Journal of Biotechnology xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Journal of Biotechnology

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

Automated piecewise power-law modeling of biological systems

Anna Machina[a,b,1,2], Arkady Ponosov[b,2], Eberhard O. Voit[a,*]

[a] Integrative BioSystems Institute and Wallace H. Coulter Department of Biomedical Engineering, Georgia Institute of Technology, 313 Ferst Drive, Suite 4103, Atlanta, GA 30332-0535, United States
[b] Centre for Integrative Genetics (CIGENE), CIGENE, Arboretveien 6, 1432 Ås, Norway

ARTICLE INFO

Article history:
Received 20 August 2009
Received in revised form
16 December 2009
Accepted 21 December 2009
Available online xxx

Keywords:
Biochemical Systems Theory
Canonical model
Differential inclusion
Inverse problem
Machine learning
Parameter estimation
Piecewise approximation
Power-law

ABSTRACT

Recent trends suggest that future biotechnology will increasingly rely on mathematical models of the biological systems under investigation. In particular, metabolic engineering will make wider use of metabolic pathway models in stoichiometric or fully kinetic format. A significant obstacle to the use of pathway models is the identification of suitable process descriptions and their parameters. We recently showed that, at least under favorable conditions, Dynamic Flux Estimation (DFE) permits the numerical characterization of fluxes from sets of metabolic time series data. However, DFE does not prescribe how to convert these numerical results into functional representations. In some cases, Michaelis–Menten rate laws or canonical formats are well suited, in which case the estimation of parameter values is easy. However, in other cases, appropriate functional forms are not evident, and exhaustive searches among all possible candidate models are not feasible. We show here how piecewise power-law functions of one or more variables offer an effective default solution for the almost unbiased representation of uni- and multivariate time series data. The results of an automated algorithm for their determination are piecewise power-law fits, whose accuracy is only limited by the available data. The individual power-law pieces may lead to discontinuities at break points or boundaries between sub-domains. In many practical applications, these boundary gaps do not cause problems. Potential smoothing techniques, based on differential inclusions and Filippov's theory, are discussed in Appendix A.

1. Introduction

Enormous advances in molecular biology, engineering and computer science have propelled the computational analysis of biological systems into the center of systems biology, and it is becoming feasible to use systems biological methods like pathway analysis in biotechnology and, especially, metabolic engineering. The main challenge for practical applications of computational systems biology is currently the process of making the computational models fit the dynamic responses of biological systems. This challenge consists of two related sub-tasks. The first is the determination of suitable functions that are capable of describing the observed dynamics with sufficient accuracy, and the second is the identification of parameter values that render a good numerical fit between observed and modeled responses. Although these two sub-tasks are clearly dependent on each other, they are different in character. While the second task of parameter identification is primarily a technical issue, which may be difficult, but which one might expect to be solvable with brute force, if not with elegance, the task of determining appropriate functions exceeds the realm of pure computation and requires insight into the biological phenomenon under investigation, numerous assumptions and simplifications, which cannot always be validated, and some degree of ingenuity in finding or inventing functional forms that fit the need. The reason that this sub-task is complicated is that nature has not provided us with guidelines for selecting these functions, and that it is usually not feasible to reduce biological processes to elemental physical processes for which mathematical descriptions are available (Voit, 2008).

Many models of biochemical systems in the past used functional forms that were chosen from a default repertoire of candidates, which included Michaelis–Menten and Hill rate laws, their generalizations toward several substrates, inhibitors and other modulators (Schulz, 1994), or so-called *canonical* representations like power-law (Voit, 1991) and lin-log functions (Heijnen, 2005). However, it is becoming increasingly evident that the standard rate functions are not always applicable and that they, like canonical representations, are simply too inaccurate for reliable extrapolations of the model to new experimental conditions.

In an attempt to respond to these challenges, we recently proposed Dynamic Flux Estimation as a useful tool for estimating

* Corresponding author. Tel.: +1 404 385 5057; fax: +1 404 894 4243.
E-mail addresses: anna.machina@umb.no (A. Machina), arkadi@umb.no (A. Ponosov), eberhard.voit@bme.gatech.edu, voiteo@musc.edu (E.O. Voit).
[1] Tel.: +1 404 385 5057; fax: +1 404 894 4243.
[2] Tel.: +47 64 96 52 97; fax: +47 64 96 51 01.

0168-1656/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.jbiotec.2009.12.016

metabolic pathway systems (Goel et al., 2008). This method consists of two phases, of which the first is entirely model free, while the second requires the choice of a mathematical representation of all involved fluxes. More specifically, the first phase consists of solving a linear system of fluxes in such a manner that the change in each variable pool at each time point is the numerical result of fluxes entering and exiting the pool. Under ideal conditions, this solution can be obtained with straightforward methods of linear algebra; in other cases, additional biological information, complementing the observed metabolic time series, is needed (Voit et al., 2009). In either case, the solution of the first phase of DFE consists of a set of points characterizing the dynamic profile of each flux over the observed time interval. This solution allows inspection of each flux either plotted against time or against its contributing metabolites and modulators. However, since the representation is not functional, it does not allow simulations of new scenarios or extrapolations to untested conditions.

The key goal of the second phase is the conversion of these numerical representations into functional forms, which subsequently permit simulations, extrapolations, and other manipulations and analyses with the model. If the numerical representation of a flux exhibits a simple trend, a function like a Michaelis–Menten or Hill rate law, or even a simple power-law representation, may serve as an adequate model. Indeed, the arsenal of traditional biochemical process descriptions is essentially unlimited (Schulz, 1994), yet it may still be insufficient. For instance, Peskov et al. (2008) recently proposed a representation of the phosphofructokinase reaction in *Escherichia coli* that by far exceeds the complexity of any traditional rate laws. In cases where the typical candidates of biochemistry fail, there is no guidance as to what other functional forms might yield satisfactory flux fits. Without an evident functional format, one might try to exhaust a set of more complex candidate functions, but it is clear that it is logistically difficult and computationally very expensive to execute such a search.

As an alternative with minimal bias, we propose here the use of univariate or multivariate piecewise power-law functions. It is known that such functions with sufficiently many pieces are capable of modeling unknown data trends with arbitrary accuracy. It is also known that even single power-law functions are often reasonable descriptions of biological processes *in vivo*, in most cases outperforming linear and other simple functions. Thus, the search for a moderately small number of pieces, each described with a power-law function, appears to be quite natural. One might wonder whether this inference of piecewise representations really qualifies as model identification. Two arguments seem to provide affirmative answers. First, if the piecewise representation is sufficiently accurate, it may serve as an extrapolation tool that is likely much better than a single-pieced representation. Furthermore, once suitable pieces have been identified, it might at least in principle be possible to reverse-engineer a single function that provides a collective, smooth representation of all individual pieces and, thus, the data describing the process.

If the numerical flux representation depends on only one variable, the determination of suitable breakpoints and fitting intervals may be accomplished by inspection. However, this is no longer feasible if the flux depends on multiple variables. In the following we describe an algorithm, adapted from a method for piecewise linear systems analysis, that automatically dissects the flux-variable space into suitable segments within which the flux is represented with a product of power-law functions. The algorithm works well for fluxes that depend on a single variable or on a larger number of variables. It allows the specification of the number of desired segments or iteratively increases this number until an acceptable residual error is reached. The resulting piecewise representation is a direct generalization of models within the widely used modeling framework of Biochemical Systems Theory (Savageau, 1976; Voit, 2000) and introduces minimal bias due to the choice of functional forms.

2. Problem statement

We consider a system of the form

$$\dot{x}_i = V_i^+(x_1, x_2, \ldots, x_{n+m}) - V_i^-(x_1, x_2, \ldots, x_{n+m}), \quad (1)$$

where index i ($i = 1, \ldots, n$) refers to dependent variables, while higher indices ($n+1, \ldots, n+m$) refer to independent variables. The non-negative influx and efflux functions V_i^+ and V_i^- may possibly consist of sums of other functions describing different contributions to the production or degradation of x_i.

Once the functions V_i^+ and V_i^- are specified, simulations with the model in Eq. (1) are easily performed, and many methods are available for mathematical analyses of features such as local stability or parameter sensitivities at the system's steady states. However, before such analyses are feasible, it is necessary to specify the functions in the model. As discussed in Section 1, this specification consists of two parts, namely the determination of the mathematical structure of the functions and the identification of suitable parameter values.

It is by now widely recognized that the estimation of parameter values from time series data is much simplified if the data are first smoothed, because smoothing permits the computation of slopes at any desired number of points along the time trajectory of any of the system variables (Vilela et al., 2007). The slopes, in turn, can be used as a set of substitutes for the differentials on the left-hand side of each differential equation at k time points ($S_i(t_k) \approx \dot{x}_i(t_k)$) (Voit and Savageau, 1982; Varah, 1982). This procedure has two significant advantages. First, the differential equations no longer have to be integrated numerically, which usually requires substantial computational effort, often consuming between 95% and close to 100% of the entire estimation time (Voit and Almeida, 2004). Second, the slope substitution in effect decouples the system of differential equations and permits parameter estimation one equation at a time, thus facilitating simpler sequential or parallel execution. Many applications of these methods have been analyzed in the fields of genomics and metabolic pathway analysis (Chou and Voit, 2009). The latter is particularly well suited because pathways not only have a well-defined connectivity but also entail conservation of mass at nodes and within the entire system. Most comments in the following therefore refer to metabolic pathway systems, even though other biological systems are not categorically excluded.

