S-system parameter estimation for noisy metabolic profiles using Newton-flow analysis

Z. Kutalik, W. Tucker and V. Moulton

Abstract: Biochemical systems are commonly modelled by systems of ordinary differential equations (ODEs). A particular class of such models called S-systems have recently gained popularity in biochemical system modelling. The parameters of an S-system are usually estimated from time-course profiles. However, finding these estimates is a difficult computational problem. Moreover, although several methods have been recently proposed to solve this problem for ideal profiles, relatively little progress has been reported for noisy profiles. We describe a special feature of a Newton-flow optimisation problem associated with S-system parameter estimation. This enables us to significantly reduce the search space, and also lends itself to parameter estimation for noisy data. We illustrate the applicability of our method by applying it to noisy time-course data synthetically produced from previously published 4- and 30-dimensional S-systems. In addition, we propose an extension of our method that allows the detection of network topologies for small S-systems. We introduce a new method for estimating S-system parameters from time-course profiles. We show that the performance of this method compares favorably with competing methods for ideal profiles, and that it also allows the determination of parameters for noisy profiles.

1 Background

In order to reveal the functions of metabolites within the cell, genomic, metabolic and proteomic data need to be integrated with system models to allow the testing of hypotheses, and eventually provide insights into the underlying mechanisms. Time course profiles, that is, measurements of quantities such as gene activity or metabolite concentration at series of time points, are one of the most suitable data types for such an integration effort. In particular, the post-genomic era has granted us with various high-throughput methods, such as microarrays and mass spectrometry, which allow us to simultaneously measure the activities of hundreds of genes or metabolites in biochemical systems [1].

Such systems are commonly modelled using systems of ordinary differential equations (ODEs) [2–4]. A particular class of such models called S-systems have recently gained popularity in biochemical system modelling [5]. One of the main advantages of S-systems is that model parameters are intimately linked with the underlying structure of the biochemical reactions. Even so, deriving appropriate estimates for the values of these parameters tends to be computationally expensive, which can be problematic since parameter values can greatly influence the dynamics of the model.

S-system parameters are usually estimated from time-course profiles [5], and various techniques have been introduced for estimating S-system parameters [5–12]. However, with the exception of [9, 12], most of the aforementioned methods have only been tested on ideal data, that is, data absent of noise. This is probably because of the fact that noise can seriously deteriorate identification of parameters [5, 13]. An extensive review paper [14] on global optimization methods favored a stochastic algorithm, evolution strategies (ES). Recent attempts have been made to estimate S-system parameters from noisy (and real) data using simulated annealing [15], or evolutionary optimization [16].

In this paper, we show that a certain minimisation problem for S-systems appears to have an important structural feature that can be exploited to estimate S-system parameters for noisy data. In particular, for a wide range of examples, we demonstrate that there is an easily identifiable one-dimensional attractor (which contains the true solution) for the Newton flow corresponding to a standard minimization problem for S-system parameter estimation. Thus, we only need to search this one-dimensional attractor (instead of the $2n+2$ dimensional parameter space) to find the optimal parameter vector. Consequently, this attractor allows us to considerably reduce parameter search space and, as we shall see, lends itself to the estimation of S-system parameters for noisy data.

The structure of the rest of this paper is as follows. First we describe our parameter estimation algorithm for S-systems. We then illustrate its applicability by investigating its performance on previously published 4- and 30-dimensional biochemical S-systems in the presence of noise. In addition, we indicate a method that might be used to estimate a biochemical network structure in case this is not known in advance. We conclude with a discussion concerning our new method.

2 Methods

Before describing our method, we begin by reviewing some facts concerning S-systems.

doi:10.1049/iet-syb:20060064

Paper first received 6th September 2006 and in revised form 20th January 2007

Z. Kutalik is with the Department of Medical Genetics, University of Lausanne, Rue de Bugnon 27, Lausanne 1005, Switzerland

W. Tucker is with the Department of Mathematics, Uppsala University, Uppsala S-75106, Sweden

V. Moulton is with the School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, United Kingdom

E-mail: zoltan.kutalik@unil.ch

2.1 S-systems

For a biochemical system having n chemical constituents, let $x_i(t)$, $1 \le i \le n$, denote the concentration of the ith chemical at time t. An S-system relating these concentrations is a system of ODEs with the following special form

$$\dot{x}_i(t) = \alpha_i \prod_{k=1}^{n} x_k(t)^{g_{ik}} - \beta_i \prod_{k=1}^{n} x_k(t)^{h_{ik}} \qquad i = 1, 2, \ldots, n, \qquad (1)$$

where α_i, $\beta_i \ge 0$ and g_{ik}, h_{ik} are all real constants (called rate constants and kinetic orders, respectively), and $\dot{x}_i(t)$ denotes the time derivative of concentration x_i at time i (cf. [5]). An example of such a system is presented in Fig. 1. Note that the non-zero parameters determine the structure of the underlying network of biochemical reactions.

