PARAMETER ESTIMATION IN S-SYSTEM AND GMA MODELS: AN OVERVIEW

Nuno Tenazinha and Susana Vinga

KDBIO Group - INESC-ID

R Alves Redol 9, 1000-029 Lisboa, Portugal

{ntenazinha, svinga}@kdbio.inesc-id.pt

May 2009

INESC-ID Technical Report 31 / 2009

ABSTRACT

Multilevel biological data for different cellular systems are accumulating at a day basis speed. High-quality time series of gene expression, metabolite concentrations or enzyme activities now add to the steady-state data gathered in the last decades of molecular biology. Mathematical modeling plays a fundamental role in understanding these experimental data by providing structured abstractions that represent the underlying biological processes at different simplification levels. Power-law modeling within Biochemical Systems Theory (BST) has been used thoroughly in the last decades within this context. Despite of its attractiveness as an approximate modeling tool, problems still exist at the level of parameter identification in order to achieve a compromise between data description and model predictive properties. This problem has been traditionally tackled through optimization routines that cannot always guarantee the right solution or that become computationally expensive when systems grow in size, therefore lacking scalability.

This report reviews recent methodological proposals to deal with the problem of parameter estimation in models of biological systems based on BST, whether in its S-System formalism or in the GMA alternative. These approaches have made significant contributions to the decrease in computational power required in the estimation procedure as well as to the reliability of the parameter solutions. Nevertheless an ideal method that can be applied to different problems and datasets, if it exists, is still far from being found.

This work was performed at the Knowledge Discovery and Bioinformatics (KDBIO) group from INESC-ID, Lisboa under the framework of project DynaMo - Dynamical modeling, control and optimization of metabolic networks funded by FCT (PTDC/EEA-ACR/69530/2006) and under PhD fellowship SFRH / BD / 33209 / 2007 (FCT) associated to the PhD Program in Computational Biology from Instituto Gulbenkian de Ciência (sponsored by FCG, FCT, Siemens SA).

INDEX

1. INTRODUCTION

1.1. Biochemical Systems Modeling

The emergence of high-throughput technologies for producing biological data (such as microarrays, mass spectrometry or *in vivo* NMR) drove biology to a situation where describing the cellular status is possible at physiological, metabolic, proteomic and genomic levels. Unraveling how these multiple cell layers interact to produce different phenotypes is now one of the most challenging topics in modern biology and one that demands a systems approach. However, in contrast to what was traditional in the decades of molecular biology it has now become humanly unfeasible to scrutinize these massive amounts of data without resourcing to bioinformatics and computational methods. In this context, mathematical modeling emerged as a fundamental tool for understanding biological phenomena by providing structured abstractions to rationalize complex cellular behaviors. The iterative model building process coupled to validation with high-quality experimental data has brought new insightful views of cell processes.

Time series profiles, that is, measurements of quantities such as gene expression, enzyme activities or metabolite concentrations through time, allow describing dynamic responses of cells under well-defined experimental conditions, and thus are particularly interesting for this purpose because they unravel the biological networks that give rise to those responses and how they are intermingled. Experimentally time-series profiles can be obtained by imposing a controlled perturbation to a stationary system, and recording the output response of the system. From a modeling perspective, deterministic models can be built that describe the time-evolution of the system from a given initial condition upon some input perturbation. These models are considered valid when a high number of biochemical molecules is present in the cell; otherwise stochasticity should be taken into account.

At any time-point *t*, the state of the system is then described by the state-vector $\mathbf{X}(t,\mathbf{p}) = \left(X_1(t,\mathbf{p}), X_2(t,\mathbf{p}), \ldots, X_n(t,\mathbf{p})\right)$. For biological systems the state-variables X_i typically represent the concentrations of biochemical species which are both time and parameter dependent. The vector **p** contains all unknown parameters of the system.

The model is then built in the form of a set of ordinary differential equations (ODEs) such as in **Equation 1**.

$$\begin{aligned} &\frac{d\mathbf{X}(t,\mathbf{p})}{dt} = f\left(t, \mathbf{X}(t,\mathbf{p}), \mathbf{p}, \mathbf{u}(t)\right), \quad \mathbf{X}(t_0,\mathbf{p}) = \mathbf{X}_0(\mathbf{p}) \\ &\mathbf{y}(t,\mathbf{p}) = g\left(t, \mathbf{X}(t,\mathbf{p}), \mathbf{p}, \mathbf{u}(t)\right) \end{aligned} \qquad \textbf{[1]}$$

where, **u**(t) are input signals, *f* is a given vector function and **y**(*t*,**p**) represents the vector of observed variables.

For the formulation of mathematical models of biochemical systems the vector-valued function *f* described in **Equation 1** needs to be defined. Traditionally it has been built in a bottom-up or mechanicistic approach: information on individual processes and their interactions is gathered and a comprehensive model is then integrated. As an example, mass action kinetics or Michaelis-Menten kinetic rates and their variants are commonly used to describe the different cell processes. Nevertheless, besides being greatly influenced by the wealth of detailed information available, these approaches often become too complex for algebraic manipulation as networks grow in size.

Alternatively, approximate formalisms have received increasing attention in the last years because they provide approximations to the kinetic law valid in the neighborhood of a specified operating point without prior assumptions on the mechanistic aspects of the underlying processes. From these the log(linear) and lin-log approximations (refer to (Heijnen 2005)) and the power-law framework stand in the front-line. The recently proposed Saturable and Cooperative formalism (Sorribas and others 2007) also presents promising results as a new tool that allows a structured formalism for approximating fluxes that show sigmoidal dependencies on the metabolites.

This report focuses on the Power-law framework both in its S-System and GMA formats, because it has proven to be flexible enough to capture the characteristic nonlinearities of biological networks. Furthermore it has the advantage that its parameters can be readily interpreted in terms of topological and regulatory features of the network as well as the order of the reactions involved in the system (Savageau 1976; Voit 2000; Voit and Savageau 1987).

1.2. The Power-Law Framework under Biochemical Systems Theory (BST)

The Power-Law framework within Biochemical Systems Theory (BST) (Savageau 1976; Voit 2000; Voit and Savageau 1987) is derived by approximating a kinetic rate-law around a defined operating point through first-order Taylor series truncation, under logarithmic coordinates. This corresponds to a product of power-laws when reverting to Cartesian space. For a biochemical system with *n* dependent variables and *m* independent variables a power-law approximation of a given rate v_r has the form represented in **Equation 2**.

$$v_r = \gamma_r \prod_{i=1}^{n+m} X_j^{f_{rj}} \qquad [2]$$

where γ_r is a non-negative kinetic rate (also known as the turnover rate of the process), and f_{rj} is the apparent kinetic-order of the process v_r with respect to X_j. This kinetic order is equivalent to the local sensitivity or elasticity coefficient in the Metabolic Control Analysis framework (Cascante and others 1995; Curto and others 1995; Fell 1997; Sorribas and others 1995).

From the model building perspective, two different strategies have been used within BST. These are the S-System and the Generalized Mass Action approaches which both rely on the power-law approximation but making slightly different considerations on the system's operation.