While the slope substitution and decoupling method has found plenty of applicability (Chou and Voit, 2009) and was shown to be statistically sound (Brunel, 2008), it still requires in most cases the estimation of parameters in sets of nonlinear (algebraic) equations. This estimation is not always trivial. Indeed it is common that evolutionary or regression algorithms are unable to find the global optimum within a reasonable time, while global estimation methods, such as branch-and-bound methods, are computationally expensive and sometimes difficult to implement (Guillen-Gosalbez and Sorribas, 2009; Polisetty et al., 2006). A second common issue is that entire ensembles of solutions may model the data with similar residual errors, thus causing model identification problems (Gutenkunst et al., 2007a,b; Raue et al., 2009). These ensembles may form more or less elliptical regions whose center is the optimal solution, but it is also possible that distinctly different solution sets are indistinguishable with respect to the residual error. Finally, a less-recognized issue is the compensation of errors among different terms within an equation, among sets of equations, or even within a single term (Raue et al., 2009; Goel, 2009). In the simplest case, two parameters p and q may always appear in the same combination, such as p/q, which precludes their individual identification. In other

Please cite this article in press as: Machina, A., et al., Automated piecewise power-law modeling of biological systems. J. Biotechnol. (2010), doi:10.1016/j.jbiotec.2009.12.016

cases, conserved quantities are much more complex and difficult to detect and characterize. Error compensation is a hideous problem: the fit to one or a few training datasets may be quite good, because a badly estimated parameter in one term is compensated by an incorrect parameter in a different term. However, if the resulting model is used for extrapolations to new conditions, the compensation may no longer hold and the model fit becomes unacceptable.

Dynamic Flux Estimation (DFE) addresses this issue of error compensation (Goel, 2009). DFE begins with a model-free estimation phase, where a linear system of fluxes is constructed at each time point. These fluxes correspond to the functions affecting the dynamics of each variable (as in Eq. (1)), but only their values are taken at one point at a time, and a functional specification is not required. Thus, at time point t_k, equations of the following type are constructed:

$$\dot{x}_i(t_k) = Influx_1(t_k) + Influx_2(t_k) + \cdots - Efflux_1(t_k) - Efflux_2(t_k) - \cdots. \quad (2)$$

Under ideal conditions (see Goel et al., 2008), the collection of all these linear equations for all variables can be solved directly with method of linear algebra, and the result is a set of points characterizing the dynamic profiles of all fluxes in the system. In other words, one obtains model-free representations of all fluxes as discrete time series. For example, if $Efflux_2$ in Eq. (2) depends on variables x_3 and x_6 and if we rename this flux for simplicity as V, one obtains a dataset of the form $\{x_3(t_k), x_6(t_k), V(t_k);\ k = 1, \ldots, K\}$. From this result, one obtains an impression of V as a function of time and/or as a function of x_3 and x_6.

The second phase of DFE is model based. Here, the task is to assign functional forms to the flux profiles that were obtained in the first phase. Ideally, such a function should perfectly match the plot of the flux versus time and also versus its alleged substrates and modulators. If a suitable functional form is known or can be assumed with some justification, this matching step is easily accomplished with a nonlinear regression algorithm. However, if no such form is known, it is unclear how to proceed.

A possible default candidate is a power-law representation of the form

$$V_i = \gamma_i \prod_{j=1}^{n} x_j^{f_{ij}}, \quad (3)$$

where the rate constant γ_i is non-negative and the kinetic orders f_{ij} are real numbers. These representations were proposed as useful approximations for metabolic systems 40 years ago and have been the method of choice in many applications. They form the basis of a modeling framework that is now widely recognized as *Biochemical Systems Theory* (BST; Voit, 1991, 2000; Savageau, 1969a,b, 1970, 1976; Torres and Voit, 2002). Power-law representations are intriguing because they are nonlinear and cover an enormously rich repertoire of functional responses when they are embedded in differential equations (Savageau and Voit, 1987), yet individually permit a simple logarithmic transformation to linearity. It was clear from the beginning of BST and its roots in Taylor approximation theory that power-law functions are local representations that are useful for modeling the dynamics of variables operating within a moderate range about their normal states, but that they may incur unacceptable errors if one tries to fit complex functional shapes that extend over wide ranges of variation in the involved metabolites and modulators.

Because power-law functions can be seen as linearizations in logarithmic space, their range of applicability may be extended in two ways. First, higher order terms could be considered in order to improve the accuracy of approximation (Cascante et al., 1991). However, while theoretically reasonable, this strategy leads to convoluted representations that are rather inconvenient for further analyses and modeling purposes. As an alternative, one may consider piecewise representations. For univariate functions, it is relatively easy to determine such representations, and they have been used, for instance, in the analysis of gene circuits (Savageau, 2001, 2002). However, for multivariate functions, the suitable determination of a piecewise power-law representation becomes difficult. In the following, we present an algorithm that automatically identifies appropriate piecewise power-law representations.

Thus, our goal is the following: given metabolic time series data corresponding to an unknown, univariate or multivariate function, compute a piecewise power-law representation that is suitable in a sense that it fits the data within an acceptable error while consisting of the smallest possible number of pieces. The problem of a piecewise power-law regression, as posed here, is not straightforward, because, given a set of samples of a flux V, which is possibly affected by noise, one has to compute both a partitioning into regions in the space of metabolites and the behavior of the unknown function V over each of them. Because of the "mixed nature" of the problem, classical approximation techniques cannot be directly applied, and it is instead necessary to develop a specific, customized method. The principles of such a method were introduced a few years ago for linear systems, where the regions in the space of variables are polyhedra and the behavior of the function V in each region is approximated by a linear expression (Ferrari-Trecate and Muselli, 2002; Ferrari-Trecate et al., 2001a).

3. Methods

3.1. An algorithm for piecewise linear regression

Ferrari-Trecate and Muselli (2002) and Ferrari-Trecate et al. (2001a) introduced a machine learning method for piecewise linear regression based on Artificial Neural Networks (ANNs). The solution of such a machine learning problem involves the reconstruction of an unknown function W: $X \to Y$ from a finite set M of samples of W (the so-called *training set*), which is possibly corrupted by noise. Upon training, the result is tested against a set of data that was not used during training and is typically called the *validation set*. ANNs may be applied to two groups of such problems according to the range of values assumed by the output Y. For Boolean-like output (0 or 1; on or off) or for output with a limited number of elements, one speaks of a classification problem, while output coded by a continuous variable requires the solution of a regression task. The identification of an optimal partition that is of interest in (Ferrari-Trecate and Muselli, 2002; Ferrari-Trecate et al., 2001a), as well as here, lies right at the border between classification and regression, because the input space X has to be subdivided into a small number of disjoint regions and the behavior of the unknown, continuous function W over each of these regions has to be generated. Thus, the method proposed in Ferrari-Trecate and Muselli (2002) and Ferrari-Trecate et al. (2001a) combines local estimation, clustering in weight space, multi-categorical classification, and linear regression, which we will substitute by power-law regression. The original method may be summarized as follows.

Let X be a polyhedron in the n-dimensional space $\Re_+^n \setminus \{0\}$ and let $\{X_s\}_{s=1}^S$ be a polyhedral partition of X, *i.e.*, $X_i \cap X_j = \emptyset$ for every i, $j = 1, \ldots, S$ and $\bigcup_{i=1}^S X_i = X$. The target of piecewise linear regression is to reconstruct an unknown function f: $X \to \Re_+$ assuming linear behavior in each region X_s:

$$f(x) = z_s = w_{s0} + \sum_{j=1}^{n} w_{sj} x_j, \quad (4)$$

when only a training set M containing m samples (x_k, y_k), $k = 1, \ldots, m$ is available. The output y_k gives an evaluation of $f(x_k)$ subject to noise in $x_k \in X$, and the region X_s to which x_k belongs is not known in

Please cite this article in press as: Machina, A., et al., Automated piecewise power-law modeling of biological systems. J. Biotechnol. (2010), doi:10.1016/j.jbiotec.2009.12.016

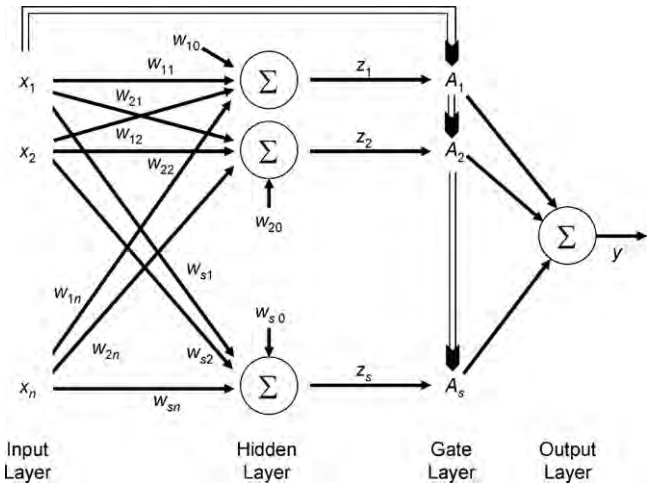

Fig. 1. Diagram of a neural network realizing a piecewise linear function (adapted from Ferrari-Trecate and Muselli, 2002; Ferrari-Trecate et al., 2001a).

advance. Scalar weights $w_{s0}, w_{s1}, \ldots, w_{sn}$, for $s = 1, \ldots, S$, uniquely characterize the function f, and their estimation is the target of the piecewise linear regression problem. For notational purposes, the weights are collected in a vector w_s. Since the regions X_s are polyhedral, they can be defined by a set of l_s linear inequalities of the type:

$$a_{sj0} + \sum_{k=1}^{n} a_{sjk} x_k \leq 0. \quad (5)$$

The scalars a_{sjk}, for $j = 1, \ldots, l_s$ and $k = 0, 1, \ldots, n$, can be collected in a matrix A_s whose estimation thus becomes the target of the reconstruction process for every $s = 1, \ldots, S$. Discontinuities are allowed and indeed are often present at the boundaries between two regions X_s.

Following (Ferrari-Trecate and Muselli, 2002; Ferrari-Trecate et al., 2001a), a neural network realizing a piecewise linear function f of this kind can be modeled as shown in Fig. 1.