In this paper, we consider the following parameter estimation problem as described, for example, by [7]: Given measurements x_{ij} $(i = 1, \ldots, n, j = 1, \ldots, N)$ of the concentration $x_i(t_j)$ of the ith chemical at time t_j, and estimates dx_{ij} of the rate of change $\dot{x}_i(t_j)$ of x_i at time t_j, determine the S-system parameter values α_i, β_i, g_{ik}, h_{ik} so that the solution to the S-system will come as close as possible to these values. It should be emphasized that our algorithm requires all x_{ij} to be known. Note that we will first consider the problem in which some of the parameters α_i, β_i, g_{ik}, h_{ik} have already been set to zero, corresponding to a model consisting of a fixed network of biochemical reactions. We will discuss the more difficult problem of deciding which parameters to set to zero (so as to decide which network of reactions to choose) at the end of the results section.

2.2 Main steps in parameter estimation

Our approach for parameter estimation consists of the following four steps:

1. As described by [7, 17], for each $1 \le i \le n$ and $1 \le j \le N$ the values dx_{ij} are used to decouple the S-system (1) into $n \cdot N$ algebraic equalities.
2. For the ith set of N equations obtained in step (1) by fixing i, $1 \le i \le n$, a least-squared minimisation problem is set-up. In particular, for

$$f(\mathbf{p}_i) = \sum_{j=1}^{N} \left(dx_{ij} - \alpha_i \prod_{k=1}^{n} x_{kj}^{g_{ik}} + \beta_i \prod_{k=1}^{n} x_{kj}^{h_{ik}} \right)^2 \qquad (2)$$

and

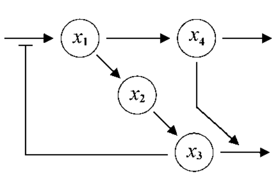
$$\mathbf{p}_i = (\alpha_i, g_{i,1}, \ldots, g_{i,n}, \beta_i, h_{i,1}, \ldots, h_{i,n})$$

where we aim to find a value for $\mathbf{p}_i$ that minimizes $f(\mathbf{p}_i)$ subject to constraints

$$\mathbf{l}_i \le \mathbf{p}_i \le \mathbf{u}_i$$

with $\mathbf{l}_i$ and $\mathbf{u}_i$ constants. The values of $\mathbf{l}_i$ and $\mathbf{u}_i$ are often derived from specialist knowledge of the biochemical system in question [7, 8].
3. Each constrained optimization problem from step (2) is solved using the interior-reflective Newton method [18]. Our experimental and theoretical investigations strongly suggest that the Newton flow [19] of the minimization problem contains a one-dimensional attractor in parameter space (the dimensions of which are made up by α_i, g_{i1}, g_{i2}, ..., g_{in}, β_i, h_{i1}, h_{i2},...h_{in}), which can be written in the form

$$\gamma_j(w) = \begin{cases} w & j = 1 \\ a_{j-1}/(b_{j-1} + w) & j = 2, \ldots, n+1 \\ w + a_0 & j = n+2 \\ c_{j-n-2}/(d_{j-n-2} + w) & j = n+3, \ldots, 2n+2, \end{cases} \qquad (3)$$



$$\begin{aligned} \dot{x}_1 &= 12x_3^{-0.8} - 10x_1^{0.5} \\ \dot{x}_2 &= 8x_1^{0.5} - 3x_2^{0.75} \\ \dot{x}_3 &= 3x_2^{0.75} - 5x_3^{0.5}x_4^{0.2} \\ \dot{x}_4 &= 2x_1^{0.5} - 6x_4^{0.8} \end{aligned}$$

Fig. 1 *Schematic representation of a 4D pathway with its governing equations*

a Branched pathway with activations and inhibitions
b Corresponding S-system. See [7] for more details

