Theoretical Biology and Medical Modelling

BioMed Central

Research

Parameter estimation in biochemical systems models with alternating regression

I-Chun Chou[1], Harald Martens[2] and Eberhard O Voit*[1]

Address: [1]The Wallace H. Coulter Department of Biomedical Engineering at Georgia Institute of Technology and Emory University, 313 Ferst Drive, Atlanta, GA, 30332, USA and [2]CIGENE/Norwegian U. of Life Sciences, P.O.Box 5003, N – 1432 Ås, Norway

Email: I-Chun Chou - gtg392p@mail.gatech.edu; Harald Martens - harald.martens@matforsk.no; Eberhard O Voit* - eberhard.voit@bme.gatech.edu

* Corresponding author

Published: 19 July 2006

Theoretical Biology and Medical Modelling 2006, **3**:25 doi:10.1186/1742-4682-3-25

Received: 27 April 2006
Accepted: 19 July 2006

This article is available from: http://www.tbiomed.com/content/3/1/25

Abstract

Background: The estimation of parameter values continues to be the bottleneck of the computational analysis of biological systems. It is therefore necessary to develop improved methods that are effective, fast, and scalable.

Results: We show here that *alternating regression* (AR), applied to S-system models and combined with methods for decoupling systems of differential equations, provides a fast new tool for identifying parameter values from time series data. The key feature of AR is that it dissects the nonlinear inverse problem of estimating parameter values into iterative steps of linear regression. We show with several artificial examples that the method works well in many cases. In cases of no convergence, it is feasible to dedicate some computational effort to identifying suitable start values and search settings, because the method is fast in comparison to conventional methods that the search for suitable initial values is easily recouped. Because parameter estimation and the identification of system structure are closely related in S-system modeling, the AR method is beneficial for the latter as well. Specifically, we show with an example from the literature that AR is three to five orders of magnitudes faster than direct structure identifications in systems of nonlinear differential equations.

Conclusion: Alternating regression provides a strategy for the estimation of parameter values and the identification of structure and regulation in S-systems that is genuinely different from all existing methods. Alternating regression is usually very fast, but its convergence patterns are complex and will require further investigation. In cases where convergence is an issue, the enormous speed of the method renders it feasible to select several initial guesses and search settings as an effective countermeasure.

Background

Novel high-throughput techniques of molecular biology are capable of producing *in vivo* time series data that are relatively high in quantity and quality. These data implicitly contain enormous information about the biological system they describe, such as their functional connectivity and regulation. The hidden information is to be extracted with methods of parameter estimation, if the structure of the system is known, or with methods of structure identification, if the topology and regulation of the system are not known. The S-system format within Biochemical Systems Theory (BST; [1-4]) is recognized as a particularly effective modeling framework for both tasks, since it has a mathematically convenient structure and because every parameter has a uniquely defined meaning and role in the biological system. Due to the latter feature, the typically complex identification of the pathway structure reduces to a parameter estimation task, though in a much higher-dimensional space. Still, like most other biological models, S-system models are nonlinear, so that parameter estimation is a significant challenge. Here, we propose a method called *alternating regression* (AR), which we combine with a previously described decoupling technique [5]. AR is fast and rather stable, and performs structure identification tasks between 1,000 and 50,000 times faster than methods that directly estimate systems of nonlinear differential equations (*cf.* [6]).

Methods

Modeling framework

In the S-system formulation within BST, X_i denotes the concentration of metabolite *i*, and its change over time, $\dot{X}_i$, is represented as the difference between one production and one degradation term, both of which are formulated as products of power-law functions.*

(* Footnote: Throughout the paper, metabolite concentrations are represented as upper-case italics (*X*). An upper-case boldface variable (**L**) represents a matrix of regressor columns and a lower-case boldface variable (**y**) represents a regressand column in a linear multivariate statistical regression model.)

The generic form of an S-system is thus

$$\dot{X}_i = \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}, i = 1,2,...,n. \qquad (1)$$

The rate constants α_i and β_i are non-negative and the kinetic orders g_{ij} and h_{ij} are real numbers with typical values between -1 and +2. The S-system format allows the inclusion of independent variables, but because these are typically known in estimation tasks and constant, they can be merged with the rate constants [4]. S-systems have been discussed many times [3,4,7,8] and need no further explanations here.

Decoupling of differential equations

Suppose the S-system consists of *n* metabolites X_1, X_2, ..., X_i, ..., X_n, and for each metabolite, a time series consisting of *N* time points t_1, t_2, ..., t_k, ..., t_N has been observed. If we can measure or deduce the slope $S_i(t_k)$ for each metabolite at each time point, we can reformulate the system as *n* sets

$$S_i(t_1) \approx \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}}(t_1) - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_1),$$

$$S_i(t_2) \approx \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}}(t_2) - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_2),$$

$$\vdots$$

$$S_i(t_k) \approx \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}}(t_k) - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_k),$$

$$\vdots$$

$$S_i(t_N) \approx \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}}(t_N) - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_N).$$

Thus, for the purpose of parameter estimation, the original system of *n* coupled differential equations can be analyzed in the form of *n* × *N* uncoupled algebraic equations [4,9].

The uncoupling step renders the estimation of slopes a crucial step. If the data are more or less noise-free, simple linear interpolation, splines [10-12], B-splines [13], or the so-called three-point method [14] are effective. If the data are noisy, it is useful to smooth them, because the noise tends to be magnified in the slopes. Established smoothing methods again include splines, as well as different types of filters, such as the Whittaker filter (see [15] for a review), collocation methods [16], and artificial neural networks [17,18]. In order to keep our illustration of the AR method as clean as possible, we assume that true slopes are available and elaborate on issues of experimental noise in the *Discussion*.