1.2.1 The S-System Modeling Framework

The S-System framework within BST focuses on the constituent pools, aggregating all influxes (or effluxes) into a given pool described by a single power-law term. Thus, for a biochemical system having *n* constituents, the generic form of an S-System is as follows in **Equation 3:**

$$\dot{X}_i = \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}} \text{ with } i = 1,2,\dots,n \qquad \textbf{[3]}$$

where X_i denotes the concentration of constituent *i*. Its time derivative $\dot{X}_i$is expressed as the difference between a production term and a degradation term. These two terms are products of power-law functions and contain two different types of parameters: the non-negative rate constants α and β relative to metabolite $\dot{X}_i$, that quantify the turnover rate of each process with magnitude depending on the scales of the system, and the real-valued kinetic orders *g* and *h*, which can be directly interpreted as the kinetic orders of the corresponding chemical reactions. However they cannot be seen as stoichiometric coefficients anymore, but rather as a measure of the influence of a particular component in the rate of the process. Negative kinetic orders reflect inhibitory effects while their magnitude quantifies the strength of the effect. A kinetic order of zero corresponds to no direct effect. Therefore, the structure of the network can be inferred in a straightforward fashion if the parameter values of the S-System are known.

The particular S-System structure also allows a straightforward steady-state analysis (by setting **Equation 1** equal to zero) through the determination of explicit steady-state solutions relating the system's fixed points with its internal parameters.

1.2.2. The GMA Modeling Framework

In the GMA formulation, each reaction having a direct effect in the process is considered independently and thus no flux aggregation occurs. Instead, each flux is individually linearized in logarithmic coordinates, which in Cartesian coordinates corresponds to a product of power-law functions that contains those and only those variables that directly affect the flux, raised to an exponent – its kinetic order. The product also contains a positive rate constant that determines the magnitude of the flux or speed of the process. The mathematical formulation of any GMA model is thus as follows in **Equation 4**:

$$\dot{X}_i = \sum_{p=1}^{P_i} \pm \gamma_{ip} \prod_{j=1}^{n} X_j^{f_{ipj}} \text{ with } i = 1,2,\dots,n \qquad \textbf{[4]}$$

where γ_{ip} is the rate constant of the process and f_{ipj} its kinetic order. In this case, the number of reactions per differential equation may be different for each species.

In mathematical terms the GMA representation can be seen as a generalized formalism that includes S-Systems (GMAs with at most one positive and one negative term – aggregating individual fluxes into the net processes of production and consumption), linear stoichiometric models, and combinations of both. Here lies one of the main advantages of GMA formalism over S-Systems: in the latter, if one of the contributing fluxes is null, all the pool influx will be set to zero due to the power-law products involved. However this flux aggregation in S-Systems allows their closed-form steady state solutions that cannot be derived for GMAs. Furthermore, S-Systems are more amenable to a black-box approach where no information about the topology is available, since the structure is always the same, independently of the specific reactions present. Another important difference to point out when trying to estimate model parameters is that, given the dimension of the system, S-Systems have a bounded number of possible parameters to be recovered, 2n(n+1), whereas this is not the case for GMA systems. Thus estimating which parameters are zero-valued (which variables have no direct effect in the processes) is of particular importance when using this formalism – and having *a priori* information on the network topology can greatly facilitate the estimation task.

1.3. Optimization methods as tools for reverse modeling

In this top-down approach most of the times the system is treated as a black box to which a stimulus is applied. If we record measurements of the output response (in the form of time series) we can try to fit a mathematically structured model to the data using some optimization algorithm, in a reverse modeling problem. For a given set of experimental measurements $\mathbf{y}_i \equiv \mathbf{y}(t_i, \mathbf{p})$ available at discrete time points t_i, i =1,…,N, the general reverse modeling problem can be stated as to find the parameter set $\hat{\mathbf{p}}$ that minimizes a chosen evaluation function quantifying the distance between the experimental data and the solutions resulting from integrating the system in **Equation 1** (typically in the least-squares sense). This problem can be dealt with optimization routines that search the whole parameter space while trying to minimize the evaluation function.

The optimization task can be considered at the moment the major bottleneck in the whole modeling process and the criticality of this step has been generally underestimated. The *a priori* available information on the system sets the difficulty of the optimization problem: if the topology and regulatory interactions are known, the problem reduces to a parameter estimation task; in the other extreme, if no information exists, the more difficult problem of network structure inference is posed. The main difference between both problems is the higher dimensionality of the parameter space to search when inferring structure.

This inverse problem should be simple if a suitable model is given. But despite of the huge increase in the amount and quality of the available data, the task of extracting significant information from it has not revealed to be straightforward. Difficulties exist both at the conceptual and at the computational levels. For instance, the choice of mathematical framework must ensure that the dynamics are captured with sufficient accuracy but still allowing the biological interpretation of parameters. Since S-Systems are intrinsically nonlinear, estimating their parameters is basically to numerically solve a system of non-linear differential equations and this is a difficult and computationally expensive task – unsuccessful optimizations may occur due to trapping of the search algorithm in local minima of the parameter space, lack of convergence or even a convergence speed that becomes unfeasible for large systems.

The standard search algorithms for overcoming these drawbacks have been gradient based non-linear regression, like the Newton (e.g (Seber and Wild 2003)) or the Levenberg-Marquardt methods (Levenberg 1944; Marquardt 1963), or direct search, being the Hooke and Jeeves (Hooke and Jeeves 1961) and the Nelder and Mead (Nelder and Mead 1965) algorithms the most popular. The convergence of these methods is faster when aiming at local minima of the objective function but if one is interested in a global minimum, with objective functions that are likely to have several local minima, stochastic optimization methods can be suitable alternatives and have proven to yield the most computationally interesting results – strategies making use of genetic algorithms or evolutionary programming have been quite common for parameter optimization in ODE-based models of biochemical networks ((Kikuchi and others 2003; Mendes and Kell 1998; Moles and others 2003). As an example of the difficulties encountered, the algorithm proposed by (Kikuchi and others 2003), efficiently estimates the parameters of an S-System with five variables but despite the small system size and the noise-free data, each loop of the algorithm took around 10 hours on a cluster of 1040 Pentium III processors (933 MHz).

Hence, gathering *a priori* knowledge on the system's topology or building educated guesses for some of the model parameters (or imposing significant restricted ranges) is one of the best ways to improve the quality of the estimation task by reducing the high dimensionality of the parameter space typical of S-Systems and GMA models. It must be noted that preprocessing of the data is *per se* an intricate step, due to noisy time series datasets that must be dealt with frequently.

1.4. Scope of the report

The aim of the report herein presented is to give a state of the art on the available methodologies to deal with the parameter estimation/optimization process in S-System and GMA based models. It will focus primarily in recent efforts that have in some way contributed to improve the currently available methodologies, some of them only tested against artificial data but with very promising results. First a brief description of the power-law based frameworks is made. Subsequently, the methods review starts with decoupling approaches that were proposed to increase computation speed of the optimization routines and which are now used in most of the methods that will follow. Focus then shifts to methods that are deterministic in nature such as alternating regression, eigenvector optimization, global optimization with branch-and-reduce and the interval analysis based methods. Methods based on evolutionary programming and simulated annealing follow afterwards.

