PARAMETER ESTIMATION AND NETWORK IDENTIFICATION IN METABOLIC PATHWAY SYSTEMS

A Dissertation
Presented to
The Academic Faculty

by

I-Chun Chou

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Bioinformatics/Biomedical Engineering

Georgia Institute of Technology
December 2008

PARAMETER ESTIMATION AND NETWORK IDENTIFICATION IN METABOLIC PATHWAY SYSTEMS

Approved by:

Dr. Eberhard O. Voit, Advisor
Department of Biomedical Engineering
Georgia Institute of Technology

Dr. Mark Borodovsky
Department of Biomedical Engineering
Georgia Institute of Technology

Dr. Robert Butera
Department of Biomedical Engineering
Georgia Institute of Technology

Dr. Melissa Kemp
Department of Biomedical Engineering
Georgia Institute of Technology

Dr. Haesun Park
College of Computing
Georgia Institute of Technology

Date Approved: July 23, 2008

To Mom, Dad, and Chiaolong

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor and mentor, Dr. Eberhard O. Voit, for his guidance, support, and encouragement throughout my graduate studies. Without his kindness and willingness to give me a chance to pursue my goals, it would have been impossible for me to come this far. He is always willing to share his experience and I have learned innumerable lessons and insights on research and other academic issues from him. Even though he was constantly busy, Dr. Voit always found time to listen to my questions, even the little ones, and to give advice. I also appreciate his patience when I was stuck and when progress was slow. I am especially thankful for his technical and editorial advice during the completion of this dissertation.

I am indebted to my committee members: Drs. Mark Borodovsky, Robert Butera, Melissa Kemp, and Haesun Park, for providing many valuable comments during my dissertation work. My special thanks go to Dr. Mark Borodovsky for his kind help with overcoming numerous obstacles when I started, especially during the early years of the Bioinformatics Ph.D. program. I would like to thank Dr. Haesun Park for her numerous discussions and suggestions regarding improvements of the matrix computations in my algorithms.

My sincere thanks are extended to Dr. Harald Martens at CIGENE, Norway, for his valuable insights, inputs and inspiring discussions toward the development of the alternating regression method. I also like to thank Dr. Jonas S. Almeida and Marco Vilela at the University of Texas M.D. Anderson Cancer Center, for initiating the eigenvector optimization work and for their great and insightful discussions.

I would be amiss if I didn’t thank my wonderful fellow researchers: Gautam Goel, for being so energetic and encouraging all the time. His special crazy ways of thinking and energetic personality have led me to see problems in different ways. Dr. Siren Veflingstad, for her generous advice in scientific matters and technical skills. Weiwei Yin, Dr. Zhen Qi, Jialiang Wu, and Yun Lee for their friendships and for creating a warm, welcoming lab atmosphere.

Last but not least, I would like to thank my husband Chiaolong, who has been by my side through all of this. I thank him for his unconditional support, unimaginable patience, and for always being so positive. I am deeply grateful to all my family members. They are my irreplaceable safe haven with their unconditional dedication, support, and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF SYMBOLS AND ABBREVIATIONS

3-AR	3-way Alternating Regression
ACO	Ant Colony Optimization
ANN	Artificial Neural Network
AR	Alternating Regression
BST	Biochemical Systems Theory
CDF	Cumulative Density Function
CMC	Correlation Metric Construction
CPU	Central Processing Unit
DE	Differential Evolution
DFE	Dynamic Flux Estimation
DGA	Distributed Genetic Algorithm
DOA	Dynamic Optimization Approach
eAPS	enhanced Aggregation Pheromone System
EC	Evolutionary Computation
EMC	Entropy Metric Construction
EO	Eigenvector Optimization
EP	Evolutionary Programming
ERM	Entropy Reduction Method
ES	Evolutionary Strategy
FBA	Flux Balance Analysis
GA	Genetic Algorithm
GLSDC	Genetic Local Search with distance independent Diversity Control
GMA	Generalized Mass Action

GOGA	Grid-Oriented Genetic Algorithm
GP	Genetic Programming
GUI	Graphical User Interface
HDE	Hybrid Differential Evolution
HPLC	High Performance Liquid Chromatography
IGA	Intelligent Genetic Algorithm
iTEA	intelligent Two-stage Evolutionary Algorithm
LMS	Least Mean Square
LS	Local Search
MA	Memetic Algorithm
MADMan	Munich, Atlanta, DiliMAN
MCA	Metabolic Control Analysis
MDS	Multidimensional Scaling
MGG	Minimal Generation Gap
MILP	Mixed Integer Linear Programming
MS	Mass Spectrometry
NHANES III	The 3rd National Health and Nutrition Examination Survey
NLP	Nonlinear Programming Problem
NMR	Nuclear Magnetic Resonance
NSS-EA	Network Structure Search-Evolutionary Algorithm
ODE	Ordinary Differential Equation
OLSR	Ordinary Least Squares Regression
OSA	Orthogonal Simulated Annealing
PCR	Principal Component Regression
PDF	Probability Density Function

PLSR	Partial Least Squares Regression
PSO	Particle Swarm Optimization
PSS	Pseudo Steady State
RBFN	Radial Basis Function Network
RCGA	Real Coded Genetic Algorithm
SA	Simulated Annealing
SC Formalism	Saturable and Cooperative Formalism
SD	Standard Deviation
SGA	Simple Genetic Algorithm
SOA	Static Optimization Approach
SQP	Sequential Quadratic Programming
SSE	Sum of Squared Error
TCA	Tricarboxylic Acid Cycle
UNDX	Unimodal Normal Distribution Crossover

SUMMARY

Cells are able to function and survive due to a delicate orchestration of the expression of genes and their downstream products at the genetic, transcriptomic, proteomic, and metabolic levels. Since metabolites, the end products of gene expression, are ultimately the causative agents for physiological responses and responsible for much of the functionality of the organism, a comprehensive understanding of cell functioning mandates deep insights into how metabolism works. However, the regulation and dynamics of metabolic networks are often too complex to allow intuitive predictions, which thus renders mathematical modeling necessary as a means for assessing and understanding metabolic systems.

The construction of mathematical models for metabolic pathways is challenging, and a particularly complicated task is the estimation of model parameters and the identification of network structure. Recent advancements in modern high-throughput techniques are capable of producing time series data that characterize dynamic metabolic responses and enable us to tackle estimation and identification tasks using "top-down" or "inverse" approaches. However, extracting information regarding the structure and regulation of the system described by these data is difficult. The challenges can be generally categorized in four problem areas, namely: data related issues, model related issues, computational issues, and mathematical issues.

To develop improved methods for inverse modeling that are effective, fast, and scalable, this work proposes two novel algorithms namely *Alternating Regression* (AR) and *Eigenvector Optimization* (EO), both applied to S-systems in *Biochemical Systems Theory* (BST). The AR method employs a decoupling technique for systems of

differential equations and dissects the complex nonlinear parameter estimation task into iterative steps of linear regression by utilizing the fact that power-law functions are linear in logarithmic space. AR is very fast in comparison to conventional methods and works well in many applications. In cases where convergence is an issue, the fast speed renders it feasible to dedicate some computational effort to identifying suitable start values and search settings. AR is beneficial for the identification of system structure in S-systems as well.

A modification of the AR algorithm is *3-way Alternating Regression* (3-AR), which was applied here to parameter estimation in S-distributions that form a statistical distribution family motivated by S-systems. 3-AR preserves the properties of AR but iterates the algorithm between three phases of linear regression. The 3-AR algorithm is very fast and performs well for artificial, error-free and noisy datasets, as well as for random samples generated from traditional statistical distributions and for observed raw data.

The EO method is an extension of AR that is based on a matrix formed from multiple regression equations of the linearized decoupled S-systems. In contrast to AR, EO operates initially only on one term, whose parameter values are optimized completely before the complementary term is estimated. It was demonstrated that the EO algorithm converges fast and can be expected to converge in most cases, without necessarily requiring knowledge of the network structure. Furthermore, EO is easily extended to the optimization of network topologies with stoichiometric precursor-product constraints among equations.

To integrate all existing techniques and make inverse modeling more effective, this work proposes an operational "work-flow" that guides the user through the estimation process, identifies possibly problematic steps, and suggests corresponding solutions based on the specific characteristics of the various available algorithms. A significant consequence and advantage of the combined approach is that the result often consists of multiple parameter sets that are all consistent with the data and that can lead to hypotheses for further theoretical and experimental investigation. Finally, the work described here discusses a recent *Dynamic Flux Estimation* (DFE) approach, which resolves open issues of model validity and quality beyond residual errors. The necessity of fast solutions to biological inverse problems is discussed in the context of concept map modeling, which allows the conversion of hypothetical network diagrams into mathematical models.

CHAPTER 1

INTRODUCTION AND LITERATURE REVIEW

1.1 Overview

A key to understanding how living cells function is to understand how genes and their products carry out their functions at the genetic, transcriptomic, proteomic, and metabolic levels in a global context. The components at the most downstream level, the metabolites or, collectively, the metabolome, are the end products of gene expression; they actually yield much of the utility to the organism and permit instantaneous physiological responses. For instance, metabolism is responsible for the generation of energy, synthesis of building blocks for the assembly of functional biomolecules, degradation of toxic substrates, and transduction of external signals to the genome. Therefore, it is quite evident that a good understanding of cell functioning is closely related to how metabolism works.

The cell's metabolic network is the collection of all metabolic pathways, each of which is composed of a series of biochemical reactions catalyzed by enzymes, and requires other cofactors in order to function properly. The metabolic pathways are not independent of each other. Instead, metabolites within one pathway may serve as precursors or regulate steps in another pathway. Hence, because metabolic pathways usually consist of many components which are coupled through multiple reactions and regulatory interactions, metabolic networks are complex and highly interrelated. Even if we take a step backward and look at only one part of the network, a single metabolic pathway may still be too complex to allow intuitive predictions. As an example, consider glycolysis, the first metabolic pathway, discovered back in 1859 when Pasteur found that certain cell extracts can cause fermentation. After almost one and half centuries of extensive research on glycolysis, we may say that this pathway is well understood but there are still many open questions regarding the regulation of these pathways remain unsolved (Teusink *et al.*, 2000; Hynne *et al.*, 2001; Voit *et al.*, 2006a; Voit *et al.*, 2006b).

The identification of the structure of metabolic networks has been the emphasis of intense research for several decades and led to a substantial knowledge of the processes that determine the biochemical and physiological properties of an organism. The challenges of truly understanding the functioning of metabolite pathways may be characterized by two aspects (Voit and Schwacke, 2007). First, the biological systems are usually nonlinear, which makes predictions difficult even for simple unbranched pathways that are regulated by a few inhibitory signals. The cell typically regulates its metabolic pathway in two ways: (1) metabolites within a pathway can directly regulate each other at the metabolic level; (2) metabolites may also affect the expression of genes or modification of proteins per signaling. The regulation within the metabolic level is much faster than regulatory mechanisms based on gene expression. Therefore, even though most metabolic pathway studies account only for regulation within the metabolic level, the coordination of regulation at different levels is ultimately unavoidable. However, multi-level controls are not well understood and further increase the complexity of metabolic systems and increase the necessity of modeling.

Second, cells tend to maintain homeostasis or “find their way out of problems,” such as the undue accumulation of unneeded metabolites. For instance, all organisms have control mechanisms that easily adapt to their environment or to changes of states. This adaptation is realized by the fact that the same components in the cell may have different functions and many of the important cellular control functions exhibit considerable redundancy. As a result, some unexpected pathways may branch out and lead metabolites to other fates, or change the proportion of fluxes between routes. These variations make the dynamics of metabolism complicated and render it clear that mathematical modeling is necessary for understanding the regulation of metabolic networks.

The typical approach to mathematical model construction of metabolic pathways consists of five phases: collection of information on network structure and regulation, mathematical model framework selection, parameter values estimation, model diagnostics, and model application. The first phase is dedicated to developing hypotheses regarding network structure. In this phase we need to identify what components and interactions of the system are to be included in the model. The results are usually visualized as diagrams with nodes denoting the components and arrows representing interactions between them. The second phase includes the choice of a mathematical modeling framework and the formulation of suitable equations. The process usually starts with converting the “wire-diagram” or “network topology” obtained from the first phase into equations. These typically form a set of ordinary differential equations that represent the velocities or fluxes in symbolic forms based on the mathematical framework of choice. After the symbolic modeling equations are formulated, the third phase is to determine the appropriate numerical parameter values that make the model consistent with experimental observations. Once the initial model is obtained, the fourth phase is dedicated to diagnostics of the model, before we can rely on it for applications in the last phase, such as making predictions, generating hypotheses, or designing additional biological experiments. The modeling process may look quite straightforward. However, in most cases it is not linear but a cyclic process which may require the return to earlier phases.

Among these phases, the most challenging task is the estimation of parameter values. This task has attracted scientists from all over the world who dedicate considerable efforts on this aspect, and it is also the focus of my work in this dissertation. One should keep in mind that parameter estimation is not an isolated task, but closely related to the other phases in the modeling process. For instance, the size and accuracy of the hypothetic model obtained in the first phase may alter the difficulty of parameter estimation and also affect later analyses and the interpretation of the results. Furthermore, the choice of a modeling framework naturally influences the degree of simplicity, feasibility, and practicability in parameter estimation.

The development of parameter estimation methods is driven by the availability of experimental data. The methods for analyzing varied types of data are distinctly different, and, conversely, the nature of suitable data for variant estimation methods is rather different. Traditionally, the kinetic properties of a single step within metabolic pathway have been presented in the terminology of enzyme kinetics, and predominantly as a Michaelis-Menten rate law. Using these types of "local" descriptions of model components and merging them into one comprehensive model is referred to as a "bottom-up" approach.

Steady-state data are also used in parameter estimation. This type of analysis is generally based on experiments that measure the responses of a biological system after a small perturbation around the steady state.

Recent advancements in experimental tools of biology enable us to tackle the parameter estimation task using a "top-down" approach in a more comprehensive manner. These tools are able to generate time series data or "global" data of metabolites, sometimes even under different conditions, such as initial concentrations or upon various gene knock-outs. The detailed processes and issues of the traditional methods and newly developed techniques in parameter estimation will be addressed in the following sections.

Based on the general flow of the modeling process, as described above, the parameter estimation methods are employed together with a symbolic model that is constructed after the first phase of the modeling process. In other words, before the parameter estimation step is started, the topology of the network and its corresponding symbolic model are set up and they are assumed to be correct with relatively high confidence. However, in reality sometimes the true topology of the metabolic pathway is not fully understood or it is even far from complete. Under these circumstances the task consists of the inference of the metabolic network topology and its regulation from metabolic data. Generally speaking, this general structure identification task is much more difficult than parameter estimation, which is already very hard. One should note in this context that there is no clear boundary between parameter estimation and structure identification. Indeed, parameter estimation is a component of structure identification. Conversely, a good structure prediction reduces the complexity of parameter estimation. The top-down approach described before may also use time series data to identify pathways with structures that are not fully known or whose regulation is obscure.

To further discuss the issues related to the inference of metabolic network connectivity and the determination of parameter values that describe the dynamics of a network model from metabolic data, key sub-topics are briefly outlined here and will be elaborated in the subsequent sections.

I. *Modeling approach*: To construct the mathematical model of a metabolic pathway, an important step is to select a mathematical form which can capture the phenomenon of interest. In Section 1.2 I will review the rationale and special demand of mathematical models for metabolic pathway modeling and introduce some of the representative modeling frameworks, such as stoichiometric model, the law of mass action, the Michaelis-Menten rate law, and canonical models. The goal of this phase is to choose a suitable kinetic model to represent the dynamics of a metabolic pathway.

II. *Kinetic model construction*: After the kinetic modeling framework has been decided, the next step is to determine the parameter values in the model. In Section 1.3 I will review some approaches for parameter estimation, including forward (bottom-up) modeling, using steady-state data, and inverse (top-down) modeling. The challenges of inverse modeling and some current optimization strategies will also be reviewed briefly in this section.

III. *Parameter estimation techniques in the top-down modeling approach*: As an extension of Section 1.3, some details of pertinent algorithms will be reviewed in Section 1.4. The methods include those that are used to attack the main problem of optimizing parameter values against the observed time series data, as well as others that circumvent the costly integration of differential equations, smooth noisy data and estimate slopes, constrain the parameter search space, or reduce the complexity of the inference task.

IV. *Inference of network structure*: In Section 1.5 I introduce some of the most relevant structure identification methods, namely the determination of the Jacobian matrix, direct observations, correlation-based approaches, simple-to-general and general-to-specific modeling, and time series data analysis using the framework of Biochemical Systems Theory (BST).

1.2 Modeling approach

1.2.1 Model requirements

In the previous sections I have briefly shown the necessity of using mathematical and computational methods for analyzing and understanding the regulation of metabolic networks. The question thus shifts toward the search for the most useful mathematical frameworks and tools. Mathematical modeling and control theory have a long history in engineering. However, the demands and specific requirements in modeling biological systems are quite different and require the adaptation and extension of present methods and also the development of additional tools in order to be suited for modeling biological phenomena. The peculiarities of biological system modeling can be generally described in five aspects (Voit and Schwacke, 2007). First, the biological processes and interactions are highly nonlinear and complex. Thus, a mathematical structure is needed that can capture nonlinearities and does not *a priori* exclude relevant biological phenomena.

Second, dynamic responses of biological systems are particularly interesting. Therefore, a suitable mathematical model will have to be time dependent, which almost always requires formulation as a set of differential equations. Third, real biological systems are usually composed of different levels of components and interactions with relatively large numbers. The ability to scale a mathematical framework to handle increasingly larger biological models is necessary. Fourth, biological systems may have stochastic features when there are only few molecules involved. Under this condition, the fundamental laws of kinetics and thermodynamics are no longer applicable and the biological behavior becomes difficult to predict. Thus, in addition to grasping a deterministic phenomenon, the mathematical model should also be able to capture stochastic behaviors when these dominate the process. And fifth, biological reactions rarely happen in a homogeneous environment but are restricted to organelles or compartments. This feature is sometimes important, and therefore the ability of handling spatial process is necessary for a comprehensive mathematical analysis.

By now it has been made clear that biological systems are complex and this may give the impression that one should include every feature and every detail when it comes to modeling. However, it is impossible to be complete and decisions must be made as to what types of simplifications and approximations are necessary. Besides, the aim of developing a model is not just finding a valid description of the system, but also maintaining some degree of convenience for analysis and manipulation. Therefore, the decisions on simplifications and approximations constitute a compromise between several factors, such as the validity of describing the system, mathematical convenience, and importantly, the goal of modeling.