Alternating regression

The decoupling of the system of differential equations allows us to estimate the S-system parameters α_i, g_{ij}, β_i, and h_{ij} $(i, j = 1,2,...,n)$ one equation at a time, using slopes and concentration values of each metabolite at time points t_k. The proposed method called *alternating regression* (AR) has been used in other contexts such as spectrum reconstruction and robust redundancy analysis [19,20], but, to the best of our knowledge, not for the purpose of parameter estimation from time series. The overall flow of the method is shown in Figure 1. Adapted to our task of S-system estimation, AR works by cycling between two phases of multiple linear regression. The first phase begins with guesses of all parameter values of the degradation term in a given equation and uses these to solve for the

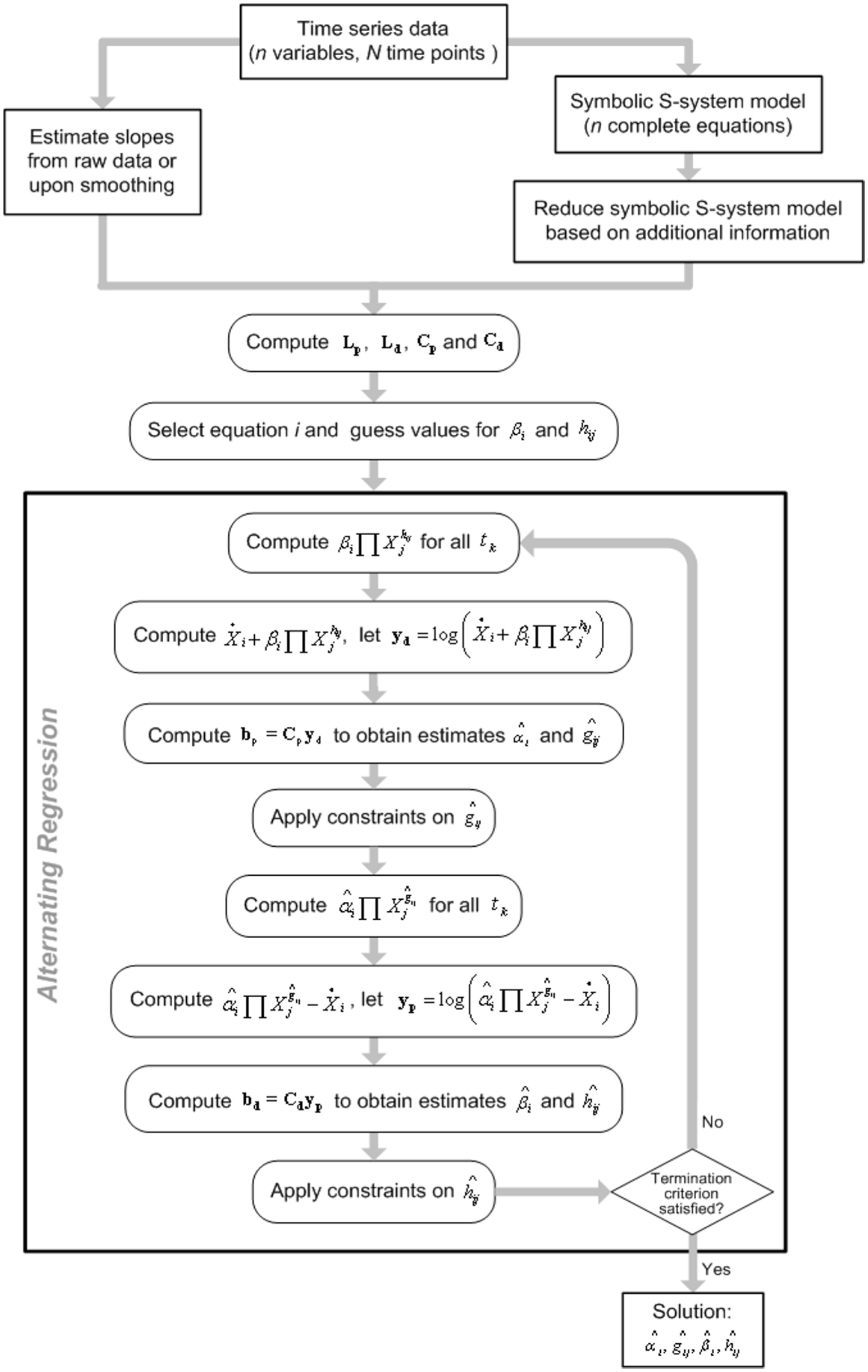
Time series data
(*n* variables, *N* time points)
Estimate slopes from raw data or upon smoothing
Symbolic S-system model (*n* complete equations)
Reduce symbolic S-system model based on additional information
Compute $\mathbf{L_p}$, $\mathbf{L_d}$, $\mathbf{C_p}$ and $\mathbf{C_d}$
Select equation *i* and guess values for β_i and h_{ij}
Alternating Regression
Compute $\beta_i \prod X_j^{h_{ij}}$ for all t_k
Compute $\dot{X}_i + \beta_i \prod X_j^{h_{ij}}$, let $\mathbf{y_d} = \log\left(\dot{X}_i + \beta_i \prod X_j^{h_{ij}}\right)$
Compute $\mathbf{b}_p = \mathbf{C}_p \mathbf{y}_d$ to obtain estimates $\hat{\alpha}_i$ and $\hat{g}_{ij}$
Apply constraints on $\hat{g}_{ij}$
Compute $\hat{\alpha}_i \prod X_j^{\hat{g}_{ij}}$ for all t_k
Compute $\hat{\alpha}_i \prod X_j^{\hat{g}_{ij}} - \dot{X}_i$, let $\mathbf{y_p} = \log\left(\hat{\alpha}_i \prod X_j^{\hat{g}_{ij}} - \dot{X}_i\right)$
Compute $\mathbf{b_d} = \mathbf{C_d y_p}$ to obtain estimates $\hat{\beta}_i$ and $\hat{h}_{ij}$
Apply constraints on $\hat{h}_{ij}$
Termination criterion satisfied?
No
Yes
Solution:
$\hat{\alpha}_i, \hat{g}_{ij}, \hat{\beta}_i, \hat{h}_{ij}$

Figure 1
Logistic flow of parameter estimation by alternating regression.

parameters of the corresponding production term. The second phase takes these estimates to improve the prior parameter guesses or estimates in the degradation term. The phases are iterated until a solution is found or AR is terminated for other reasons.

In pure parameter estimation tasks, the structure of the underlying network is known, so that it is also known which of the S-system parameters are zero and which of the kinetic orders are positive or negative. Thus, the search space is minimal for the problem. Nonetheless, the same method of parameter estimation can in principle also be used for structure identification. In this case, the estimation is executed with an S-system where no parameter is *a priori* set to zero and all parameters have to be estimated. As an intermediate task, it is possible that only some of the structure is known. This information can again be used to reduce the search space. If it is known, for instance, that variable X_j does not affect the production or degradation of X_i, the corresponding parameter value g_{ij} or h_{ij} is set equal to zero, or X_j is taken out of the regression. One can thus reduce the regression task either by constraining the values of some *g*'s or *h*'s throughout the AR or by selecting a subset of regressors at the beginning, *i.e.*, by taking some variables out of the regression. Similarly, if a kinetic order is known to represent an inhibiting (activating) effect, its range of possible values can be restricted to negative (positive) numbers. This constraining of kinetic orders, while not essential, typically improves the speed of the search. It is imaginable that a kinetic order is constrained too tightly. In this case, the solution is likely to show the kinetic order at the boundary, which is subsequently relaxed.