As previously noted, the solution of the piecewise linear regression requires a combination of classification and regression: the classification aims at finding matrices A_s to be inserted into the gate layer of the neural network, whereas the latter provides the weight vector w_s as input to the hidden layer connections (Fig. 1).

Ferrari-Trecate and Muselli (2002) and Ferrari-Trecate et al. (2001a) solved the problem in four steps that correspond to specific tasks and are outlined below.

1. Local regression
   For every $k = 1, \ldots, m$ do the following:
   1a. Form the set C_k containing the pair (x_k, y_k) and the samples $(x, y) \in M$ associated with $c - 1$ nearest neighbors x to x_k.
   1b. Perform linear regression to obtain the weight vector v_k of a linear unit fitting the samples in C_k.
2. Clustering
   Perform a clustering process in the space $\Re^{n+1}$ to subdivide the set of weight vectors v_k into S groups V_s.
3. Classification
   Build a new training set M' containing the m pairs (x_k, s_k), where V_{s_k} is the cluster containing v_k. Train a multi-categorical classification method to produce the matrices A_s for the regions X_s.
4. Regression
   For every $s = 1, \ldots, S$ perform linear regression on the samples $(x, y) \in M$ with $x \in X_s$ to obtain the weight vector w_s for the s^{th} unit in the hidden layer.

In the following, we are interested in power-law functions rather than linear functions and, correspondingly, use either linear regression in log space or power-law regression. In order to make the necessary alterations to the linear method as clear as possible, Table 1 shows, side by side, the steps in the different procedures.

In broad strokes, the linear and power-law methods are structured as follows (excluding self-evident log-transformations):

1. Associate to each data point a local linear model (Step 1);
2. Aggregate local models with similar features into clusters (Step 2);
3. Classify data points corresponding to local models in the same cluster and estimate the regions (Step 3);
4. Estimate the parameter vectors (Step 4).

Each step faces distinct challenges and assumptions, which may be summarized as follows:

1. Step 1 requires the specification of the number of neighboring points c, which is needed for each local regression. The bigger the number c, the bigger is the number of mixed points (*i.e.*, points providing spurious information about the true model). For this reason, one should like to keep c as low as possible. At the same time, one needs to choose c large enough to counteract the effects of noise on the accuracy of the local models. Thus, the choice of c is a trade-off.
2. The goal of Step 2 is to determine S clusters. This is accomplished via *clustering algorithms*, that require S as an input. In some cases, the number of desired clusters is given *a priori*. However, if S is not known, it may theoretically be estimated from the dataset. According to Ferrari-Trecate et al. (2003), this can be done by adopting clustering algorithms where the number of clusters is not fixed *a priori* but automatically estimated (Fritzke, 1997).
3. At Step 3 each cluster is expected to collect all local models with similar features, data points are classified, and sub-model datasets are built accordingly. The regions $\{X_s\}_{s=1}^{S}$ can be found by resorting to *pattern-recognition* algorithms. The main disadvantage is that the fastest pattern-recognition algorithms may leave gaps in the input domain X, when the dimension of X is greater than one. More precise algorithms that do not leave holes are available, but they are considerably slower and require more memory. Specifically, for the identification of a small number of pieces with a large number of data points, the fast Proximal Support Vector Classification (PSVC) algorithm is recommended, and results are obtained within seconds or minutes on a standard PC. However, if the dimension of the problem is greater than one, it is not guaranteed that the union of regions will cover the input domain. To avoid gaps, one may use Multi-category Robust Linear Programming (MRLP), if access to professional Linear Programming (LP) and Quadratic Programming (QP) solvers like CPLEX is available.
4. Conceptually, this is the easiest step. The data points in each region $\{X_s\}_{s=1}^{S}$ can be used for estimating the parameter vectors of the linear (or, respectively, power-law) regression. It is apparent that X_s must contain enough data points to perform both the local regressions and the estimation of scalar parameters composing a parameter vector. If a cluster contains few points, it is discarded and the number of modes is reduced.

3.2. Piecewise Power-Law Regression

Given the general strategy in the previous section and in Table 1, it is now relatively straightforward to adapt the original linear method to power-law models. This adaptation begins with a logarithmic transformation of the data. In logarithmic coordinates,

Please cite this article in press as: Machina, A., et al., Automated piecewise power-law modeling of biological systems. J. Biotechnol. (2010), doi:10.1016/j.jbiotec.2009.12.016

Table 1
Steps toward piecewise linear and piecewise power-law models.

Method	Original linear method	Piecewise power-law model based on linear regression	Piecewise power-law model based on power-law regression
		Logarithmic transformation	
		Local regression (Step 1) Clustering (Step 2) Classification (Step 3)	
	Linear regression (Step 4)	Linear regression	Inverse-logarithmic transformation
		Inverse-logarithmic transformation	Power-law regression
Key features	Output is a piecewise linear approximation to a function. The sum of squared errors is E1.	Output is a piecewise power-law approximation with the sum of squared errors E2 < E1 (provided the unknown function is nonlinear).	Output is a piecewise power-law approximation with the sum of squared errors E3 < E2 < E1 (provided the unknown function is nonlinear).

the piecewise power-law function becomes piecewise linear, and we therefore implement the first three steps of local regression, clustering, and classification of the above algorithm with the log-transformed data. This transition allows us to obtain a polyhedral partition of the input domain in logarithmic space. Implementation of Step 4 (linear regression) yields the weight parameters in Eq. (4) for the data in logarithmic coordinates. These parameters are the rate constant and the kinetic orders of the target representation in piecewise power-law format. When we apply the inverse-logarithmic transformation, the polyhedral partition that is produced by the algorithm for logarithmic data becomes a partition of the initial input domain that is bounded by nonlinear surfaces given by

$$a_{sj0} + \sum_{k=1}^{n} a_{sjk} \ln x_k \leq 0, \tag{6}$$

where the coefficients a_{sjk} ($k = 0, 1, \ldots, n$; $s = 1, \ldots, S$) were obtained in the classification Step 3 of the algorithm. In this manner we obtain a piecewise power-law approximation to an unknown function, where sub-domains of the partition are defined by Eq. (6).

The procedure has all the advantages of linear regression. However, because the logarithmic transformation distorts the error structure of the problem, the least squares error for the resulting piecewise power-law is in general less accurate than the corresponding error obtained by power-law regression of the original data. As a partial remedy, it is therefore advantageous to modify Step 4 of the algorithm by applying a power-law regression to the original data over each of the S regions in Eq. (6). Even though the regression is now nonlinear, the increase in computational difficulty is modest, due to the partition in small pieces. In other words, the partition of the initial domain localizes the process of approximation to sub-domains, thereby reducing the risk of failure of the power-law regression. While this modification reduces error distortion to some degree, it does not affect Steps 1-3, in which the boundaries of the domains are determined. As a consequence, the domains are optimal in log space but not necessarily in Cartesian space and, for instance, result in better fits for small values of metabolite concentrations, which become more pronounced in the logarithmic representation. Expressed differently, the method is optimized toward relative, rather than absolute errors, which in many cases in biology is actually an advantage. We will see the consequences of this distortion in the first example of Section 4. The same example demonstrates that the distortion becomes less significant as the algorithm uses higher numbers of segments.

The modified algorithm thus proceeds as follows. Let X be the image of X in the n-dimensional logarithmic space R^n and let $\{X_s\}_{s=1}^{S}$ be a polyhedral partition of X, *i.e.*, $X_i \cap X_j = \emptyset$ for every $i, j = 1, \ldots, S$ and $\bigcup_{i=1}^{S} X_i = X$. Let M be the image of M in logarithmic space R^n.

1. Logarithmic transformation
   Transform the data points (x_k, y_k) $(k = 1, \ldots, m)$ logarithmically to $x_k = \ln x_k$, $y_k = \ln y_k$.
2. Local regression
   For every $k = 1, \ldots, m$ do the following:
   2a. Form the set C_k containing the pair (x_k, y_k) and the samples $(x, y) \in M$ associated with $c - 1$ nearest neighbors x to x_k.
   2b. Perform a linear regression to obtain the weight vector v_k of a linear unit fitting the samples in C_k.
3. Clustering
   Perform a clustering process in the space R^{n+1} to subdivide the set of weight vectors v_k into S groups V_s.
4. Classification
   Build a new training set M' containing the m pairs (x_k, s_k), where V_{s_k} is the cluster containing v_k. Train a multi-categorical classification method to produce the matrices A_s for the polyhedral regions X_s.
5. Inverse-logarithmic transformation
   Exponentially transform the data back to Cartesian space: $x_k = \exp(x_k)$, $y_k = \exp(y_k)$. Utilize matrices A_s to obtain the partition of the input domain given by Eq. (6).
6. Regression
   For every $s = 1, \ldots, S$ perform power-law regression on the samples $(x, y) \in M$ with $x \in X_s$ to obtain the rate constant and the kinetic orders for each sub-domain given by Eq. (6).

3.3. Implementation as MatLab toolbox

The Hybrid Identification Toolbox (HIT) is a free MatLab toolbox for regression with piecewise linear maps. HIT implements the clustering-based algorithms described in (Ferrari-Trecate et al., 2001b, 2003; Ferrari-Trecate and Muselli, 2003; Ferrari-Trecate and Schinkel, 2003). In addition, HIT provides facilities for plotting and validating the identified models. HIT uses routines of the MPT toolbox (Kvasnica et al., 2004) for handling polytopes and solving Linear Programming (LP) and Quadratic Programming (QP) problems. These toolboxes can be used to implement the first four steps of the modified algorithm, which produce the partition of the input domain. As soon as the partitioning is known, one can perform a power-law regression, using MatLab or other software.