One has to decide what kind of model is suitable for the objectives and the experimental data by considering four properties: dynamic or static, continuous or discrete, deterministic or stochastic, and spatial or homogeneous (Veflingstad *et al.*, 2008). In metabolic pathway modeling, we are usually interested in the dynamic and continuous changes of metabolites. Therefore, a dynamic and continuous model is typically preferred over a static and discrete model. In addition, if we are primarily interested in average model responses rather than extreme or highly unlikely cases, the stochastic aspect is usually ignored and a deterministic model will be sufficient. Typically stochastic phenomena are more eminent in gene regulatory networks than in metabolic networks because there are only a few molecules involved in gene interactions. Furthermore, if the spatial aspects are not particularly important, we can ignore them and assume the environment is homogenous.

Thus, if a dynamic, continuous, deterministic, and homogenous model is chosen to represent the behavior of a metabolic pathway, the temporal changes of metabolites can be formulated as a generic set of ordinary differential equation of the form

$$\dot{X}_i = V_i^+ - V_i^- = V_i^+(X_1,\ldots,X_n) - V_i^-(X_1,\ldots,X_n), \quad i = 1,\ldots,n, \tag{1.1}$$

where X_i denotes the concentration of a metabolite or metabolite pool and n is the number of metabolites in the system. The functions V_i^+ and V_i^- represent the reaction rates or fluxes coming in and going out of the metabolite pool X_i. This general framework has numerous alternatives and applications in metabolic pathway modeling depending on the functions used to describe V_i^+ and V_i^-. I will briefly review some of the modeling approaches in the following sections.

1.2.2 Stoichiometric models

Mathematical models describing metabolic pathways can be constructed with a focus either on stoichiometry or kinetics. The stoichiometric property itself is time invariant. It is a simple translation of the wire diagram that describes the network topology into a matrix which represents how metabolites are converted into other metabolites. There are two important features of the elements in a stoichiometry matrix, the sign and the value. The sign represents the direction of material flow, for instance, whether the reaction increases or decreases the concentration of a certain metabolite pool. If a metabolite and a reaction are unrelated, the corresponding element is zero. The value indicates the stoichiometric relationship and must be an integer. For instance, if one unit of substrate molecules breaks down into two product molecules, the gain in product is coded as +2. Stoichiometric models thus use the stoichiometry matrix **N**, multiplied with a vector of fluxes v, to describe the dynamics of the metabolite concentrations in a vector S using a set of ordinary differential equations. Each of the equations represents a biochemical reaction and the set taken together expresses the dynamics of the metabolite concentrations as

$$\frac{dS}{dt} = \mathbf{N} \cdot v \,. \tag{1.2}$$

Detailed description of stoichiometric models can be found in a number of journal articles and books (Gavalas, 1968; Heinrich and Schuster, 1996; Stephanopoulos *et al.*, 1998; Palsson, 2006).

The main application of stoichiometric models is to determine the rates of the fluxes v in the metabolic network. The flux determination methods can be generally divided into three categories depending on the type of experimental data. First, in most analyses, stoichiometric models are studied in the steady state, where all material flow into the pool equals the material flow out the pool, by assuming that the flux rates are constant. Under this assumption, the left hand sides of the equations in Eq. (1.2) become zero and the system of differential equations becomes a set of linear algebraic equations. If the stoichiometric matrix is full rank, it is straightforward to calculate the fluxes. However, it is usually the case that there are more unknown fluxes than equations, so that the system of linear equations is underdetermined.

Flux balance analysis (FBA) inherits the properties of stoichiometric approach but adds some features like imposing mathematical constraints to find the feasible or optimal distribution of fluxes. The background of FBA is reviewed in Palsson (Palsson, 2006) and the development of variations is summarized in Kauffman *et al*. (Kauffman *et al.*, 2003). The modeling process in FBA consists of four steps: system identification, mass balance, defining measurable fluxes, and optimization. Mass balance is the application of conservation of mass which is a distinctive property in metabolic pathways and not applicable to gene regulatory networks. For instance, the total number of moles of carbon in the system is conserved during the time of reaction. Therefore, by accounting for material flows entering and leaving each metabolite pool in the pathway, one can determine the material distribution and also identify some flows which might have been unknown or difficult to measure in the experiment. In the optimization step an objective function is proposed, for instance, to maximize the yield of certain metabolites of interest while minimizing nutrient utilization. Then, the objective function is obtained using standard algorithms such as linear programming. The main advantages of both the stoichiometric model and FBA are their matrix representation and linearity at the steady state, which make the analysis relatively easy since there are numerous well-established analytical methods that support this kind of analysis. Several examples have shown that FBA is capable of assessing the theoretical capabilities and operative modes of metabolic systems in the absence of kinetic information (*cf.* (Selkov *et al.*, 1997; Bono *et al.*, 1998; Edwards and Palsson, 2000; Forster *et al.*, 2003; Palsson, 2006)).

Stoichiometric models are sometimes studied under the pseudo-steady-state (PSS) assumption in cases where the concentrations of metabolites rapidly adjust to new levels (Yang *et al.*, 2002; Okamoto, 2008; Teixeira *et al.*, 2008). This PSS approximation was shown to be valid for most intracellular metabolites (Vallino and Stephanopoulos, 1993). Under this assumption, it is reasonable to neglect the instantaneous changes of metabolites and set the rate of change to zero.

When the complete time course of metabolite changes is available, the flux distribution at each time point can be determined under the PSS assumption (Vallino and Stephanopoulos, 1993) or without (Goel *et al.*, submitted). Different from the standard application where only the steady state data are used, the metabolite change rates in the latter case are not necessary zero and can be deduced by slope determination or direct measurements (Goel *et al.*, submitted). Once the left-hand sides of Eq. (1.2) are substituted by instantaneous changes, the fluxes at each time point can be determined directly if the stoichiometric matrix is full rank. However, similar to the standard steady-state application, in many cases the system is underdetermined.

Mahadevan and coworkers (Mahadevan *et al.*, 2002) extended traditional FBA to account for dynamics and presented two different formulations for dynamic FBA: the dynamic optimization approach (DOA) and the static optimization approach (SOA). DOA involves optimization over the entire time period of interest to obtain time profiles of fluxes and metabolite levels. SOA involves dividing the batch time into several time intervals and solving the instantaneous optimization problem at the beginning of each time interval. By testing the methods in the analysis of diauxic growth in *Escherichia coli*, the authors concluded that SOA was computationally simpler to implement provided all of the constraints were linear, whereas DOA was more flexible and suitable for the incorporation of experimental data.

By now I have shown that using the stoichiometric property together with the rate changes of metabolites is successful in studying the flux distribution in metabolic pathways. There are other applications of stoichiometric models, such as the inclusion of regulation by multiplication of stoichiometric matrices with binary regulation matrices, which represent the turning on and off of additional (regulatory) processes (Palsson, 2006). However, the main advantage of the approaches above is that they focus almost exclusively on the connectivity structure of the system and the fluxes distribution and do not require kinetic information. Therefore, the predictive power is limited due to the lack of nonlinearity such as regulatory signals and other nonlinear dynamic interactions, which can only be included in the formulation of a kinetic model.

1.2.3 Kinetic models of pathway steps

When detailed information is available about the kinetics of the specific metabolic reaction, it is possible to describe its dynamics by incorporating kinetic properties with the known stoichiometry of metabolic pathways (Gombert and Nielsen, 2000). A first step toward combining the stoichiometric property with kinetic features is to investigate the appropriate functions to represent the simple flux quantities V_i^+ and V_i^- in Eq. (1.1). Many functional forms of have been proposed, but the most prevalent are formulations based on the law of mass action, Michaelis-Menten rate laws, and different types of canonical models.

Mass action systems

Models based on the law of mass action are typically used to describe reaction networks consisting of elementary reactions. The rate of a given elementary reaction is proportional to the product of concentrations of all variables reacting in the elementary process and is generally formulated as the basis function

$$v = k\prod_{g=1}^{n} X_i^{g_i}\ ,\ i = 1, 2,\ldots, n, \tag{1.3}$$

where k is the rate constant which is always positive and g_i are kinetic orders which are non-negative integer numbers that reflect the numbers of molecules involved in the reaction. The advantage of models based on the law of mass action is that it can be determined directly from the elemental reactions and their stoichiometry, if the information is known. However, in most realistic cases the reactions are not elemental but catalyzed by enzymes, not well understood, or experimentally inaccessible in detail. Therefore, the equations are hard to set up and the parameters of the model are difficult to obtain.

Michaelis-Menten and similar rate laws

The Michaelis-Menten model (Michaelis and Menten, 1913) and its variations are among the most commonly used representations for kinetic modeling in metabolic pathways. The model is based on the concept that a substrate and an enzyme form a transient complex which either dissolves to return the two or leads to the formation of a product and the release of the enzyme. The modeling of enzyme reactions in this type of approach is simplified considerably under the quasi-steady-state approximation assumption, which states that the intermediate complex does not change appreciably over time. Even though the Michaelis-Menten based rate laws are straightforward to set up, complete description of more complex enzyme mechanisms may become massive if several substrates or reactions are involved, even in moderately large biochemical systems (Schulz, 1994). As the result, the mathematical analyses become very complex and the parameter estimation requires an undue amount of experimental data (Veflingstad *et al.*, 2008). In addition to issues caused by technical problems, the model results are difficult to interpret and thus useful information is hard to extract to understand the underline biological system (Heinrich and Rapoport, 1974).

Canonical models

As discussed in the previous sections, the predictive ability of stoichiometric models is limited because nonlinearities due to regulation are not included in the model. To improve the model, detailed kinetic information of the pathway is needed. However, if we incorporate the dynamic information using the *ad hoc* models such as those based on traditional kinetic rate laws to describe the flux rates, the task quickly becomes cumbersome mathematically and impractical in reality. Therefore, to find a good compromise which can capture the dynamics while keeping the mathematics simple, it is often beneficial to search for a “canonical” nonlinear model whose structure is fixed and whose individuality comes from its parameter values. Besides, these homogeneous structures are more or less size independent and therefore allow the same types of analyses and diagnostics. I will present some details of canonical models in the next section.

1.2.4 Canonical models

Arguably the most promising canonical nonlinear models in metabolic modeling are S-system and Generalized Mass Action (GMA) system structures within *Biochemical Systems Theory* (BST) (Savageau, 1969b; Savageau, 1969a; Savageau, 1976; Voit, 2000a; Torres and Voit, 2002). These models are constructed by approximating fluxes with products of power-law functions, which are mathematically grounded in the well-established approximation theory of Taylor. In the S-system formalism, each equation has a particularly simple format: The change in system variables is given as one set of influxes minus one set of effluxes, and each set is collectively written as one product of power-law functions as

$$\dot{X}_i = \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}} \text{ , } i = 1, 2, \ldots, n, \qquad (1.4)$$

where X represents the variable (metabolite) and n denotes the number of variables in the system. The non-negative numbers α_i and β_i are *rate constants* which quantify the turnover rate of the production or degradation, respectively. The real numbers g_{ij} and h_{ij} are *kinetic orders* which reflect the strengths of the effects that the corresponding variables X_j have on a given flux term. A positive value signifies an activating or augmenting effect exerted by X_j, a negative value signifies an inhibitory effect. A kinetic order of zero implies that the corresponding variable X_j does not have an effect on a given flux.

In the GMA formalism, instead of aggregating all influxes and all effluxes into one term each, all influxes and effluxes are approximated individually with power-law terms such that

$$\dot{X}_i = \sum_{p=1}^{p_i} \left(\pm \gamma_{ip} \prod_{j=1}^{n} X_j^{f_{ipj}} \right), \, i = 1, 2, \ldots, n, \tag{1.5}$$

where the rate constants γ_{ip} are non-negative and the kinetic orders f_{ipj} may have any real values as in the S-system form. It should be noted that differences between these two formulations only exist at branch points, whereas all other steps are identical.

BST models have a number of important advantages which have been discussed in detail (Savageau, 1969b; Savageau, 1969a; Savageau, 1976; Voit, 1993; Voit, 2000a). Among the beneficial features, four are particularly crucial here. First, these systems are rich enough in structure to capture virtually any nonlinearity including complex oscillations and chaos. Second, symbolic BST models can be set up without mechanistic information on the underlying system, but if information is available, it can be used to simplify the symbolic representation. Third, the highly structured format facilitates mathematical and numerical analyses. These analyses include computations associated with steady states, sensitivity, stability, as well as dynamic features. Fourth, BST models are characterized by a one-to-one relationship between parameters and structural features. Thus, if structural features are known, it is explicitly clear where they will appear in the BST models. Conversely, if a parameter has been identified, its interpretation in terms of structural properties is immediate. This feature is especially crucial for structure identification and parameter estimation of metabolic model. The reasons will be discussed in detail in Section 1.4.4. The power-law models were initially used to model metabolic pathways, but this formalism is also shown to be satisfied in modeling several other kinds of biological systems, including genetic networks, multi-level systems, and cell signaling (Savageau, 2001; Atkinson *et al.*, 2003; Vera *et al.*, 2007).

An alternative canonical form is “lin-log approximation” which was introduced by Hatzimanikatis and Bailey (Hatzimanikatis and Bailey, 1996) and expanded by Visser and Heijnen (Visser and Heijnen, 2002). This form is based on taking the logarithm of each metabolite concentration and enzyme activity in relationship to a corresponding reference value. The lin-log model constitutes an extension of Metabolic Control Analysis (MCA), a theoretical framework for analyzing control and regulation in metabolic networks close to their steady state (Kacser and Burns, 1973; Fell, 1997).

Another recently proposed canonical form is the *Saturable and Cooperative Formalism* (SC formalism) (Sorribas *et al.*, 2007), which is derived based on Taylor approximation in a special transformation space defined by power-inverses and logarithms of power-inverses. The SC formalism is shown to have the properties of cooperativity and saturation, which are absent in other canonical formalisms. In addition, unlike the other formalisms where the approximation is valid only around small enough deviations from the operating point, the SC formalism is expected to be accurate over a wider range around the operating point if the approximated functions are saturated.

The choice of an S-system, GMA, lin-log, or SC formalism depends on the information available and on the purpose of modeling. For instance, GMA systems are basically stoichiometric models that incorporate kinetic information using power-law approximation. Therefore, GMA systems are often closer to biochemical intuition, compared to S-system formalism. However, the GMA format does not allow the algebraic calculation of steady states, which is important for certain analyses. The SC formalism may be seen as a good tool in numerical simulations since it provides greater accuracy. However, similar to the GMA models, the straightforward algebraic analysis which can easily be done in S-system models is lacking in models based on the SC formalism. The lin-log model shares the advantage with the GMA format that the sum of terms is close to biochemical intuition and also has the benefit of the S-system format by allowing algebraic calculations of its steady states. However, it can not represent certain nonlinear behaviors since the structure is essentially linear (Savageau, 1998). The BST representations become more inaccurate for very high substrate concentrations, while the lin-log approximation results in greater errors for substrate values close to zero (Wang *et al.*, 2007; del Rosario *et al.*, 2008a). One should keep in mind that both BST and lin-log approximations have one aspect in common, namely that both approaches are local approximations and guaranteed to perform well as long as the variables stay within a reasonable range.

1.3 Kinetic model construction

After we collect the information of network structure and choose the mathematical model framework to describe the metabolic system, a symbolic model which is described as a set of ordinary equations can be derived. The next step is to assign numerical values to all parameters in the model. There is no unique recipe for the task of parameter estimation. In fact, the estimation problem is in general complicated and it continues to be the bottleneck of biomathematical modeling.

In this section, I will review some of the recent methods developed for parameter estimation: the forward (bottom-up) approach, estimation from steady-state data, and inverse (top-down) modeling using time-series data. The nature of suitable data for each type of estimation is rather different, and so are the methods of analysis. One should note that none of these approaches will be completely replaced by the others. Instead, they will and should complement each other. In the future, a combined strategy may become the standard, because it has much greater potential leading to suitable models than either approach by itself.

1.3.1 Forward or bottom-up modeling

Before the rapid development of high-throughput experimental tools, essentially all metabolic models were developed from “local” kinetic information of biochemical or physiological responses in a reductionist manner. Specifically, biologists around the world worked on characterizing one particular enzyme or transport step at a time in the traditional manner. They purified the enzyme, studied its characteristics, determined optimal temperature and pH ranges, and quantified cofactors, modulators, and secondary substrates. Isolated from these laboratory experimenters, modelers converted this information into a mathematical rate law. Once enough information had been collected of all rate laws, the modeler attempted to merge all this information into an integrative mathematical model. If done right this “forward” or “bottom-up” process might lead to a model representation of the pathway that exhibits the same features as reality, at least qualitatively, if not quantitatively (Voit, 2004; Goel *et al.*, 2006; Mao *et al.*, 2008). Some recent studies that used this forward approach in BST include: the TCA cycle in *Dictyostelium discoideum* (Shiraishi and Savageau, 1992); the citric acid cycle (Torres, 1994; Torres *et al.*, 1996); fermentation in *Saccharomyces cerevisiae* (Cascante *et al.*, 1995; Curto *et al.*, 1995; Sorribas *et al.*, 1995); purine metabolism (Curto *et al.*, 1997; Curto *et al.*, 1998a; Curto *et al.*, 1998b); the Maillard-glyoxylase network with formation of advanced glycation end products (Ferreira *et al.*, 2003); the trehalose cycle (Voit, 2003); the ferredoxin system with information from protein structure for model identification (Alves *et al.*, 2004); and sphingolipid metabolism in *Saccharomyces cerevisiae* (Alvarez-Vasquez *et al.*, 2004; Alvarez-Vasquez *et al.*, 2005; Alvarez-Vasquez *et al.*, 2007). In almost all of these cases, the strategy consisted of setting up a symbolic model, estimating local parameters, studying the integration of all individual rate laws into a comprehensive model, testing the model, and making refinements to some of the model structure and the parameter values.

While theoretically straightforward, there are several disadvantages of this approach. The main issue is that a considerable amount of local kinetic information is needed and that this information is often obtained from different organisms, different species, and collected under different experimental conditions. Therefore, more often than not the “integrated result” is not consistent with biological observations. Furthermore, this process of construction and refinement is very labor intensive and requires a combination of biological and computational expertise that is still rare (Goel *et al.*, 2006; Mao *et al.*, 2008).