2. RECENT DEVELOPMENTS IN PARAMETER ESTIMATION OF POWER-LAW BASED MODELS

2.1. Decoupling Methods

Two of the leading challenges in the parameter estimation process are the development of efficient optimization methods but also of powerful numerical integration tools to solve differential equations. Failure in numerical integration is a major concern when performing optimization tasks besides the fact that it is time consuming, requiring in excess of 95% of the total search time and this percentage may even approach 100% if the differential equations are stiff (Voit and Almeida 2004).

Different groups have made important contributions, developing new methods to reduce this computational burden. (Kimura and others 2005) present a decomposition strategy to divide the inference problem into several subproblems; (Voit and Almeida 2004) propose a derivative method, based in slope approximations to be used as measured data, splitting the differential equations into sets of algebraic relations; (Tsai and Wang 2005) also present a way of decoupling the system into algebraic equations through a modified collocation method.

a) Decomposition Method

(Kimura and others 2005) present a way to reduce the high-dimensionality of the inference problem in S-System models by decomposing it into a set of smaller subproblems. They define the canonical problem as an optimization aiming at the minimization of an objective function based in the sum of the squared relative error and then divide it into several subproblems, each corresponding to a state variable. Thus for an S-System with *n* constituents, the objective function is as in **Equation 5**:

$$f_i = \sum_{t=1}^{N} \frac{\left(Xi(t) - Xe,i(t)\right)^2}{Xe,i_{max}^2} \text{ with } i = 1,2,\ldots,n;\ t = 1,2\ldots,N \qquad \textbf{[5]}$$

where $X_{e,i}(t)$ is the experimentally observed concentration of species *i*, from the time-series data and $X_i(t)$ is the numerically determined concentration of species *i*, by solving the system defined in **Equation 6**:

$$\dot{X}_i = \alpha_i \prod_{j=1}^{n} Y_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} Y_j^{h_{ij}} \text{ with } i = 1,2,\ldots,n \qquad \textbf{[6]}$$

such that $Y_j = \begin{cases} X_j, & if\ j = i \\ \hat{X}_j, otherwise \end{cases}$

where $\hat{X}_j$ is an estimated concentration value obtained by direct estimation from the time-series data (either by spline interpolation or local linear regression, depending on the absence or presence of noise in the dataset, respectively).

As previously noted, the optimization problem as defined in **Equation 1**, is *2n(n+1)* dimensional. By applying this decomposition, we end up having *n* subproblems with dimension *2(n+1)* and the solutions of **Equation 6** will be more proximate to the solutions from **Equation**

1 depending on the accuracy of the curves given as observed concentrations. The optimization formulated in this manner is amenable to parallel computation of the decoupled differential equations which can greatly reduce the computation time of the search algorithm.

b) Derivative Method

A new direction in approaching the parameter estimation task was recently introduced by (Voit and Almeida 2004) following ideas in (Voit and Savageau 1982), where the authors propose the decoupling of the system into a set of separate algebraic equations, by interpreting the time-derivatives at each time point as a slope measured from the data. This method does not require the integration of the set of differential equations and thus optimization algorithms can become much faster. As an example, if one has an S-System model with *n* components, with data measurements at *N* time points, the decomposition in **Equation 7** can be made:

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

with $i = 1,2,\ldots,n$ and $k = 1,..,N$ **[7]**

where S_i = $S_i(t_k)$ is the estimated slope of metabolite X_i at time point t_k. The optimization problem is thus reformulated from a set of n differential equations to *n x N* algebraic equations. This reformulation allows the independent analysis of each equation, which allied to the easiness of manipulating algebraic equations, greatly reduces the time requirements of parameter estimation.

It is extremely important to have accurate calculations of the slopes for the method to be efficient because most of the time derivatives calculations amplifies the error present in noisy datasets. Thus, depending on the type of data, spline interpolation or smoothing algorithms can be used for this purpose, e.g. (Vilela and others 2007) propose a smoothing algorithm capable of dealing with different noise structures that has the advantage of providing a closed-form solution for computing the derivatives of the smoothed signal.

c) Modified Collocation Method

As an alternative way of decoupling the set of differential equations into an approximating set of algebraic equations, (Tsai and Wang 2005) suggest a modified collocation method. Collocation methods are traditionally used for finding numerical solutions of ODEs. The underlying idea is to choose a finite-dimensional space of candidate solutions (usually in the form of polynomials), and a number of points in the domain, named collocation points, that satisfy the given system at the collocation points. Hence, the state variables X_i (as defined in **Equation 1**), are spanned by a set of shape functions $\phi_j(t)$ as shown in **Equation 8**:

$$X_i(t) = \sum_{j=1}^{N} x(t_k)\phi_j(t_k) \qquad \textbf{[8]}$$

where $x(j)$ is an expansion coefficient of *X(t)* and ϕ_j is a set of polynomial shape functions.

The authors illustrate their method by using the simplest shape function, the interpolation polynomial in the Lagrange form, expressed in **Equation 9**, and applied to *N* collocation points:

$$x(j) = x(j-1) + 0.5\eta_j\{f[x(j),p] + f[x(j-1),p]\} \quad \textbf{[9]}$$
$$= F[x(j), x(j-1), p] \quad \text{with } j\text{=1,2,…,}N$$

where *x(j)* is a vector of expansion coefficients at the *j*-th collocation point and is equal to the solution *X(t)* at time $t = t_j$; *f[x(j),p]* is a vector function of expansion coefficients at the *j*-th collocation point, η_j is the time interval between the *j*-th and the (*j*-1)-th collocation points. To increase the speed of the algorithm, the authors further approximate the system by substituting the measured data (eventually smoothed) at each collocation point in the right-hand side of **Equation 9**, such that (see **Equation 10**):

$$\begin{bmatrix} X_1(t_j) \\ \vdots \\ X_n(t_j) \end{bmatrix} \cong \begin{bmatrix} X_{e,1}(t_{j-1}) \\ \vdots \\ X_{e,n}(t_{j-1}) \end{bmatrix} + 0.5\eta_j \left\{ \begin{bmatrix} f_1(X_e(t_j), p) \\ \vdots \\ f_n(X_e(t_j), p) \end{bmatrix} + \begin{bmatrix} f_1(X_e(t_{j-1}), p) \\ \vdots \\ f_n(X_e(t_{j-1}), p) \end{bmatrix} \right\} \quad \textbf{[10]}$$

with *j*=1,2,…,*N*

By doing this, the reverse problem is reformulated into a system of *n x N* decoupled algebraic equations, saving a lot of computational time because no numerical integration of differential equations is needed.

A known problem of all these decoupling strategies is that care must be taken with the mass conservation relations between differential equations. Since most of the times different rate-laws are shared among several ODEs, constraints must be imposed to account for these effects before running the optimization algorithm, or these relations will be lost when decoupling the system.