To estimate the parameters of the i^{th} differential equation, the steps of the AR algorithm are as follows:

{1} Let $\mathbf{L_p}$ denote an *(n+1)* × *N* matrix of logarithms of regressors X_i, defined as

$$\mathbf{L_p} = \begin{bmatrix} 1 & \log(X_1(t_1)) & \cdots & \log(X_i(t_1)) & \cdots & \log(X_n(t_1)) \\ 1 & \log(X_1(t_2)) & \cdots & \log(X_i(t_2)) & \cdots & \log(X_n(t_2)) \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ 1 & \log(X_1(t_k)) & \cdots & \log(X_i(t_k)) & \cdots & \log(X_n(t_k)) \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ 1 & \log(X_1(t_N)) & \cdots & \log(X_i(t_N)) & \cdots & \log(X_n(t_N)) \end{bmatrix} \qquad (3)$$

$\mathbf{L_p}$ is used in the first phase of AR to determine the parameter values of the production term. Additional information on the system, if it is available, reduces the width of $\mathbf{L_p}$. For instance, if $X_2$and X_4do not affect the production of X_1in a four variable system, Eq. (3) reduces to

$$\mathbf{L_p} = \begin{bmatrix} 1 & \log(X_1(t_1)) & \log(X_3(t_1)) \\ 1 & \log(X_1(t_2)) & \log(X_3(t_2)) \\ \vdots & \vdots & \vdots \\ 1 & \log(X_1(t_k)) & \log(X_3(t_k)) \\ \vdots & \vdots & \vdots \\ 1 & \log(X_1(t_N)) & \log(X_3(t_N)) \end{bmatrix} \qquad (4)$$

Analogous to $\mathbf{L_p}$, let $\mathbf{L_d}$ denote the *(n+1)* × *N* matrix of regressors used in the second phase of AR to determine the parameter values of the degradation term. $\mathbf{L_p}$ and $\mathbf{L_d}$ are the same when the variables used in two phases of AR are identical.

{2} Compute the matrices

$$\mathbf{C_p} = (\mathbf{L_p}^{\mathbf{T}}\mathbf{L_p})^{-1}\mathbf{L_p}^{\mathbf{T}} \qquad (5)$$

$$\mathbf{C_d} = (\mathbf{L_d}^{\mathbf{T}}\mathbf{L_d})^{-1}\mathbf{L_d}^{\mathbf{T}} \qquad (6)$$

which are invariant throughout the iterative process.

{3} Select values for β_i and h_{ij} in accordance with experience about S-system parameters (*cf.* [4]: Ch. 5) and make use of any available information constraining some or all h_{ij}.

{4} For all t_k, k = 1, 2, ..., *N*, compute $\beta_i \prod_{j=1}^{n} X_j^{h_{ij}}$, using values $X_j(t_k)$ from the observed or smoothed time series measurements.

{5} Compute the *N*-dimensional vector $\mathbf{y_d} = \log\left(S_i(t_k) + \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_k) \right)$ (k = 1, 2, ..., *N*) containing transformed "observations" on the degradation term. *Note*: It is possible to compute $\mathbf{y_d}$ for all *n* traces simultaneously so that $\mathbf{Y_d}$ becomes an *n* × *N* matrix with columns $\mathbf{y_d}$.

{6} Based on the multiple linear regression model

$$\mathbf{y_d} = \mathbf{L_p}\mathbf{b_p} + \varepsilon_{\mathrm{p}} \qquad (7)$$

estimate the regression coefficient vector $\mathbf{b_p}$= [$\hat{\alpha}_i$, $\hat{g}_{ij}$, j = 1, 2,..., *n*]' by regression over the *N* time points. In other words, this step leads to an estimation of parameters in sets of equations of the type $y_{d,k} = \alpha_i + \sum_{j=1}^{n} g_{ij} \log\left(X_j\left(t_k \right) \right) + \varepsilon_{i,l}$. Specifically, compute $\mathbf{b_p}$ as

$$\mathbf{b_p} = (\mathbf{L_p}^T\mathbf{L_p})^{-1}\mathbf{L_p}^T\mathbf{y_d} = \mathbf{C_p}\mathbf{y_d} \quad (8)$$

according to Eqs. (3–5).

{7} Constrain some or all $\hat{g}_{ij}$, if outside information on the model suggests it.

{8} Using the observed values of $X_j(t_k)$, compute $\hat{\alpha}_i \prod_{j=1}^{n} X_j^{\hat{g}_{ij}}$ for all t_k, $k = 1, 2, ..., N$.

{9} Compute the N-dimensional vector $\mathbf{y_p} = \log\left(\alpha_i \prod_{j=1}^{n} X_j^{g_{ij}}\left(t_k \right) - S_i\left(t_k \right) \right)$ containing the transformed "observations" associated with the production term.

{10} Based on the multiple linear regression model

$$\mathbf{y_p} = \mathbf{L_d}\mathbf{b_d} + \varepsilon_\mathbf{d} \quad (9)$$

and in analogy to step {6}, estimate the regression coefficient vector $\mathbf{b_d} = [\ \hat{\beta}_i,\ \hat{h}_{ij}\ , j = 1, 2,..., n]'$ by regression over the N time points as

$$\mathbf{b_d} = \mathbf{C_d}\mathbf{y_p} \quad (10)$$

{11} Constrain some or all $\hat{h}_{ij}$, if outside information on the model suggests it.

{12} Iterate Steps {4} - {11} until a solution is found or some termination criterion is satisfied.

At each phase of AR, lack-of-fit criteria are estimated and used for monitoring the iterative process and to define termination conditions. In this paper we use the sum of squared y-errors (SSE_d and SSE_p) as optimization criteria for the two regression phases, *i.e.* we compute

$$log\left(SSE \right) = log\left(\sum_{k=1}^{N} \left(\mathbf{y_k} - \hat{\mathbf{y}}_\mathbf{k} \right)^2 \right), \qquad (11)$$

where $\hat{\mathbf{y}} = \mathbf{L} \times \mathbf{b}$, $\mathbf{L}$ equals $\mathbf{L_p}$ or $\mathbf{L_d}$, and $\mathbf{b}$ is the solution vector $\mathbf{b_p}$ or $\mathbf{b_d}$, estimated through regression and modified by constraints reflecting structural information. We use the logarithm of *SSE* because it is superior in illustrating small changes in the residual error.