One of the default assumptions of the algorithm is that the number of sub-domains S in the target solution is given. However, this may not always be desirable. For instance, one may want to create a minimal partition that satisfies a maximally acceptable error. If S is unknown, it may be estimated from the dataset. According to

Please cite this article in press as: Machina, A., et al., Automated piecewise power-law modeling of biological systems. J. Biotechnol. (2010), doi:10.1016/j.jbiotec.2009.12.016

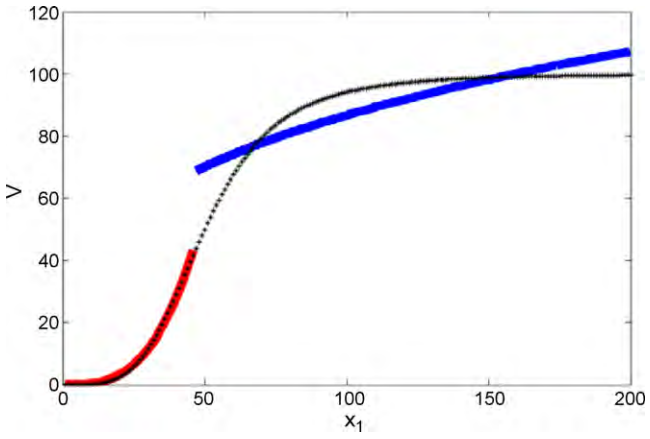

Fig. 2. Two-piece power-law fit to the Hill function in Eq. (7). See text for explanations.

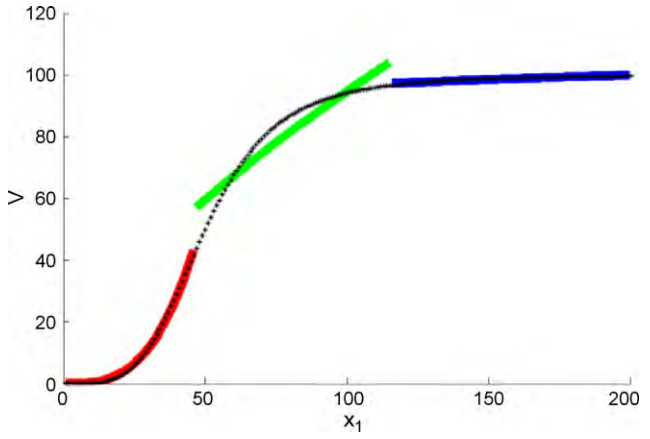

Fig. 3. An improved fit is obtained if the interval of higher concentrations from the previous fit (Fig. 2) is automatically split into two. The new breakpoint is $x_1 = 117$ and the residual error is SSE = 43.

Ferrari-Trecate et al. (2003), this can be done by adopting clustering algorithms where the number of clusters is not fixed *a priori* and is automatically estimated (Fritzke, 1997). As a simple alternative, one may run the algorithm with increasing values for S until an acceptable solution is reached. This strategy is quite reasonable because the algorithm is fairly fast on a standard PC. An example is presented in Section 4.

Another parameter to be specified is the number c of nearest neighbors (see 2.a of the modified algorithm). HIT provides facilities for the selection of S and c through cross-validation. For details about tuning the parameter c the reader is referred to Ferrari-Trecate et al. (2003).

4. Results

4.1. Piecewise power-law regression of a univariate function

For reasons of easy illustration and visualization, it is beneficial to study the algorithm for modeling a univariate function. It is known that a single power-law function cannot model S-shaped dynamics well, and we therefore use as a base function the Hill rate law

$$V(x_1) = \frac{V_m x_1^h}{K_m^h + x_1^h} \tag{7}$$

with arbitrarily chosen Hill coefficient $h = 4$, maximal velocity $V_m = 100$, and Michaelis constant $K_m = 50$. The artificial dataset consists of 200 points that are uniformly spread out over the interval [1, 200].

As a first illustration, we specify the number of sub-domains as $S = 2$. The result of the algorithm is a good fit for small values of x_1 and a less appealing fit for large values (Fig. 2). The breakpoint at $x_1 = 47$ is obtained directly by the algorithm, and the sum of squared errors in this case is SSE = 84. Inspection makes it clear that the interval of lower values of V is better represented than the interval containing higher values of V. The imbalance is due to the fact that the partitioning algorithm works on logarithms of the data and therefore on relative errors, which are more pronounced for small concentrations x_1.

One may proceed in two ways. Either, one may specify $S = 3$ and redo the analysis. Or, one may retain the fit for the interval [0, 47] and compute a two-piece approximation of the remaining interval [47, 200]. The result of the latter strategy is a good representation of high values [117, 200] and a reasonable representation of the center interval (Fig. 3). The residual error is now SSE = 43, which roughly corresponds to a 50% error reduction. A third application of the algorithm to this center interval leads to a four-piece representation with a much reduced error of SSE = 14 (Fig. 4).

An obvious question is the minimally required number of data points. While the question itself is important and valid, its answer is surprisingly complicated, because it depends on a number of factors, including the complexity of the function, the complexity in relation to the noise in the data, and the desired smoothness. For instance, the identification of an exponential function without much noise may only require three or four data points, while a damped oscillation or a more complicated function may need dozens or more points, even if the noise level is low. Another factor influencing the minimum number of data points is their spatial distribution. If many data points are clustered within a small domain, they do not convey much information. Similarly, if many time courses are available but represent only a small portion of the possible space of values of the variables, not much is gained from additional, similar data.

As an example, Fig. 5 shows fits to the Hill function (Eq. (7); compare with Figs. 2–4) obtained with 50, 20, and 10 error-free points, respectively.

4.2. Piecewise power-law regression of a multivariate function

As a more complicated example, suppose we had analyzed N time series with K time points each from a metabolic pathway

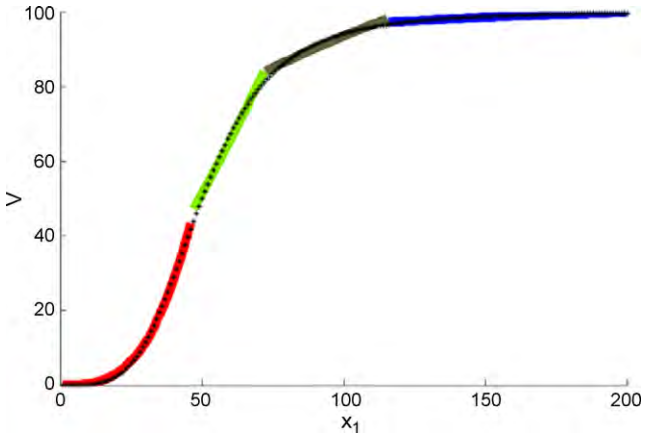

Fig. 4. A much improved fit is obtained if the former center interval (Fig. 3) is further split into two. The new residual error is SSE = 14.

Please cite this article in press as: Machina, A., et al., Automated piecewise power-law modeling of biological systems. J. Biotechnol. (2010), doi:10.1016/j.jbiotec.2009.12.016

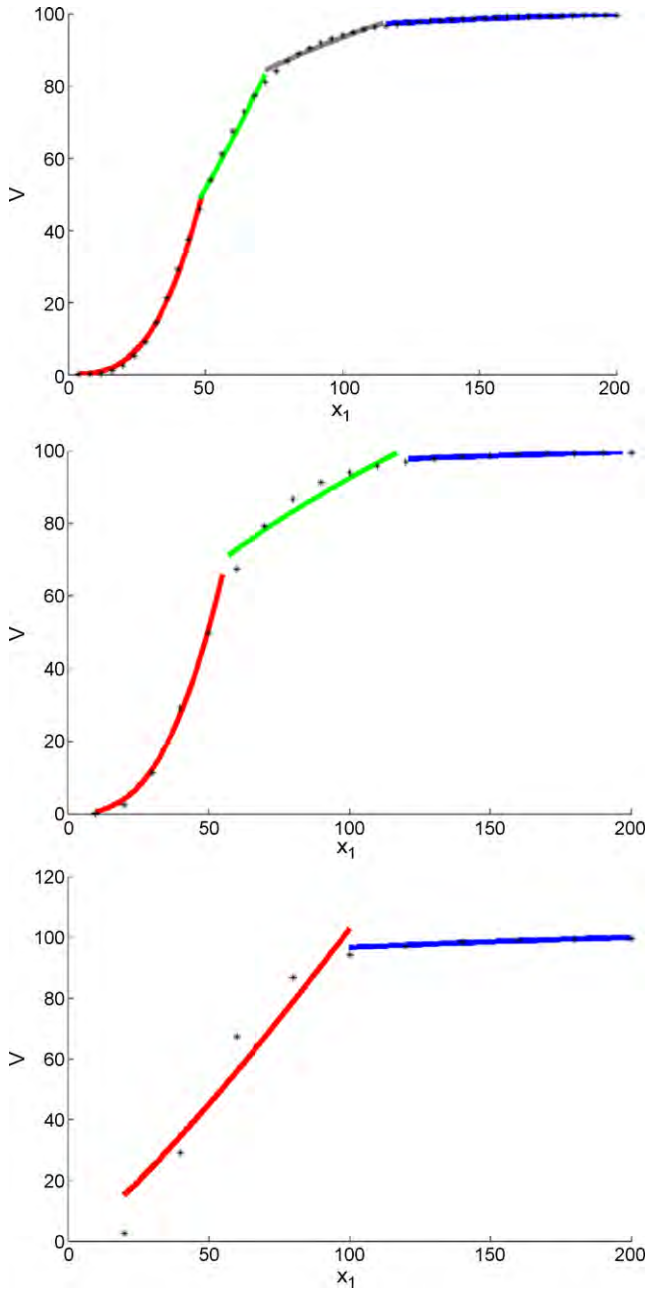

Fig. 5. Piecewise power-law fits to the Hill function in Eq. (7) when smaller numbers of data points are available than in Figs. 2–4. Reducing the number of data points to 50 (top panel) does not alter the earlier results much. If the number of data points is further reduced to 20 (center panel), four pieces can no longer be estimated, and the best option is a representation with three pieces. Finally, only 2 pieces are possible for a dataset containing 10 points (bottom panel).