1.3.2 Using steady-state data

If a system operates preferentially at a steady state, the parameters of the model can be estimated using steady-state data, including steady-state concentrations, fluxes of material flows at steady state, and logarithmic gains. Estimations of parameter values from steady-state data are generally based on observing how a biochemical system responds to (infinitesimally) small perturbations around the steady state. There are basically two approaches can be taken. First, parameter values can be obtained by direct experimental measurements of how a variable affects the fluxes coming in and going out of the metabolite pool. Suppose the flux rate and metabolite concentrations in steady state of one particular biochemical process are known. One can then slightly alter the concentration of a variable systematically while keeping the other variables constant. The result of these experiments can be plotted as flux rate versus metabolite concentrations in logarithm coordinate. Thus, in the case of power-law systems, the kinetic order of the variable can be measured easily as the slope of the line in the logarithmic plot obtained by linear regression (*e.g.*, (Wanders *et al.*, 1984; Curto *et al.*, 1997; Curto *et al.*, 1998a; Curto *et al.*, 1998b)). Under ideal circumstances, sufficient experimental measurements can be collected to allow the regression analysis. However, the data usually contain noise and consist of only a few measurements, which make the regression more vulnerable to experimental uncertainties. Second, parameter values can be estimated by experimental measurements of logarithmic gains (*e.g*. (Kacser and Burns, 1979; Sorribas and Cascante, 1994)). This approach is based on perturbing variables in the interesting portion of the pathway and recording the corresponding changes after perturbations. The information about modulation including flux rates and concentrations is collected to calculate the kinetic orders.

1.3.3 Inverse or top-down modeling

Much of the information necessary for parameter estimation depends not only on steady-state measurements or simple perturbations around the steady-state, but on measurements for all metabolites at sequential points in time that may include considerable deviations from the steady state. Modern high-throughput techniques of biology are capable of producing this type of time series data and have begun to offer distinct alternative options for modelling metabolic systems, namely the "top-down" or "inverse" approach. The experimental tools which allow the generation of dynamic metabolite concentration profiles presently include nuclear magnetic resonance (NMR), mass spectrometry (MS), high performance liquid chromatography (HPLC), and flow cytometry (see review in (Voit, 2004)). In contrast to the "local" data obtained from traditional experiments, the clear advantages of using "global" data are that the information is collected within the same organism, obtained under the same experimental condition, and sometimes even *in vivo*. These data contain enormous information on the structure and regulation of the biological system they describe. However, this information is mostly implicit, and it is very challenging to extract it from these data because the complexity and nonlinearity of biological networks. There are several distinct challenges of this approach, some of which are readily anticipated, while others are surprising and puzzling. I describe these challenges in detail in the next section.

1.3.4 Challenges of the top-down modeling approach

The challenges of model identification from time series data are both on the biological and the computational sides. They can be generally categorized in four problem areas, namely: data related issues, model related issues, computational issues, and mathematical issues (Voit, 2004; Voit *et al.*, 2005; Voit *et al.*, 2006b).

Data related issues

Typical biological datasets usually contain noise, measurement errors, and are seldom complete. Consider a biological dataset containing the concentration measurements of variables (metabolites) of interest over time. For example, n variables (metabolites) $X_1, X_2, \ldots, X_i, \ldots, X_n$ are measured and for each metabolite a time series consisting of m time points $t_1, t_2, \ldots, t_k, \ldots, t_m$ has been observed. Therefore, the dataset can be represented as an $m \times n$ matrix where n denotes the total variables and m denotes the total time points. There are several scenarios regarding missing data points. First, the data points are sparsely missing. Second, the measurements of all variables at a certain time point t_k are missing, which corresponds to the missing of a whole row in the matrix. The situation can happen when the experimentalists miss the collection of sampling at a certain time point. Third, the entire traces of some known variables are missing due to technical limitations or simply undetectable because the concentrations are too low. This situation corresponds to the missing of the whole column in the matrix. Fourth, potentially important system components are not measured or the investigators are not even aware of these components. Therefore, those variables are not measured nor been included in the model. This is typically the cause of "leakage" or some unexpected phenomena seen in the profile or the model. Among these scenarios, the first and second situations are relatively easy to tackle but last two are rather difficult.

Even if the time series are complete, they are usually noisy. Furthermore, another problem of the data is uncertainties about the particular experimental conditions at the time of observation. For instance, external influences like temperature may perturb the reaction mechanism. Therefore, it is very important that the temperature is carefully monitored. Besides, a good understanding of all sources of inaccuracy inherent in the experimental apparatus and measurement is needed. These uncertainties should be taken into account as these will affect the parameter estimation and predictive accuracy of the resulting model. The other potential problems in the dataset are that the data matrix is ill-posed, which may be caused by collinearity between time series data, or that the time series is non-informative, *e.g.*, consists essentially of constant time profiles.

Model related issues

The inverse problem requires a mathematical model that captures the dynamics of the data in a suitable fashion. However, there is an unlimited variety of nonlinear structures and mathematical formulations that could be potential candidates for the optimal data representation. I have introduced some of the modeling frameworks and their pros and cons in Section 1.2. Here I highlight the specific challenges in model selection in top-down modeling approaches.

It seems that there are good reasons for selecting particular model formalisms which are proposed as representations of the underlying chemical reactions. However, this mechanistic approach is not always appropriate. The reasons are, first, that it is usually not the case that the high-throughput time series data are of sufficient quality to be able to suggest the underlying reaction mechanism. Second, sometimes the underlying mechanisms which generate the data are little known. Third, traditional kinetic rate functions, such as the Michaelis-Menten rate law, are not necessary the best choice for *in vivo* data (Savageau, 1995). In this case, the aim of nonlinear modeling is somewhat different; it may be more appropriate to take a generic approach. That is, to choose models which are more or less crude abstractions of reality based on criteria like: ability to capture certain mathematical features of a data set, simplicity of representing the data, mathematical tractability, interpretability of mathematical results within the biological realm. Such criteria may practically more important than deep theory. It is clear that this selection is supported by rational considerations but that it also involves abstractions, assumptions, arbitrariness and, to some degree, personal taste.

Computational issues

The estimation process itself is very challenging computationally. The first typical problem is computational capacity, which is characterized by the size and complexity of the system and usually translates into the number of equations and variables in the model. In addition, because the describing models are usually nonlinear and typically formulated as systems of differential equations, the optimization of their parameters is far more complex than in linear regression and there is seldom an analytical solution (Mendes and Kell, 1998). Corresponding nonlinear methods are usually not straightforward and lead to challenging issues, such as slow algorithmic progress toward the error minimum and lacking convergence or convergence to local minima due to the complicated error surfaces. Furthermore, the integration of differential equations is usually needed during the optimization process. The integration may be very consuming especially when the system is stiff. Other computational challenges include the distinction between direct and indirect effects, characterization of intermediate steps and time delays, consideration of heterogeneity, and stochasticity of biological systems, which is seldom addressed in nonlinear models.

Mathematical issues

A further source of problems comes from issues of mathematical redundancy in the models. These derive from the fact that different sets of parameter values can produce responses that fit the experiment data about equally well, for instance, due to numerical compensation between a rate constant and the kinetic orders in a particular data fit (Berg *et al.*, 1996; Sands and Voit, 1996). Other issues include: distinctly different, yet numerically equivalent[i] solutions (Voit, 1992a); non-equivalent solutions with similar error; invalid assumptions regarding the chosen process descriptions; error compensation

[i] Here ‘equivalence’ of different mathematical solutions means that there exist transformation groups under whose action the solutions remain unchanged.

within and among flux descriptions and within and among equations. I will describe some of these issues in greater detail in Chapter 6 (Section 6.1).

Despite these challenges, the inverse approach based on *in vivo* time series data is certainly worthwhile, because these data are the most accurate reflections of what cells and organisms really do, in a global manner. Therefore, the development of methods to overcome these challenges is extremely important.

1.3.5 Current solution strategies of top down modeling

Responding to the challenges outlined above, the development of modeling techniques using global dynamic data has focused on the following tasks: (1) The development of strategies for the pre-handling and diagnosis of input time series data; (2) the choice of symbolic models that capture the dynamics of biological systems and are mathematically tractable; (3) the actual algorithmic development of methods for extracting information from (often noisy) biological time series data sets; and (4) the creation of diagnostic tool to avoid mathematical compensation within or between terms in order to find more valid model. I will briefly describe some of the current achievements in these tasks as following.

Data preprocessing

One of the most frequent data related issues in top-down modeling is that biological time series data are incomplete or not even available for some of the molecular species. An extreme case in this category is concept map modeling, which our group recently proposed as a useful link between experimental biology and biological systems modeling and analysis (Goel *et al.*, 2006). Concept map modeling leads to very uncertain time series on which inference and hypothesis generating schemes are based. Concept map modeling requires the collaboration between biologist and modeler. Based on the known or hypothesized connectivity and regulatory information regarding a static concept map, the biologist designs a regulated connectivity diagram of processes comprising the biological system of interest and also provides semi-quantitative information on stimuli and measured or expected responses of the system. The modeler converts this information through methods of forward and inverse modeling into a mathematical construct that can be used for simulations and to generate and test new hypotheses. Then the biologist-modeler team collaboratively interprets the results and devises improved concept maps. Our group is presently developing a Matlab® based software package BSTBox, to support this concept and various other modeling activities (Goel, 2008).

Symbolic models selection

The S-system and GMA models in BST framework have been shown to be a promising representation for biological systems modeling (see Section 1.2.4 for review). Therefore, throughout this chapter I will primarily focus my discussions on BST representations and their parameter estimation algorithms.

Optimization algorithms

As a consequence of the pressing needs and high rewards, many groups around the world have begun to develop optimization algorithms for inverse tasks of parameter estimation. However, so far none of these methods is perfect, or even sufficiently effective, for the majority of realistic cases. The computational solutions to biological inverse problems typically require a combination of techniques that include methods to attack the main problem of optimizing parameter values as well as supporting algorithms, such as methods for circumventing the costly integration of differential equations, smoothing overly noisy data, constraining the parameter search space, or reducing the complexity of the inference task. These techniques will be reviewed in detail in Section 1.4.

Diagnostic tools for mathematical redundancy

Mathematical redundancies in the model may occur within or between fluxes and equations. The compensation between fluxes can be partially avoided if each of the fluxes in the model is obtained. These techniques will be reviewed in detail in Chapter 6 (Section 6.2).

1.4 Parameter estimation techniques in top-down modeling approach

In this section I review some of the recently developed techniques in top-down parameter estimation for BST models.

1.4.1 Repeatedly solving differential equations

Most prominently, solving inverse problems requires the development of efficient algorithmic methods for determining optimal estimates. Many of the standard methods involve solving the differential equations directly, which requires a lot of computational effort. As an example indicative of the problem at hand, consider a direct attempt to estimate the parameters of a five-variable system of ordinary equations from noise-free time series data with a genetic algorithm (Kikuchi *et al.*, 2003). This group used a cluster of 1,040 CPUs, which ran for ~10 hours for each loop of the estimation program. Needing 7 loops, the entire estimation time thus was roughly 70,000 PC-hours.

Analyzing this dire situation, the distinct tasks within the optimization were clocked in detail with the result that parameter searches involving differential equations are very time consuming because easily 95% of the time spent is used on integrating the equations, while relatively little time is used to compute gradients toward the optimal estimates (Voit and Almeida, 2004). In fact, if the underlying model is stiff, the computation time may increase to almost 100%, and even if the model is not stiff, the likelihood is high that some trial solutions during the algorithmic process could make it stiff (Voit and Almeida, 2004). Therefore, it is very important to speed up the evaluation of differential equations.

Slope estimation and decoupling of the differential equations significantly alleviate the problem. An early implementation of this method was accomplished by manually estimating slopes from observed time series data and substituting them for the derivatives in the differential equations (Voit and Savageau, 1982a; Voit and Savageau, 1982b; Voit, 2000a). This substitution entirely eliminates the need to integrate differential equations, because the estimation is now executed on systems of algebraic equations. Furthermore, the equations become uncoupled so that they can be assessed in parallel or one at a time. Voit and Almeida (Voit and Almeida, 2004) used this slope-estimation-decoupling strategy for improving efficiency to avoid the need for solving the differential equations in S-system format. The set of equations was then used with a nonlinear search method to estimate parameter values. The slope-estimation-decoupling idea has subsequently been combined with various methods such as genetic algorithms, simulated annealing, swarm methods, interval analysis, and a number of hybrid methods. One drawback of this approach is that, if the data are noisy, it may not be easy to obtain good measurements or estimates of the slopes. The slope estimation methods will be reviewed in detail in Section 1.4.2. However, it may still be advantageous to use this approach, since the roughly obtained estimates may be used as good initial guesses for standard nonlinear optimization methods. Other advantages of the decoupling approach are reviewed in Voit and Almeida (Voit and Almeida, 2004). An application of the slope-estimation-decoupling strategy is described in detail in Chapter 2 (Section 2.2.2).

In a different implementation, the decoupling allowed solving and fitting of one differential equation at a time instead of solving the entire system. Maki *et al.* (Maki *et al.*, 2002) proposed this “step-by-step” strategy and Kimura *et al*. (Kimura *et al.*, 2004; Kimura *et al.*, 2005) introduced a similar concept called “decomposition,” which decomposes the large network inference problem into sub-problems. In both methods, the variables contributing to the single differential equation being integrated are substituted with the actual observed time series data or with smoothed analogues and thus used as off-line inputs to the decoupled system. This approach significantly reduced the computation time. For instance, using the same artificial five-variable datasets as Kikuchi *et al*. (Kikuchi *et al.*, 2003) did, Kimura and co-workers ran the algorithm on a single CPU with far less computing time requiring only about 59 minutes to optimize each subproblem.

A drawback of decoupling and decomposition approaches is that each subproblem is solved independently, a procedure which does not allow the exchange of information between subproblems. For instance, the variables serving as off-line data in one equation are actually solved in another equation. Thus, if the value of one variable is updated during optimization, the information should be incorporated into optimization processes of the other subproblem. This feature is especially important when there is considerable noise. Kimura *et al*. (Kimura *et al.*, 2005) proposed to solve the decomposed subproblems simultaneously using a cooperative coevolutionary algorithm. Since the decomposed subproblems interact with each other through their calculated time series data, the inferred model is more likely to represent the dynamics.

In order to reduce the number of numerical integration steps, Matsubara *et al.* (Matsubara *et al.*, 2006) proposed to use a radial basis function network (RBFN) for parameter estimation. RBFN is a type of artificial neural network (ANN) that uses radial basis functions as activation functions; it has been shown to be able to approximate nonlinear time series data effectively (Rank, 2003). In order to examine the performance of RBFN, Matsubara and co-workers proposed two schemes: one is using a simple genetic algorithm (SGA) with numerical integration, and the other is RBFN with simple GA included in the input data selection phase. Both schemes were examined in metabolic pathways using Michaelis-Menten equations. While SGA improves the fitness between parameterized model and time series data and integrates every time during optimization, RBFN predicts the optimal parameter values by learning the relationship between parameters and fitness values using slopes to replace derivatives and integrates the system only once at the last step. Therefore, numerical integrations used to evaluate the fitness are reduced from many to one. The results indicated that the RBFN scheme halved the computation time and increased the optimization successful rate.

An alternative approach avoiding numerical integration is a modified collocation method, which converts ordinary differential equations into algebraic equations which directly adopt the measured data to approximately yield dynamic profiles at sampling points. This approximation not only reduces computation time, but also decouples the equations so that parallel computation is allowed for the parameter estimation. This modified collocation method was combined with hybrid differential evolution (HDE) to determine the global solution of an estimation task (Tsai and Wang, 2005). Again, applying this type of “uncoupling” strategy in combination with other estimating methods reduced the computation time dramatically.

1.4.2 Slope estimation

As a crucial part of the slope-estimation-decoupling strategy, decent estimates of the slopes are required, but they are not always easy to obtain. If the data are more or less noise-free, simple linear interpolation, splines (de Boor, 1978; de Boor *et al.*, 1993; Green and Silverman, 1994), B-splines (Seatzu, 2000), the so-called three-point method (Burden and Faires, 1993), or even hand fitting (Voit and Savageau, 1982b) is effective. If the data are noisy, it is useful to smooth them, because the noise tends to be magnified in the slopes. Established smoothing methods again include splines, as well as different types of filters. Artificial neural networks (ANNs) have been shown to be useful in a number of applications of biochemical pathways modeling (Almeida, 2002). Voit and Almeida (Almeida and Voit, 2003; Voit and Almeida, 2004) proposed the data preprocessing with a “universal function” that is computed by training an ANN. The main advantage of using ANN to smooth the time traces is that the resulting universal function can be made to fit the data arbitrarily closely and that it has an algebraic format for which the slope can be computed straightforwardly (Mendes and Kell, 1996; Almeida, 2002). Furthermore, the universal output function provides an unlimited number of interpolated data points within the time interval of interest. Other advantages of ANN are reviewed in Almeida (Almeida, 2002) and Voit and Almeida (Voit and Almeida, 2004). The ANN method was shown to determine the smoothed traces very efficiently even if the data contained considerable noise, as long as the true trend was well represented. However, the interpolating function resulting from the ANN solution is a superposition of sigmoidal functions and has the tendency to lead to artifacts in the derivatives, which cause slight, but undesirable bias during the smoothing process, even when the deviations are not visually obvious in the smoothed traces.

Another popular filter is the Savitzky-Golay or Whittaker filter which was proposed over eighty years ago (Whittaker, 1923). Much more recently, Eilers presented a matrix form of this older implicit method call a "perfect filter" (Eilers, 2003). Vilela and co-workers further explored the use of Rényi's second-order entropy of the cross-validation error entropy as optimization criterion for configuring the Whittaker-Eilers smoother (Vilela *et al.*, 2007). The filter, implemented in the software AutoSmooth, can be used to extract signals and derivatives from time series with non-stationary noise structure.

1.4.3 Constraining the parameter searching space

To ensure that the results of a parameter estimation fall within reasonable ranges, constraining the maximally permitted values of parameters is usually needed throughout the optimization processes, or even for guessing the initial values. The simplest way of constraining a parameter value is to restrict the range of each parameter in the model. For instance, in BST representations, the structural features of a system are mapped onto parameters of models in a unique fashion as described in Section 1.2.4. Therefore, if the network structure is known, whether the kinetic order of a variable X_j is positive, negative, or zero, could be determined immediately by characterizing its influence (activation, inhibition, or no effect) on variable X_i. Furthermore, the rate constants in BST are always non-negative. Particularly in metabolic pathway, the kinetic orders are real numbers with typical values between -1 and +2.

Parameter values could also be constrained by other values in the equations. For instance, the values of production and degradation term in S-system models (Eq. (1.4)) could be constrained by the derivatives (or slopes) to some degree after decoupling. Since these two terms on the right hand side of Eq. (1.4) are always non-negative, if the slopes are negative, the values of degradation term must be greater than or equal to the absolute value of slopes in order to make production terms non-negative. Inversely, if the slopes are positive, the values of production term must be greater than or equal to the value of slopes to ensure the degradation term positive. Detail description of this application will be reviewed in Chapter 2 (Section 2.3.2).

Some other supporting techniques aim to reduce the parameter searching space including: Kutalik *et al*. (Kutalik *et al.*, 2007) characterized a one-dimensional basin of attraction containing the true optimum with minimal error; Tucker and Moulton (Tucker and Moulton, 2006) proposed a method based on interval analysis which allows exhausting searches of the entire set of parameter values with a finite number of steps; Tucker *et al*. (Tucker *et al.*, 2007) used constraint propagation to find the possible ranges of parameter values, thus significantly constraining the parameter search space.