It is known that collinearity may affect the efficiency of multivariate linear regressions. We therefore also implemented methods of principal component regression (PCR), partial least squares regression (PLSR) and ridge regression [21]. For the cases analyzed here, these methods did not provide additional benefit.

Results and discussion

For illustration purposes, we use a didactic system with four variables that is representative of a small biochemical network [5]. A numerical implementation with typical parameters is

$$\begin{array}{ll} \dot{X}_1 = 12X_3^{-0.8} - 10X_1^{0.5} & X_1\left(t_0 \right) = 1.4 \\ \dot{X}_2 = 8X_1^{0.5} - 3X_2^{0.75} & X_2\left(t_0 \right) = 2.7 \\ \dot{X}_3 = 3X_2^{0.75} - 5X_3^{0.5}X_4^{0.2} & X_3\left(t_0 \right) = 1.2 \\ \dot{X}_4 = 2X_1^{0.5} - 6X_4^{0.8} & X_4\left(t_0 \right) = 0.4 \end{array} \qquad (12)$$

The system is first used to create artificial datasets that differ in their initial conditions (Table S1 of *Additional file*1). In a biological setting, these may mimic different stimulus-response experiments on the same system. For example, they could represent different nutrient conditions in a growth experiment. Figure 2 shows the branched pathway, along with a selection of time course data (dataset 1) and slopes.

In order not to confuse the features of AR with possible effects of experimental noise, we use true metabolite concentrations and slopes; we compute the latter directly from Eq. (12) at each time point. We initially assume that we have observations at 50 time points, but discuss cases with fewer points and with noise later.

Performance of AR

Given the time series data of X_i and S_i at every time point t_k, the AR algorithm is performed for each metabolite, one at a time. Figure S1 summarizes various patterns of convergence observed. Generally we can classify the convergence patterns into four types: 1) convergence to the true value; 2) convergence to an incorrect value; 3) no convergence; typically the value of α_i (or β_i) continuously

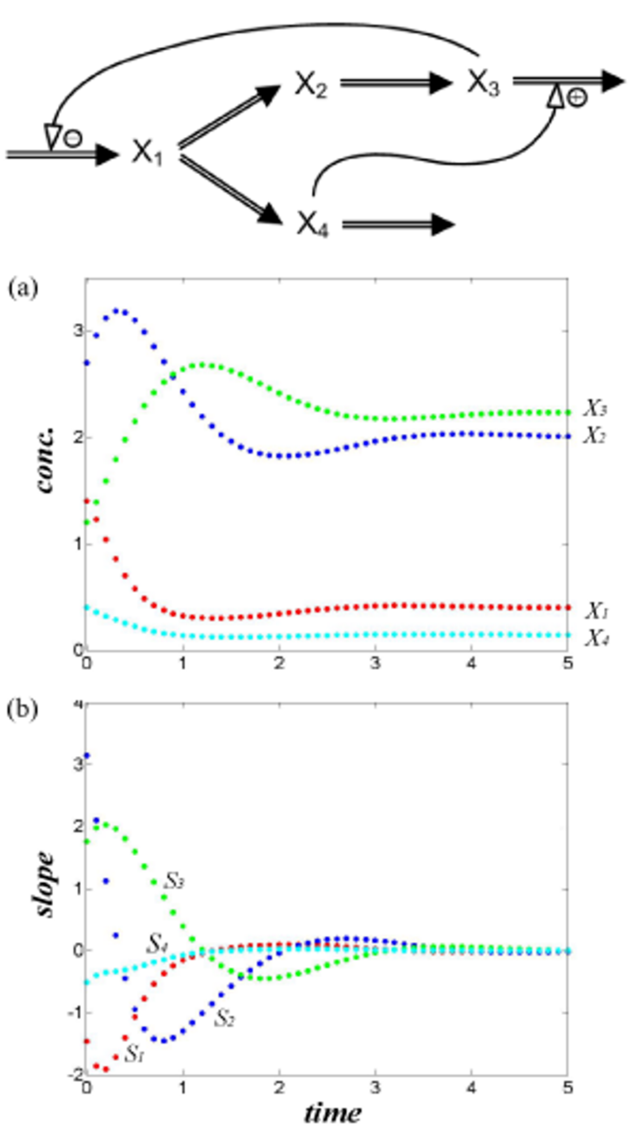

Figure 2

Test system with four dependent variables. (a) time courses computed with initial values in Eq. (12) (use dataset 1 in Table S1); (b) corresponding dynamics of slopes. Typical units might be concentrations (*e.g.*, in mM) plotted against time (*e.g.*, in minutes), but the example could as well run on an hourly scale and with variables of a different nature.

increases while all g_{ij} (or h_{ij}) gradually approach zero, while in some other cases g_{ij} and the corresponding h_{ij} increase (or decrease) in a parallel manner; 4) termination during AR, due to some of the observations $\mathbf{y}_\mathrm{d}$ (or $\mathbf{y}_\mathrm{p}$) taking on complex values.

As is to be expected, the speed of convergence depends on the initial guesses, the variables used as regressors, the constraints, and the data set. After a few initial iterations, the approach of the true value is usually, though not always, strictly monotonic. In some cases, the error initially decreases rapidly and subsequently enters a phase of slower decrease. It is also possible that convergence is non-monotonic, that the algorithm converges to a different point in the search space, or that it does not converge at all. Convergence to the wrong solution and situations of no convergence are particularly interesting. In the case of no convergence, the solution arrives at unreasonable parameter values that grow without bound; this case is very easy to detect and discard. By contrast, the search may lead to a solution with wrong parameter values, but a satisfactory residual error. Thus, the algorithm produces a wrong, but objectively good solution. It is close to impossible with *any* algorithm to guard against this problem, unless one can exclude wrong solutions based on the resulting parameter values themselves. This is actually greatly facilitated with S-systems because all parameters have a clearly defined meaning in terms of both their sign and magnitude, which may help spot unrealistic solutions with small residual error.

