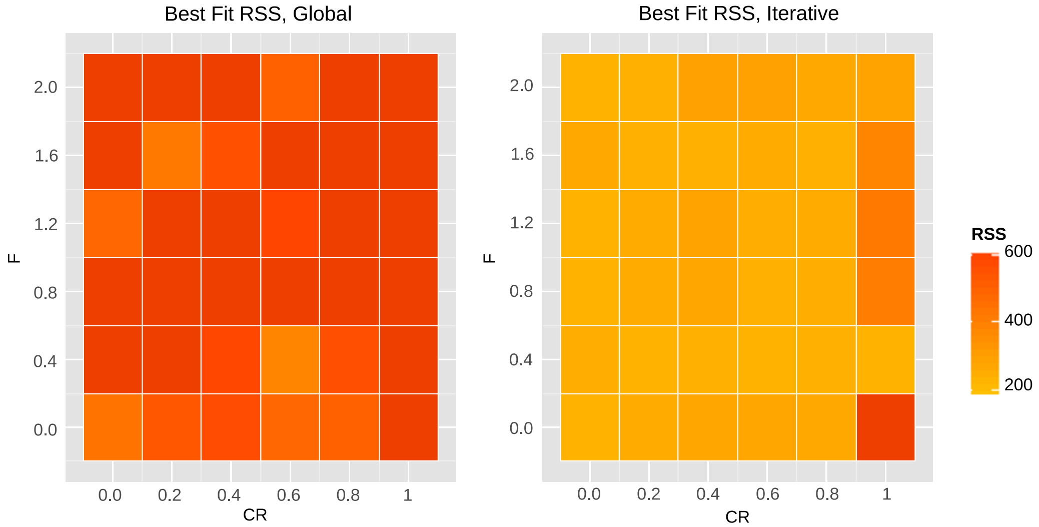

Fig. 6. Comparison of the best fits achieved for parameters of the model in Fig. 1 (system 01 in Table 2) using the *iterative* and the *global* fitting procedure with different values for the DE parameters *F* and *CR*. The *RSS* values are indicated by the colour. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.5. *Computational efficiency*

In order to estimate the computational efficiency of the different methods, computation times were recorded. Fig. 8 shows the CPU time until termination of the algorithm in seconds for optimisations of the systems in Table 2 and the numbers of parameters. Regardless of the complexity of the system and the number of parameters, there is a significant advantage in terms of computational efficiency for the iterative method. The advantage is higher in more complex systems, as the computation time grows faster with the number of parameters for the global than for the iterative method.

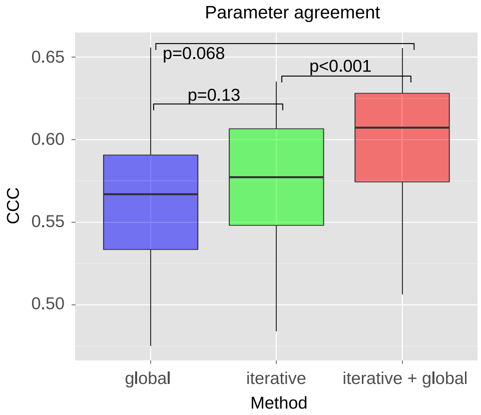

Fig. 7. Concordance correlation coefficient for parameters of the model in Fig. 1 (system 01 in Table 2) optimised by Simulated Annealing instead of Differential Evolution. Comparison of the best fits achieved in 50 attempts to estimate model parameters using the *global* (blue), the *iterative* (green), and the *iterative* + *global* (red) fitting procedure. Statistical significance was tested using the Welch two-sample t-test. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