1.4.4 Reducing the complexity of the inference task

The typical approach of modeling is to collect network information and translate the wire-diagram to a symbolic model, where there is only limited number of parameters since the biological systems are usually sparsely connected (*cf.* (Jeong *et al.*, 2000) and see Section 1.5 for detail description). However, when the topology of the system is unknown or only partially known, one can only derive a full symbolic model with all free parameters. When the system is relatively small, it is feasible to explore all possibility to find the optimum. When the number of variables and parameters grows, all methods of parameter estimation eventually run into problems caused by "combinatorial explosion," which makes the estimation process extremely difficult and the solutions problematic. This explosion can be tamed to some degree by constraining the connectivity within the system by systematically identifying the network structure or gradually "pruning" unlikely connection during optimization process. The structure identification techniques will be reviewed in detail in Section 1.5. In this section, I focus only on the parameter pruning methods.

The rationale behind the pruning techniques is closely related to the characteristic of BST models. As briefly mentioned in Sections 1.2.4, structure identification tasks can be translated into parameter estimation problems if the parameter values directly map to the network, as it is the case with BST representations. To recall this mapping, the kinetic orders g_{ij} and h_{ij} for S-systems quantify the regulatory effect of variable X_j on the production or degradation of variable X_i. If the magnitude of the corresponding kinetic orders are very small or close to zero, the connection between variable X_j and the dynamics of X_j is likely to be negligible. Therefore, these low intensity connections can be purged during optimization, which not only helps to detect a reasonable and parsimonious model of the true pathway structure, but also reduces the parameter search space for further optimization.

The simplest manner of "pruning" a possibly highly connected network is to define a threshold for the absolute value of each type of parameter, below which values are set to zero (Voit and Almeida, 2004; Vilela *et al.*, 2008). In addition, since the likelihood that a variable exists in both the production and degradation terms with non-zero values in the S-system model is low, the smaller of the kinetic orders is more likely to be zero and the value of the other one is adjusted accordingly (Voit and Almeida, 2004).

Some authors have suggested more sophisticated methods for this pruning process. As an extension of the objective functions described before, various articles have applied sums of the absolute values of kinetic orders as a penalty term in the cost function. Thus, this basic pruning method for BST models penalizes all small kinetic orders and prevents the model from finding false-positive interactions that unrealistically inflate the model (Kikuchi *et al.*, 2003; Voit and Almeida, 2003). To improve this condition further, Kimura and co-workers (Kimura *et al.*, 2004; Kimura *et al.*, 2005) introduced a different penalty term by rearranging kinetic orders in ascending order based on their absolute values. Furthermore, accounting for the observation that very few factors modulate both the production and degradation of a specific variable, Noman and Iba (Noman and Iba, 2005b) proposed an alternative representation of the penalty term.

No matter what kind of penalty term is chosen, pruning approaches have a common drawback. Namely, the weighted coefficient in the penalty term needs to be carefully tuned since it affects the results of the structure identification task. So far there is no clear guidance about how to set suitable penalty weights. Stochastic ranking may be used to alleviate this difficulty since it aims to balance the error and penalty term in the objective function (Runarsson and Yao, 2000). However, this method requires an additional parameter to define the probability of the error term for comparisons in ranking. Cho *et al*. (Cho *et al.*, 2006) proposed a distinctly different way to retain the sparseness feature in biological pathways without adding extra terms to the objective function, namely the S-tree representation. The S-tree is a tree representation of the S-system, where the number of sub-trees corresponds to the number of ordinary differential equations in the system. Each sub-tree is divided into two parts; the left part represents the production term and right part represents the degradation term. The depth of the S-tree is always three and the root node at depth zero. Since S-tree modeling is intrinsically suitable for representing sparse networks, an S-tree together with genetic programming has the potential to infer network topology and find parameter values in a more efficient way without any *a priori* knowledge or adding penalty term. To avoid assigning a coefficient weight to the penalty term, Liu and Wang recently proposed an alternative method based on multi-objective optimization (Liu and Wang, 2008). Instead of minimizing the residual error using a single objective function either in concentrations or slopes, they minimized the concentration error, slope error, and interaction measure simultaneously. The authors proved that the algorithm guarantees the minimum solution for the constrained problem to achieve the minimum interaction network for the inference problem. The approach avoided assigning a penalty weight for sums of magnitude of kinetic orders.

The pruning methods are used in the optimization problem that determine the parameter values, as described in the next section.

1.4.5 Algorithms for determining optimal parameter estimates

The parameter estimation task is traditionally formulated as a function optimization problem that minimizes an objective function measuring a generalized distance between experimental data and model predictions. The Euclidean distance is the most commonly used and often refers to a least-squares error criterion. Other fitness evaluation methods include information based criteria (Shin and Iba, 2003; Noman and Iba, 2006). Two objective functions are typically used for parameter estimation in BST models: a concentration error based objective function and a slope error based objective function (*e.g.* (Tsai and Wang, 2005)). The concentration error based objective function is a straightforward calculation of the sum of squared distances between the metabolite measurements and the predictions. The simulation profiles are usually obtained by applying a numerical integration method to solve the differential equations like Eq. (1.1). The integration process can be computational costly, especially if the system is stiff (see

Section 1.4.1). As an alternative, the slope error based objective function employs the decoupling technique as described in Section 1.4.1 and uses the slope information for evaluating fitness of the function. That is, it calculates the sum of squared errors between the measured slopes from the raw data (or upon smoothing) and the predicted slopes.

In Section 1.4.4 I review some pruning methods which improve the objective function and constraining the connectivity during the optimization process. Independent of pruning, the most prominent methods for parameter estimation from time series data can generally be grouped as: gradient-based methods, stochastic search algorithms, and others that do not belong to the first two groups. Several articles have been published in the recent literature describing computational methods for the inverse problem of extracting information from time series data using BST, but no method so far has risen to the top as the clear general winner in terms of efficiency, robustness and reliability. I will review some of these optimization methods in the following paragraphs and summarized in Table 1.1.

Gradient-based Algorithms

Some of the commercial gradient-based methods have been applied in a novel fashion for finding the parameter values using BST models. Marino and Voit (Marino and Voit, 2006) proposed an algorithm which comprises three modules: model generation, parameter estimation or model fitting, and model selection. The initial plausible models are generated in a step-by-step manner upon decoupling and limiting connectivity (see Section 1.5.5 for detail). After the set of ODEs is decoupled, each differential equation is fitted separately using the Levenberg-Marquardt while replacing the other variables with raw data of smoothed traces.

Kutalik *et al*. (Kutalik *et al.*, 2007) proposed a Newton-flow optimization method for parameter estimation in S-system models. The method starts with decoupling the differential equations and setting up an objective function for each equation. The next step is to select suitable start guesses and bounds for parameters and run a Newton method to obtain several points in the parameter space that correspond to reasonable solutions. The authors found that the solution space contains a one-dimensional attractor. Thus standard regression allowed them to estimate the parameters of this attractor. Afterward, the Newton method was performed again using the initial guesses lying on the estimated attractor to find the true optimal of the parameter values. The interesting feature of this method is that most (or maybe even all) good parameter solutions seem to lie on one-dimensional manifolds within the high-dimensional parameter space. Optimization along this curve is comparatively easy. A potential problem of the method is that the original initial guesses for the parameters must lie within the basin of attraction of the one-dimensional manifold. Otherwise, each run may lead to disjoint sections of the parameter space.

Because biological systems are usually nonlinear, the problem of parameter estimation can be stated as a nonlinear programming problem (NLP) subject to nonlinear differential-algebraic constraints (Moles *et al.*, 2003). Because of its nonlinear and constrained nature, this inverse problem is usually non-convex. Therefore, most of the traditional nonlinear algorithms involving gradient methods run the risk of getting trapped in local optima, depending upon the degree of system nonlinearity and the initial starting point (Mendes and Kell, 1998). Polisetty *et al.* (Polisetty *et al.*, 2006) employed a branch-and-bound algorithm to convert the inverse problem in the GMA formalism into a convex optimization problem in order to obtain a global solution.

Stochastic search algorithms

There are many different kinds of stochastic methods for global optimization. They include evolutionary computation (EC), simulated annealing (SA), adaptive stochastic methods, clustering methods, and other meta-heuristics, such as ant colony optimization (ACO) and particle swarm optimization (PSO). These algorithms have been applied to parameter estimation tasks with the goal of finding global solutions, especially in the context of identifying the structures of gene regulatory networks (Moles *et al.*, 2003).

Evolutionary computation techniques, also known as biological inspired methods, include genetic algorithms (GAs), evolutionary programming (EP), evolution strategies (ES), genetic programming (GP), as well as many of their variants. They are attractive because they have an increased potential of finding global optima. Genetic algorithms (GAs) have been shown to be useful and practical in parameter estimations of biological systems (*e.g*. (Mendes and Kell, 1996; Park *et al.*, 1997; Moles *et al.*, 2003; Voit and Almeida, 2003)). Using the conventional simple genetic algorithm (SGA), Tominaga *et al.* inferred parameter values of a small network, but only with a very limited number of parameters and the convergence rate was low (Tominaga *et al.*, 2000). The SGA typically has two problems: early convergence in the fast stage of the search and evolutionary stagnation in the last stage. Kikuchi *et al.* (Kikuchi *et al.*, 2003) enhanced the SGA by using a more robust real coded genetic algorithm (RCGA) and improved the conventional cost function by adding a penalty term to prune unlikely connections in the system using the S-system formalism. In addition, they employed a novel crossover method and introduced a gradual optimization strategy in the procedure. The results showed the algorithm successfully inferred the network structure with faster convergence rate, optimization speed, and with more predictable parameters compare to the traditional GA. However, the approach turned out to be computationally very costly because of numerical integration of the entire differential equations (see Section 1.4.1).

Other modifications were made to improve the efficiency of SGA using time series data in S-system form. Examples include: a hybrid algorithm of SGA with Modified Powell method (Okamoto *et al.*, 1998); a hybrid algorithm of SGA for static Boolean networks applied to an S-system with steady state and temporal data (Maki *et al.*, 2001); and a combination of RCGAs with unimodal normal distribution crossover

(UNDX) and minimal generation gap (MGG) to optimize parameters in S-systems (Ueda *et al.*, 2001; Ueda *et al.*, 2002; Nakatsui *et al.*, 2003). Daisuke and Horton optimized an S-system model with a distributed genetic algorithm (DGA) with "scale-free" properties (Daisuke and Horton, 2006). Ho *et al*. (Ho *et al.*, 2007) proposed an intelligent two-stage evolutionary algorithm (iTEA), which used an intelligent GA (IGA) to solve decomposed ODEs independently, then combined all solutions from each subproblem and used an orthogonal experimental design-based simulated annealing algorithm (OSA) to refine the solution.

Spieth and co-worker (Spieth *et al.*, 2004b; Spieth *et al.*, 2004a) proposed a memetic algorithm (MA) consisting of two parts: a local search (LS) with an evolutionary strategy (ES) for parameter estimation, and a global GA based search (GS) framework for structure identification, where the former is embedded within the later part. They tested the algorithm in an S-system model and the results showed that MA was better suitable for inferring genetic networks than a standard ES or GA. In follow-up work, they showed that the feedback coordination from LS to GS can even improve the performance of MA (Spieth *et al.*, 2005).

Kimura *et al*. (Kimura *et al.*, 2004) used an evolutionary algorithm called Genetic Local Search with distance independent Diversity Control (GLSDC) combined with the decomposition strategy using the S-system formalism. The proposed method included an estimation technique for the initial gene expression level and enabled the reconstruction of medium-scale genetic networks with noisy data. They also showed that the combination with a cooperative coevolutionary algorithm can further improve the accuracy of prediction (Kimura *et al.*, 2005). Okamoto's group also proposed evolutionary search techniques, such as the Network-Structure-Search Evolutionary Algorithm (NSS-EA) and its variant, the Grid-Oriented Genetic Algorithm Framework (GOGA Framework). They employed an S-system as the underlying mathematical model and used a GA as search engine to infer network structure (Morishita *et al.*, 2003; Ono *et al.*, 2004; Imade *et al.*, 2005).

Noman and co-workers recently incorporated their previously developed techniques and presented a memetic algorithm for inferring gene regulatory networks (Noman and Iba, 2005b; Noman and Iba, 2005a; Noman and Iba, 2005c; Noman and Iba, 2006; Noman and Iba, 2007). They used differential evolution (DE) along with a hill-climbing local-search method in their evolutionary algorithm. An information criterion-based fitness evaluation was introduced instead of the conventional least squared error approach.

Tsai and Wang (Tsai and Wang, 2005) used hybrid differential evolution (HDE) for estimating a satisfactory, though not optimal solution, and then used the solution as the initial value for a gradient-based optimization method to obtain refined solutions. As described in Section 1.4.1, they used a modified collocation method to avoid direct numerical integration. In their recent work, they also implemented HDE combined with a multiple-objective optimization approach (see Section 1.4.4 for review) to inferring biochemical networks in S-system format (Liu and Wang, 2008).

Genetic programming (GP) has also been employed to find the topology of metabolic pathway from time-series data (*e.g.* (Koza *et al.*, 2001)). The ordinary GP is not always effective in finding the parameter values because the method relies mainly on the combination of randomly generated constants. Sagamoto and Iba (Sakamoto and Iba, 2001) therefore used a least mean square (LMS) method along with ordinary GP to improve the situation, using an S-system as one example. Their results showed that the fitness values decreased faster in the early phase with the LMS method compared to the non-LMS method, since the former seemed to provide a better seeds for GP search. In contrast to GA algorithms, which usually require defining equations before optimization, GP provides a general approach for finding arbitrary equations from time series data without any knowledge of the equation.

Table 1.1. Comparison of representative algorithms for inverse problems in BST models.

Authors	Year	Main Methods	Model Format	Examples
Kikuchi *et al.*	2003	• Simple genetic algorithm (SGA) • Penalty term • Numerical integration	S-system	(a)
Voit and Almeida	2004	• Decoupling • ANN smoothing • Slope approximation	S-system	(b)
Kimura *et al.*	2004	• Decomposition method (Maki *et al.* 2002 proposed similar idea) • Numerical integration with local linear regression	S-system	(a) (c)
Kimura *et al.*	2005	• Decomposition • Cooperative coevolutionary algorithm	S-system	(a) (c) (d)
Tsai and Wang	2005	• Modified collocation method (converted to algebraic equation) • Decoupling	S-system	(a) (e)
Marino and Voit	2006	• Decoupling • Limit connectivity • Gradient-based method	S-system	(b)
Daisuke and Horton	2006	• Distributed genetic algorithm (DGA) • Scale-free property	S-system	(a) (f)
Cho *et al.*	2006	• S-tree based genetic programming (GP)	S-system	(a) (g) (h)
Kim *et al.*	2006	• Genetic programming to estimate slopes and avoid numerical integration	S-system	(b)
Tucker and Moulton	2006	• Interval analysis	S-system	(a) (b) (i)
Polisetty *et al.*	2006	• Branch-and-bound strategy	GMA	(j) (k)
Noman and Iba	2007	• Information criteria-based fitness evaluation • Differential evolution (DE) along with local search heuristics	S-system	(a) (l) (m)
Gonzalez *et al.*	2007	• Simulated annealing (SA)	S-system	(b) (n)
Kutalik *et al.*	2007	• Newton-flow method	S-system	(b) (c)
Tucker *et al.*	2007	• Constraint propagation	S-system GMA	(b) (j)
Marin-Sanguino *et al.*	2007	• GMA optimizer • Geometric programming	GMA	(k) (o)
Liu and Wang	2008	• Modified collocation and slope approximation for each subsystem	S-system	(a) (c) (p) (q) (r)

(a) Five variables gene regulatory network (Hlavacek and Savageau, 1996); (b) Four variables didactic system (Voit and Almeida, 2004); (c) Thirty variables system (Maki *et al.*, 2001); (d) cDNA microarray data of *Thermus thermophilus* HB8 strains; (e) Cascade three variable system (Tsai and Wang, 2005); (f) Experimental data (GDS404) (Daisuke and Horton, 2006); (g) Yeast anaerobic fermentation pathway (Vera *et al.*, 2003); (h) SOS DNA repair system in *E. coli* (Sutton *et al.*, 2000); (i) Three variable system (Voit, 2000a); (j) Branched pathway with several feedback inhibition (Voit, 2000a); (k) Anaerobic fermentation pathway in *Saccharomyces cerevisiae* (Curto et al., 1995); (l) Twenty variable system (Noman and Iba, 2007); (m) Yeast cell-cycle microarray data (Cho *et al.*, 1998); (n) cadBA in *E. coli* (Kuper and Jung, 2005); (o) Tryptophan operon in *E. coli* (Xiu *et al.*, 2002); (p) Kinetics model of ethanol fermentation (Wang *et al.*, 2001); (q) Circadian oscillations of period protein in drosophila (Ingalls, 2004); (r) Embryonic gene regulatory network in zebrafish (Huang *et al.*, 2006).

Sugimoto and co-workers (Sugimoto *et al.*, 2005) implemented GP along with adding a penalty term to the cost function and introducing numeric mutations to the conventional procedure. They tested this method by predicting two equations of metabolic reaction regarding adenylate kinase and phosphofructokinase in Michaelis-Menten formation, the equation of which is hard to derive if the underlying mechanism is not known. While their results showed that the algorithm can predict the equations which have relatively simple forms, the method is still very time consuming.

Kim *et al*. (Kim *et al.*, 2006) adopted a pre-processing symbolic regression step in GP to avoid time consuming numerical integration, since the estimation of slopes for each time series data point can be obtained from the results of GP. Cho and co-workers (Cho *et al.*, 2006) took advantage of the fact that GP has an evolving tree structure for given data and proposed S-tree based genetic programming for parameter estimation and structural identification in S-system models. As introduced in Section 1.4.4, this approach intrinsically accounts for the sparseness of the biological network. Therefore, even though no *a priori* knowledge about the network is known, the S-tree based GP can still identify the underlying structure rather efficiently without adding a penalty term in the objective function.

As seen in the previous paragraphs, a considerable number of recently published papers applied evolutionary algorithms to tackle the inverse problem using BST models. However, so far there is no clear comparison among these algorithms regarding their efficiency, robustness, and accuracy. Moles *et al*. (Moles *et al.*, 2003) compared some stochastic global optimization methods using the case study of a biochemical model, which consisted of 36 parameters and was formulated as a set of eight ODEs. Nevertheless, the model was formulated as Michaelis-Menten type equations, not in BST representations. Spieth *et al*. (Spieth *et al.*, 2006) compared six evolutionary algorithms in three model frameworks: linear weight matrices, S-systems, and H-systems, where one fitness function was used to evaluate the convergence of algorithms. A comprehensive comparison of EAs is still needed.