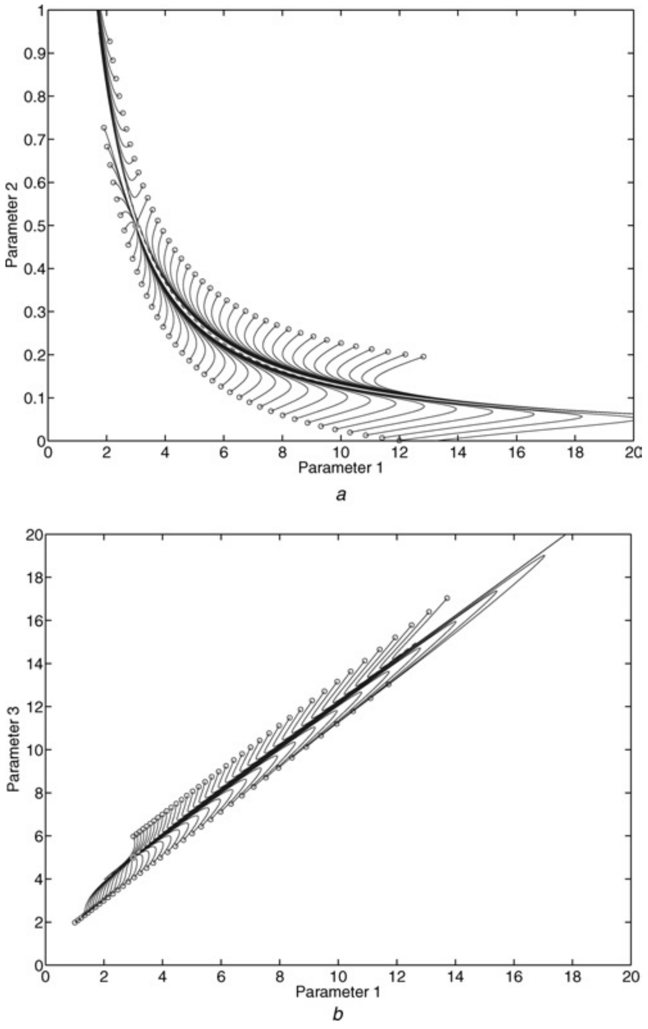


Fig. 2 *Two types of attractor projections*

Two dimensional projections of the phase plane of the Newton-flow of the minimisation problem associated with a 4-dimensional S-system The hypothesised attractor is marked with a dotted line, the trajectories are shown in a solid line and the light dot on the attractor represents the global optimum
(a) parameter α_3 vs. g_{33}; (b) parameter α_3 vs. β_3

where a_j, b_j, c_j, d_j are real constants (for an example of a typical phase plane projection see Fig. 2). Variable j loops through the parameter indices: $j = 1$ means α_i, $j = 2{:}g_{i1}$, $j = 3{:}g_{i2}$, $\ldots$, $j = n+1{:}g_{in}$, $j = n+2{:}\beta_i$, $j = n+3{:}h_{i1}$, $j = n+4{:}h_{i2}$, $\ldots$, $j = 2n+2{:}h_{in}$. To rigorously prove the existence of an attractor of this form in general appears to be a difficult mathematical problem. However, we have been able to show that such an attractor exists for various S-systems, including the 4-dimensional example presented in [7], a 30-dimensional system from [5], and a non-sparse 7-dimensional S-system (see Supplementary Material).

Since the Newton method depends on initial guesses, we first generate 40 uniformly distributed random guesses (i.e. the coordinates of the vectors are drawn from a uniform distribution in the bounding box of the parameters). After running the Newton method (for maximum 50 iterations) on each of these initial guesses, we obtain several points in the parameter space, which we assume to lie near to the hypothesised attractor (see Fig. 2). Using the robust bisquare regression (implemented in Matlab) in two-dimensions, we then use these points to estimate the parameters a_j, b_j, c_j and d_j in (3). If each regression produces a goodness of fit R^2 value greater than 0.9 we proceed to step (4) (otherwise, we stop, as probably either the given model is unable to describe the data sufficiently well, or the available data is insufficient for reliable parameter estimation).

4. Finally, using the attractor equations estimated in step (3), we perform the Newton algorithm again with initial guesses lying on the estimated attractor. In particular, we split each bounding interval for α, $[\mathbf{l}_i, \mathbf{u}_i]$, into M equal parts $(\mathbf{l}_i, = w_0 \leq w_1 \leq \cdots \leq w_M = \mathbf{u}_i)$ and run the Newton algorithm again for each of the initial points $(\gamma(w_0), \ldots, \gamma(w_M))$. This results in new estimates for the S-system parameters. Our final parameter estimate is then taken to be the one for which the cost function f takes on its lowest value.

3 Results and discussion

To assess the applicability of our method, we tested it on simulated data sets obtained from S-systems models of biological systems described in the literature, as in example [7]. In particular, for a given S-system, and randomly generated initial concentrations, $x_i(t_1)$, profiles were obtained by numerical integration of (1) over the time interval $[T_0, T_1]$. Measurement points were selected to be equidistant on logarithmic scale since measurements during the highly variant initial phase carry more information for the parameter estimation [3]. Specifically, time points were defined by the formula

$$t_j = T_1 \frac{c^{j-1} - 1}{c^{N-1} - 1} \quad j = 1, 2, \ldots, N$$

where c and N are constants. We set $T_0 = 0$, $T_1 = 20$, and $N = 20$. We chose $N = 20$ to remain within realistic experimental limits. [7] used 6 data sets, each of $N = 100$ time point samples, while in [9] 20 data sets were used each containing $N = 11$–14 samples. We put $c = 1.2$ to allow the collection of 20 data points well before the system approached its equilibrium state.

Subsequently Gaussian noise was added to both the true concentration and derivative values to obtain simulated measurements. In particular, we took

$$\begin{aligned} x_{ij} &= x_i(t_j)(1 + r_j) \\ dx_{ij} &= \dot{x}_i(t_j)(1 + s_j) \end{aligned} \qquad (4)$$

where $r_j \sim \mathcal{N}(0, \sigma^2)$ and $s_j \sim \mathcal{N}(0, (2\sigma)^2)$ (with $\mathcal{N}(a, b^2)$ denoting a normal distribution with mean value a and standard deviation b). The relative error among the slope estimates can vary considerably, and depends on the method by which they are derived. In our simulations–using a spline fitting method–we obtained a median slope estimate deviating by approximately $\mathbf{2}\sigma$ from the real slope values. Therefore, we assumed that the slope estimates could have twice as much relative noise as the observed measurement values. Note that even though the relative error in the slope estimation is not homogeneous, high relative noise occurs mostly for near zero slopes. However, such terms are almost negligible in the objective function optimization, and so do not greatly effect our parameter estimation.