Reasons for AR not to converge are sometimes easily explained, but sometimes obscure. For instance, the slope-minus-degradation or -production expressions in steps {5} and {9} of the algorithm may become negative, thereby disallowing the necessary logarithmic transformation. As a consequence, the regression terminates. If this happens, it usually happens during the first or the second iteration, and the problem is easily solved when the initial β or α is increased. In other cases, AR converges for one dataset, but not for another, even for the same model. This sometimes happens if datasets have low information content, for instance, if the dynamics of a variable is affected by a relatively large number of variables, but the observed time course is essentially flat or simple monotonic. In this case, convergence is obtained if one adjusts the constraints on some of the parameter values or selects a different set of regressors (see below). Of importance is that each iteration consists essentially of two linear regressions so that the process is fast. Thus, even the need to explore alternative settings is computationally cheap and provides for an effective solution to the convergence problem.

Patterns of convergence

The speed and pattern of convergence depend on a combination of several features, including initial guesses for all parameters and the datasets. Overall, these patterns are very complicated and elude crisp analytical evaluations. This is not surprising, because even well-established algorithms like the Newton method can have basins of attraction that are fractal in nature (*e.g.*, [22]). A detailed description of some of these issues, along with a number of intriguing color plates describing well over one million ARs, is presented in Additional file 1.

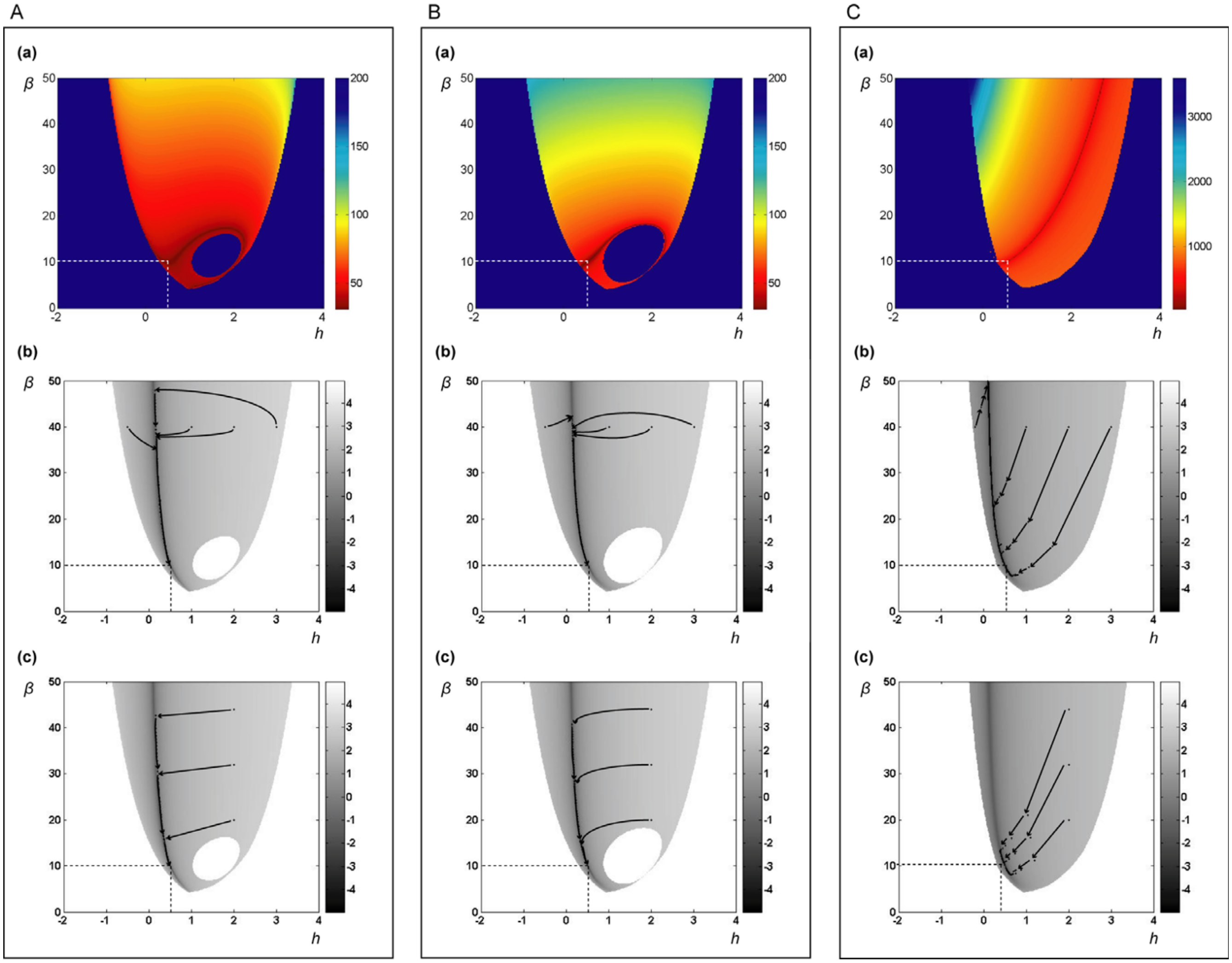

Figure 3

Summary of convergence patterns of AR. Panel A: all variables are initially used as regressors and constraints are imposed afterwards; Panel B: regression with the "union" of variables of both terms; Panel C: only those variables that are known to appear in the production or degradation term, respectively, are used as regressors. Row (a): speed of convergence; the color bars represent the numbers of iterations needed to converge to the optimum solution; Rows (b) and (c): 2D view of the error surface superimposed with convergence trajectories with different initial values of β and *h*; the color bars represent the value of *log(SSE)*. The intersections of dotted lines indicate the optimum values of parameters β and *h*.

Effect of initial parameter guesses

Figure 3 combines results from several sets of initial guesses of β_i and h_{ij} (the results of the second phase of AR are not shown, but are analogous). The data for this illustration consist of observations on the first variable of datasets 4, 5 and 6 (see Table S1 in the Additional file1). These are processed simultaneously as three sets of algebraic equations at 50 time points. Thus, the parameters α_1, g_{13}, β_1, and $h_{11}$of the equation

$$\dot{X}_1 = \alpha_1 X_3^{g_{13}} - \beta_1 X_1^{h_{11}} \qquad (13)$$

are to be estimated. As a first example, we initiate AR with all variables (X_1, ..., X_4) as regressors, but constrain the kinetic orders g_{11}, g_{12}, and $g_{14}$to be zero after the first phase of the regression, and the kinetic orders h_{12}, h_{13}, and h_{14}after the second phase, in accordance with the known network structure.