Simulated annealing (SA), colony optimization (ACO), and particle swarm optimization (PSO) are also stochastic optimization methods. Simulated annealing, a physically inspired method, is created in a way to simulate the cooling process of metal or glass. SA can behave as a global or local optimization search and automatically switches from a global to a local search when the "temperature" goes down. Gonzalez *et al*. (Gonzalez *et al.*, 2007) adapted SA for S-systems parameter estimation from time series data. They tested the algorithm using three artificial datasets under the assumption that the structure was known or unknown, by solving the entire set of ODEs or upon decoupling. They also applied the algorithm to a real biological system.

Ant colony optimization (ACO) was inspired by the behavior of ants in finding short paths from their colony to food sources. ACO is a probabilistic technique for solving computational problems which can be reduced to finding good paths through nodes in a graph. Zuñiga *et al*. (Zuñiga *et al.*, 2008) adapted ACO for S-system models by treating each metabolite as a node in a graph and inferring how other nodes were connected to it. Their preliminary results showed that ACO was able to reduce the connectivity of the network. They also proposed an enhanced aggregation pheromone system (eAPS), which is an extension of ACO, for parameter estimation tasks.

Particle swarm optimization (PSO) is a stochastic, population-based evolutionary computation algorithm. The original form of PSO algorithm, which is motivated by social-psychological principles such as bird flocking and fish schooling, was first described by Eberhart and Kennedy (Eberhart and Kennedy, 1995). In PSO, each potential solution is represented as a particle. A collection of potential solutions is called a swarm which consists of particles that fly around in a multidimensional search space. During flight, each particle adjusts its position according to its own experience and also collaborates with its neighboring particles through communication. When a particle encounters a promising solution, the surrounding area of the solution is further explored by the swarm. Therefore, PSO combines local search methods with global search methods. Naval *et al.* (Naval *et al.*, 2006) further adapted PSO to scan the parameter space of a BST model

<u>Other algorithms</u>

Some methods that aim to reduce the parameter search space using BST formalisms are described in Section 1.4.3 (Tucker and Moulton, 2006; Kutalik *et al.*, 2007; Tucker *et al.*, 2007). Specifically for linear parts of pathways, a technique of "peeling" terms (Lall and Voit, 2005) can be applied to models in BST to convert the nonlinear parameter estimation task into a series of linear regression tasks.

Some other methods which were developed recently for inverse problems in biological systems are (Stelling *et al.*, 2002; Yeung *et al.*, 2002; Liao *et al.*, 2003; Rank, 2003; Thomas *et al.*, 2004; Tran *et al.*, 2005; Srividhya *et al.*, 2007). However, these methods are not yet implemented for BST applications.

Among these parameter estimation methods, so far no single method has risen to the top and can be declared the clear winner. A cursory comparison of parameter estimation algorithms in biochemical pathways has been published, but only two networks were considered and both of them were not yet implemented for BST applications (Moles *et al.*, 2003). del Rosario and co-workers (del Rosario *et al.*, 2008b) recently proposed a benchmarking framework for comparing current parameter estimation algorithms using BST frameworks. The details of the framework will be described in Chapter 5 (Section 5.1).

1.5 Inference of network structure

As mentioned in Sections 1.1, the traditional approach of modeling is to collect network information and build up a stoichiometric model by converting the "wiring diagram," which describes the metabolic pathway, into a set of equations. The translation can more or less reflect the real system as long as the diagram is more or less complete. However, in reality, the information on network connectivity is sometimes only partially known and seldom fully understood. Therefore, the identification of components and interactions of the system that need to be included in the model and to develop hypotheses regarding the network structure is a crucial step in the modeling process (Veflingstad *et al.*, 2008).

The need for valid system identification can be described in three aspects. First, wrong hypotheses regarding variables and interactions to be included in the model may lead to wrong interpretations of the results. Second, overly complex models may provide good approximations to the time series data used for estimation but are unlikely to perform as well when tested on new datasets, due to over-fitting. Third, the inclusion of too many components and interactions in the model eventually run into problems caused by combinatorial explosion, which means that any computational techniques will eventually be overwhelmed by the rapidly increasing number of equations, variables, and interactions between variables in large systems.

Fortunately, biology offers a counteracting and very beneficial feature: namely the likelihood that a real biochemical networks is fully connected is very low, because most metabolites are connected only to a limited number of other metabolites, and usually through fewer than four or five reactions (Jeong *et al.*, 2000; Wagner and Fell, 2001; Milo *et al.*, 2002). To take advantage of this fact of nature, it must therefore be our goal to precede any estimation attempt with a concerted effort to limit objectively the number of candidate (structural and functional) connections within a system, thereby *a priori* reducing the parameter space that must be searched. This feature is crucial since structure identification and parameter estimation are closely related tasks which complement each other. In this section, I review some of the structure identification techniques, namely the determination of the Jacobian matrix after small perturbations around operating points, direct observation of time profiles, a correlation-based approach, a “simple-to-general and general-to-specific” modeling strategy, and various additional methods using time series data within the BST framework.

1.5.1 Methods based on the Jacobian matrix

Much of the information necessary for identifying network structure depends on dynamic experiments. One type of these experiments is the measurement of transient responses of the system after a small perturbation from steady state. When the perturbation is close enough to the equilibrium, the system behaves roughly linearly. Thus, the Jacobian matrix of the corresponding linearization can be determined and reveals the connectivity of the network. In the past two decades, several attempts have been made to obtain the Jacobian matrix from experimental observations. Chevalier and co-workers (Chevalier *et al.*, 1993) solved the Jacobian by applying multilinear least-square fitting to perturbed data. This approach is straightforward but very sensitive to noise and missing data points, because the crucially important differencing procedure is prone to generating large errors.

To avoid instabilities due to numerical differentiation, Chevalier and co-workers suggested using an integral representation, which expressed this solution in terms of eigenvectors and eigenvalues and solved the equation using nonlinear regression (Chevalier *et al.*, 1993). The advantage of this approach is that no differentiation is needed and hence the slopes do not need to be estimated. However, the drawback of this method is that the fit to a sum of exponentials with undetermined exponents is sometimes numerically problematic, and the nonlinear regression does not necessarily provide a solution which fits the data.

To overcome this difficulty, Sorribas *et al.* (Sorribas *et al.*, 1998) suggested to reformulate the integral representation of the target function by reducing it to a multilinear regression problem. As the result, the eigenvalues of the Jacobian in the previous method can be easily calculated. However, the computation of eigenvalues is very sensitive to noise and rounding error, making the method unreliable unless the multiplicities of the eigenvalues are exactly known. In order to avoid this problem, Díaz-Sierra and co-workers (Díaz-Sierra *et al.*, 1999) proposed a variation to the previous methods, in which they directly obtained the Jacobian by expanding it in its Taylor-series without searching for eigenvalues. This methods yielded faster convergence.

All methods mentioned in the previous paragraphs are based on linear approximation, which is valid as long as the perturbation from steady state remains relatively small. On one hand, the range of deviation needs to be small enough to yield a sufficiently accurate representation. However, on the other hand, the perturbation must be large enough to generate measurable responses. To alleviate this dilemma, Veflingstad *et al.* (Veflingstad *et al.*, 2004) suggested using the entire time course and fit the data in a piecewise linear fashion, using as an example an S-system within BST. In this case, the time series is subdivided into appropriate time intervals and within each subset, linearization is computed about a chosen operating point. Therefore, instead of focusing on one operating point, most reference states are different from the steady state. The results show the piecewise approach is more likely to capture the relationship between variables in the system and can tolerate larger perturbations. The authors also showed that the collection of estimated coefficients resulting from different variations of linearization provided very strong clues about which variables were likely to be involved in a given equation and which were not. These clues reflect likely parameter ranges or likely constraints on parameter values of the true model. However, this method does not identify parameter values *per se*. For instance, as shown in Eqs. (6)-(8) of their paper (Veflingstad *et al.*, 2004), it does not allow a distinction between various combinations of g_{ij} and h_{ij} in the S-system form because only their difference is being assessed as a single parameter. However, with this formation, if information of the Jacobian matrix and both the concentration and fluxes at steady state are known, the difference between g_{ij} and h_{ij} can be directly calculated (Kitayama *et al.*, 2006). If the difference has a magnitude that is significantly different from 0, it is likely that one of the kinetic orders is zero, because it is rare that a variable influences both production and degradation of the same variable. Therefore, if one can detect which connection may be omitted, the kinetic order can be computed straightforwardly.

Hatzimanikatis, Floudas and Bailey (Hatzimanikatis *et al.*, 1996b; Hatzimanikatis *et al.*, 1996a) indirectly contributed to the topic of structure identification per linearization by optimizing not only the production of yield in an S-system at steady state, as it has been done many times (*e.g.* (Voit, 1992c; Torres and Voit, 2002)), but by also optimizing its regulatory structure. This numerical and structure optimization task led to a mixed integer linear programming (MILP) approach, for which standard software is available.

1.5.2 Direct observation

Unlike the previous methods for determining the Jacobian matrix by examining the linear properties on small amplitude perturbation near one or more operating points, the network connectivity can be deduced to some degree from direct observations on responses to perturbations of arbitrary amplitude made at different locations in the network. Vance and co-workers (Vance *et al.*, 2002) proposed a strategy based on perturbing different components in a network and showed that relationships between the perturbed component and the remaining components may be deduced by observation of features in the response profile. These features include the order and size of the extreme values of the unperturbed components in response to the perturbed component, and the initial slopes of the time series at the perturbation. The former reflects the topological distances among the perturbed components and the remaining components in the network, while the latter reveals whether the components are directly affected by the perturbed one or not. This distinction is accomplished by checking if the initial slopes are nonzero or zero upon perturbation. Vance *et al.* showed that this approach works well in some artificial networks including branching, feedback, and regulatory interactions. This method was also applied to an *in vitro* experiment with a glycolysis system, where the authors measured concentration changes in the reactor following impulse changes of different reaction metabolites (Torralba *et al.*, 2003). From the experimental time series data the authors were able to identify some of the causal connectivities among the metabolites in the reaction pathway. Even though the method performed well in the synthetic time series and with experimental data from relatively small systems, this approach may not be applicable to more complicated networks, where the interpretation of profiles and the network reconstruction must be expected to be much harder.

1.5.3 Correlation-based approach

Some other approaches have been suggested for the reconstruction of chemical reaction networks. Arkin and co-workers (Arkin and Ross, 1995; Arkin *et al.*, 1997) showed how correlations among components measured in the system may be used to infer or reconstruct a chemical reaction pathway. The approach, termed correlation metric construction (CMC), is based on the calculation and analysis of a time-lagged multivariate correlation function of time series data that are subjected to a series of random, large amplitude changes in the input concentration. The correlation information is used to construct the distance matrix and interpreted using a two-dimensional graph obtained with a projection technique called multidimensional scaling (MDS). The graph represents the connectivity and the strength of interactions among the species in the network. For instance, the shorter distances in the graph imply stronger connections and longer distance represent weaker interactions. The approach was also tested experimentally on a part of an *in vitro* glycolysis system containing eight enzymes and fourteen metabolites (Arkin *et al.*, 1997). Along the same lines, Samoilov and collaborators (Samoilov *et al.*, 2001) proposed methods, named entropy metric construction (EMC) and entropy reduction method (ERM), for the analysis of correlations between species from time series data and the inference of their underlying network.

1.5.4 Simple-to-general and general-to-specific modeling

As briefly mentioned in the introduction of this section, overly complex models may fit the data very well since increasing the complexity of the model naturally allows more freedom to provide a better fit to the data, for instance, in terms of the sum of squared errors. However, an over-inflated model typically does not perform well when tested on new data. This problem is known as over-fitting. One approach for restricting model complexity and to find the optimal model size is to add a penalty term to the cost function that is minimized. The optimal model can then be determined by finding the one that minimizes the aggregate cost function (Akaike, 1974). The consequent problem of using this approach is how to proceed with convergence with respect to model complexity. One approach, namely “simple to general,” calls for starting with a simple model and adding one term at a time until a minimal cost function is found (*e.g.* (Judd and Mees, 1995)). In the opposite direction, the “general to specific” strategy initially includes everything possible in the model and then gradually eliminates terms until the minimum in the cost function is found (Hendry and Krolzig, 2003). Crampin and co-workers (Crampin *et al.*, 2004b; Crampin *et al.*, 2004a) used these two approaches of model constriction to extract kinetic information from time series data. Although their result suggested that the general-to-specific algorithm outperforms the simple-to-general approach, they indicated that when the number of chemical species included in the model is large (~10), the numbers of possible elementary reactions are massive thus making the computation difficult. Therefore, it is desirable to limit the size of the basic set below a reasonable upper bound using knowledge of the network connectivity, because metabolic networks are generally sparsely connected (Jeong *et al.*, 2000).

1.5.5 Using time series data

So far I have reviewed the methods of structure identification mainly based on the temporal data obtained from perturbations around the operating point or changes correspond to randomly inputs. In this section, I review methods of structure identification using time series data. The parameter estimation from time series data usually requires considerable computational effort, especially when the structure is unknown. Therefore, in addition to the task of inferring the topology itself, one important benefit of developing good structure identification strategies is to ameliorate the problem in parameter estimation by limiting the analysis to the most likely connections in advance and thus reduce the search space and providing good initial guesses.

For the identification of structure from time series data, the BST models seems particularly useful, especially if not much additional information about the metabolic network is available. The advantages and features of BST representations have been reviewed in Sections 1.2.4 and 1.4.4 and need no more description here.

In addition to the pruning techniques reviewed in Section 1.4.4, pruning can also be achieved based on biological insight. Almeida and Voit (Almeida and Voit, 2003) suggested making maximal use of other *a priori* biological information that might be available in addition to the time series data. As an example, Voit and Savageau (Voit and Savageau, 1982a) analyzed a yeast fermentation system in several variations that corresponded to hypotheses regarding the existence of specific processes and regulatory signals and studied the improvement in error with statistical methods.

In a more generic fashion of “inverse pruning,” and pursuing the “specific to general” strategy, Marino *et al*. (Marino and Voit, 2006) proposed an algorithm based on reconstructing equations in a gradual progress manner. First the set of differential equations is decoupled into single differential equations. The model generation scheme is then applied separately to each differential equation, starting from the minimal (and most parsimonious) model, and increasing the number of variables step by step automatically in the equations using the S-system representation, until a maximally allowed level of connectivity is reached. By choosing a modest connectivity index, the combination of plausible models is greatly reduced. Arguing that the vast majority of metabolites is involved in only a few reactions (Jeong *et al.*, 2000), this algorithm terminates much sooner than one might expect. In some sense, this method is similar to the "simple-to-general" approach described in the previous section.

Daisuke and Horton (Daisuke and Horton, 2006) also utilized the "scale-free" property of networks (Barabási *et al.*, 2000; Podani *et al.*, 2001) to restrict the connectivity in biological systems during optimization procedure. Their results showed that the restriction increased the conversion ratio while reducing the average number of generations and reducing both false positive and false negative estimations of links in the network. Zuñiga *et al*. (Zuñiga *et al.*, 2008) recently proposed to apply ant colony optimization (ACO) on the network inference problem using the S-system formalism. Their preliminary results showed that, starting with a fully connected network, ACO was able to recover the connectivity of the network.

1.6 Dissertation overview

In spite of the considerable amount of methods that have been proposed regarding the inverse modeling problem recently, every method has its pros and cons, and so far none of them can be declared as the clear general winner in terms of efficiency, robustness and reliability, for the majority of realistic cases. There are still challenges and open questions in the data related issues, model related issue, computational issues, and mathematical issues. Therefore, to develop improved methods for inverse modeling that are effective, fast, and scalable, this work proposes two novel algorithms, *Alternating Regression* (AR) and *Eigenvector Optimization* (EO), for parameter estimation and structure identification in metabolic pathways. A novel *3-way Alternating Regression* (3-AR) is also proposed here to parameter estimation in S-distributions. To integrate all existing techniques and make inverse modeling more effective, this work proposes an operational "work-flow" that guides the user through the estimation process, identifies possibly problematic steps, and suggests corresponding solutions based on the specific characteristics of the various available algorithms. Finally, the work described here discusses a recent *Dynamic Flux Estimation* (DFE) approach, which resolves open issues of model validity and quality beyond residual errors. The overview of corresponding chapters and appendices are shown in Table 1.2.

Table 1.2. Dissertation overview.

Chapter	Content	Related appendixes
2[i]	Parameter estimation in biochemical systems models with alternating regression	**A**: Additional documentation of parameter estimation using alternating regression in S-systems
3[ii]	Parameter estimation of S-distributions with alternating regression	
4[iii]	Parameter optimization in S-system models using eigenvector optimization	**B**: Additional documentation of parameter estimation using eigenvector optimization in S-systems
5[iv]	Inverse modeling approach and parameter estimation strategies	
6[v]	Conclusions and future work	

i. Adapted from: Chou, I-C., Martens, H., and Voit, E. O. (2006) Parameter estimation in biochemical systems models with alternating regression. *Theor. Biol. Med. Model.*, 3**,** 25.

ii. Adapted from: Chou, I-C., Martens, H., and Voit, E. O. (2007) Parameter estimation of S-distributions with alternating regression. *Stat. Operations Res. Transactions* (*SORT*), 31(1), 55-74.

iii. Adapted from: Vilela, M., Chou, I-C., Vinga, S., Vasconcelos, S. T. R., Voit, E. O., and Almeida, J. S. (2008) Parameter optimization in S-system models. *BMC Syst. Biol.*, 2,35.

iv. Some of the material was presented at International Conference on Molecular Systems Biology 2008 (ICMSB08) in the Manila, Philippines (Chou *et al.*, 2008).

v. Some of the material are adapted from: Goel, G., Chou, I-C., Voit, E. O. (submitted) System estimation from metabolic time series data.

CHAPTER 2

PARAMETER ESTIMATION IN BIOCHEMICAL SYSTEMS MODELS WITH ALTERNATING REGRESSION[ii]

2.1 Introduction

Novel high-throughput techniques of molecular biology are capable of producing *in vivo* time series data that are relatively high in quantity and quality. These data implicitly contain enormous information about the biological system they describe, such as their functional connectivity and regulation. The hidden information is to be extracted with methods of parameter estimation, if the structure of the system is known, or with methods of structure identification, if the topology and regulation of the system are not known. The S-system format within BST (see Chapter 1 (Section 1.2.4) for review) is recognized as a particularly effective modeling framework for both tasks, since it has a mathematically convenient structure and because every parameter has a uniquely defined meaning and role in the biological system. Due to the latter feature, the typically complex identification of the pathway structure reduces to a parameter estimation task, though in a much higher-dimensional space. Still, like most other biological models, S-system models are nonlinear, so that parameter estimation is a significant challenge. In this chapter, I discuss a novel method called *alternating regression* (AR), which is particularly effective in combination with a previously described decoupling technique (Voit and Almeida, 2004). AR is fast and rather stable, and performs structure identification tasks between 1,000 and 50,000 times faster than methods that directly estimate systems of nonlinear differential equations (*cf.* (Kikuchi *et al.*, 2003)).