2.2. Alternating Regression

Alternating regression has been proposed by (Chou and others 2006) as a fast new strategy to parameter estimation and structure identification within the S-System framework. Its key feature is that it dissects the nonlinear inverse problem of parameter estimation into iterative steps of linear regression. After decoupling the system in the sense proposed by (Voit and Almeida 2004), the algorithm alternates between estimating the parameters of the production term and the degradation term of a single S-System equation at a time. Guesses from one phase are used to improve the estimates of the other phase iteratively until a solution is found or the algorithm is stopped under certain criteria. The method is also suited for structure identification when no parameters in the S-System are *a priori* set to zero. The algorithmic details for estimating the parameters of the ith differential equation are as follows:

[Alternating Regression]

1. Create the *(n+1) x N* matrices L$_P$ and L$_D$ containing the regressors of the production and degradation terms, respectively, of X_i. Only those X_i that directly affect the production term (those with non-zero kinetic order) are included in L_P. L_D is defined analogously but with X_i affecting the degradation term.

$$L_P = \begin{bmatrix} 1 & log\,(X_1(t_1)) & \dots & log\,(X_i(t_1)) & \cdots & log\,(X_n(t_1)) \\ 1 & log\,(X_1(t_2)) & \dots & log\,(X_i(t_2)) & \cdots & log\,(X_n(t_2)) \\ \vdots & \vdots & & \vdots & & \vdots \\ 1 & log\,(X_1(t_k)) & \dots & log\,(X_i(t_k)) & \cdots & log\,(X_n(t_k)) \\ \vdots & \vdots & & \vdots & & \vdots \\ 1 & log\,(X_1(t_N)) & \dots & log\,(X_i(t_N)) & \cdots & log\,(X_n(t_N)) \end{bmatrix}$$

2. Compute the matrices C_P and C_D, invariant throughout the iterative process
$$C_P = (L_P^T L_P)^{-1} L_P^T \;, C_D = (L_D^T L_D)^{-1} L_D^T$$
3. Select initial values for β_i and h_{ij}, making use of available information for constraining some or all h_{ij}
4. FOR all t_k, k = 1,2,…, N compute $\beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_k)$ using values $X_j(t_k)$ from the observed or smoothed time series data
5. Compute the N-dimensional vector, y_D: $y_D = log\,(S_i(t_k) + \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_k))$, k =1,2,..,N
6. Estimate the parameters of the production term by determining vector b_P:
$$b_P = [\hat{\alpha}_i, \hat{g}_{ij}, j = 1,2, \dots, n]^T = C_P y_D$$
7. Perform the analogous regression for the degradation term using the parameter estimations used in the previous step:
$$y_D = log\,(\alpha_i \prod_{j=1}^{n} X_j^{g_{ij}}(t_k) - \; S_i(t_k))$$
$$b_D = [\hat{\beta}_i, \hat{h}_{ij}, j = 1,2, \dots, n]^T = C_D y_P$$
The components of b_D will be used as estimations for the parameters of the degradation term in the next iteration.
8. Calculate the logarithm of the sum of squared-errors (SSE):
$$log(SSE) = log\left(\sum_{k=1}^{N} \left(y_D(k) - L_P b_P(k)\right)^2 + \sum_{k=1}^{N} \left(y_P(k) - L_D b_D(k)\right)^2\right)$$
9. ITERATE UNTIL the termination criteria are satisfied (for example log(SSE) is smaller than a specified value or the maximum number of iterations is reached)

The algorithm was successfully demonstrated with artificial data generated for a didactic biochemical network with 4 dependent variables and 17 parameters. The authors claim that the strongest point of alternating regression is its use of iterative linear regressions that renders the algorithm extremely fast and, in combination with decoupling methods, makes it many times faster than an algorithm that directly estimates systems of nonlinear differential equations simultaneously. Nevertheless issues on patterns of convergence and convergence speed are not well understood yet, and no necessary and sufficient criteria for convergence are known. Issues on the suitability of alternating regression for noisy datasets were recently addressed (Beyer 2008) and despite of still being relatively fast, some difficulties have arisen, with the algorithm not being able to recover the parameter values of the tested models used. Even though the curve fits were reasonable, predictions for untested conditions were unsuccessful.

2.3. Eigenvector Optimization

Recently (Vilela and others 2008; Vilela and others 2009) proposed a new method motivated by Alternating Regression for the problem of model identification when no information on the system's topology exists. The method decouples the system into sets of algebraic equations and uses multiple linear regression coupled to a sequential quadratic programming optimization routine. Unlike Alternating Regression, which iteratively computes the parameters from both terms of the S-System to improve the estimates successively, this new approach first focuses only on one of the terms, estimating both its rate constant (α or β) and kinetic order (g or h) accurately before the other term is estimated through linear regression. The estimation of the first term's parameters relies on the following notion – as in Alternation Regression, the decoupled S-System format can be rewritten such that:

$$L_P b_P = y_P \qquad \textbf{[11]}$$

where L_P, b_P and y_P are as defined in points 1, 5 and 6, respectively, of Alternating Regression. It is clear that, if matrix L_P is invertible, than the parameter vector b_P can be computed through $b_P = (L_P^T L_P)^{-1} L_P^T y_P$. Combined with **Equation 11**, this means that:

$$(L_P^T L_P)^{-1} L_P^T y_P = y_P \Leftrightarrow C_P y_P = y_P \qquad \textbf{[12]}$$

Where C_P is the same matrix as in point 2 of Alternating Regression. **Equation 12** implies that the vector y_P must be an eigenvector of matrix C_P, with associated eigenvalue 1. The proposed method thus relies in optimizing a cost function based in the logarithm of the squared residuals between the left and right sides of **Equation 12**. This is, find the optimal eigenvector $\hat{y}_P = C_P y_P$ such that the evaluation function *F* in **Equation 13**, is minimized. The authors suggest a Sequential Quadratic Programming routine to perform this constrained minimization step.

$$F = \log\left((y_P - \hat{y}_P)^T (y_P - \hat{y}_P)\right) \qquad \textbf{[13]}$$

The algorithmic details go as follows.

[Eigenvector Optimization]

1. Create the *(n+1) x N* matrices L_P and L_D containing the regressors of the production and degradation terms, as in the Alternating Regression algorithm.

2. Compute the matrices C_P and C_D, invariant throughout the iterative process
$$C_P = (L_P^T L_P)^{-1} L_P^T \ , C_D = (L_D^T L_D)^{-1} L_D^T$$
3. Select initial values for β_i and h_{ij}. The authors suggest the computation of h_{ij} given an initial estimate of β_i using a linear regression under logarithmic space on the expression $\beta_i \prod_{j=1}^{n} X_j^{h_{ij}} = \varepsilon - S_i^-$, where S_i^- represents negative slope values computed from the time-series. If the slope vector does not contain negative values, than ε should be discarded.
4. Run the Sequential Quadratic Programming routine on function *F* on **Equation 13**, using the form of the gradients of *F* (see (Vilela and others 2008) Methods section).
5. Calculate the other term parameters using multiple linear regression and using the optimized parameters computed from the eigenvector optimization.

This new method is not free from the convergence problems already found on Alternating Regression, especially when ill-posed problems are at stake (when collinearity exists between columns of the regression matrix L_P). However, the method was tested against several systems with different sizes (2, 4 and 5-state variable sizes), and apart from the ill-posed cases, it showed the same convergence pattern. This result is clear from the convergence tests performed by the authors and thus it appears to be a computationally efficient alternative to other methods herein discussed, allowing a quick exploration of the parameter space (on the order of minutes).