system involving some number of metabolites. The differences between the N time series may be the result of different initial conditions, such as different amounts of substrate input. Under the action of the pathway system, each metabolite exhibits some time trend, which is affected directly or indirectly by some or all of the other variables. Suppose we had analyzed the data with DFE and that one of the processes in the system, V, was known to depend *directly* only on two of the system variables, x_1 and x_2, which themselves were affected by other variables. The result of the DFE analysis that is pertinent here would then consist of N data sets $\{x_1(t_k), x_2(t_k), V(t_k)\}_n$ $(k = 1, \ldots, K;\ n = 1, \ldots, N)$. The true function V, which we however pretend not to know, is a two-variable

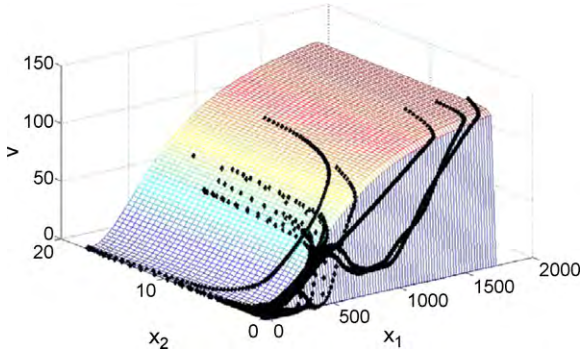

Fig. 6. Different datasets correspond to points that form trend lines lying on the surface given by Eq. (8). The shape and location of the trend lines depend, for instance, on the initial conditions of the pathway model, in which x_1 and x_2 are just two of several metabolites.

Hill function of the form

$$V(x_1, x_2) = \frac{V_m x_1^h x_2^h}{(K_{m1}^h + x_1^h)(K_{m2}^h + x_2^h)} \tag{8}$$

with the arbitrarily chosen Hill coefficient $h = 2$ for both components, a maximal velocity $V_m = 150$, and Michaelis constants $K_{m1} = 600$ and $K_{m2} = 0.1$.

Each dataset corresponds to a single time course and forms a dotted trend line on the three-dimensional surface that is given by Eq. (8) in the space of x_1, x_2, and V (Fig. 6). In reality this surface is unknown. Even with 13 complete datasets, the observed data constitute a rather sparse sample of the surface, demonstrating how difficult the quest for the minimal number of necessary data is (Fig. 6). In our example, the data are noise free, but noise is no hindrance to the partitioning and approximation method and is therefore ignored here for clarity.

The operating domain for the algorithm is specified as $\Omega = [\min(x_1), \max(x_1)] \times [\min(x_2), \max(x_2)]$. Fitting the data with a single two-variable power-law over the entire domain Ω yields an SSE of 425 (results not shown). In order to reduce SSE, we execute the piecewise power-law method for two sub-domains ($S = 2$). The resulting representations in log space and Cartesian space are shown in Figs. 7 and 8. The residual error in this case is SSE = 319

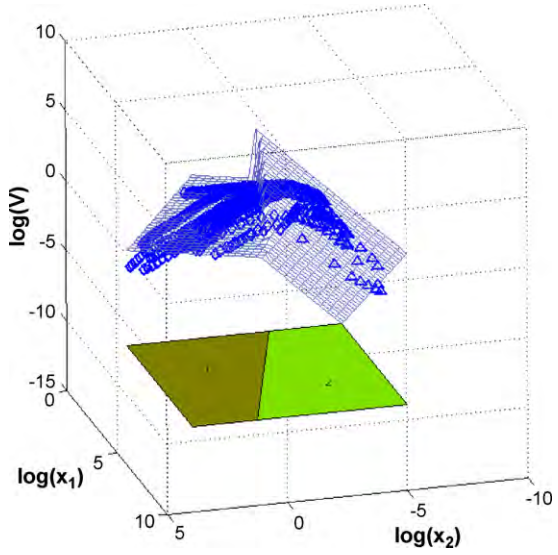

Fig. 7. Two-domain power-law representation for 13 datasets sampling the space (x_1, x_2, V) as trend lines (see text for explanations). The representation shown is in logarithmic coordinates. See Fig. 8 for a Cartesian representation.

Please cite this article in press as: Machina, A., et al., Automated piecewise power-law modeling of biological systems. J. Biotechnol. (2010), doi:10.1016/j.jbiotec.2009.12.016

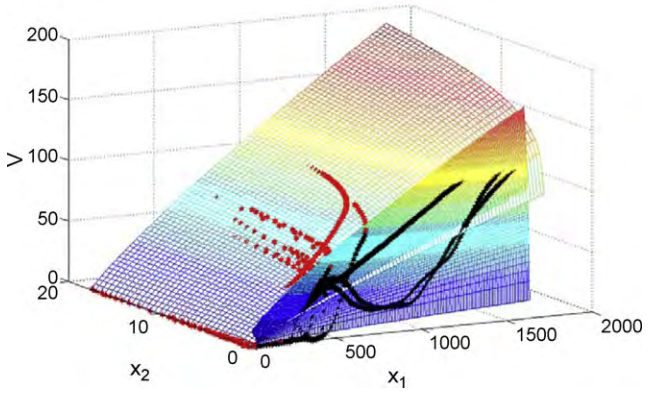

Fig. 8. Two-domain power-law representation for 13 datasets sampling the space (x_1, x_2, V) as trend lines (see text for explanations). In contrast to Fig. 7, the representation here is in Cartesian coordinates.

which corresponds to a reduction of about 25% over the single power-law fit.

As the next illustration we increase the number of sub-domains to $S = 4$. This higher resolution reduces SSE to 176, which corresponds to about 40% of the single-domain fit, but one notes that the algorithm begins to suffer from an insufficient number of data points in some of the sub-domains. The results in log space and Cartesian space are shown in Figs. 9 and 10. One also notes the gaps in representation. These gaps are caused by MatLab's fast Proximal Support Vector Classification pattern-recognition algorithm for reconstructing the regions. If it is necessary to fill these gaps, a slower method such as Multi-category Robust Linear Programming (MRLP) must be employed.

It is not easy to judge the quality of approximation from these global plots. It is therefore useful to show how well individual time courses ($x_1(t_k)$, $x_2(t_k)$, $V(t_k)$) are represented by the four-piece power-law representation. Six examples are given in Fig. 11, where the black symbols represent the original data, while corresponding piecewise approximations are shown in different colors. One can see that the approximation quality depends on the particular dataset as well as the concentrations of x_1 and x_2.

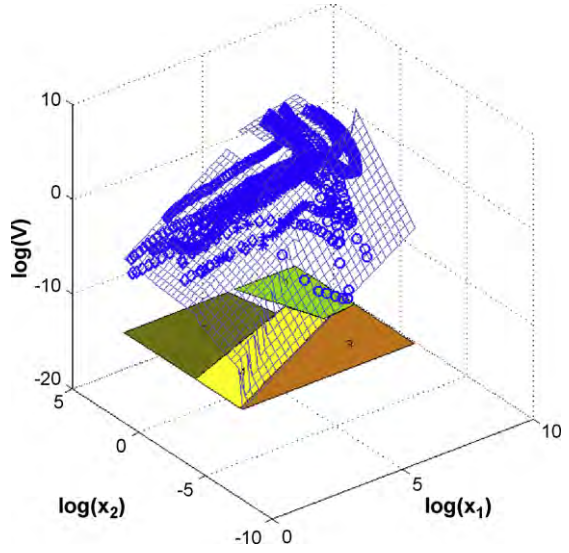

Fig. 9. Four-domain power-law representation for 13 datasets sampling the space (x_1, x_2, V) as trend lines (blue symbols; see text for explanations). The corresponding representation in Cartesian space is shown in Fig. 10. The fit with four pieces is significantly improved over a fit with two pieces (Fig. 7), with the SSE being reduced from 319 to 176.

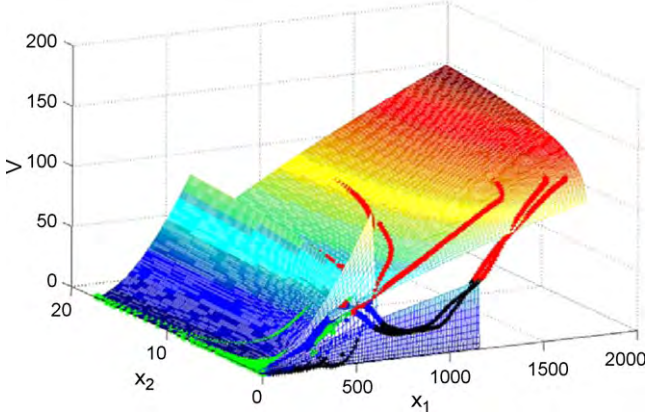

Fig. 10. Four-domain power-law representation for 13 datasets sampling the space (x_1, x_2, V) as trend lines. The corresponding representation in log space is shown in Fig. 9. The fit with four pieces is much better than the corresponding fit with two pieces (Fig. 8).

The algorithm is not limited to two dimensions and can, in principle, be applied to any number of dependent variables. Such examples are difficult to visualize and require rich datasets. Nonetheless, if such data are given, the algorithm performs the classification and piecewise representation quite quickly.