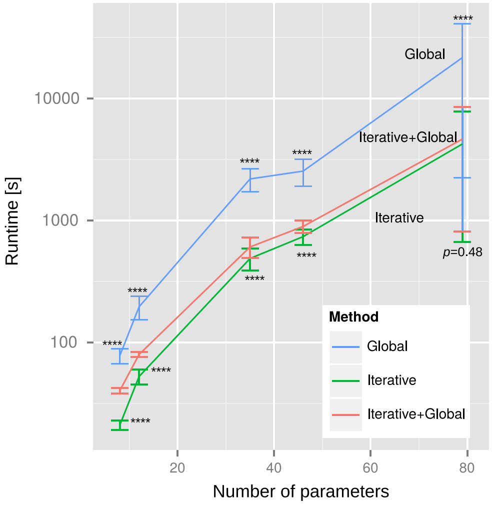

Fig. 8. Computation times in seconds CPU time in 50 attempts to estimate parameters of the model in Fig. 1 (system 01 in Table 2) using the *global* (blue), the *iterative* (green), and the *iterative* + *global* (red) fitting procedure. The algorithm was terminated if no significant improvement (more than 0.005%) could be detected over 100 generations of the optimisation algorithm. Asterisks indicate the statistical significance between the *iterative* and the *global* method (top) and between the *iterative* and the *iterative* + *global* method (bottom); ****: $p < 0.0001$ as determined by the Welch two-sample t-test. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5. Conclusion

In the given class of parameter estimation problems from biological networks, iteratively optimising model parameters for single equations by the classical Differential Evolution algorithm while considering data for all other nodes as input to the system offers significantly improved performance compared to using the same optimisation algorithm and the same dataset to estimate all parameters simultaneously. This was shown in terms of the required CPU time, the quality of the fit produced by estimated parameters and distance from the true parameters. The estimates gained by the iterative routine were consistently closer to the true parameters. Further optimisation in this region of the parameter space improved the quality of fit, but not the parameter estimates.

The proposed iterative method robustly provided advantages in different network topologies. This advantage increased with the size of the networks and was confirmed for both tested interaction functions.

Under practical aspects, the procedure is limited by the availability of data, since it requires known data for all modelled quantities. An iterative strategy could be imagined for cases where data for one or more state variables are missing that would involve simultaneous fitting of a subset of the state variables with a larger number of parameters.

Acknowledgements

This work was supported by the Helmholtz Graduate School for Infection Research, by the Human Frontier Science Program, by the project GerontoShield in the framework of the BMBF GerontoSys initiative, by the projects SYSIMIT and SysStomach in the framework of the BMBF eMED initiative, and by iMed — the Helmholtz Initiative on Personalized Medicine.