2.4. Global Optimization with Branch-and-Reduce

(Polisetty and others 2006) present a way of using deterministic optimization to find global solutions for the parameter estimation task in GMA-based models. Consider a given nonconvex nonlinear optimization problem as set in **Equation 14** with n dependent variables:

$$min_{\gamma,f} \ \|\boldsymbol{e}\|_r \qquad \textbf{[14]}$$

such that $\dot{X}_i = h_i(X(t),\gamma,f) - \ e_i(t)$ with $i = 1,2,\dots,n; \ \ r > 0; \ \ t = 1,2,\dots,P$

where *r* is the *r*-norm considered for minimization, *P* is the number of data points sampled at time *t*, h_i are the GMA rate functions that define the production or consumption rates for species *i*, given the parameters $\gamma$and *f*; and $e_i(t)$ are the errors associated with each constraint equation for species *i* at time *t*. Note that the objective function is linear, but the nonconvexity results from the equality constraints, which are non-linear.

To solve the problem formulated above, a branch-and-bound variant method is suggested coupled with convexification strategies. Before submitting the model to the global optimization a process of converting the GMA format to a set of linear systems with logarithmic constraints occurs. Since in Cartesian space the equations consist in sums of power-law terms and each power-law term also becomes linear in a multivariate logarithmic space, the strategy makes use of this dual linearity features, reducing the remaining linearities to simple logarithmic constraints. Despite still being nonconvex, these constraints can be underestimated by simple functions, converting the task into a linear problem. This reformulation is presented in the set of **Equations 15**.

$$min\ \sum\nolimits_{\forall i,k} e_i(t_k) \qquad \textbf{[15]}$$

such that

$$\dot{X}_i(t) - \sum\nolimits_k \pm z_{ik}(t) \leq\ e_i(t) \qquad [15a]$$

$$-\dot{X}_i(t) + \sum\nolimits_k \pm z_{ik}(t) \leq\ e_i(t) \qquad [15b]$$

$$w_{ik}(t) = log\ (z_{ik}(t)) \qquad [15c]$$

$$w_{ik}(t) = \Gamma_{ik} + \sum\nolimits_{j=1}^{n+m} f_{ijk} log\ (X_j(t)) \quad [15d]$$

where the two inequalities in **Equation 15a** and **15b** arise because when minimizing the absolute value of some function, constraints of the form *|f(x)|= e* can be rewritten as two inequality constraints *f*(x) ≤ e and -*f*(x) ≤ $e.$ $z_{ik}(t) = \gamma_{ik} \prod_{j=1}^{n+m} (X_j(t))^{f_{ijk}}$ are new variables defined to the convexification purpose. The values of $\dot{X}_i(t)$ and $X_i(t)$ are assumed to be known from the observed data, for example from smoothing methods. If we further apply logarithms to each definition of the variables *z*, it will result in the variables *w* defined in **Equations 15c** and **15d**, where Γ_{ik} is the logarithm of the rate constant. Notice that Γ_{ik}, f_{ijk} and $e_i(t)$ only appear in linear constraints whereas *w* and *z* are related through a simple non-linear expression.

After this initial step, a branch-and-reduce method is applied. This method generates upper and lower bounds for the objective function, at the global solution. It is a variant of the traditional branch-and-bound algorithm with bound tightening techniques for accelerating the convergence of the algorithm – unfeasible or suboptimal parts of the feasible region are eliminated by using range reduction techniques such as optimality-based and feasibility-based range reduction tests or interval analysis techniques.

[Branch-and-Reduce]

1. Generate the boundaries: the lower bound is created by solving the convex relaxation of the original nonlinear problem. Any local minimizer for the original non-linear problem may serve as an initial upper bound for the objective function value. If the lower bound is sufficiently close to the upper bound, within a pre-specified ε tolerance, the algorithm terminates.
2. If not, introduce partitions in the feasible region generating new lower bounds for the new partitions. Use a fathoming criteria to verify if a given partition needs further processing or if it can be removed from consideration:
   a. If the relaxed problem associated with the partition is infeasible, adding additional constraints will not make it feasible; the partition itself is infeasible and hence can be removed from further consideration;

b. If the objective function value of the relaxed problem associated with the current partition is greater or equal to the best solution found so far, then the partition can be removed from further consideration.

3. Any feasible solution to the original problem may serve as an upper bound for the global solution. The algorithm terminates when the lower bounds for all partitions either exceed or are sufficiently close to the best upper bound. At this point, a global optimum has been determined within the originally preset bounds on the parameter search space. This global optimum is the best value of the objective function (but it is noteworthy that multiple points in the parameter space may lead to equivalent values of the objective function)

The algorithm was tested in two artificial networks: a didactic example (3 dependent variables, 1 independent variable and 14 parameters) and in a more complex model of the anaerobic fermentation pathway of *Saccharomyces cerevisiae* (5 dependent variables, 9 independent variables and 19 parameters). For the systems chosen the branch-and-reduce algorithm is fast and reliable and the authors claim this to be advantageous when compared with local solvers, such as non-linear regression algorithms, which may not be able to converge to the global solution when the parameter search space is large, or the error surface is ragged. A slightly altered methodology (using a Mixed Integer NonLinear Programming Algorithm) was demonstrated to be feasible in (Polisetty and others 2008), to optimize the yield in two different regulated metabolic networks.

2.5. Deterministic Approaches with Interval Analysis and Newton-Flow Analysis

In (Tucker and Moulton 2006) the authors present a deterministic approach to parameter estimation based on interval analysis which allows the analysis of entire sets of parameters, exhausting the global search within a finite number of steps, and overcoming the known drawbacks when approaching this issue as a global minimization problem. The methodology makes use of the decoupling strategy proposed by (Voit and Almeida 2004) but extends it by adopting the computation of ranges of slopes for entire domains of parameters. If *$[p_i]$* denotes a hypercube in the parameter space $\mathbb{P}_\mathrm{i}$ (each component of *$[p_i]$* is an interval), then for any point $p_i \in [p_i]$ we have a vector field *f* such that:

$$f_i(X(t_k); p_i) \in F_i(X(t_k); [p_i])$$

where F_i is a set-valued function that contains all possible slopes corresponding to parameters taken from *$[p_i]$*. Thus if a given sample point *$X(t_k)$* produces a range of slopes S_i such that $S_i \notin F_i(X(t_k); [p_i])$, then no parameter in *$[p_i]$* can have generated the sample data; it is then said that the parameter hypercube *$[p_i]$* violates the *cone condition*, and this is a simple criterion to reject regions of the parameter search space $\mathbb{P}_\mathrm{i}$. The proposed algorithm can be generally described as follows:

[Interval Analysis for S-System Parameter Estimation]

1. For a collection of sample data $\{X_{ij};S_i\}_{i,j}$, generated from an S-System with parameter $p^* = (p_1^*,\ldots,p_d^*)$, the search is divided into d independent component-wise searches for $p_1^*,\ldots,p_d^*$. The searches can be performed in parallel because they are completely independent.
2. Initialize each global parameter region for the search as $\mathbb{P} = ([p_1],\ldots,[p_{2(d+1)}])$, where the bounds can be set based in *a priori* biochemical knowledge.
3. The idea is to adaptively partition each space $\mathbb{P}_\mathrm{i}$ into successively smaller hypercubes, retaining the ones satisfying the cone condition. If this occurs either the parameter box diameter is smaller than a predefined tolerance value, in which case the result is stored, or the hypercube is bisected along its widest component, and the two resulting hypercubes are fed to the algorithm again. For a given level of resolution, the process is terminated and the result is a collection of hypercubes $[p_i^{(1)}],\ldots,[p_i^{(n)}]$ each of which satisfies $\mathfrak{T}([p_i^{(j)}]) = 1$ where

$$\mathfrak{T}([p_i^{(j)}]) \overset{\text{def}}{=} \wedge_{j=0}^{N}\left(S_i \in F_i(X(t_k);[p_i])\right)$$

   is a Boolean function that returns 1 if $[p_i] \in \mathbb{P}_\mathrm{i}$ satisfies the cone condition at all sample times, and 0 otherwise.