5. Discussion

Until about a decade ago, biomathematical modeling was in some sense easier, because the comparatively poor quality of data allowed for substantial latitude. For instance, even rather crude approximations were often still considered consistent with the modeled data, due to large experimental error bands. Recent developments in molecular and high-throughput biology have changed this situation. It is now feasible to measure comprehensive metabolic time series, sometimes even *in vivo* (Neves et al., 2000), and the resulting data are often so good and plentiful that inferior model descriptions can no longer survive and that it quickly becomes evident when the deviations between model and data are systematic, rather than statistic. Furthermore, methods like Dynamic Flux Estimation (DFE; Goel et al., 2008) reach beyond standard fitting routines by identifying individual flux representations and their dependency not only just on time but also on its contributing metabolites and modulators. While a significant step forward, this result leads to a new challenge, namely the identification of specific functional forms matching the inferred numerical flux profile. Experience in our lab indicates that these profiles are often much more complicated than previously assumed and that simple Michaelis–Menten, Hill, or power-law functions are sometimes, but not always, capable of representing them appropriately (Goel, 2009). This insight, in turn, suggests the need for either an intensified, yet targeted search for suitable local representations of metabolic processes, or an expansion of the simple canonical approximations that in the past were found to be useful defaults.

In this article we have discussed the second of the two options. Specifically, we showed how piecewise power-law representations for unknown functional relationships between fluxes and metabolites can automatically be constructed with a customized adaptation of software that had been proposed for piecewise linear systems analysis (Ferrari-Trecate and Muselli, 2002; Ferrari-Trecate et al., 2001a). This expansion is directly in line with Biochemical Systems Theory (Voit, 1991, 2000; Savageau, 1976; Torres and Voit, 2002) and the next logical step beyond piecewise definitions of single power-law functions (*e.g.*, Savageau, 2001, 2002).

Please cite this article in press as: Machina, A., et al., Automated piecewise power-law modeling of biological systems. J. Biotechnol. (2010), doi:10.1016/j.jbiotec.2009.12.016

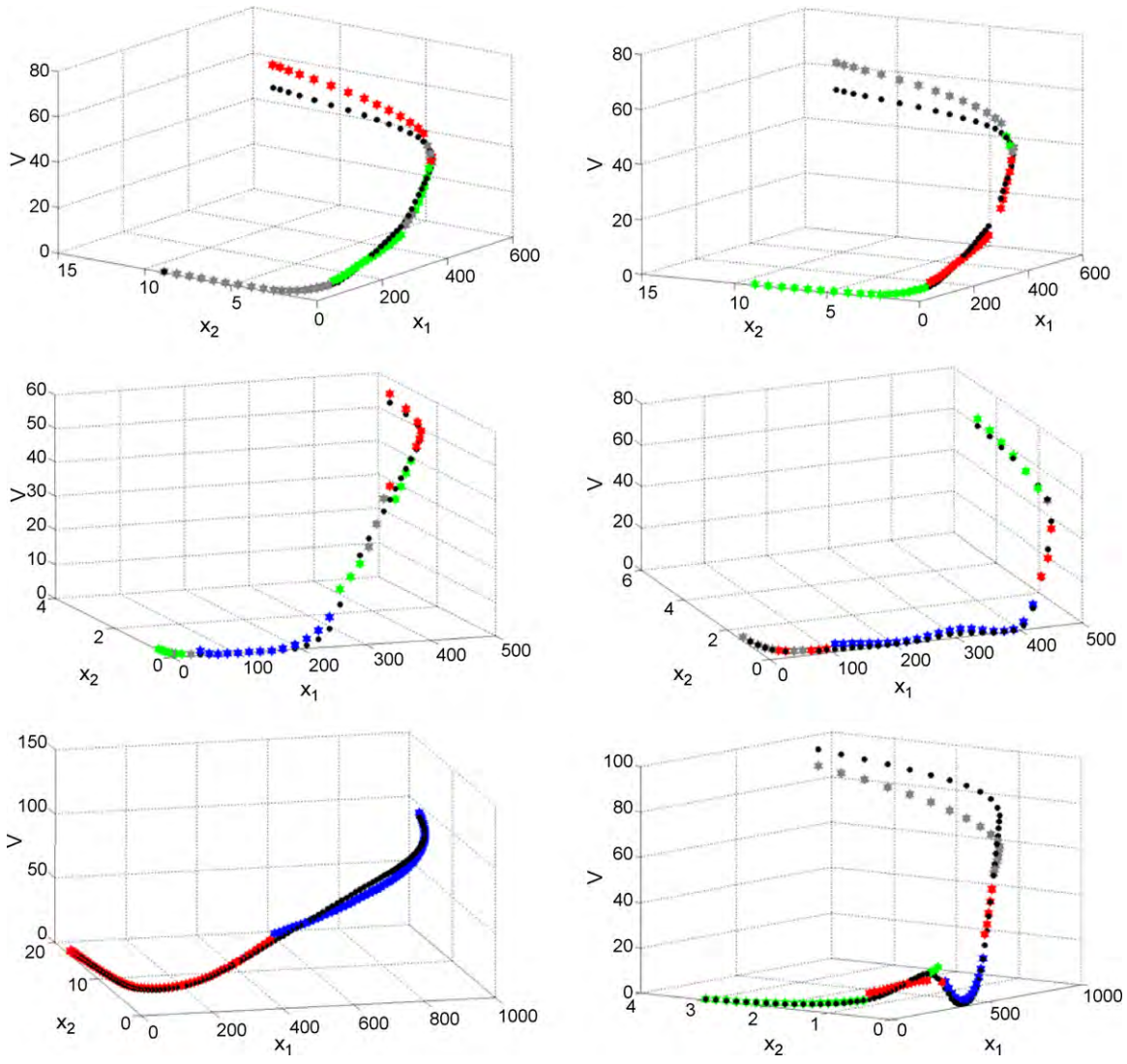

Fig. 11. Comparison of individual time trends in V as a function of its substrates x_1 and x_2. Depending on the particular dataset, the shape of V is quite different, and so is the quality of the piecewise power-law approximations. Error-free data are shown as black symbols and pieces of approximation in different colors.

The proposed algorithm resolves the main difficulty in reconstructing piecewise power-law representations, namely the simultaneous (integer-valued) subdivision of the variable space and the optimized (real-valued) estimation of parameters within each subdivision. Interestingly, this dual task is solved automatically and requires only a few settings of operational parameters, which however is not a problem. The number of clusters, S, can either be predetermined, increased sequentially, or estimated with an adaptive algorithm (Fritzke, 1997), and the number of neighboring points for each local regression (parameter c) can be tuned through cross-validation with a built-in option of the HIT software in Matlab (Ferrari-Trecate et al., 2003).

Theoretically, the proposed software permits arbitrarily accurate representations of univariate or multivariate processes. In reality, however, the method is limited by the number of data points and their representation of the manifold on which they lie. Metabolic time series that correspond to trend lines within a narrow sub-manifold constitute limited samples that may allow the inference of the sub-manifold, but not of the entire manifold, which might be of interest for extrapolation studies. Obviously, the method is also limited by the complexity of the manifold itself. If this manifold is smooth and monotonic, a few sub-domains with their own power-law models might be sufficient. By contrast, a ragged manifold will require many more sub-domains and quickly lead to situations where the data samples are no longer representative and sufficiently comprehensive.

By its nature, the proposed method usually leads to representations with gaps or discontinuities at the boundaries between sub-domains. These discontinuities come in two types. First, the illustrative examples identify gaps in the operating domain. These gaps are entirely due to the choice of a fast pattern-recognition algorithm by the software and can be circumvented with more complex and much slower methods. The second type of discontinuity results from the fact that the algorithm does not require the power-law representations to be continuous on the boundaries of sub-domains. In most practical applications, these discontinuities will be of no major concern. However, if the discontinuities are indeed undesirable, inspection of the collection of pieces might suggest a suitable nonlinear function that could capture the entire range of variation in variables. If so, this candidate function can be reverse-engineered and parameterized from the pieces, and the result is smooth throughout. In this sense, the proposed method is indeed a means for model identification, even if it is somewhat indirect. As an alternative, Appendix A discusses means of addressing

Please cite this article in press as: Machina, A., et al., Automated piecewise power-law modeling of biological systems. J. Biotechnol. (2010), doi:10.1016/j.jbiotec.2009.12.016

this situation based on differential inclusions and Filippov's theory (Filippov, 1985).

While issues such as discontinuities should be investigated further, the proposed piecewise power-law representation, as it is described here, offers a welcome and relatively unbiased alternative to a potentially unlimited search for suitable functions.

Acknowledgments

This work was supported in part by a National Heart, Lung and Blood Institute Proteomics Initiative (Contract N01-HV-28181; D. Knapp, PI), a Molecular and Cellular Biosciences Grant (MCB-0517135; E.O. Voit, PI) from the National Science Foundation, and a Biological Energy Science Center (BESC) grant from the Department of Energy (M. Keller, PI). The first author was partially supported by the Norwegian State Educational Loan Fund (Lånekassen) and by Centre of Integrative Genetics (CIGENE). 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.

Appendix A.

The proposed piecewise power-law regression method naturally leads to discontinuities between neighboring sub-domains. In many cases, these jumps may not cause problems: one simply defines variables within sub-domains and uses one of the boundary values. Nevertheless, the question arises of whether it is possible in principle to define continuous solutions. A general answer is not trivial and requires concepts from Filippov's theory and differential inclusions (Filippov, 1985). These concepts are sketched out in the following.