[ii] This chapter is adapted from: Chou, I-C., Martens, H., and Voit, E. O. (2006) Parameter estimation in biochemical systems models with alternating regression. *Theor. Biol. Med. Model.*, 3, 25.

2.2 Methods

2.2.1 Modeling framework

As modeling framework for AR I use the S-system formulation within BST, which is especially suitable for AR, because each equation contains at most two terms. The significance of this fact will become evident later in this chapter. For the purposes of estimation, I assume that all independent variables, which are typically constant, are merged with the rate constants, so that the system contains as many equations as variables. Thus, the form to be parameterized is

$$\dot{X}_i = \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}},\ i = 1,2,\ldots,n. \tag{2.1}$$

The notation and parameters in this system were discussed in Chapter 1 (Section 1.2.4).

2.2.2 Decoupling of differential equations

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

$$\begin{aligned}
S_i(t_1) &\approx \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}}(t_1) - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_1),\\
S_i(t_2) &\approx \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}}(t_2) - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_2),\\
&\vdots\\
S_i(t_k) &\approx \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}}(t_k) - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_k),\\
&\vdots\\
S_i(t_m) &\approx \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}}(t_m) - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_m).
\end{aligned} \tag{2.2}$$

Thus, for the purpose of parameter estimation, the original system of *n* coupled differential equations can be analyzed in the form of *n*×*m* uncoupled algebraic equations

(Voit and Savageau, 1982b; Voit, 2000a). The uncoupling step renders the estimation of slopes a crucial step. Methods of slope estimation from raw data or upon smoothing have been reviewed in Chapter 1 (Section 1.4.2). In order to keep our illustration of AR as clean as possible, I initially assume that true slopes are available and elaborate on issues of experimental noise in Section 2.3.

2.2.3 Alternating regression

The decoupling of the system of differential equations permits the estimation of S-system parameters α_i, g_{ij}, β_i, and h_{ij} ($i, j=1,2,\ldots,n$) one equation at a time, using slopes and concentration values of each metabolite at time points t_k. The proposed method, called *alternating regression* (AR), has been used in other contexts such as spectrum reconstruction and robust redundancy analysis (Karjalainen, 1989; Oliveira *et al.*, 2004), but, to the best of my knowledge, not for the purpose of parameter estimation from time series. Adapted to our task of S-system estimation, AR works by cycling between two phases of multiple linear regression. The first phase begins with guesses of all parameter values of the degradation term in a given equation and uses these to solve for the parameters of the corresponding production term. The second phase takes these estimates to improve the prior parameter guesses or estimates in the degradation term. The phases are iterated until a solution is found or AR is terminated for other reasons.

In pure parameter estimation tasks, the structure of the underlying network is known, so that it is also known which of the S-system parameters are zero and which of the kinetic orders are positive or negative. Thus, the search space is minimal for the problem. Nonetheless, the same method of parameter estimation can in principle also be used for structure identification (see Chapter 1 (Sections 1.2.4 and 1.4.4) for review). In this case, the estimation is executed with an S-system where no parameter is *a priori* set to zero and all parameters are estimated. As an intermediate task, it is possible that only some of the structure is known. This information can again be used to reduce the search space. If it is known, for instance, that variable X_j does not affect the production or degradation of X_i, the corresponding parameter value g_{ij} or h_{ij} is set equal to zero, or X_j is taken out of the regression. One can thus reduce the regression task either by constraining the values of some *g*'s or *h*'s throughout the AR or by selecting a subset of regressors at the beginning, *i.e.*, by taking some variables out of the regression. Similarly, if a kinetic order is known to represent an inhibiting (activating) effect, its range of possible values can be restricted to negative (positive) numbers. This constraining of kinetic orders, while not essential, typically improves the speed of the search. It is imaginable that a kinetic order is constrained too tightly. In this case, the solution is likely to show the kinetic order at the boundary, which is subsequently relaxed.

2.2.3.1 Steps of the AR algorithm

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

{1} Let $\mathbf{L_1}$ denote an $m\times(n+1)$ matrix of logarithms of regressors X_i, defined as

$$\mathbf{L_1}=\begin{bmatrix} 1 & \log\left(X_1\left(t_1\right)\right) & \cdots & \log\left(X_i\left(t_1\right)\right) & \cdots & \log\left(X_n\left(t_1\right)\right) \\ 1 & \log\left(X_1\left(t_2\right)\right) & \cdots & \log\left(X_i\left(t_2\right)\right) & \cdots & \log\left(X_n\left(t_2\right)\right) \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ 1 & \log\left(X_1\left(t_k\right)\right) & \cdots & \log\left(X_i\left(t_k\right)\right) & \cdots & \log\left(X_n\left(t_k\right)\right) \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ 1 & \log\left(X_1\left(t_m\right)\right) & \cdots & \log\left(X_i\left(t_m\right)\right) & \cdots & \log\left(X_n\left(t_m\right)\right) \end{bmatrix}. \tag{2.3}$$

$\mathbf{L_1}$ is used in the first phase of AR to determine the parameter values of the production term. Additional information on the system, if it is available, reduces the width of $\mathbf{L_1}$. For instance, if X_2 and X_4 do not affect the production of X_1 in a four variable system, Eq. (2.3) reduces to

$$\mathbf{L_1} = \begin{bmatrix} 1 & \log\left(X_1\left(t_1\right)\right) & \log\left(X_3\left(t_1\right)\right) \\ 1 & \log\left(X_1\left(t_2\right)\right) & \log\left(X_3\left(t_2\right)\right) \\ \vdots & \vdots & \vdots \\ 1 & \log\left(X_1\left(t_k\right)\right) & \log\left(X_3\left(t_k\right)\right) \\ \vdots & \vdots & \vdots \\ 1 & \log\left(X_1\left(t_m\right)\right) & \log\left(X_3\left(t_m\right)\right) \end{bmatrix}. \tag{2.4}$$

Analogous to $\mathbf{L_1}$, let $\mathbf{L_2}$ denote the $m\times(n+1)$ matrix of regressors used in the second phase of AR to determine the parameter values of the degradation term. $\mathbf{L_1}$ and $\mathbf{L_2}$ are the same when the variables used in two phases of AR are identical.

{2} Compute the matrices

$$\mathbf{C_1} = \left(\mathbf{L_1}^{\mathrm{T}}\mathbf{L_1}\right)^{-1}\mathbf{L_1}^{\mathrm{T}}, \tag{2.5}$$

$$\mathbf{C_2} = \left(\mathbf{L_2}^{\mathrm{T}}\mathbf{L_2}\right)^{-1}\mathbf{L_2}^{\mathrm{T}}, \tag{2.6}$$

which are invariant throughout the iterative process.

{3} Select values for β_i and h_{ij} in accordance with experience about S-system parameters (*cf.* (Voit, 2000a): Ch. 5) and make use of any available information constraining some or all h_{ij}.

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

{5} Compute the m-dimensional vector $\mathbf{y_1} = \log\left(S_i(t_k) + \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_k)\right)$ $(k = 1, 2,\ldots, m)$ containing transformed "observations" on the degradation term[iii].

{6} Based on the multiple linear regression model

$$\mathbf{y_1} = \mathbf{L_1}\hat{\mathbf{b}}_1 + \boldsymbol{\varepsilon}_1, \tag{2.7}$$

[iii] It is possible to compute $\mathbf{y_1}$ for all n traces simultaneously so that $\mathbf{Y_1}$ becomes an $m\times n$ matrix with columns $\mathbf{y_1}$.

estimate the regression coefficient vector $\hat{\mathbf{b}}_\mathbf{1} = \begin{bmatrix} \log(\hat{\alpha}_i) & \hat{g}_{i1} & \hat{g}_{i2} & \cdots & \hat{g}_{in} \end{bmatrix}^\mathrm{T}$ by regression over the *m* time points. In other words, this step leads to an estimation of parameters in sets of equations of the type $y_{1,k} = \log(\hat{\alpha}_i) + \sum_{j=1}^{n} \hat{g}_{ij} \log\left(X_j(t_k)\right) + \varepsilon_{i,k}$.

Specifically, compute $\hat{\mathbf{b}}_\mathbf{1}$ as

$$\hat{\mathbf{b}}_\mathbf{1} = \left(\mathbf{L}_\mathbf{1}{}^\mathrm{T}\mathbf{L}_\mathbf{1}\right)^{-1} \mathbf{L}_\mathbf{1}{}^\mathrm{T}\mathbf{y}_\mathbf{1} = \mathbf{C}_1\mathbf{y}_\mathbf{1}, \tag{2.8}$$

according to Eqs. (2.3-2.5).

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

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

{9} Compute the *m*-dimensional vector $\mathbf{y}_\mathbf{2} = \log\left(\hat{\alpha}_i \prod_{j=1}^{n} X_j^{\hat{g}_{ij}}(t_k) - S_i(t_k)\right)$ containing the transformed "observations" associated with the production term.

{10} Based on the multiple linear regression model

$$\mathbf{y}_\mathbf{2} = \mathbf{L}_2\hat{\mathbf{b}}_\mathbf{2} + \boldsymbol{\varepsilon}_\mathbf{2} \tag{2.9}$$

and in analogy to step {6}, estimate the regression coefficient vector

$\hat{\mathbf{b}}_\mathbf{2} = \begin{bmatrix} \log(\hat{\beta}_i) & \hat{h}_{i1} & \hat{h}_{i2} & \cdots & \hat{h}_{in} \end{bmatrix}^\mathrm{T}$ by regression over the *m* time points as

$$\hat{\mathbf{b}}_\mathbf{2} = \mathbf{C}_2\mathbf{y}_\mathbf{2}. \tag{2.10}$$

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

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

At each phase of AR, lack-of-fit criteria are estimated and used for monitoring the iterative process and to define termination conditions. For the purposes here I use the sum of squared *y*-errors (SSE_1 and SSE_2) as optimization criteria for the two regression phases, *i.e.* one computes

$$log(SSE) = log\left(\sum_{k=1}^{m}\left(\mathbf{y_k} - \hat{\mathbf{y}}_\mathbf{k}\right)^2\right), \tag{2.11}$$

where $\hat{\mathbf{y}} = \mathbf{L} \times \hat{\mathbf{b}}$, **L** equals $\mathbf{L_1}$ or $\mathbf{L_2}$, and $\hat{\mathbf{b}}$ is the solution vector $\hat{\mathbf{b}}_\mathbf{1}$ or $\hat{\mathbf{b}}_\mathbf{2}$, estimated through regression and modified by constraints reflecting structural information. I use the logarithm of *SSE* because it is superior in illustrating small changes in the residual error. The overall flow of the method is shown in Figure 2.1.

It is known that collinearity may affect the efficiency of multivariate linear regressions. I therefore also implemented methods of principal component regression (PCR), partial least squares regression (PLSR) and ridge regression (Martens and Naes, 1989). For the cases analyzed here, these methods did not provide additional benefit.

2.2.3.2 Matrix computation representation of AR algorithm

The AR algorithm can be reformulated using the matrix computation representation. For the first phase of AR, $\mathbf{L_1}$ is a $m \times (n+1)$ matrix and $\mathbf{y_1}$ is a $m \times 1$ vector. The values of β_i and h_{ij} in the first iteration are guessed to obtained $\mathbf{y_1}$. The problem of finding the estimates then becomes a minimization problem

$$\min \left\|\mathbf{L_1 x_1} - \mathbf{y_1}\right\|_2, \tag{2.12}$$

where $\hat{\mathbf{x}}_\mathbf{1}$ can be computed as in Step {6}

$$\hat{\mathbf{x}}_\mathbf{1} = \left(\mathbf{L}_\mathbf{1}^\mathrm{T}\mathbf{L_1}\right)^{-1}\mathbf{L}_\mathbf{1}^\mathrm{T}\mathbf{y_1}. \tag{2.13}$$

However, this approach augments numerical error cause by floating point errors. As an alternative, QR decomposition can be used to avoid this situation as

$$\mathbf{L_1} = \mathbf{Q_1}\begin{pmatrix}\mathbf{R_1}\\ \mathbf{0}\end{pmatrix}, \tag{2.14}$$

$$\mathbf{R_1}\hat{\mathbf{x}}_\mathbf{1} = \left(\mathbf{Q}_\mathbf{1}^\mathrm{T}\mathbf{y_1}\right)(1:n+1), \tag{2.15}$$

$$\mathbf{\hat{x}_1} = \mathbf{R_1} \backslash \left[\left(\mathbf{Q_1^T y_1} \right) (1:n+1) \right], \tag{2.16}$$

where $\mathbf{\hat{x}_1} = \left[\log(\hat{\alpha}_i) \quad \hat{g}_{i1} \quad \hat{g}_{i2} \quad \cdots \quad \hat{g}_{in} \right]^{\mathrm{T}}$. Use the estimates $\mathbf{\hat{x}_1}$ as the initial guess for the second phase of regression, $\mathbf{y_2}$ can be computed as in Step {9}. The least square problem can be formulated as in Eq. (2.12) and computer in the same fashion as Eqs. (2.14-2.16)

$$\min \left\| \mathbf{L_2 x_2 - y_2} \right\|_2, \tag{2.17}$$

$$\mathbf{L}_2 = \mathbf{Q}_2 \begin{pmatrix} \mathbf{R}_2 \\ \mathbf{0} \end{pmatrix}, \tag{2.18}$$

$$\mathbf{R}_2 \mathbf{\hat{x}}_2 = \left(\mathbf{Q}_2^{\mathrm{T}} \mathbf{y}_2 \right) (1:n+1), \tag{2.19}$$

$$\mathbf{\hat{x}}_2 = \mathbf{R}_2 \backslash \left[\left(\mathbf{Q}_2^{\mathrm{T}} \mathbf{y}_2 \right) (1:n+1) \right], \tag{2.20}$$

where the estimates of β-term are obtained $\mathbf{\hat{x}_2} = \left[\log\left(\hat{\beta}_i\right) \quad \hat{h}_{i1} \quad \hat{h}_{i2} \quad \cdots \quad \hat{h}_{in} \right]^{\mathrm{T}}$. The matrices $\mathbf{L_1}$ and $\mathbf{L_2}$ generally are identically which include the measurements of all variables. However, since all or part of the structure information is known before the parameter estimation step, some of the variables are excluded from the regression to constrain the search space as described in Step {1}.

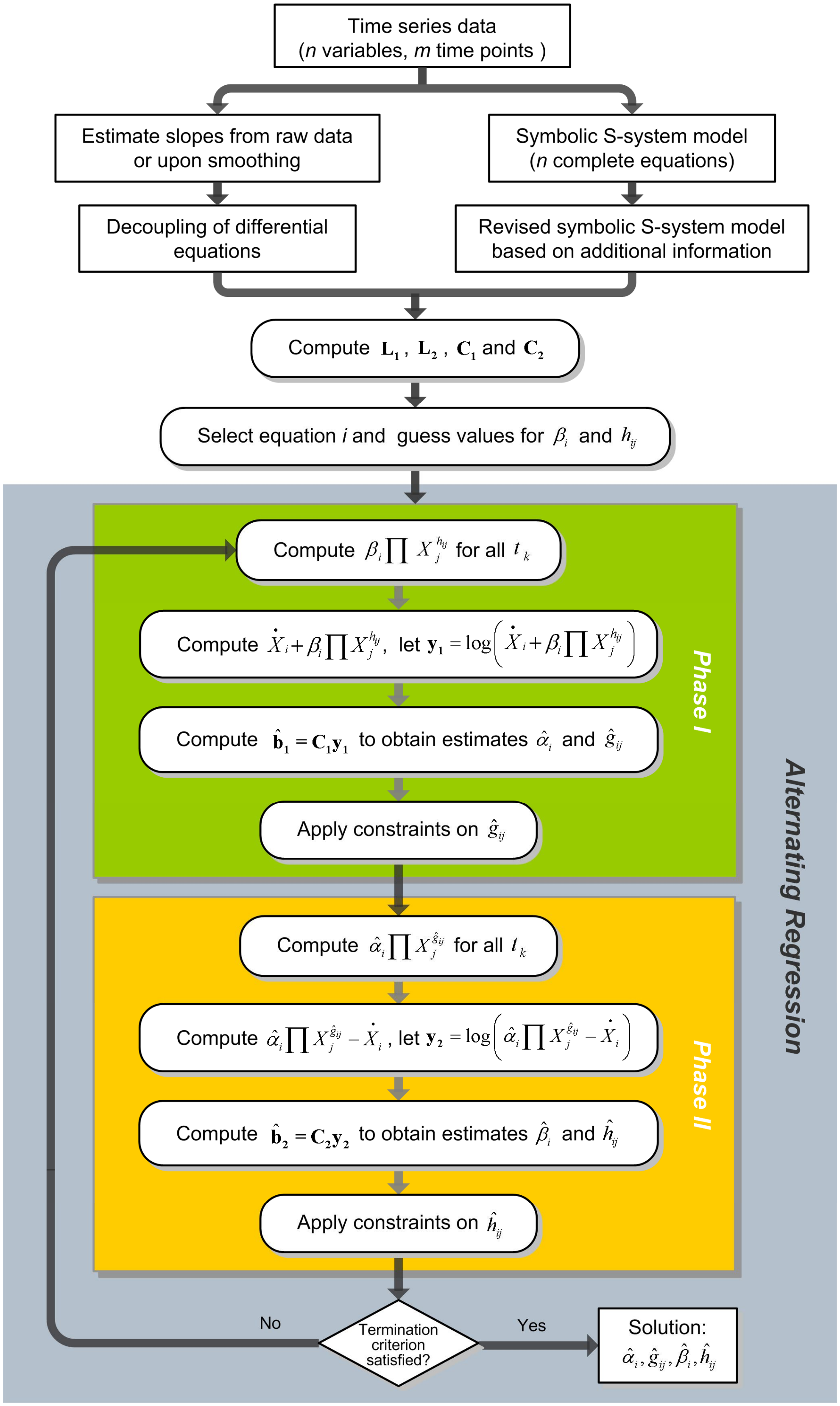

Time series data
(n variables, m time points)
Estimate slopes from raw data or upon smoothing
Symbolic S-system model (n complete equations)
Decoupling of differential equations
Revised symbolic S-system model based on additional information
Compute $\mathbf{L}_1$, $\mathbf{L}_2$, $\mathbf{C}_1$ and $\mathbf{C}_2$
Select equation i and guess values for β_i and h_{ij}
Compute $\beta_i \prod X_j^{h_{ij}}$ for all t_k
Compute $\dot{X}_i + \beta_i \prod X_j^{h_{ij}}$, let $\mathbf{y}_1 = \log\left(\dot{X}_i + \beta_i \prod X_j^{h_{ij}}\right)$
Compute $\hat{\mathbf{b}}_1 = \mathbf{C}_1 \mathbf{y}_1$ to obtain estimates $\hat{\alpha}_i$ and $\hat{g}_{ij}$
Apply constraints on $\hat{g}_{ij}$
Phase I
Compute $\hat{\alpha}_i \prod X_j^{\hat{g}_{ij}}$ for all t_k
Compute $\hat{\alpha}_i \prod X_j^{\hat{g}_{ij}} - \dot{X}_i$, let $\mathbf{y}_2 = \log\left(\hat{\alpha}_i \prod X_j^{\hat{g}_{ij}} - \dot{X}_i\right)$
Compute $\hat{\mathbf{b}}_2 = \mathbf{C}_2 \mathbf{y}_2$ to obtain estimates $\hat{\beta}_i$ and $\hat{h}_{ij}$
Apply constraints on $\hat{h}_{ij}$
Phase II
Alternating Regression
No
Termination criterion satisfied?
Yes
Solution:
$\hat{\alpha}_i, \hat{g}_{ij}, \hat{\beta}_i, \hat{h}_{ij}$

Figure 2.1. Logistic flow of parameter estimation by alternating regression.