The methodology was tested in several S-Systems in the task of reconstructing the model structure (this is, not knowing *a priori* the zero-valued parameters). The algorithm has a runtime of several hours for a system of 4 or 5 equations and gives accurate parameter estimations, but is not yet able to handle noisy time series data. The authors also envision that the method should be scalable to a larger class of problems, including GMA models.

More recently, (Kutalik and others 2007) presented an interesting new approach that ameliorates the interval analysis algorithm in terms of computational performance and in handling noisy time series data. They show, for several S-System models, that there is a one-dimensional attractor (containing the true parameter set) in the Newton-flow corresponding to the standard minimization problem in parameter estimation. Although the existence of this attractor is not mathematically proven, the authors propose the form shown in **Equation 16**.

$$\gamma_j(w) = \begin{cases} w, & j=1 \\ \frac{a_{j-1}}{b_{j-1}+w}, & j=2,\ldots,n+1 \\ w + a_0, & j=n+2 \\ \frac{c_{j-n-2}}{d_{j-n-2}+w}, & j=n+3,\ldots,2n+2 \end{cases} \qquad \textbf{[16]}$$

where a_j, b_j, c_j and d_j are real constants and variable j loops through the parameter indices, this is, j = 1 corresponds to α_i, j = 2 to g_{i1}, j = 3 to g_{i2},…, j = n+1 to g_{in}, j = n+2 corresponds to β_i, j = n+3 to h_{i1}, until j = 2n+2 that corresponds to h_{in}.

Thus it suffices to determine this attractor, instead of searching the complete parameter space, to find a global optimal parameter set. The authors also propose a variant of this method to extend it to the task of identifying network topologies. The algorithmic details are described below.

[Searching the Attractor in Newton-flow for S-System Parameter Estimation]

1. Start by decoupling the differential system in the sense proposed by (Voit and Almeida 2004), into *n x N* algebraic inequalities, using the slopes S_i
2. FOR the ith set of *N* equations for the *n* available time points, define the optimization problem as a least-squared minimization problem:
$$f(p_i) = \sum_{j=1}^{N} \left(S_i(t_j) - \alpha_i \prod_{k=1}^{n} X_{kj}^{g_{ik}} - \beta_i \prod_{k=1}^{n} X_{kj}^{h_{ik}} \right)^2$$
where $p_i = (\alpha_i, g_{i,1}, \ldots, g_{i,n}, \ldots, \beta_i, h_{i,1}, \ldots, h_{i,n})$ and the aim is to find a set p_i that minimizes $f(p_i)$ subject to upper and lower bounds set by a priori biochemical knowledge.
3. Generate 40 uniformly distributed initial random guesses within the constrained parameter boxes, which will be used to feed the interior-reflective Newton minimization algorithm – the resulting optimized parameters are assumed to lie in the neighborhood of the hypothesized attractor.
4. With a 2-dimensional regression, use the previous estimations to determine the parameters a_j, b_j, c_j and d_j from **Equation 16**. If a goodness of fit R^2 greater than 0.9 is achieved proceed to step 5.
5. Once the attractor equations are estimated, use these to perform again the Newton minimization algorithm. Start by dissecting the bounding interval for α, $[l_i, u_i]$ into *M* equal parts, this is, $[l_i = w_0 \leq w_1 \leq \cdots \leq w_M = u_i]$ and performing the optimization with initial guesses $\gamma(w_0), \ldots, \gamma(w_M)$ - this will yield new estimates for the S-System parameters
6. The global optimum will be chosen amongst the results of step 5, corresponding to the parameter set whose cost function evaluation is minimal.

The methodology was tested with 4-variables and 30-variables artificial networks and was shown to be efficient even for noisy datasets because the attractor in the Newton flow appears to be relatively insensitive to noise. If this conjectured attractor is true (even if the authors do not mathematically prove its existence) it would guarantee that the methodology always results in a global optimum for the parameter space.

2.6. Evolutionary Multi-Objective Optimization

(Liu and Wang 2008), suggest a multi-objective optimization based in an evolutionary algorithm in order to infer realizable S-System structures. As suggested elsewhere (Kikuchi and others 2003; Kimura and others 2005), this task benefits from the assignment of a penalty factor to the cost function based on the sums of the magnitudes of kinetic orders. However there is no obvious way of choosing an admissible penalty weight for this purpose and so, the authors circumvent this need by posing the inference problem as a simultaneous minimization to find the parameter set *p* in the feasible region Ω, (see **Equation 17**):

$$min_{\, p\in\Omega}\{J_1, J_2, J_3\} \qquad \textbf{[17]}$$

where $$J_1 = \frac{1}{nN_s}\sum_{i=1}^{n}\sum_{j=1}^{N_s}\frac{\left(Xe,i(t_j)-Xi(t_j)\right)^2}{Xe,i_{max}^2}$$

$$J_2 = \frac{1}{nN_s}\sum_{i=1}^{n}\sum_{j=1}^{N_s}\frac{\left(\dot{X}e,i(t_j)-\dot{X}i(t_j)\right)^2}{\dot{X}e,i_{max}^2}$$

$$J_3 = \sum_{i=1}^{n}\sum_{j=1}^{l}\left(|g_{ij}| + |h_{ij}|\right)$$

The three functions J_1, J_2 and J_3 are, respectively, the concentration error (in the least-squared sense), the slope error (in the least squared sense) and the interaction measure. The concentration error is employed to measure the goodness-of-fit of the model, $X_i(t)$, to experimental time-series data, $X_{e,i}(t)$; the slope error is used to judge the accuracy of the net rates ($\dot{X}_i(t)$) whereas the interaction measure sums up magnitudes of the kinetic orders, in order to have a measure to prune the structure of the S-System.

They use Pareto optimality and the ε-constraint method (Sakawa 1993) to convert the multi-objective optimization into a single-optimization problem (taking one criterion as the objective function and letting the rest be inequality constraints), as reformulated in **Equation 18**.

$$min_{\, p\in\Omega}\, J = J_3(p)/J_3^E + \omega\langle max\{C_1(p), C_2(p)\}\rangle_+^2 \qquad \textbf{[18]}$$

such that $$C_1(p) = J_1(p)/J_1^E - 1 \leq 0$$

$$C_2(p) = J_2(p)/J_2^E - 1 \leq 0$$

The values J_i^E (i = 1,2,3) in **Equation 18** refer to the expected values for the concentration error, the slope error and the interaction measure, respectively. The bracket operation is defined as $\langle C(p)\rangle_+ = max\ \{C(p), 0\}$; the second term uses a weighted penalty when the parameter set p is not feasible; if any or both C_1 and C_2 are positive, the worst value is used; if both inequality constraints are feasible ($max\ \{C_1, C_2\} \leq 0$) the penalty is null ($\langle C(p)\rangle_+ = 0$). To improve the search for feasible parameters, the penalty parameter, ω, can be set to be greater than $1/max\{J_1^E, J_2^E\}$.