In our implementation we used the Matlab command `lsqnonlin` to carry out the Newton minimisation.

3.1 4-dimensional example

We first consider the following 4-dimensional example that was studied by [7] (see also [17]), as depicted in Fig. 1.

To investigate the performance of our method we applied it to simulated data with various relative noise levels: $\sigma = 0$, 0.02, 0.05, 0.1, 0.2. We simulated 16 noisy data sets (4 replicates of 4 data sets each with different initial conditions). Thus for each substrate, x_i $(i = 1, \ldots, 4)$ we have $20 \times 16 = 320$ equations. A typical collection of profiles is shown in Fig. 3.

As in [7], parameters α_i, β_i were assumed to lie in the interval [0, 20], and g_{ik}, h_{ik} in [−1, 1]. However, we found that the results generated by our method were not greatly affected by the choice of bounds.

The parameters of the system (1) are shown in Table 1. To find the average performance of our algorithm, we generated eight replicates of the 16 data sets (for each noise level) and ran our algorithm for the eight realisations of the complete experiment. We then computed the median of the relative errors for each of the eight runs for each parameter. These median relative errors for all parameters are summarised in Table 1. The average of all (median) relative parameter errors were 0%, 3.9%, 10.04%, 18.53%, 37.49% for noise levels 0%, 2%, 5%,10%, 20%, respectively. Note

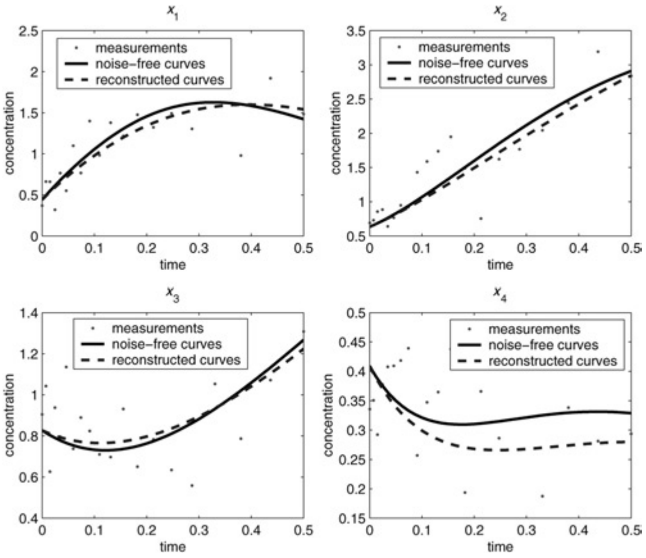


Fig. 3 *Reconstructed concentration profiles*

Typical concentration profiles of the 4-dimensional example
Measurement data are obtained by adding 20% relative noise to the noise-free concentration curves

Table 1: Median relative error of the parameters for different noise levels

Parameters									
12.00	–	–	−0.80	–	10.00	0.50	–	–	–
8.00	0.50	–	–	–	3.00	–	0.75	–	–
3.00	–	0.75	–	–	5.00	–	–	0.50	0.20
2.00	0.50	–	–	–	6.00	–	–	–	0.80
Relative error of estimates for 0% noise									
0	–	–	0	–	0	0	–	–	–
0	0	–	–	–	0	–	0	–	–
0	–	0	–	–	0	–	–	0	0
0	0	–	–	–	0	–	–	–	0
Relative error of estimates for 2% noise									
0.0391	–	–	0.032	–	0.0474	0.0419	–	–	0
0.0241	0.0154	–	–	–	0.0606	–	0.041	–	0
0.0484	–	0.0379	–	–	0.0324	–	–	0.0557	0.0283
0.0588	0.0488	–	–	–	0.0082	–	–	–	0.0424
Relative error of estimates for 5% noise									
0.0101	–	–	0.0313	–	0.0092	0.0163	–	–	–
0.0657	0.0523	–	–	–	0.1923	–	0.1253	–	–
0.2859	–	0.2023	–	–	0.1734	–	–	0.2883	0.0412
0.0683	0.0365	–	–	–	0.0162	–	–	–	0.0919
Relative error of estimates for 10% noise									
0.1221	–	–	0.1344	–	0.1544	0.1328	–	–	–
0.1331	0.0639	–	–	–	0.3344	–	0.1421	–	–
0.2912	–	0.254	–	–	0.1568	–	–	0.4611	0.1602
0.2134	0.142	–	–	–	0.0567	–	–	–	0.197
Relative error of estimates for 20% noise									
0.5261	–	–	0.4849	–	0.612	0.422	–	–	–
0.2839	0.1694	–	–	–	0.6576	–	0.3333	–	–
0.2691	–	0.2539	–	–	0.1402	–	–	0.697	0.333
0.2769	0.3305	–	–	–	0.2528	–	–	–	0.3303

that these values were obtained when the true underlying topology was assumed to be known (i.e. when all the zero parameters are all assumed to be equal to 0).