2.3 Results and Discussion

For illustration purposes, I use a didactic system with four variables that is representative of a small biochemical network (Voit and Almeida, 2004). A numerical implementation with typical parameters is

$$\begin{array}{ll} \dot{X}_1 = 12X_3^{-0.8} - 10X_1^{0.5} & X_1(t_0) = 1.4 \\ \dot{X}_2 = 8X_1^{0.5} - 3X_2^{0.75} & X_2(t_0) = 2.7 \\ \dot{X}_3 = 3X_2^{0.75} - 5X_3^{0.5}X_4^{0.2} & X_3(t_0) = 1.2 \\ \dot{X}_4 = 2X_1^{0.5} - 6X_4^{0.8} & X_4(t_0) = 0.4 \end{array} . \qquad (2.21)$$

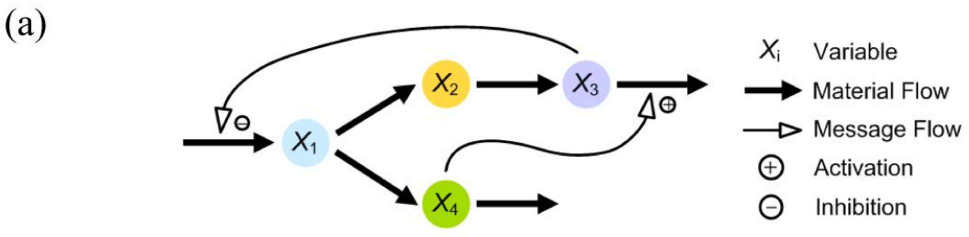
The system is first used to create artificial datasets that differ in their initial conditions (Table 2.1). In a biological setting, these may mimic different stimulus-response experiments on the same system. For example, they could represent different nutrient conditions in a growth experiment. Figure 2.2 shows the branched pathway and the symbolic model representation, along with a selection of time course data and slopes.

In order not to confuse the features of AR with possible effects of experimental noise, I use true metabolite concentrations and slopes and compute the latter directly from Eq. (2.21) at each time point. I initially assume that there are observations at 50 time points, but discuss cases with fewer points and with noise later.

In the following sections I describe the main results of this example using the AR algorithm. Some additional results are shown in Appendix A.

Table 2.1. Sets of initial concentrations used for the creation of artificial datasets.

Dataset	$X_1(t_0)$	$X_2(t_0)$	$X_3(t_0)$	$X_4(t_0)$
1	1.4	2.7	1.2	0.4
2	0.4	2.0	4.5	0.1
3	0.2	0.3	2.2	0.01
4	2.0	2.0	2.2	0.1
5	1.4	1	0.2	3.0
6	4.0	1.0	3.0	4.0



(b)

Variables \ Dependence	Production Term					Degradation Term				
	α	X_1	X_2	X_3	X_4	β	X_1	X_2	X_3	X_4
X_1	α_1	0	0	g_{13}	0	β_1	h_{11}	0	0	0
X_2	α_2	g_{21}	0	0	0	β_2	0	h_{22}	0	0
X_3	α_3	0	g_{32}	0	0	β_3	0	0	h_{33}	h_{34}
X_4	α_4	g_{41}	0	0	0	β_4	0	0	0	h_{44}

$\boldsymbol{\alpha}, \boldsymbol{\beta}$ Rate constants	🟪 Activation (positive)	🟨 Positive number
$\boldsymbol{g}, \boldsymbol{h}$ Kinetic orders	🟦 Inhibition (negative)	**0**s No affect

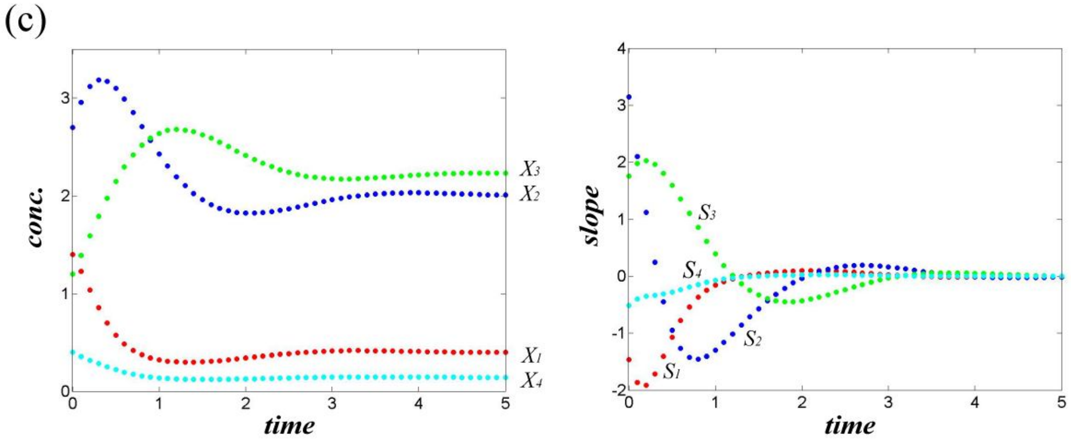


Figure 2.2. Test system with four dependent variables.
(a) Didactic system with four variables that represents a small biochemical network; (b) the symbolic model in S-system representation. The rate constants α_i and β_i are non-negative and the kinetic orders g_{ij} and h_{ij} are real numbers. A kinetic order of zero implies no effect of the corresponding variable X_j on X_i, whereas positive implies activating or augmenting and negative implies inhibiting; (c) time courses computed with initial values in Eq. (2.21) (use dataset 1 in Table 2.1) and its corresponding dynamics of slopes. Typical units might be concentrations (*e.g.*, in mM) plotted against time (*e.g.*, in minutes), but the example could as well run on an hourly scale and with variables of a different nature.

2.3.1 Performance of AR

Given the time series data of X_i and S_i at every time point t_k, the AR algorithm is performed for each metabolite, one at a time. Figure 2.3 summarizes various patterns of convergence observed. Generally one can classify the convergence patterns into four types: (1) convergence to the true value; (2) convergence to an incorrect value; (3) no convergence; typically the value of α_i (or β_i) continuously increases while all g_{ij} (or h_{ij}) gradually approach zero, while in some other cases g_{ij} and the corresponding h_{ij} increase (or decrease) in a parallel manner; (4) termination during AR, due to some of the observations $\mathbf{y_1}$ (or $\mathbf{y_2}$) taking on complex values.

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

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

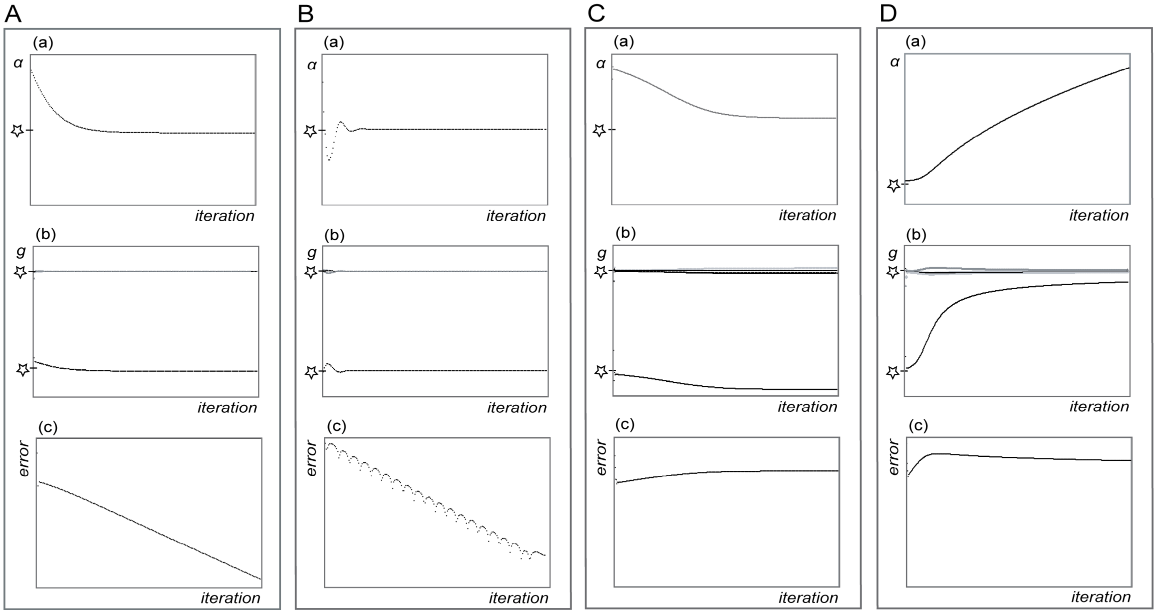


Figure 2.3. Generic patterns of convergence of AR.
Panel A: monotonic convergence to the true value; Panel B: non-monotonic convergence to the true value; Panel C: convergence to a different value; Panel D: no convergence. Row (a): rate constant α; Row (b): kinetic order g; Row (c): log of residual error. The asterisk represents the true value of α or g. See Section 2.3.1 for detailed description.

2.3.2 Patterns of convergence

The speed and pattern of convergence depend on a combination of several features, including initial guesses for all parameters and the datasets. Overall, these patterns are very complicated and elude crisp analytical evaluations. This is not surprising, because even well-established algorithms like the Newton method can have basins of attraction that are fractal in nature (*e.g.*, (Epureanu and Greenside, 1998)). A detailed description of some of these issues, along with a number of intriguing color plates describing well over one million ARs, is presented in Appendix A (Section A.3).

Effect of initial parameter guesses

Figure 2.4 combines results from several sets of initial guesses of β_i and h_{ij} (the results of the second phase of AR are not shown, but are analogous). The data for this illustration consist of observations on the first variable of datasets 4, 5 and 6 (see Table 2.1). These are processed simultaneously as three sets of algebraic equations at 50 time points. Thus, the parameters α_1, g_{13}, β_1, and h_{11} of the equation

$$\dot{X}_1 = \alpha_1 X_3^{g_{13}} - \beta_1 X_1^{h_{11}} \tag{2.22}$$

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

Figure 2.4A(a) shows the "heat map" of the convergence, where the x- and y-axes represent the initial guesses of h_{11} and β_1, respectively, and the color bar represents the number of iterations needed for convergence. Since I am using noise-free data, the residual error should approaches 0, which corresponds to $-\infty$ in logarithmic coordinates. I use -7 instead as one of the termination criteria, which corresponds to a result very close to the true value, but allows for issues of machine precision and numerical inaccuracies. Once this error level is reached, AR stops and the number of iterations is recorded as a measure for the speed of convergence. The unusual shape of a "martini with olive" is due to the following. The deep blue outside area indicates an inadmissible domain, where the initial parameter guess causes one or more of the terms

$S_i(t_k) + \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_k), k = 1, 2, ..., m$ in step {5} to become negative, so that the logarithm,

$\mathbf{y}_1$, becomes a complex number and the regression cannot continue. The line separating admissible and inadmissible domains is thus not smooth but shows the envelope of several pieces of power-law functions where the β-term is smaller than the (negative) slope at some time point. The "olive" inside the glass is also inadmissible. In this case, the chosen initial value causes the term $\hat{\alpha}_i \prod_{j=1}^{n} X_j^{\hat{g}_{ij}}(t_k) - S_i(t_k), k = 1, 2, ..., m$ in step {9} to become negative, so that $\mathbf{y}_2$ becomes complex and AR terminates during the second phase. This type of termination usually, though not always, happens during the first iteration. In order to prevent it, one may *a priori* require that

$$S_1(t_k) + \beta_1 X_1^{h_{11}}(t_k) > 0 \tag{2.23}$$

for every t_k, such that the logarithm is always defined. This is possible through the choice of a sufficiently large value for the initial guess of β. The magnitude of β should be reasonable, however, because excessive values tend to slow down convergence. As a matter of practically, one may start with a value of 5 or 10 and double it if condition in Eq. (2.23) is violated.

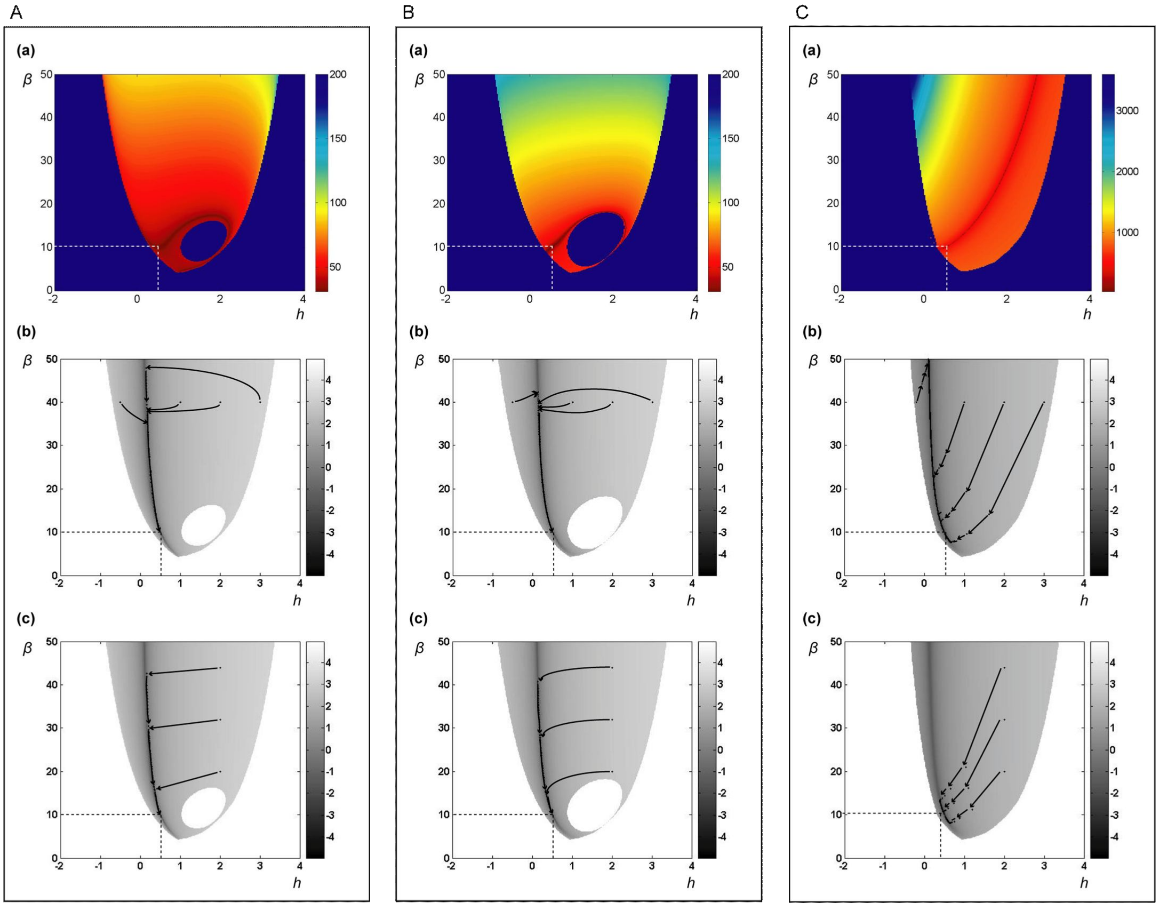


Figure 2.4. Summary of convergence patterns of AR.
Panel A: all variables are initially used as regressors and constraints are imposed afterwards; Panel B: regression with the "union" of variables of both terms; Panel C: only those variables that are known to appear in the production or degradation term, respectively, are used as regressors. Row (a): speed of convergence; the color bars represent the numbers of iterations needed to converge to the optimum solution; Rows (b) and (c): 2D view of the error surface superimposed with convergence trajectories with different initial values of β and h; the color bars represent the value of *log*(*SSE*). The intersections of dotted lines indicate the optimum values of parameters β and h.

Use of different variables as regressors

Panel A in Figure 2.4 shows results where all variables are initially used as regressors, but where their kinetic orders are constrained to zero after each iteration, if they are known to be zero. As alternatives, Panels B and C show results of using different variable combinations as regressors under otherwise identical conditions. In Panel B, both phases of AR use all variables as regressors that appear in either the production or the degradation term of the equation. In Panel C I make full use of pre-existing knowledge of the pathway structure and include in each term only the truly involved variables. Interestingly, this choice of regressors has a significant effect on convergence.

Compared with the case in of Figure 2.4A(a), the speed of convergence is slower in Figure 2.4B(a) and much slower in Figure 2.4C(a), even though this represents the “best-informed” scenario. The time needed to generate the graphs in Figures 2.4A(a), 2.4B(a), and 2.4C(a) for all shown 60,000 initial values is 72, 106, and 1,212 minutes, respectively. Thus, supposing that roughly half of the start points are inadmissible and require no iteration time, the average convergence time in Figure 2.4A(a) is 0.144 seconds, whereas it is 0.212 seconds in Figure 2.4B(a) and 2.424 seconds in Figure 2.4C(a). The pattern of convergence is affected by the datasets used. As another example, Figure 2.5 shows results of regressions with dataset 5.