Figure 3A(a) shows the "heat map" of the convergence, where the x- and y-axes represent the initial guesses of $h_{11}$and β_1, respectively, and the color bar represents the number of iterations needed for convergence. Since we use noise-free data, the residual error should approaches 0, which corresponds to -∞ in logarithmic coordinates. We use -7 instead as one of the termination criteria, which corresponds to a result very close to the true value, but allows for issues of machine precision and numerical inaccuracies. Once this error level is reached, AR stops and the number of iterations is recorded as a measure for the speed of convergence. The unusual shape of a "martini with olive" is due to the following. The deep blue outside area indicates an inadmissible domain, where the initial parameter guess causes one or more of the terms $S_i\left(t_k\right) + \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}\left(t_k\right), k = 1,2,\ldots,N$ in step {5} to become negative, so that the logarithm, $\mathbf{y_d}$, becomes a complex number and the regression cannot continue. The line separating admissible and inadmissible domains is thus not smooth but shows the envelope of several pieces of power-law functions where the β-term is smaller than the (negative) slope at some time point. The "olive" inside the glass is also inadmissible. In this case, the chosen initial value causes the term $\hat{\alpha}_i \prod_{j=1}^{n} X_j^{\hat{g}_{ij}}\left(t_k\right) - S_i\left(t_k\right), k = 1,2,\ldots,N$ in step {9} to become negative, so that $\mathbf{y_p}$ becomes complex and AR terminates during the second phase. This type of termination usually, though not always, happens during the first iteration. In order to prevent it, one may *a priori* require that

$$S_1\left(t_k\right) + \beta_1 X_1^{h_{11}}\left(t_k\right) > 0 \qquad (14)$$

for every t_k, such that the logarithm is always defined. This is possible through the choice of a sufficiently large value for the initial guess of β. The magnitude of β should be reasonable, however, because excessive values tend to slow down convergence. As a matter of practically, one may start with a value of 5 or 10 and double it if condition (14) is violated.

Use of different variables as regressors

Panel A in Figure 3 shows results where we initially use all variables as regressors, but constrain their kinetic orders to zero after each iteration, if they are known to be zero. As alternatives, Panels B and C show results of using different variable combinations as regressors under otherwise identical conditions. In Panel B, both phases of AR use all variables as regressors that appear in either the production or the degradation term of the equation. In Panel C we make full use of our knowledge of the pathway structure and include in each term only the truly involved variables. Interestingly, this choice of regressors has a significant effect on convergence.

Compared with the case in of Figure 3A(a), the speed of convergence is slower in Figure 3B(a) and much slower in Figure 3C(a), even though this represents the "best-informed" scenario. The time needed to generate the graphs in Figures 3A(a), 3B(a), and 3C(a) for all shown 60,000 initial values is 72, 106, and 1,212 minutes, respectively. Thus, if we suppose that roughly half of the start points are inadmissible and require no iteration time, the average convergence time in Figure 3A(a) is 0.144 seconds, whereas it is 0.212 seconds in Figure 3B(a) and 2.424 seconds in Figure 3C(a). The pattern of convergence is affected by the datasets used. As another example, Figure S2 shows results of regressions with dataset 5.

Error surface

Rows (b) and (c) in Figure 3 Panels A, B, and C show heat maps of *log(SSE)*, where darker dots indicate smaller errors. The true minimal value of *log(SSE)* for our noise-free data is -∞, but for illustration propose, we plot it only to -5. Pseudo-3-D graphs of the error surface are shown in Figure S3 with views from two angles.

Convergence trajectories

Paths toward the correct solution may be visualized by plotting and superimposing the solution at every regression step onto the corresponding heat maps, with arrowheads indicating the direction of each trajectory (Figures 3A(b,c), 3B(b,c), and 3C(b,c)). For the first set of illustrations, four different initial values of $h_{11}$are chosen, while the value of $\beta_1$is always 40. For the second set of illustrations, four different initial values of $\beta_1$are chosen, while the value of $h_{11}$is always 2. Interestingly, independent of the start values, only two iterations are needed to reach a point very close to the valley of the error surface where the true solution is located. After the dramatic initial jump, all solutions follow essentially the same trajectory with small steps toward the true solution. We can also link the observations of Figure 3A(b) and 3A(c) to the result in 3A(a). For the same β_1, a start point in the right part the graph causes AR to jump to a more distant location on the trajectory, thus requiring more iterations to converge to the true solution.

It might be possible to speed up convergence in the flat part of the error surface, for instance by using history-based modeling based on conjugated gradients or partial least squares regression [21]. These options have not been analyzed.

Accuracy and speed of solution

The previous sections focused on the first equation of the S-system model in Eq. (12) and Figure 2. We used the AR

Table 1: Estimated parameter values of the S-system model of the pathway in Figure 2 using *log(SSE)* < -7 as termination criterion. [a] Regressor: A: all variables used as regressors and subsequently constrained; B: use of "union" variables as regressors (see *Text*); C: fully informed selection of regressors (see *Text*). [b] time (secs) needed to converge to the solution with *log(SSE)* < -7. [c] Convergence results according to AR algorithm: *: convergence to the true solution; **: convergence to different solution; *: no convergence. [d] time after running 1,000,000 iterations. See Eq. (12) for optimal parameter values and the *Additional file* for further comments.**

	Regressor[a]	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	*log(SSE)*	Time(sec)[b]	Note[c]
X_1	A	12.00	0.00	0.00	-0.80	-0.00	10.00	0.50	-0.00	0.00	0.00	-6.84	0.58	*
	B	12.03	-0.00	0	-0.80	0	10.04	0.50	0	0.00	0	-7.00	2.39	*
	C	12.00	0	0	-0.80	0	9.99	0.50	0	0	0	-6.95	0.17	*
X_2	A	44.50	-0.00	-0.02	-0.04	0.11	31.48	0.03	0.14	0.05	-0.13	0.51	1071.58[d]	**
	B	8.01	0.50	0.00	0	0	3.01	-0.00	0.75	0	0	-7.00	0.97	*
	C	8.01	0.50	0	0	0	3.01	0	0.75	0	0	-7.00	69.05	*
X_3	A	3.00	0.00	0.75	-0.00	-0.00	5.00	-0.00	0.00	0.50	0.20	-9.44	0.03	*
	B	7.29	0	0.37	-0.00	-0.00	8.76	0	-0.00	0.19	0.04	-4.04	1117.14[d]	**
	C	2.98	0	0.75	0	0	5.00	0	0	0.51	0.20	-7.01	0.50	*
X_4	A	96.80	0.01	0.01	-0.00	0.00	100.00	-0.00	-0.01	0.00	0.02	-3.83	4.59	***
	B	98.29	0.06	0	0	0.00	100.00	-0.00	0	0	0.01	-5.85	341.94	***
	C	2.016	0.50	0	0	0	5.99	0	0	0	0.80	-6.97	84.91	*

algorithm in the same manner to estimate all other parameters. Again, three sets of regressors were used for every variable. For simplicity of discussion, we describe the results from using dataset 1 of Table S1, always using as initial guesses $\beta_i = 15$ and $h_{ij} = 1$. The results are listed in Tables 1, S2 and S3.