As expected, the error in parameter estimation increases as the relative noise level amplifies. We also found that the average relative parameter error is proportional to the relative noise level of the data. This fact is supported by theoretical findings presented in the Discussion. It can be clearly seen in Table 1 that for all noise levels the largest mean error occurred for the parameters in the third equation ($\dot{x}_3 = 3x_2^{0.75} - 5x_3^{0.5}x_4^{0.2}$), which is not surprising as it involves the most parameters.

Using our algorithm it took 90 seconds (on a single Pentium 4 PC: Dell Poweredge 2800 Bi-dual Xeon 2.8 Ghz with 4Gb RAM) to estimate the 17 unknown parameters for the above 4-dimensional example. The run time is almost independent of the noise level.

3.2 30-dimensional example

To investigate the performance of our method when applied to a higher-dimensional example we applied it to the 30-dimensional S-system corresponding to a gene network described in [9] (originally published in [20]). Note that here instead of metabolite concentration profiles we model gene activities over time.

We employed exactly the same setup as presented by [9] (for details see the Supplementary Material): initial concentrations were randomly taken from [0, 2]; 20 data sets of 11 sampling points were generated; 2%, 5%, 10%, 20% relative Gaussian noise has been added to the simulated data.

Note that since we decouple the system, the complexity of the parameter estimation is determined by the sparsity of the network. Thus, even though the system is 30-dimensional, due to its sparsity the parameter estimation does not require much more computational effort than the 4-dimensional example.

As before, we repeated our test for 8 runs to measure the mean performance of our parameter estimation. The relative error for each type parameter is presented in Table 2. As in the case of the 4-dimensional example, the median relative error shows a more or less linear relationship with the relative noise level. In [9] only the topology was estimated, therefore no direct comparison can be drawn with our results.

The total run time of our algorithm to estimate all the 128 unknown parameters of the above 30-dimensional example was between 10–14min on a Pentium 4 PC (Bi-dual Xeon 2.8 Ghz, 4 Gb RAM, Dell Poweredge 2800). The run time increases somewhat with higher noise levels.

Table 2: Median relative error of the parameters of the 30-dimensional example for various relative noise level

Median relative error					
Noise (%)	α	g	β	h	Median error
2	0.0111	0.0115	0.0140	0.0082	0.0106
5	0.0566	0.0290	0.0689	0.0403	0.0494
10	0.1841	0.1320	0.2389	0.1361	0.1521
20	0.5231	0.3217	0.7029	0.3357	0.3754

Parameters are grouped into rate constants and kinetic orders. Medians are computed across all equations for the respective group (α, g, β and h) in Table 1 in the Supplementary Material. The last column is the median of all relative errors in the full table

3.3 *Identifying a network topology*

Up to this point we have only used our algorithm to estimate parameters in case we have assumed *a priori* which terms h_{ik} and g_{ik} to set to zero (corresponding to a fixed network of interactions). We now propose a way to extend our method to estimate parameters in case the underlying network is not assumed to be known in advance, only that $h_{ik} \cdot g_{ik} \neq 0$ for all i, j. This latter assumption is vital, otherwise we run into identifiability problems due to the extremely high correlation between h_{ik} and g_{ik}.

Note that due to the decoupling of the ODE system, we are in essence searching for networks all of whose edges contain a certain vertex. Let $\mathcal{N}_i$ denote the collection of networks that have at least one directed edge ending at vertex i. For example, in Fig. 1, for $i = 3$ we have 2 edges ending at x_3.

Our algorithm can now be extended in the following way. We run steps (i)–(iii) of our algorithm for minimal networks in $\mathcal{N}_i$ (ordered by network inclusion). We then discard all networks that produce an R^2 value lower than 0.9 in step (iii) and also contain (under inclusion) any of the discarded networks. After this, we carry out step (iv) for the remaining minimal networks and record the one(s) whose $f(\mathbf{p}^*)$ value is the lowest. Subsequently, we discard all networks that we have tested so far, and repeat the same steps for networks that are minimal among those which we have not yet looked at. The pseudo code of the search algorithm can be found in Section 4 the Supplementary Material.

This method appears to work well in practice since redundant networks (i.e. networks that contain a nested network which fits the data equally well) can be quickly identified from their low R^2 values. Due to the fact that in the presence of a redundant parameter the minimisation problem is under-determined, for different values of the redundant parameter we obtain different attractors. Hence, different initial guesses converge to different attractors, causing low R^2 values when trying to fit a single attractor to the set of the points obtained by 40 Newton iterations to the randomly scattered initial guesses. For non-redundant networks we use our objective function (f) to select the ones that give the best fit(s).

We applied our method to the 4-dimensional S-system above, assuming that we did not know a priori which h_{ik}, g_{ik} parameters were 0 (thus yielding 40 parameters), with 4 different initial conditions, each with 20 sampling points with 0%, 2%, 5%, 10%, 20% relative Gaussian noise. We replicated each data set eight times.