Let us consider the generic vector equation

$$\dot{x} = f(x) \equiv V^+(x) - V^-(x), \tag{A1}$$

where $x=(x_1, x_2, \ldots, x_n)$ belongs to the open operating domain Ω whose closure $\bar{\Omega}$ is contained in the set $\Re^n_+ = \left\{(x_1, x_2, \ldots, x_n)\right\}, x_j > 0, j = 1, 2, \ldots, n$. Assume that after having performed the piecewise power-law regression we arrived at the following result: the influx (or efflux) function $V^+(x)$ (or $V^-(x)$) is approximated by $V^+ = \gamma^+ \prod_{j=1}^n x_j^{f_{ij}^+}$ if $x \in \Omega_i^+$ (or by $V^- = \gamma^- \prod_{j=1}^n x_j^{f_{ij}^-}$ if $x \in \Omega_i^-$), where the closures $\bar{\Omega}_i^\pm$ of the disjoint open subsets $\Omega_i^\pm \subset \Omega^\pm, i \in I^\pm$ satisfy $\bigcup_{i \in I^\pm} \bar{\Omega}_i^\pm \supset \Omega^\pm$. In other words, the open sets $\Omega_i^\pm$ constitute a partition of the set $\Omega^\pm$, which does not contain gaps as they appeared in Fig. 9; that is, we assume that the regression algorithm automatically removes possible gaps from the partition.

The approximation leads to the following vector equation

$$\dot{x} = \hat{f}(x) \equiv \hat{V}^+(x) - \hat{V}^-(x), \tag{A2}$$

where the functions $\hat{V}^\pm(x)$ are defined as follows

$$\hat{V}^\pm(x) = V_i^\pm(x) \equiv \gamma_i^\pm \prod_{j=1}^n x_j^{f_{ij}^\pm} \quad \text{if } x = (x_1, x_2, \ldots, x_n) \in \Omega_i^\pm, \ \ i \in I^\pm.$$

This representation simply formalizes the fact that we merged different pieces of the functions $\hat{V}^\pm(x)$ that are defined differently over different operating sub-domains. As any of these pieces is an output of the automatically performed regression procedure, the functions $\hat{V}^\pm(x)$ may be discontinuous on the borders of the operating sub-domains, *i.e.*, on the sets $\Omega^\pm \backslash \bigcup_{i \in I^\pm} \Omega_i^\pm$, respectively. One also notes that the regression does not guarantee that the partitions $\Omega_i^\pm, i \in I^\pm$, coincide. This implies that the discontinuity set of the function $\hat{f}(x)$ in (A2) may be as big as the union of the respective discontinuity sets of the functions $\hat{V}^\pm(x)$. Hence the solutions of the vector equation (A2) are only well defined in any of the open subsets $\Omega_i^+ \cap \Omega_k^-$ provided that this subset is non-empty. More precisely, we are able to define mathematically and find (theoretically or numerically) a piece $x_{ij}(t)$ of the entire solution of the vector equation (A2) for time points t, where $x_{ik}(t) \in \Omega_i^+ \cap \Omega_k^-$. In this case, the solution satisfies the equation having the following power-law representation:

$$\dot{x} = \gamma_i^+ \prod_{j=1}^n x_j^{f_{ij}^+} - \gamma_k^- \prod_{j=1}^n x_j^{f_{kj}^-}. \tag{A3}$$

This representation is valid only for separate pieces of the entire trajectory, namely for those satisfying $x_{ik}(t) \in \Omega_i^+ \cap \Omega_k^-$.

Now the natural question arises of whether and how it is possible to join these pieces. The resulting trajectory should be continuous, being an approximation to the continuous solution $x(t)$ of the vector equation (A1). But unlike $x(t)$, this approximation may be non-smooth outside the subsets $\Omega_i^+ \cap \Omega_k^-$, as the right-hand side of the approximating vector equation (A2) is discontinuous there. An answer to the question of how to "glue together" the different pieces is not always trivial, even if we sacrifice the property of smoothness. In fact, it requires a more thorough treatment than we presented before.

We observe first that the solution to (A3) is not unique, because it still requires an initial value and an initial time. Assume that we know how to find any of these solutions and that we are able to check for which values of t the solutions belong to the sub-domain $\Omega_i^+ \cap \Omega_k^-$. We know as well that joining the pieces of the solution should produce a continuous function. Let us first consider the case where this problem can be easily solved. Assume that we have two adjacent sub-domains $G_1 = \Omega_{i1}^+ \cap \Omega_{k1}^-$ and $G_2 = \Omega_{i2}^+ \cap \Omega_{k2}^-$ having Γ as the common piece of the boundary. Assume further that the piece of the solution which belongs to G_1 hits Γ at some point χ at some time point t_0. In this case we may try to define a natural extension of the solution by solving the corresponding vector equation (A3) in G_2 under the additional initial value condition $x(t_0) = \chi$. If the resulting solution proceeds into the sub-domain G_2, then our problem is solved, at least as long as the solution belongs to G_2 for $t > t_0$. By definition, this solution is continuous in both G_1 and G_2, but its derivative may have a jump at $t = t_0$. This kind of solutions is well known in the theory of switching systems. Sometimes one calls the set Γ the *transparent* piece of the boundary, because the solution just travels through Γ like a ray of light.

However, this is not the only scenario. Assume, for instance, that the solution to the corresponding Eq. (A3) satisfying the initial condition $x(t_0) = \chi$ *does not* belong to G_2 for $t > t_0$. In other words, the solutions of the respective equations in the sub-domains G_1 and G_2 approach $\chi \in \Gamma$ from both sides, which means that the point χ is attractive. From the mathematical point of view both solutions stop at χ, but we know that this cannot be true, as no solution of the original vector equation (A1) can stop. Again, from the theory of switching systems it is known that such χ gives rise to so-called "sliding motions" along the piece Γ, which in this case is called "black." This situation is generic in a sense that it cannot be destroyed by small, even smooth, perturbations. In our setting such a situation would mean that the solutions cannot simply be joined at χ. To obtain a proper solution we need to trace the sliding motion along Γ which however cannot be constructed directly from the representation (A3). The solution may still leave Γ after some time, entering G_2 at some point χ_1 at time t_1, so that we will be able to calculate it as the solution of the corresponding Eq. (A3) subject to the initial condition $x(t_1) = \chi_1 (t > t_1)$ and provided that we know how to calculate t_1 and χ_1. The latter is only possible if

we know exactly the behavior of the solution along Γ, which is a non-trivial problem in its own. Thus, the main disadvantage of the piecewise power-law regression method lies in the properties of the approximating solutions in the vicinity of the boundaries between the operating sub-domains. This disadvantage balances the considerable appeal of this method, namely that the power-law representation (A3) is often very efficient and fits biological data well.

The question therefore arises as to how to construct approximating solutions in the discontinuity set of vector equation (A2). A natural way to do this consists of using smooth approximations of Eq. (A2), which replace the right-hand side $\hat{f}(x)$ with a continuous approximation. However, such a replacement makes it impossible to exploit the representation in (A3) and thus removes the advantage of using the power-law formalism. In addition, one is left with very steep nonlinearities which are often difficult to handle.

An alternative approach goes back to Filippov's theory and is based upon differential inclusions, *i.e.*, differential equations with multi-valued right-hand sides. This approach enables us to define continuous trajectories of the approximating solutions without sacrificing the local power-law representations. The price we have to pay is non-uniqueness of the solutions themselves. The following offers a brief explanation for this phenomenon, based on ideas from Chapter 2 of Filippov's monograph (Filippov, 1985).

Assume that a point $x \in \Omega$ is a limit point for the sub-domains $G_m = \Omega_{im}^+ \cap \Omega_{km}^-$ ($m = 1, 2, \ldots, M$), but not for any other sub-domain $G = \Omega_i^+ \cap \Omega_k^-$, *i.e.*, $x \in \bar{G}_m$ ($m = 1, 2, \ldots, M$), but $x \notin \bar{G}$ for any other sub-domain. Inside each G_m the vector equation (A2) can be represented as in (A3):

$$\dot{x} = f(x, m) \equiv \gamma_{im}^+ \prod_{j=1}^{n} x_j^{f_{ijm}^+} - \gamma_{km}^- \prod_{j=1}^{n} x_j^{f_{kjm}^-}. \tag{A4}$$

Let $F(x)$ be the least convex subset of $\Re^n$ containing all vectors $f(x, m)$ ($m = 1, 2, \ldots, M$), *i.e.*, $F(x) = co\{f(x, m) | m = 1, 2, \ldots, M\}$, and let us consider the following differential inclusion:

$$\dot{x} \in F(x) \tag{A5}$$

in the operating domain Ω. A solution to the inclusion (A5) on an interval $[a, b]$ is an absolutely continuous function $x(t)$, $t \in [a, b]$ which satisfies $\dot{x}(t) \in F(x(t))$ almost everywhere on $[a, b]$.

According to Filippov's theory, the initial value problem $x(t_0) = x_0$ has a solution for some $t > t_0$ provided that $x_0 \in \Omega$. The solution either exists for all $t > t_0$, or it blows up at some instant t_1: $x(t) \to \infty$ as $t \to t_1$. However, this solution is not unique in general.

If x belongs to some sub-domain, *i.e.*, if $x \in \Omega_i^+ \cap \Omega_k^-$, then by definition $F(x) = \gamma_i^+ \prod_{j=1}^n x_j^{f_{ij}^+} - \gamma_k^- \prod_{j=1}^n x_j^{f_{kj}^-}$ and we obtain the power-law representation (A3). This means that inside any $\Omega_i^+ \cap \Omega_k^-$ the solutions of (A5) coincide with the solutions of (A3). But the inclusion (A5) gives us much more information about the solutions: it states how to join the separate pieces without losing control over continuity. Thus, any solution of the inclusion (A5) may be viewed as a continuous approximation of the solution to the vector equation (A1). Moreover, this approximating solution coincides with the solutions of Eq. (A3) on the respective sub-domains. Thus, we have solved (at least theoretically) the problem of how to define solutions of collections of differential equations obtained with piecewise power-law regression.