For every variable, at least one of the three choices of regressors leads to convergence to the correct solution. Convergence is comparably fast, even if we require a very high accuracy for termination ($log(SSE) < -20$) (see Table S2). If we relax the accuracy to $log(SSE) < -7$ or $log(SSE) < -4$, the solution is still very good, but the solution time is noticeably decreased (Tables 1 and S3). However, the false-positive rate increases slightly for $log(SSE) < -4$. As a compromise, we use $log(SSE) < -7$ as termination criterion for the remainder of this paper.

Interestingly, the speed of convergence is fastest for the strategy "A" of using all variables as regressors; however, the failure rate in this case is also the highest. In contrast, the slowest speed of convergence is obtained for the correct regressors ("C"), where AR always converges to the right solution. The regressor set "B" is between "A" and "C" in terms of speed and ability to yield the correct optimum. For cases that don't converge to the right solution one easily adapts the AR algorithm by choosing different start values, slightly modifying constraints, or choosing different regressors in addition to the three types used above. The probability of finding the correct solution is increased if different datasets are available for sequential or simultaneous estimation. The same was observed for other estimation methods (*e.g.*, [5]).

Structure identification

The previous sections demonstrated parameter estimation for a system with known structure. Similar to this task is the identification of the unknown structure of a pathway from time series data, if one uses S-systems as the modeling framework [5]. The only difference is that very few or no parameters at all can *a priori* be set to zero or constrained to the positive or negative half of the search space. A totally uninformed AR search of this type often leads to no convergence. However, since each AR is fast, it is feasible to execute many different searches, in which some of the parameters are allowed to float, while others are set equal to zero.

Table S6 shows the results of exhausting all combinations of constraints to determine those that yield convergence. The total time for this exhaustive search is just over one hour. This is furthermore reduced if some *a priori* information is available. As an alternative to an exhaustive search, one may obtain constraining information from a prior linearization of the system dynamics [23]. This method does not identify parameter values per se, but provides very strong clues on which variables are likely to be involved in a given equation and which not. In the example tested, this method provided an over 90% correct classification of the relevant variables in each equation (see Table S7). Using this inference information, the total time was reduced to 53 minutes.

Finally, it is possible to sort parameter combinations by their empirical likelihood of inclusion in an equation [24]). For instance, a metabolite usually affects its own degradation but usually has no effect on its own production. Thus, a reasonable start is the parsimonious model $\dot{X}_i = \alpha_i - \beta_i X_i^{h_{ii}}$ with $g_{ii} = 0$ and $h_{ii} > 0$. In subsequent runs, free-floating variables (parameters) are added, one at a time. This strategy reduced the total time from one hour to under 3 minutes (see Table S8). As illustration, and for a second, independent example, we used the strategy of Veflingstad *et al.* [23] to determine the regulatory structure and parameter values of a gene regulatory network model [25] that has become a benchmark in the field. Kikuchi and collaborators [6] identified the structure of this model by using a genetic algorithm acting directly on the five differential equation of the model. Using a cluster of 1,040 CPUs, the solution required about 70 hours. We generated time series data from the model, using 0.5 as initial concentration for all five variables. The solution time needed for exhausting all constraint combinations for all variables and an error tolerance of *log(SSE)* = -7 was 81.2 min on a single PC. Interestingly, the false-positive rate in this case was higher in this system as compared to the example above. The time needed for the hierarchical strategy proposed by Marino and Voit [24] was 6.38 mins. The parameter values of metabolites X_1, X_2, X_4, and X_5 were found correctly, but the parameters associated with X_3 were not all identified, even though the error satisfied our termination criterion (*log(SSE)* < -7), indicating that a different solution with essentially zero-error exists in this equation. This result interestingly echoes the result based on linearization, as proposed by Veflingstad *et al.* [23]. The reason is probably that X_2 contributes to both the production term and the degradation term of X_3 with the same kinetic order (-1) and that the time course is not very informative. Also similar to Veflingstad's results, when we used different initial concentrations to perturb $X_2$and X_3 more strongly, AR yielded the correct solution.

Conclusion

Biological system models are usually nonlinear. This renders the estimation of parameter values a difficult problem. S-systems are no exception, but we have shown here that their regular structure offers possibilities for restructuring the estimation problem that are uniquely beneficial. Specifically, the combination of the previously described method of decoupling with the alternating regression technique proposed here dramatically reduces estimation time. Since the AR algorithm essentially consists of iterative linear regressions, it is extremely fast. This makes it feasible to explore alternative settings or initial guesses in cases where a particular initiation fails to lead to convergence.

Methods of parameter estimation, and the closely related task of structure identification, naturally suffer from combinatorial explosion, which is associated with the number of equations and the much faster increasing number of possible interactions between variables, which show up as parameters in the equations. The proposed method of decoupling behaves much better in this respect than most others (*cf.* [5,24]). In practical applications, the increase in the number of combinations is in most cases vastly less than theoretically possible, because the average connectivity of a biological network is relatively small ($<<O(n^2)$; *e.g.*, [26]).

The patterns of convergence are at this point not well understood. Some issues were discussed in the *Results* section and others are detailed in Additional file 1. From these numerical analyses it is clear that convergence depends in a very complicated fashion on the dataset, the constraints, the choice of regressors, and the structure and parameter values of the system. Given that even the convergence features of the Newton algorithm are not fully understood [22], it is unlikely that simple theorems will reveal the convergence patterns of AR in a general manner.