As an illustration of our results, we consider the third equation ($\dot{x}_3(t) = \alpha_3 \prod_{j=1}^{n} x_j(t)^{g_{3j}} - \beta_3 \prod_{j=1}^{n} x_j(t)^{h_{3j}}$). For 5% relative noise, 25 out of the 56 possible networks with 5 unknowns (i.e. out of the $2n+2$ possible variables $2n-3$ are set to zero and 5 variables are non-zero) proved to be non-redundant. However, amongst the maximal networks only one out of the 70 possible network structures proved to be non-redundant. Thus, instead of the 162 possible networks only 62 remained to be compared in step (iv). Intriguingly, out of these networks the true network (that is, the network in Fig. 1) came only second compared to the network corresponding to the equation

$$\dot{x}_3 = \alpha_3 x_1^{g_{31}} x_2^{g_{32}} - \beta_3 x_4^{h_{34}}$$

in seven out of eight trials, while being the best candidate only once. This network differs from the true network in only one edge: metabolite x_3 is governed by x_1 instead of auto-regulating itself.

In Table 3 we present a summary for each noise level and subproblem of the rank of the true underlying model structure amongst the 162 possible networks. Although the differences in the residuals were statistically significant (p-value <0.05), even for the 15th ranked network ((1), 10% noise) the residuals of the true network were less than twice as large as the best performing model. Still, a sudden drop can be seen in the network identification success between 5% and 10% relative noise: the complete 4D-system topology ranked as 4th (=1 * 2 * 2 * 1) at 5% noise level, while 135th (=3 * 3 * 15 * 1) at 10% noise level. This is probably due to the high cross-correlation between the parameters. In case of noise-free data (0% noise) both the network and the parameter reconstruction were perfect.

Note that the total run time for estimating the 4-dimensional example with unknown topology was 2.5 h on the same Pentium 4 PC machine mentioned above. This compares well with [7] where, for the same 4-dimensional example, but without noise, it took 15 min (machine not specified), and [17] where a runtime of 2.5 h (1200 MHz Intel Pentium M processor using 384 MB of RAM) was reported, without noise, using interval analysis.

As pointed out above, due to decoupling we treated the n ODEs separately. Thus the run time of our algorithm increases only linearly as a function of reactants n. The other source of complexity is the sparsity of the network. Namely, the maximum number of chemicals influencing a certain chemical in the system k, corresponding to the number of edges entering a node. The advantage of our proposed method for network searching is that we start checking from the simplest topologies (those that do not contain any other network) to the more complex ones (see also [21] where a similar topology searching strategy is proposed). Since our method spots redundant topologies we only need to check up to the true complexity (plus one), and not all existing topologies. The maximum number of Newton iterations in steps (ii)–(iii) was selected to be 10 times the number of unknown parameters in the given equation to find the dependencies between parameters. In

Table 3: Topology ranking

Equations	0%	2%	5%	10%
1	1st	1st	1st	3rd
2	1st	1st	2nd	3rd
3	1st	1st	2nd	15th
4	1st	1st	1st	1st

our experience this iteration number selection was sufficient to find good estimates for attractor curves. A similar network construction algorithm was proposed by [22].

Based on these algorithm settings, using the Stirling formula [23], the computational complexity of our algorithm is approximately $\mathcal{O}(\sum_{j=1}^{k}(\frac{n}{j})^{j})$. As expected, it is polynomial in terms of the number of reactants (n), and exponential in the maximum degree of the network (k). In practice, however, for biochemical systems the underlying network structure tends to be sparse and therefore we expect $k \leq n$.

4 Conclusions

We have presented a new method for estimating S-system parameters, that can be applied to noisy data. In particular, we have shown for 4- and 30-dimensional examples that we can reliably reconstruct the parameters for an underlying model. Moreover, for the 4-dimensional example we have shown that for moderate noise (5%) we can more often than not identify the underlying network associated with the model.

The run time of our method compares favorably with other contemporary methods in the literature. For a fixed network the running time is directly proportional to the dimension of the underlying system, which should prove useful compared with other approaches. This is a direct consequence of the fact that the attractor of the Newton flow appears to be one-dimensional. Also, having successfully recovered parameters for a 30-dimensional example having high network complexity (the largest that we have found in the literature on S-system parameter estimation), we have demonstrated that our method can probably perform well for higher dimensional (not too dense) problems.

The success of our method, even for relatively high-noise levels, can be attributed to the fact that the conjectured attractor appears to only slightly change even for high noise levels (Fig. 4). This is supported by theoretical evidence presented by [24]. Another advantage of our method is that, in case of S-system models, the attractor curve is straight-forward to fit. If our conjecture were true it would guarantee that our method would always obtain global optimum. Some of the initial (40) guesses in step (iii) may converge to suboptimal solutions, but the rest of them should be enough to reconstruct the one-dimensional attractor – which contains the true solution of the S-system-based on its known form for S-systems.