The authors use a Hybrid Differential Evolution (HDE) algorithm (refer to (Tsai and Wang 2005; Wang and Sheu 2000) to minimize each objective function towards finding the global solution. The algorithmic details follow.

[Multi-Objective Optimization]

1. Calculate the expected values, J_1^E and J_2^E, for the concentration error and slope error using HDE to minimize its single-objective parameter estimation problem (in **Equation 17**); let each expected value be its corresponding minimum error criterion
2. Compute the sum of the magnitude of kinetic orders for each single-objective parameter estimation problem. Let each expected value of the interaction measure for each single-objective problem be J^E_{31} and J^E_{32}, and J_3^E set as the maximum of these values

3. Set the parameter ω, such that $\omega > 1/max\{J_1^E, J_2^E\}$, and solve the inference problem in **Equation 18** using HDE. Let the minimum solution be $p*$.
4. If $\langle C(p^*)\rangle_+$ is smaller than a pre-defined tolerance value, go to step 5; otherwise stop the algorithm
5. Sort the kinetic orders, g_{ij} and h_{ij}, using the score $|g_{ij}|/max\{|g_{ij}|,|h_{ij}|\}$ and $|h_{ij}|/max\{|g_{ij}|,|h_{ij}|\}$, respectively
6. Delete the smaller kinetic orders with scores less than the assigned value and repeat step 1-4 to infer the pruned model.

The authors demonstrate the efficiency of this iterative algorithm with two case-studies: inferring an artificial network with 5 dependent variables, 3 independent variables and 28 parameters both with noise-free and noisy datasets, and a 4-variable model of a batch fermentation process with *Saccharomyces diastaticus*. Despite of the relatively successful results the authors acknowledge that the pruning strategy may not be adequate for inferring whether genetic interactions are fragile or robust, because the interaction measure may not be suitable for such high sensitive systems. They suggest instead the use of dynamic sensitivities of state variables with respect to the parameters g_{ij} and h_{ij} in the multi-objective minimization problem.

2.7. Simulated Annealing

The heuristic optimization method commonly known as Simulated Annealing (Kirkpatrick and others 1983) has particular features that render it interesting for the parameter estimation task in S-Systems. Depending on the 'temperature program' used, the method can behave as a global or a local optimization approach, switching to the latter as the pseudo-temperature goes down. Different candidate solutions (of quality different from the current solution) can be accepted during the algorithm iterations, thus allowing transitions out of the local optima typical of the parameter search space. (Gonzalez and others 2007) explore these features by coupling the SA algorithm with an appropriately constructed perturbation function for the parameter search. For example, **Equation 19** describes how the candidate solutions for the kinetic orders are randomly perturbed in the course of the optimization process:

$$p^{(n)} = p^{(n-1)} + l \times \log\left(\sqrt{e} + 1\right) \times N(\bar{x}, \sigma) \qquad \textbf{[19]}$$

where p is the kinetic order (g_{ij} or h_{ij}), e is the current residual error, and l is an externally optimized constant. $N(\bar{x}, \sigma)$ is a function that returns a random number from a Gaussian distribution with mean $\bar{x}$ and standard deviation σ, here set to 0 and 1, respectively. The term $l \times log\left(\sqrt{e} + 1\right)$ is suggested by the authors because there exists a quasi-linear relationship between a perturbation in a given parameter and the current total error of the optimization which is captured by this factor (c.f. (Gonzalez and others 2007)).

A general description of the SA implementation proposed by the authors goes as follows:

[Simulated Annealing]

1. Generate an initial estimate, $p^{(0)}$, assigning random values to each parameter (within the constraining boundaries)
2. Initialize the pseudo-temperature T (should be enough to completely 'melt' the initial parameter estimates). Authors suggest a value of 1000 as initial value.
3. WHILE $T > T_{min}$ (T_{min} = 10^{-5})
    a. Generate a candidate solution *p'* by applying the perturbation function (**Equation 19**) for each of the kinetic parameters, initialized in *p*

    b. Compute the residual errors as
    $e = \frac{1}{nN}\sum_{i=1}^{n}\left(\hat{X}_i(t) - X_i(t)\right)^2$ with *i* = 1,2,…,*n* and *t* = 1,2,..,*N*
    where $X_i(t)$ is the concentration of species *i*, at timepoint *t*, from the time series data and $\hat{X}_i(t)$ is the estimation of that concentration obtained by numerical integration of the system with the particular candidate *h'*.
    Set Δe to *e(p')-e(p)*

    c. IF $\Delta e \leq 0$, accept the candidate solution *p'*, OTHERWISE accept the candidate solution based in the probability *P = exp{-Δe/T}* (notice that higher temperatures and lower changes in the error lead to higher probabilities for the acceptance of *p'*)

    d. Lower the variable *T* (the authors used a pseudo-temperature regime of gradually decrementing T by 96% of the current value)
4. END WHILE

The authors demonstrate the efficiency of the algorithm in three artificial networks designed to simulate different topologies and behaviors and also in the *cadBA* system of *Escherichia coli* against experimental data. They also propose a way of adapting the method for the purpose of structure identification by running an initial estimate of all the model parameters, using a pruning rule to set the zero-valued ones and then performing the algorithm to re-estimate them.

3. CONCLUSION

Multilevel biological data for different cellular systems are accumulating at a day basis speed. High-quality time series of gene expression, metabolite concentrations or enzyme activities now add to the steady-state data gathered in the last decades of molecular biology. Systems biology task now is to make sense of all this knowledge to help understand the rules that govern cellular behaviors in an integrated fashion.

Biochemical Systems Theory has been around for 30 years and has proven to be a successful mathematical framework for modeling biological systems. Much has been discussed both on its S-System variant and on the Generalized Mass Action form, their advantages as well as their shortcomings are acknowledged by the modeling community. Nevertheless the task of building the model is still frequently hindered when estimating its parameters, and as the networks under study grow in size, the associated combinatorial explosion in the parameter search space is unavoidable. Hence, the computational power has become a limiting step for modelers. More efficient and easy to automate methodologies are required to overcome this.

Decoupling methods have been suggested by different researchers in order to relieve the computational costs associated with having to numerically integrate large systems of differential equations. These techniques decompose the optimization problem into smaller subproblems that can be dealt either through parallel computation of the separate solutions or by transforming the system of differential equations into a larger set of algebraic equations which represent a minor burden in computation time. The equations can then be solved independently yielding 'local' estimates for the parameters of each equation. Nevertheless it is recurrent that the collection of the resulting 'local' estimates not always leads to a satisfactory fit of the complete, integrated set of differential equations. And even when this is the case, seldom is the model able to predict system's responses under untested conditions, severely compromising its generalization properties and thus its overall usability. Notwithstanding this appears to be an unavoidable drawback that renders the optimization process feasible for realistic biological systems sizes. The major improvements in computation speed brought by decoupling strategies have been used in most of the latest algorithms as shown in this report. Even if the parameter solutions are very sensitive to the noise structure in the data, they can provide rough guesses to feed anew to other optimization routines. For example (Gennemark and Wedelin 2007) proposed a heuristic search algorithm to identify model structure and estimate its parameters which makes use of the decoupled system to prune the model space by evaluating local structures. For the estimation process a standard nonlinear least squares method is fed with parameter guesses computed through the derivative method.