The approximations thus defined may be non-unique. To see why this is so, let us go back to the conventional method of calculating approximating solutions by replacing the right-hand side $\hat{f}(x)$ with its smooth approximations $\hat{f}_\alpha(x)$ where the parameter α indicates how good the approximation $\hat{f}_\alpha(x)$ is, *i.e.*, $\hat{f}_\alpha(x) \to \hat{f}(x)$ as $\alpha \to 0$. The smoothness of $\hat{f}_\alpha(x)$ guarantees the existence of a unique solution $x_\alpha(t)$ of the initial value problem $\dot{x} = \hat{f}_\alpha(x), x(a) = x_0$, say, on the interval $[a, b]$. According to Filippov's theory, the set $\{x_\alpha\}$ is compact in the topology of the uniform convergence on $[a, b]$, and any converging subsequence $\{x_{\alpha_\nu}\}$ approaches one of the solutions $x(t)$ of the differential inclusion (A5) satisfying the same initial condition $x(a) = x_0$. However, different approximations may give different limit solutions, so that non-uniqueness of the Filippov solutions means simply non-uniqueness of approximating solutions in the piecewise power-law regression algorithm. The latter is in turn the result of a very special shape of the approximation (A2): we know how the solutions look like inside any of the operating sub-domains $\Omega_i^+ \cap \Omega_k^-$ due to the power-law representation (A3). However, between the operating sub-domains the behavior of the approximating trajectories becomes more uncertain, which is reflected in the possible non-uniqueness of the solutions outside these sub-domains.

In summary, the paper prescribes how to obtain piecewise power-law representations, even in high-dimensional spaces, but the problem of constructing *smooth* approximating solutions numerically is not solved. Some algorithms based on singular perturbation analysis may be useful for this purpose and can be found in (Plahte and Kjoglum, 2005). Similarly, characterization of the convergence of the approximating solutions, *i.e.*, the solutions of the differential inclusion (A5) to the solutions of the vector equation (A1), is beyond the scope of this paper. The major difficulty of this characterization is the mean-square convergence of the approximations in the piecewise power-law regression, which is distinct from the standard uniform convergence used in the theory of differential equations and inclusions. It is likely that additional assumptions on the influx and efflux functions $V^+(x)$ and $V^-(x)$ are needed in order to prove convergence of the approximating solutions.

References

Brunel, N.J.-B., 2008. Parameter estimation of ODE's via nonparametric estimators. Electr. J. Stat. 2, 1242–1267.
Cascante, M., Sorribas, A., Franco, R., Canela, E.I., 1991. Biochemical Systems Theory: increasing predictive power by using second-order derivative measurements. J. Theor. Biol. 149, 521–535.
Chou, I.-C., Voit, E.O., 2009. Recent developments in parameter estimation and structure identification of biochemical and genomic systems. Math. Biosci. 219, 57–83.
Ferrari-Trecate, G., Muselli, M., 2002. A new learning method for piecewise linear regression. In: Lecture Notes in Computer Science: Proceedings of the International Conference on Artificial Neural Networks, vol. 2415, pp. 444–449.
Ferrari-Trecate, G., Muselli, M., 2003. Single-linkage clustering for optimal classification in piece-wise affine regression. In: Engell, S., Gueguen, H., Zaytoon, J. (Eds.), IFAC Conference on the Analysis and Design of Hybrid Systems (ADHS 03). Saint-Malo, France.
Ferrari-Trecate, G., Schinkel, M., 2003. Conditions of optimal classification for piecewise affine regression. In: Proc. 6th Int. Workshop on Hybrid Systems: Computation and Control. Springer-Verlag, Berlin, Heidelberg, Prague, Czech Rep., pp. 188–202.
Ferrari-Trecate, G., Muselli, M., Liberati, D., Morari, M., 2001a. A learning algorithm for piecewise linear regression. In: Marinaro, M., Tagliaferri, R. (Eds.), Neural Nets: WIRN VIETRI-01, 12th Italian Workshop on Neural Nets. Springer, London, Vietri sul Mare, Italy, pp. 114–119.
Ferrari-Trecate, G., Muselli, M., Liberati, D., Morari, M., 2001b. Identification of piecewise affine and hybrid systems. In: Proceedings of the 2001 American Control Conference, IEEE, Piscataway, NJ, USA, Arlington, VA, pp. 3521–3526.
Ferrari-Trecate, G., Muselli, M., Liberati, D., Morari, M., 2003. A clustering technique for the identification of piecewise affine and hybrid systems. Automatica 39, 205–217.
Filippov, A.F., 1985. Differential Equations with Discontinuous Right-hand Sides. Nauka, Moscow (English transl., Kluwer, Dordrecht, 1998).
Fritzke, B., 1997. Some Competitive Learning Methods, Technical Report, Institute for Neural Computation, Ruhr-Universität Bochum.
Goel, G., 2009. Dynamic Flux Estimation: A Novel Framework for Metabolic Pathway Analysis. Department of Biomedical Engineering, Georgia Institute of Technology, Atlanta, p. 156.
Goel, G., Chou, I.-C., Voit, E.O., 2008. System estimation from metabolic time series data. Bioinformatics 24, 2505–2511.

Please cite this article in press as: Machina, A., et al., Automated piecewise power-law modeling of biological systems. J. Biotechnol. (2010), doi:10.1016/j.jbiotec.2009.12.016

Guillen-Gosalbez, G., Sorribas, A., 2009. Identifying quantitative operation principles in metabolic pathways: a systematic method for searching feasible enzyme activity patterns leading to cellular adaptive responses. BMC Bioinform. 10, 386.
Gutenkunst, R.N., Casey, F.P., Waterfall, J.J., Myers, C.R., Sethna, J.P., 2007a. Extracting falsifiable predictions from sloppy models. Ann. N. Y. Acad. Sci. 1115, 203–211.
Gutenkunst, R.N., Waterfall, J.J., Casey, F.P., Brown, K.S., Myers, C.R., Sethna, J.P., 2007b. Universally sloppy parameter sensitivities in systems biology models. PLoS Comput. Biol. 3, 1871–1878.
Heijnen, J.J., 2005. Approximative kinetic formats used in metabolic network modeling. Biotechnol. Bioeng. 91, 534–545.
Kvasnica, M., Greider, P., Baotić, M., Morari, M., 2004. Multi-Parametric Toolbox (MPT), Hybrid Systems: Computation and Control. Springer, Berlin/Heidelberg.
Neves, Ramos, Shearman, Gasson, Almeida, Santos, 2000. Metabolic characterization of *L. lactis* deficient in lactate dehydrogenase using in vivo ^{13}C NMR. Eur. J. Biochem. 267, 3859–3868.
Peskov, K., Goryanin, I., Demin, O., 2008. Kinetic model of phosphofructokinase-1 from *Escherichia coli*. J. Bioinform. Comput. Biol. 6, 843–867.
Plahte, E., Kjoglum, S., 2005. Analysis and generic properties of gene regulatory networks with graded response functions. Physica D 201, 150–176.
Polisetty, P.K., Voit, E.O., Gatzke, E.P., 2006. Identification of metabolic system parameters using global optimization methods. Theor. Biol. Med. Model. 3, 4.
Raue, A., Kreutz, C., Maiwald, T., Bachmann, J., Schilling, M., Klingmuller, U., Timmer, J., 2009. Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. Bioinformatics 25, 1923–1929.
Savageau, M.A., 1969a. Biochemical systems analysis. I. Some mathematical properties of the rate law for the component enzymatic reactions. J. Theor. Biol. 25, 365–369.
Savageau, M.A., 1969b. Biochemical systems analysis. II. The steady-state solutions for an n-pool system using a power-law approximation. J. Theor. Biol. 25, 370–379.
Savageau, M.A., 1970. Biochemical systems analysis. 3. Dynamic solutions using a power-law approximation. J. Theor. Biol. 26, 215–226.
Savageau, M.A., 1976. Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology. Addison-Wesley Pub. Co., Advanced Book Program, Reading, MA.
Savageau, M.A., 2001. Design principles for elementary gene circuits: elements, methods, and examples. Chaos 11, 142.
Savageau, M.A., 2002. Alternative designs for a genetic switch: analysis of switching times using the piecewise power-law representation. Math. Biosci. 180, 237–253.
Savageau, M.A., Voit, E.O., 1987. Recasting nonlinear differential equations as S-systems: a canonical nonlinear form. Math. Biosci. 87, 83–115.
Schulz, A.R., 1994. Enzyme Kinetics: From Diastase to Multi-enzyme Systems. Cambridge University Press, Cambridge, New York.
Torres, N.V., Voit, E.O., 2002. Pathway Analysis and Optimization in Metabolic Engineering. Cambridge University Press, Cambridge, U.K.
Varah, J.M., 1982. A spline least squares method for numerical parameter estimation in differential equations. SIAM J. Sci. Stat. Comput. 3, 28–46.
Vilela, M., Borges, C.C., Vinga, S., Vasconcelos, A.T., Santos, H., Voit, E.O., Almeida, J.S., 2007. Automated smoother for the numerical decoupling of dynamics models. BMC Bioinform. 8, 305.
Voit, E.O. (Ed.), 1991. Canonical Nonlinear Modeling. S-System Approach to Understanding Complexity. Van Nostrand Reinhold, New York, NY.
Voit, E.O., 2000. Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists. Cambridge University Press, New York.
Voit, E.O., 2008. Modeling networks using power-laws and S-systems. Essays Biochem. 45, 29–40.
Voit, E.O., Almeida, J., 2004. Decoupling dynamical systems for pathway identification from metabolic profiles. Bioinformatics 20, 1670–1681.
Voit, E.O., Savageau, M.A., 1982. Power-law approach to modeling biological systems. III. Methods of analysis. J. Ferment. Technol. 60, 233–241.
Voit, E.O., Goel, G., Chou, I.-C., da Fonseca, L., 2009. Estimation of metabolic pathway systems from different data sources. IET Syst. Biol. 3 (6), 513–522.

Please cite this article in press as: Machina, A., et al., Automated piecewise power-law modeling of biological systems. J. Biotechnol. (2010), doi:10.1016/j.jbiotec.2009.12.016