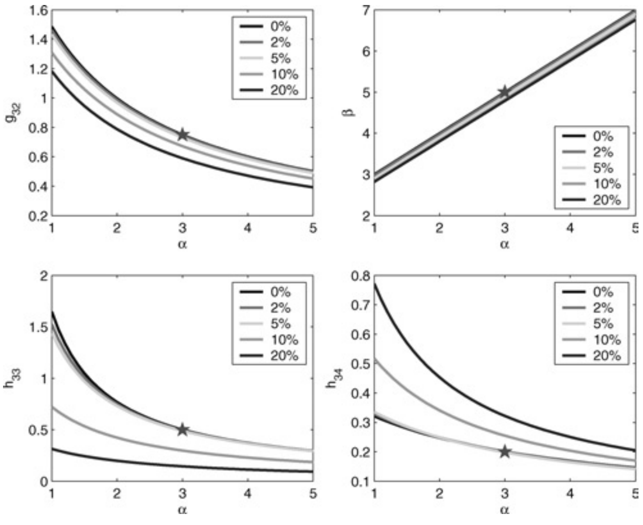


Fig. 4 *Effect of noise on the attractor*

2D projections of the attractors obtained from the 4-dimensional example data with various noise levels

Star marks the respective projection of the true parameter vector

Although we have not been able to prove the the existence of a one-dimensional attractor for the Newton flow, in the Supplementary Material we have provided some evidence that this is probably the case. This is supported by the existence of an invariant, exponentially attracting manifold for the Newton flow in general [25]. Moreover, if the attractor does not exhibit the assumed properties our method will terminate without producing a (false) result due to the R^2 check that we make in the algorithm.

The goodness-of-fit R^2 does not depend on the level of noise added to the perfect data. However, it deteriorates with the increasing complexity of the S-system, although this can be improved by increasing the number of Newton iterations.

We illustrate through our 4-dimensional example (at 5% noise level) how much each step of our algorithm ((ii)–(iv)) increases the accuracy of the parameter estimation: (1) In step (ii) we ran the Newton search with random initial guesses; (2) In step (iii) we started the search with initial guesses lying on the attractor; and (3) In step (iv) we picked the best limit point iterated in the previous step. We computed the expected value of the average parameter estimation error for the estimates produced in each step. We ran the algorithm for 10 different data sets with varying noise patterns. The average error in (1) was 3.21; in (2) 2.40; and in (3) 0.55. This result indicates that our method can achieve a more accurate parameter identification than standard Newton-methods.

When evaluating the results different sources of error need to be considered in the estimation: (i) The inherent error in any estimate which is caused by the noise in the data set, which cannot be completely avoided and only depends on the actual data set and the objective function in use; (ii) The error introduced by the numerical method we use to solve the optimization of the objective function. The first source of error can be somewhat avoided by making a better choice of objective function (f), changing initial conditions ($\mathbf{x}(0)$), or sampling time selection (t_i) [3]. Note, however that these issues are out of the scope of our paper. Considering the noise of the optimal solution as a function of these factors, a formula that gives a linear relationship between the standard deviation of the theoretically best parameter estimate and the relative noise in the data sets (σ) can be derived (see Supplementary Material). The parameter estimates in our examples reflect this linear trend.

We applied this formula to compute the theoretically expected relative error for parameter estimation. An example of this for 5% relative noise is shown in Table 4. The mean of the expected relative errors of the theoretically best estimate for 2%, 5%, 10%, 20% relative noise were 3.9%, 9.76%, 19.52%, 39.03%, respectively. Note the close agreement with the mean expected error of the estimates obtained by our method: 3.9%, 10.04%, 18.53%, 37.49%. In particular, (2) indicates that our algorithm will provide an estimate whose expected error is very similar to that of the theoretically best estimate. It follows that our method is quite likely to find a global optimum. Note that our tests showed that the performance of our algorithm drops if both kinetic orders g_{ik} and h_{ik} are non-zero for fixed i and k, that is, chemical constituent k can simultaneously inhibit and induce chemical constituent i. This is probably due to the fact that the attractor is much less stable in this case.

In future, it will be worth investigating how the properties of the attractor change for other models, such as GMAs [11]. Also, as has been mentioned above, the network

Table 4: Theoretically expected error

5 % Noise									
0.08	0	0	0.07	0	0.10	0.10	0	0	0
0.07	0.06	0	0	0	0.18	0	0.12	0	0
0.10	0	0.08	0	0	0.06	0	0	0.11	0.08
0.16	0.14	0	0	0	0.04	0	0	0	0.11

searching method that we have described is quite slow and therefore is not applicable to larger examples. Hence it will be important to develop faster tools for significantly reducing the class of plausible networks.

5 Supporting information

Additional information and further details of the results can be found in the Supplementary Material available at http://www.ietdl.org/IET-SYB.

6 Acknowledgment

This work was undertaken by Kutalik as part of his Ph.D. studies at the School of Computing Sciences, University of East Anglia, whom Kutalik would like to thank for funding his studies. In addition, Tucker would like to thank the Royal Swedish Academy of Sciences, where he is currently a Research Fellow.