The speed of convergence is also affected by the starting guesses, the choice of regressors, the constraints imposed, and the data set. From our analyses so far it seems that if initially more regressors are used than actually needed, and if they are secondarily constrained, AR converges the fastest. However, a loosely constrained selection of regressors also has a higher chance of convergence to a wrong solution or never to converge. This is especially an issue if the time series are not very informative; for instance, if the system is only slightly perturbed from its steady state. By contrast, when fewer regressors are used, the speed of convergence is slower, but the chance of reaching the optimal solution is increased. A possible explanation of this phenomenon is that more regressors offer more degrees of freedom in each regression, which results in more leeway but also in an increased chance for failure. If AR does not converge, choosing different datasets, using different regressors, or slightly relaxing or tightening the constraints often yields convergence to the correct solution. Most importantly, in all cases of convergence the solution is obtained very quickly in comparison to other methods that attempt to estimate parameters directly via nonlinear regression on the differential equations.

At this stage we have deduced optimized solutions for each metabolite separately. In other words, we have not accounted for constraints among equations, such as stoichiometric precursor-product or branch point relationships. Also, it seems that similar methods should be efficacious for the estimation of Generalized Mass Action systems [4]. These issues will be the subject of further study. We have also assumed that the data are error-free. This assumption was made to identify advantages and failures of the AR algorithm in a fashion as unobstructed as possible. Also, as we typically smooth raw data before estimating parameter values, the analysis of noisy data seems to depend more on the quality of smoothing than on AR itself. The same is the case for data that do not stem from S-system models, where the quality of the estimation is driven by the accuracy of the S-system representation. Future studies will elucidate how sensitive to experimental error the algorithm is.

Like any other estimation algorithm, AR is not a panacea. However, our studies so far provide strong indication that this algorithm is much faster than nonlinear algorithms that one can afford to test quite a number of false starts and explore multiple combinations of initial guesses.

Competing interests

The author(s) declare that they have no competing interests.

Additional material

Additional File 1

Additional file of the manuscript
Click here for file
[http://www.biomedcentral.com/content/supplementary/1742-4682-3-25-S1.pdf]

Acknowledgements

This work was supported in part by a National Heart, Lung and Blood Institute Proteomics Initiative (Contract N01-HV-28181; D. Knapp, PI), and an endowment from the Georgia Research Alliance. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsoring institutions.

References

1. Savageau MA: **Biochemical systems analysis. I. Some mathematical properties of the rate law for the component enzymatic reactions.** *J Theor Biol* 1969, **25:**365-369.
2. Savageau MA: **Biochemical systems analysis. II. The steady-state solutions for an n-pool system using a power-law approximation.** *J Theor Biol* 1969, **25:**370-379.
3. Savageau MA: **Biochemical systems analysis. A study of function and design in molecular biology.** Reading, MA, Addison-Wesley; 1976:xvii, 379.
4. Voit EO: **Computational analysis of biochemical systems: a practical guide for biochemists and molecular biologists.** Cambridge, U.K., Cambridge University Press; 2000:xii, 531.
5. Voit EO, Almeida JS: **Decoupling dynamical systems for pathway identification from metabolic profiles.** *Bioinformatics* 2004, **20:**1670-1681.
6. Kikuchi S, Tominaga D, Arita M, Takahashi K, Tomita M: **Dynamic modeling of genetic networks using genetic algorithm and S-system.** *Bioinformatics* 2003, **19:**643-650.
7. Voit EO: **Canonical nonlinear modeling : S-system approach to understanding complexity.** New York, Van Nostrand Reinhold; 1991:xii, 365.
8. Torres NV, Voit EO: **Pathway analysis and optimization in metabolic engineering.** Cambridge, U.K., Cambridge University Press; 2002:xiv, 305.
9. Voit EO, Savageau MA: **Power-Law Approach to Modeling Biological-Systems. 3. Methods of Analysis.** *J Ferment Technol* 1982, **60:**233-241.
10. de Boor C: **A practical guide to splines.** In *Applied mathematical sciences: 27* New York, Springer-Verlag; 1978:xxiv, 392.
11. de Boor C, Höllig K, Riemenschneider SD: **Box splines.** In *Applied mathematical sciences ; v 98* New York; Hong Kong, Springer-Verlag; 1993:xvii, 200.
12. Green PJ, Silverman BW: **Nonparametric regression and generalized linear models: a roughness penalty approach.** In *Monographs on statistics and applied probability ; 58* 1st edition. London; New York, Chapman & Hall; 1994:ix, 182.
13. Seatzu C: **A fitting based method for parameter estimation in S-systems.** *Dynam Systems Appl* 2000, **9:**77-98.
14. Burden RL, Faires JD: **Numerical Analysis.** 5th edition. Boston, MA, PWS Publishing Co; 1993:pp. 156-167.
15. Eilers PHC: **A perfect smoother.** *Anal Chem Anal Chem* 2003, **75:**3631-3636.
16. Tsai KY, Wang FS: **Evolutionary optimization with data collocation for reverse engineering of biological networks.** *Bioinformatics* 2005, **21:**1180-1188.
17. Almeida JS, Voit EO: **Neural-network based parameter estimation in complex biomedical system.** *Genome Inform* 2003, **14:**114-123.
18. Almeida JS: **Predictive non-linear modeling of complex data by artificial neural networks.** *Curr Opin Biotechnol* 2002, **13:**72-76.
19. Karjalainen EJ: **The Spectrum reconstruction problem: use of alternating regression for unexpected spectral compoments in two-dimensional spectroscopies.** *Chemom Intell Lab Syst* 1989, **7:**31-38.
20. Oliveira MR, Branco JA, Croux C, Filzmoser P: **Robust redundancy analysis by alternating regression.** In *Theory and Applications of Recent Robust Methods* Edited by: Hubert M, Pison G, Struyf A and Van Aelst S. Basel, Birkhauser; 2004:235-246.
21. Martens H, Naes T: **Multivariate Calibration.** Chichester, UK, John Wiley & Son Ltd.; 1989:419.
22. Epureanu BI, Greenside HS: **Fractal basins of attraction associated with a damped Newton's method.** *SIAM Rev* 1998, **40:**102-109.
23. Veflingstad SR, Almeida JS, Voit EO: **Priming nonlinear searches for pathway identification.** *Theor Biol Med Model* 2004, **1:**8.
24. Marino S, Voit EO: **An automated procedure for the extraction of metabolic network information from time series data.** *J Bioinform Comp Biol* in press.
25. Hlavacek WS, Savageau MA: **Rules for coupled expression of regulator and effector genes in inducible circuits.** *J Mol Biol* 1996, **255:**121-139.
26. Jeong H, Tombor B, Albert R, Oltval ZN, Barabasi AL: **The large-scale organization of metabolic networks.** *Nature* 2000, **407:**651-654.