References

[1] J.A. Papin, T. Hunter, B.O. Palsson, S. Subramaniam, Nat. Rev. Mol. Cell Biol. 6 (2) (2005) 99–111. URL http://www.nature.com/nrm/journal/v6/n2/full/nrm1570.html, http://dx.doi.org/10.1038/nrm1570.
[2] Z. Frankenstein, U. Alon, I.R. Cohen, Biol. Direct 1 (2006) 32. URL http://www.biologydirect.com/content/1/1/32, http://dx.doi.org/10.1186/1745-6150-1-32.
[3] U.S. Bhalla, Ravi Iyengar, Science 283 (5400) (1999) 381–387. URL http://www.sciencemag.org/cgi/doi/10.1126/science.283.5400.381, http://dx.doi.org/10.1126/science.283.5400.381.
[4] J. Haugh, Wiley Encyclopedia of Chemical Biology, Wiley, 2008, pp. 1–9. URL http://onlinelibrary.wiley.com/doi/10.1002/9780470048672.wecb646/full, http://dx.doi.org/10.1002/9780470048672.wecb646.
[5] N.J. Eungdamrong, R. Iyengar, Biol. Cell 96 (5) (2004) 355–362. URL http://www.biolcell.org/boc/096/boc0960355.htm, http://dx.doi.org/10.1016/j.biolcel.2004.03.004.
[6] B. Goldstein, J.R. Faeder, W.S. Hlavacek, Nat. Rev. Immunol. 4 (6) (2004) 445–456. URL http://www.nature.com/nri/journal/v4/n6/full/nri1374.html, http://dx.doi.org/10.1038/nri1374.
[7] I.-C. Chou, E.O. Voit, Math. Biosci. 219 (2) (2009) 57–83. URL http://www.sciencedirect.com/science/article/pii/S0025556409000583, http://dx.doi.org/10.1016/j.mbs.2009.03.002.
[8] P. Mendes, D. Kell, Bioinformatics 14 (10) (1998) 869–883. URL http://bioinformatics.oxfordjournals.org/content/14/10/869, http://dx.doi.org/10.1093/bioinformatics/14.10.869.
[9] L. Ingber, B. Rosen, Math. Comput. Model. 16 (11) (1992) 87–100. URL http://www.sciencedirect.com/science/article/pii/089571779290108W, http://dx.doi.org/10.1016/0895-7177(92)90108-W.
[10] Y. Shi, R. Eberhart, IEEE International Conference on Evolutionary Computation. Anchorage, IEEE, 1998, pp. 69–73. URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=699146, http://dx.doi.org/10.1109/ICEC.1998.699146.
[11] R. Storn, K. Price, J. Global Optim. (1997) 341–359. URL http://link.springer.com/article/10.1023/A:1008202821328, http://dx.doi.org/10.1023/A:1008202821328.
[12] C.G. Moles, J.R. Banga, K. Keller, Appl. Soft Comput. 5 (1) (2004) 35–44. URL http://www.sciencedirect.com/science/article/pii/S1568494604000559, http://dx.doi.org/10.1016/j.asoc.2004.03.011.
[13] Y. Maki, T. Ueda, M. Okamoto, N. Uematsu, K. Inamura, K. Uchida, Y. Takahashi, Y. Eguchi, Genome Inform. 13 (2002) 382–383. URL http://www.jsbi.org/pdfs/journal1/GIW02/GIW02P054.pdf.
[14] D. Marbach, C. Mattiussi, D. Floreano, Ann. NY Acad. Sci. 1158 (2009) 234–245. URL http://onlinelibrary.wiley.com/doi/10.1111/j.1749-6632.2008.03944.x/abstract, http://dx.doi.org/10.1111/j.1749-6632.2008.03944.x.
[15] T. Mestl, E. Plahte, S.W. Omholt, J. Theoret. Biol. 176 (2) (1995) 291–300. URL http://www.sciencedirect.com/science/article/pii/S0022519385701993, http://dx.doi.org/10.1006/jtbi.1995.0199.
[16] M. Meyer-Hermann, M.T. Figge, R.H. Straub, Arthritis Rheum. 60 (9) (2009) 2585–2594. URL http://onlinelibrary.wiley.com/doi/10.1002/art.24797/abstract, http://dx.doi.org/10.1002/art.24797.
[17] L.I.-K. Lin, Biometrics 45 (1) (1989) 255–268. URL http://link.springer.com/10.1007/s00500-004-0363-x, http://dx.doi.org/10.1007/s00500-004-0363-x.
[18] K.M. Mullen, D. Ardia, D.L. Gil, D. Windover, J. Cline, J. Stat. Softw. 40 (6) (2011) URL http://www.jstatsoft.org/v40/i06/.
[19] D. Eddelbuettel, R. François, J. Stat. Softw. 40 (8) (2011) URL http://www.jstatsoft.org/v40/i08/.
[20] H. Wickham, ggplot2: Elegant Graphics for Data Analysis, Springer, New York, 2009, URL http://had.co.nz/ggplot2/book.
[21] Y. Xiang, S. Gubian, B. Suomela, J. Hoeng, The R Journal 5 (1) (2013) 13–28.
[22] J. Liu, J. Lampinen, Soft Comput. 9 (6) (2004) 448–462. URL http://link.springer.com/10.1007/s00500-004-0363-x, http://dx.doi.org/10.1007/s00500-004-0363-x.
[23] J. Brest, S. Greiner, B. Bošković, M. Mernik, V. Žumer, IEEE Trans. Evol. Comput. 10 (6) (2006) 646–657. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4016057, http://dx.doi.org/10.1109/TEVC.2006.872133.