7 References

1 Almeida, J.S., and Voit, E.O.: 'Natural-network-based parameter estimation in S-system models of biological networks', *Genome Inform.*, 2003, **14**, (1), pp. 114–123

2 Cho, K.-H., and Wolkenhauer, O.: 'Analysis and modeling of signal transduction pathways in systems biology', *Biochem. Soc. Trans.*, 2003, **31**, (6), pp. 1503–1509

3 Kutalik, Z., Cho, K.-H., and Wolkenhauer, O.: 'Optimal sampling time selection for parameter estimation in dynamic pathway modeling', *Biosystems*, 2004, **75**, (1-3), pp. 43–55

4 Wolkenhauer, O., Ullah, M., Wellstead, P., and Cho, K.-H.: 'The dynamic systems approach to control and regulation of intracellular networks', *FEBS Letters*, 2005, **579**, (8), pp. 1846–1853

5 Voit, E.O.: 'Computational analysis of biochemical systems' (Cambridge University Press, 2000)

6 Seatzu, C.: 'A fitting based method for parameter estimation in S-systems', *Dynam. Sys. Appl.*, 2000, **9**, (1), pp. 77–98

7 Voit, E.O., and Almeida, J.: 'Decoupling dynamical systems for pathway identification from metabolic profiles', *Bioinformatics*, 2004, **20**, (11), pp. 1670–1681

8 Kikuchi, S., Tominaga, D., Arita, M., Takahashi, K., and Tomita, M.: 'Dynamic modeling of genetic networks using genetic algorithm and S-system', *Bioinformatics*, 2003, **19**, (5), pp. 643–650

9 Kimura, S., Ide, K., Kashihara, A., Kano, M., Hatakeyama, M., Masui, R., Nakagawa, N., Yokoyama, S., Kuramitsu, S., and Konagaya, A.: 'Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm', *Bioinformatics*, 2005, **21**, (7), pp. 1154–1163

10 Ko, C.L., Wang, F.S., Chao, Y.P., and Chen, T.W.: 'S-system approach to modeling recombinant Escherichia coli growth by hybrid differential evolution with data collocation', *Biochem. Eng. J.*, 2006, **28**, (1), pp. 10–16

11 Polisetty, P.K., Voit, E.O., and Gatzke, E.P.: 'Identification of metabolic system parameters using global optimization methods', *Theor. Biol. Med. Model*, 2006, **3**, p. 4

12 Chen, B.S., and Wang, Y.C.: 'On the attenuation and amplification of molecular noice in genetic regulatory networks', *BMC Bioinform.*, 2006, **7**, p. 52

13 Schilling, M., Maiwald, T., Bohl, S., Kollmann, M., Kreutz, C., Timmer, J., and Klingmüller, U.: 'Quantitative data generation for systems biology–the impact of randomisation, calibrators and normalisers', *IEE Systems Biol.*, 2005, **152**, (4), pp. 193–200

14 Moles, C.G., Mendes, P., and Banga, J.R.: 'Parameter estimation in biochemical pathways: a comparison of global optimization methods', *Genome. Res.*, 2003, **13**, (11), pp. 2467–2474

15 Gonzalez, O.R., Küper, C., Jung, K., Naval, P.C., and Mendoza, E.: 'Parameter estimation using simulated annealing for S-system models of biochemical networks', *Bioinformatics*, 2007, **23**, (4), pp. 480–486

16 Tsai, K.Y., and Wang, F.S.: 'Evolutionary optimization with data collocation for reverse engineering of biological networks', *Bioinformatics*, 2005, **21**, (7), pp. 1180–1188

17 Tucker, W., and Moulton, V.: 'Parameter reconstruction for biochemical networks using interval analysis', *Reliable Computing*, 2006, **12**, (5), pp. 1–14

18 Coleman, T.F., and Li, Y.: 'An interior, trust region approach for nonlinear minimization subject to bounds', *SIAM J. Optimiz.*, 1996, **6**, (2), pp. 418–445

19 Dedieu, J.-B., and Shub, M.: 'Newton flow and interior point methods in linear progrfamming', *Internat. J. Bifur. Chaos Appl. Sci. Engrg.*, 2005, **15**, (3), pp. 827–840

20 Maki, Y., Tominaga, D., Okamoto, M., Watanabe, S., and Eguchi, Y.: 'Development of a system for the inference of large scale genetic networks'. Proc. Pac. Symp., 2001, vol. 6, pp. 446–458

21 Tucker, W., Kutalik, Z., and Moulton, V.: 'Estimating parameters for generalized mass action models using constraint propagation', *Math. Biosci.*, 2007, In press

22 Marino, S., and Voit, E.O.: 'An automated procedure for the extraction of metabolic network information from time series data', *J. Bioinf. Comp. Biol.*, 2006, **4**, (3), pp. 665–691

23 Nikol'skii, S.M.: 'Encyclopaedia of mathematics' (Springer-Verlag, 2002)

24 Hauser, R., and Nedic, J.: 'The continuous Newton-Raphson method can look ahead', *SIAM J. Optimiz.*, 2005, **15**, (3), pp. 915–925

25 Janovský, V., and Seige, V.: 'Qualitative analysis of Newton's flow', *SIAM J. Num. Anal.*, 1996, **33**, (5), pp. 2068–2097