The recent methodologies highlighted in this review present promising results in circumventing the high-dimensionality of parameter space and are thus interesting approaches for both structure inference and parameter estimation in S-System and GMA models, even if some of them have so far been solely applied to artificial data. Despite of its attractiveness, a direct and efficient methodology has not been found yet. Even if out of the scope of this report, it would be interesting to test how these methods perform in the model identification task, because up to now, there is no systematic comparison of their performance. Since there are now available some benchmarks for testing model identification from time-series data (Gennemark and Wedelin 2009) this is work worth exploring in the future. Nevertheless, given the huge variety of problem domains and the intricacy of biological systems, the holy grail of optimization methods, if it exists, is still far from sight – and the role of the modeler is still one of choosing the best methodology, given the available knowledge on the particular system under study.

4. ACKNOWLEDGEMENTS

Nuno Tenazinha and Susana Vinga would like to acknowledge financial support from the PhD Program in Computational Biology from Instituto Gulbenkian de Ciência (sponsored by Fundação Calouste Gulbenkian, Siemens SA and Fundação para a Ciência e a Tecnologia) (SFRH / BD / 33209 / 2007) and by project DynaMo (PTDC / EEA-ACR / 69530 / 2006, FCT).

5. REFERENCES

Beyer W. 2008. Combining Autosmooth and Alternating Regression for the Estimation of S-System Parameters. INESC-ID Technical Report 29.

Cascante M, Curto R, Sorribas A. 1995. Comparative characterization of the fermentation pathway of Saccharomyces cerevisiae using biochemical systems theory and metabolic control analysis: steady-state analysis. Math Biosci 130,51-69.

Chou IC, Martens H, Voit EO. 2006. Parameter estimation in biochemical systems models with alternating regression. Theor Biol Med Model 3,25.

Curto R, Sorribas A, Cascante M. 1995. Comparative characterization of the fermentation pathway of Saccharomyces cerevisiae using biochemical systems theory and metabolic control analysis: model definition and nomenclature. Math Biosci 130,25-50.

Fell D. 1997. *Understanding the control of metabolism*. Portland Press, London.

Gennemark P, Wedelin D. 2007. Efficient algorithms for ordinary differential equation model identification of biological systems. IET Syst Biol 1,120-9.

Gennemark P, Wedelin D. 2009. Benchmarks for identification of ordinary differential equations from time series data. Bioinformatics 25,780-6.

Gonzalez OR, Kuper C, Jung K, Naval PC, Jr., Mendoza E. 2007. Parameter estimation using Simulated Annealing for S-system models of biochemical networks. Bioinformatics 23,480-6.

Heijnen JJ. 2005. Approximative kinetic formats used in metabolic network modeling. Biotechnol Bioeng 91,534-45.

Hooke R, Jeeves T. 1961. 'Direct search' solution of numerical and statistical problems. J Assoc Comput Mach 8,212-229.

Kikuchi S, Tominaga D, Arita M, Takahashi K, Tomita M. 2003. Dynamic modeling of genetic networks using genetic algorithm and S-system. Bioinformatics 19,643-50.

Kimura S, Ide K, Kashihara A, Kano M, Hatakeyama M, Masui R, Nakagawa N, Yokoyama S, Kuramitsu S, Konagaya A. 2005. Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm. Bioinformatics 21,1154-63.

Kirkpatrick S, Gelatt CD, Vecchi MP. 1983. Optimization by Simulated Annealing. Science 220,671-680.

Kutalik Z, Tucker W, Moulton V. 2007. S-system parameter estimation for noisy metabolic profiles using newton-flow analysis. IET Syst Biol 1,174-80.

Levenberg K. 1944. A method for the solution of certain nonlinear problems in least squares. Quarterly Journal of Applied Mathmatics II,164-168.

Liu PK, Wang FS. 2008. Inference of biochemical network models in S-system using multiobjective optimization approach. Bioinformatics 24,1085-92.

Marquardt D. 1963. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. SIAM Journal on Applied Mathematics 11,431–441.

Mendes P, Kell D. 1998. Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. Bioinformatics 14,869-83.

Moles CG, Mendes P, Banga JR. 2003. Parameter estimation in biochemical pathways: a comparison of global optimization methods. Genome Res 13,2467-74.

Nelder J, Mead R. 1965. A simplex method for function minimization. Comput J 7,308-313.

Polisetty PK, Gatzke EP, Voit EO. 2008. Yield optimization of regulated metabolic systems using deterministic branch-and-reduce methods. Biotechnol Bioeng 99,1154-69.

Polisetty PK, Voit EO, Gatzke EP. 2006. Identification of metabolic system parameters using global optimization methods. Theor Biol Med Model 3,4.

Sakawa M. 1993. *Fuzzy sets and interactive multiobjective optimization*. Plenum, New York.

Savageau MA. 1976. *Biochemical systems analysis : a study of function and design in molecular biology*. Addison-Wesley Pub. Co., Advanced Book Program, Reading, Mass.

Seber GAF, Wild CJ. 2003. *Nonlinear regression*. Wiley-Interscience, Hoboken, N.J.

Sorribas A, Curto R, Cascante M. 1995. Comparative characterization of the fermentation pathway of Saccharomyces cerevisiae using biochemical systems theory and metabolic control analysis: model validation and dynamic behavior. Math Biosci 130,71-84.

Sorribas A, Hernandez-Bermejo B, Vilaprinyo E, Alves R. 2007. Cooperativity and saturation in biochemical networks: a saturable formalism using Taylor series approximations. Biotechnol Bioeng 97,1259-77.

Tsai KY, Wang FS. 2005. Evolutionary optimization with data collocation for reverse engineering of biological networks. Bioinformatics 21,1180-8.

Tucker W, Moulton V. 2006. Parameter Reconstruction for Biochemical Networks Using Interval Analysis. Reliable Computing 12,389-402.

Vilela M, Borges CC, Vinga S, Vasconcelos AT, Santos H, Voit EO, Almeida JS. 2007. Automated smoother for the numerical decoupling of dynamics models. BMC Bioinformatics 8,305.

Vilela M, Chou IC, Vinga S, Vasconcelos ATR, Voit EO, Almeida JS. 2008. Parameter optimization in S-system models. BMC Systems Biology 2,-.

Vilela M, Vinga S, Grivet Mattoso Maia MA, Voit EO, Almeida JS. 2009. Identification of neutral biochemical network models from time series data. BMC Syst Biol 3,47.

Voit EO. 2000. *Computational analysis of biochemical systems : a practical guide for biochemists and molecular biologists*. Cambridge University Press, New York.

Voit EO, Almeida J. 2004. Decoupling dynamical systems for pathway identification from metabolic profiles. Bioinformatics 20,1670-81.

Voit EO, Savageau MA. 1982. Power-Law Approach to Modeling Biological-Systems .3. Methods of Analysis. Journal of Fermentation Technology 60,233-241.

Voit EO, Savageau MA. 1987. Accuracy of alternative representations for integrated biochemical systems. Biochemistry 26,6869-80.

Wang F, Sheu J. 2000. Multiobjective parameter estimation problems of fermentation processes using a high ethanol tolerance yeast. Chem Eng Sci 55,3685-3695.