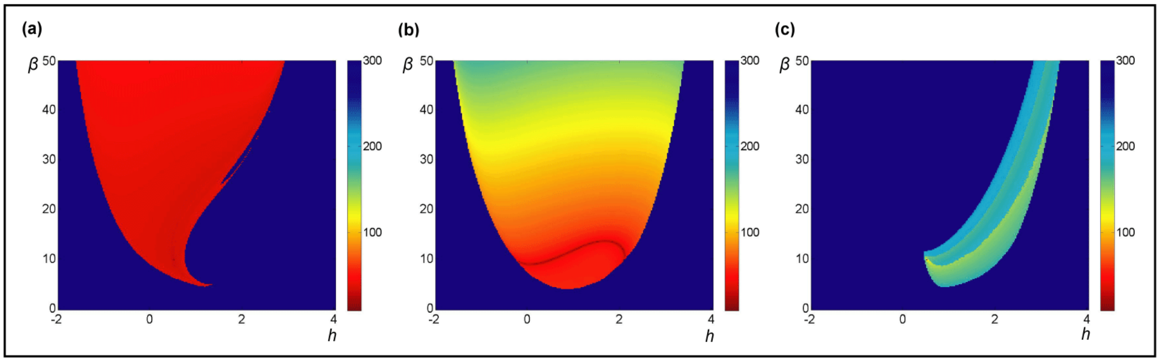


Figure 2.5. Convergence of AR for data set 5.
(a) Use all variables as regressor with secondary constraints; (b) use “union” variables as regressors that appear in either the production or the degradation term of the equation; and (c) use fully informed variable selection and include in each term only the truly involved variables.

Error surface

Rows (b) and (c) in Figure 2.4 Panels A, B, and C show heat maps of *log*(*SSE*), where darker dots indicate smaller errors. The true minimal value of *log*(*SSE*) for our noise-free data is -∞, but for illustration propose, I plot it only to -5. Pseudo-3-D graphs of the error surface are shown in Figure 2.6 with views from two angles.

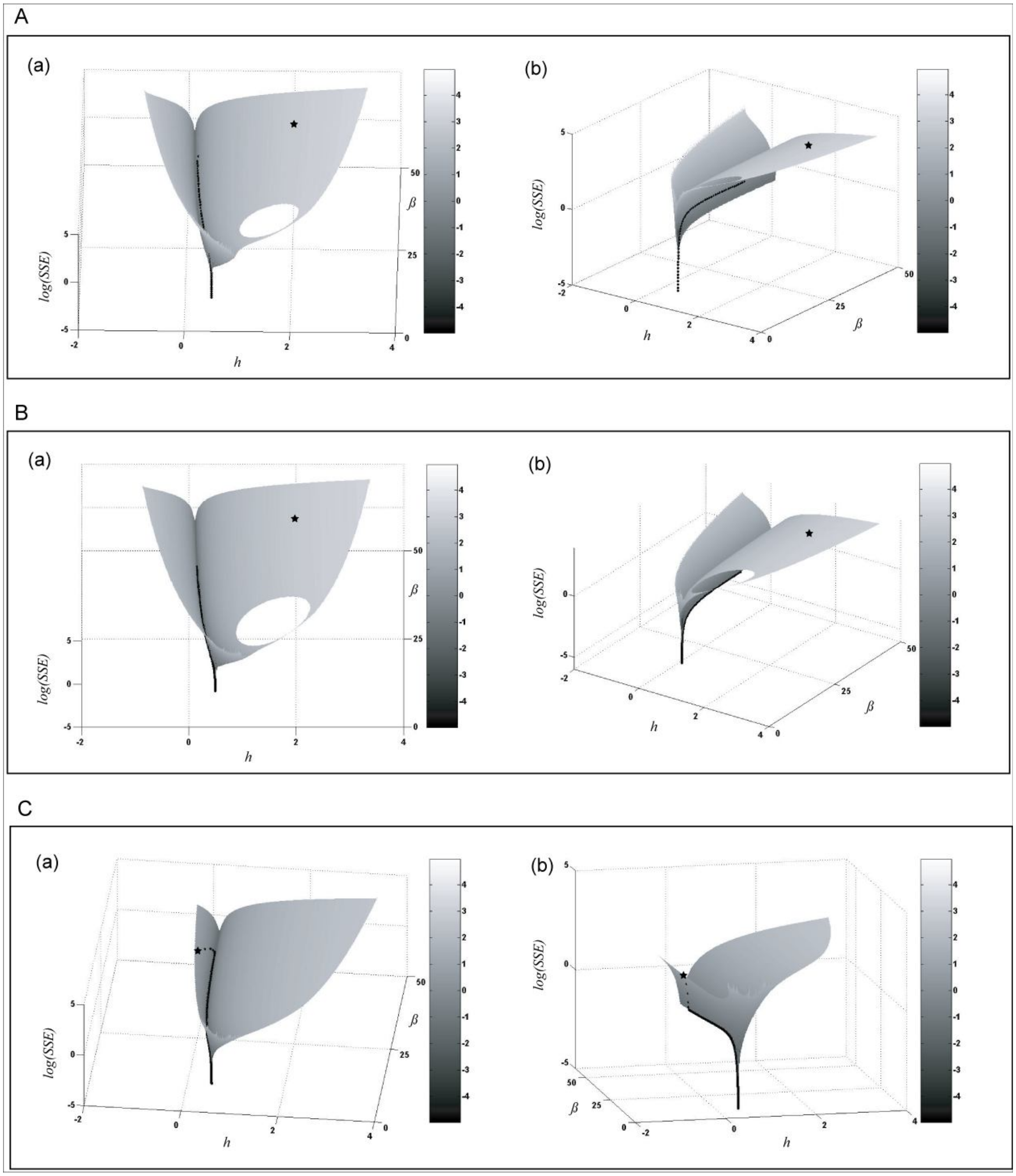


Figure 2.6. Pseudo-3D graph of the error surface for a convergence trajectory.
The graphs in Panels A, B, and C correspond to the graphs in Figure 2.4 (Panels A, B, and C), respectively. Columns (a) and (b) show views from two angles. For all three panels, in just one or few iterations, the trajectories are close to—though not exactly in—the valley of the error surface. The asterisk indicates the initial guess (β, h) = (40, 2).

Convergence trajectories

Paths toward the correct solution may be visualized by plotting and superimposing the solution at every regression step onto the corresponding heat maps, with arrowheads indicating the direction of each trajectory (Figures 2.4A(b, c), 2.4B(b, c), and 2.4C(b, c)). For the first set of illustrations, four different initial values of h_{11} are chosen, while the value of β_1 is always 40. For the second set of illustrations, four different initial values of β_1 are chosen, while the value of h_{11} is always 2. Interestingly, independent of the start values, only two iterations are needed to reach a point very close to the valley of the error surface where the true solution is located. After the dramatic initial jump, all solutions follow essentially the same trajectory with small steps toward the true solution. One can also link the observations of Figures 2.4A(b) and A(c) to the result in 2.4A(a). For the same β_1, a start point in the right part the graph causes AR to jump to a more distant location on the trajectory, thus requiring more iterations to converge to the true solution.

It might be possible to speed up convergence in the flat part of the error surface, for instance by using history-based modeling based on conjugated gradients or partial least squares regression (Martens and Naes, 1989). These options have not been analyzed.

Accuracy and speed of solution

The previous sections focused on the first equation of the S-system model in Eq. (2.21) and Figure 2.2. I used the AR algorithm in the same manner to estimate all other parameters. Again, three sets of regressors were used for every variable. For simplicity of discussion, I describe the results from using dataset 1 of Table 2.1, always using as initial guesses β_i=15 and h_{ij}=1. The main result is listed in Tables 2.2. Additional results and further comments are presented in Appendix A (Section A.1) Tables A.1 and A.2.

Table 2.2. Estimated parameter values of the S-system model of the pathway in Figure 2.2 using *log*(*SSE*)<-7 as termination criterion.

[a] Regressor: A: all variables used as regressors and subsequently constrained; B: use of "union" variables as regressors (see text in Section 2.3.2 for detail); C: fully informed selection of regressors (see text in Section 2.3.2 for detail). [b] time (secs) needed to converge to the solution with *log*(*SSE*)<-7. [c] Convergence results according to AR algorithm: *: convergence to the true solution; **: convergence to different solution; ***: no convergence. [d] time after running 1,000,000 iterations. See Eq. (2.21) for optimal parameter values and the Appendix A (Section A.1) for further comments.

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

For every variable, at least one of the three choices of regressors leads to convergence to the correct solution. Convergence is comparably fast, even if one requires a very high accuracy for termination (*log*(*SSE*)<-20) (see Table A.1). If one relaxes the accuracy to *log*(*SSE*)<-7 or *log*(*SSE*)<-4, the solution is still very good, but the solution time is noticeably decreased (Tables 2.2 and A.2). However, the false-positive rate increases slightly for *log*(*SSE*)<-4. As a compromise, I use *log*(*SSE*)<-7 as termination criterion for the remainder of this paper.

Interestingly, the speed of convergence is fastest for the strategy "A" of using all variables as regressors; however, the failure rate in this case is also the highest. In contrast, the slowest speed of convergence is obtained for the correct regressors ("C"), where AR always converges to the right solution. The regressor set "B" is between "A" and "C" in terms of speed and ability to yield the correct optimum. For cases that don't converge to the right solution one easily adapts the AR algorithm by choosing different start values, slightly modifying constraints, or choosing different regressors in addition to the three types used above. The probability of finding the correct solution is increased if different datasets are available for sequential or simultaneous estimation. The same was observed for other estimation methods (*e.g.*, (Voit and Almeida, 2004)).

2.3.3 Structure identification

The previous sections demonstrated parameter estimation for a system with known structure. Similar to this task is the identification of the unknown structure of a pathway from time series data, if one uses S-systems as the modeling framework (Voit and Almeida, 2004). The only difference is that very few or no parameters at all can *a priori* be set to zero or constrained to the positive or negative half of the search space. A totally uninformed AR search of this type often leads to no convergence. However, since each AR is fast, it is feasible to execute many different searches, in which some of the parameters are allowed to float, while others are set equal to zero.

Table 2.3 shows the results of exhausting all combinations of constraints to determine those that yield convergence. The total time for this exhaustive search is just over one hour. This is furthermore reduced if some *a priori* information is available. As an alternative to an exhaustive search, one may obtain constraining information from a prior linearization of the system dynamics (see (Veflingstad *et al.*, 2004) and Chapter 1 (Section1.5.1) for detail). This method does not identify parameter values per se, but provides very strong clues on which variables are likely to be involved in a given equation and which not. In the example tested, this method provided an over 90% correct classification of the relevant variables in each equation (see Table 2.4). Using this inference information, the total time was reduced to 53 minutes. The savings with this method in the given example are actually only modest (about 20%). Among the possible reasons are that the method does not allow distinction between effects mediated through the α-term from those mediated through the β-term and that the interaction between X_3

and X_4 (represented by g_{34} and h_{34}) is actually not identified correctly, even though Veflingstad's method gives it 66.7% support. Forcing g_{24} and h_{24} to be zero (which is predicted to be the case with 83% likelihood) leads to no convergence.

Table 2.3. Constraints on kinetic orders leading to AR convergence. Termination criterion is *log*(*SSE*)<-7.

* Time (mins) needed for testing all 256 combinations of zero and non-zero values of kinetic orders in each equation.

	Production constraint	Degradation constraints	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	Time*
X_1	[0 0 g_{13} 0]	[h_{11} 0 0 0]	12.00	0.00	0.00	-0.80	0.00	10.00	0.50	0.00	0.00	0.00	
	[0 0 g_{13} 0]	[h_{11} h_{12} 0 0]	12.00	0.00	0.00	-0.80	0.00	10.00	0.50	0.00	0.00	0.00	
	[0 g_{12} g_{13} 0]	[h_{11} 0 0 0]	12.00	0.00	0.00	-0.80	0.00	10.00	0.50	0.00	0.00	0.00	
	[g_{11} 0 g_{13} 0]	[h_{11} 0 0 0]	12.02	0.00	-0.00	-0.80	-0.00	10.02	0.50	0.00	0.00	0.00	20.82
X_2	[g_{21} 0 0 g_{24}]	[0 h_{22} h_{23} 0]	8.02	0.50	0.00	0.00	0.00	3.00	-0.00	0.75	0.00	0.00	
	[g_{21} 0 g_{23} g_{24}]	[0 h_{22} 0 0]	8.04	0.50	-0.00	-0.00	0.00	3.01	0.00	0.75	-0.00	-0.00	
	[g_{21} g_{22} 0 g_{24}]	[0 h_{22} 0 0]	7.97	0.50	0.00	-0.00	-0.00	2.99	0.00	0.75	-0.00	-0.00	8.50
X_3	[0 g_{32} 0 0]	[0 0 h_{33} h_{34}]	3.00	0.00	0.75	-0.00	-0.00	5.00	-0.00	0.00	0.5	0.2	
	[0 g_{32} g_{33} 0]	[0 0 h_{33} h_{34}]	3.00	0.00	0.75	-0.00	-0.00	5.02	0.00	-0.00	0.50	0.20	9.21
X_4	[g_{41} 0 0 0]	[0 h_{42} 0 h_{44}]	2.00	0.50	-0.00	-0.00	-0.00	6.00	0.00	-0.00	0.00	0.80	
	[g_{41} 0 0 0]	[0 h_{42} h_{43} h_{44}]	2.02	0.49	0.00	0.00	-0.00	6.02	-0.00	-0.00	0.00	0.80	
	[g_{41} 0 0 g_{44}]	[0 0 0 h_{44}]	2.06	0.49	-0.00	0.00	0.01	6.08	-0.00	0.00	-0.00	0.80	
	[g_{41} 0 g_{43} 0]	[0 h_{42} 0 h_{44}]	2.03	0.49	0.00	-0.00	-0.00	6.03	-0.00	-0.00	-0.00	0.79	
	[g_{41} g_{42} 0 0]	[0 0 h_{43} h_{44}]	2.01	0.50	0.00	0.00	-0.00	6.00	-0.00	0.00	0.00	0.80	
	[g_{41} g_{42} g_{43} 0]	[0 0 0 h_{44}]	2.02	0.49	0.00	-0.00	-0.00	6.01	-0.00	0.00	-0.00	0.79	30.60

Table 2.4. Collective inference of the gene network based on results from all linearization, according to Veflingstad *et al.* (2004).

A plus sing implies a positive influence, a minus sign implies a negative influence, and a zero implies no influence. Bold entries denote correctly identified interactions and numbers in parentheses give the fraction of models that suggest positive identification.

	X_1	X_2	X_3	X_4
X_1	**- (100 %)**	**0 (100 %)**	**- (83 %)**	**0 (83 %)**
X_2	**+ (100 %)**	**- (67 %)**	**0 (100 %)**	**0 (83 %)**
X_3	**0 (100 %)**	**+ (83 %)**	**- (83 %)**	0 (67 %)
X_4	**+ (100 %)**	**0 (100 %)**	**0 (100 %)**	**- (83 %)**

Finally, it is possible to sort parameter combinations by their empirical likelihood of inclusion in an equation (see (Marino and Voit, 2006) and Chapter 1 (Section1.5.5) for detail). For instance, a metabolite usually affects its own degradation but usually has no effect on its own production. Thus, a reasonable start is the parsimonious model $\dot{X}_i = \alpha_i - \beta_i X_i^{h_{ii}}$ with g_{ii}=0 and h_{ii}>0. In subsequent runs, free-floating variables (parameters) are added, one at a time. This strategy reduced the total time from one hour to under 3 minutes (see Table 2.5).

Table 2.5. First constraint found leading to AR convergence, starting from the most parsimonious constraint. Termination criterion is *log*(*SSE*)<-7.
* Time (mins) needed for testing all 256 combinations of zero and non-zero values of kinetic orders in each equation. [a] In the 4th place of the combination matrix 1; [b] In the 31th place of the combination matrix 2; [c] In the 20th place of the combination matrix 3; [d] In the 11th place of the combination matrix 4.

	Production constraint	Degradation constraints	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	Time*
X_1	$[0\ 0\ g_{13}\ 0]^a$	$[h_{11}\ 0\ 0\ 0]^a$	12.00	0.00	0.00	-0.80	0.00	10.00	0.50	0.00	0.00	0.00	0.02
X_2	$[g_{21}\ g_{22}\ 0\ g_{24}]^b$	$[0\ h_{22}\ 0\ 0]^b$	7.97	0.50	0.00	-0.00	-0.00	2.99	0.00	0.75	-0.00	-0.00	0.95
X_3	$[0\ g_{32}\ 0\ 0]^c$	$[0\ 0\ h_{33}\ h_{34}]^c$	3.00	0.00	0.75	-0.00	-0.00	5.00	-0.00	0.00	0.5	0.2	0.49
X_4	$[g_{41}\ 0\ 0\ g_{44}]^d$	$[0\ 0\ 0\ h_{44}]^d$	2.06	0.49	-0.00	0.00	0.01	6.08	-0.00	0.00	-0.00	0.80	0.86

As an illustration, and for a second, independent example, I used the strategy of Veflingstad *et al.* (Veflingstad *et al.*, 2004) to determine the regulatory structure and parameter values of a gene regulatory network model (Hlavacek and Savageau, 1996) that has become a benchmark in the field. Kikuchi and collaborators (Kikuchi *et al.*, 2003) identified the structure of this model by using a genetic algorithm acting directly on the five differential equation of the model. Using a cluster of 1,040 CPUs, the solution required about 70 hours. I generated time series data from the model, using 0.5 as initial concentration for all five variables. The solution time needed for exhausting all constraint combinations for all variables and an error tolerance of *log*(*SSE*)=-7 was 81.2 min on a single PC. Interestingly, the false-positive rate in this case was higher in this system as compared to the example above. The time needed for the hierarchical strategy proposed by Marino and Voit (Marino and Voit, 2006) was 6.38 mins. The parameter values of metabolites X_1, X_2, X_4, and X_5 were found correctly, but the parameters associated with X_3 were not all identified, even though the error satisfied my termination criterion (*log*(*SSE*)<-7), indicating that a different solution with essentially zero-error exists in this equation. This result interestingly echoes the result based on linearization, as proposed by Veflingstad *et al.* (Veflingstad *et al.*, 2004). The reason is probably that X_2 contributes to both the production term and the degradation term of X_3 with the same kinetic order (-1) and that the time course is not very informative. Also similar to Veflingstad's results, when I used different initial concentrations to perturb X_2 and X_3 more strongly, AR yielded the correct solution.

2.4 Conclusions

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

Methods of parameter estimation, and the closely related task of structure identification, naturally suffer from combinatorial explosion, which is associated with the number of equations and the much faster increasing number of possible interactions between variables, which show up as parameters in the equations. The proposed method of decoupling behaves much better in this respect than most others (*cf.* (Voit and Almeida, 2004; Marino and Voit, 2006)). In practical applications, the increase in the number of combinations is in most cases vastly less than theoretically possible, because the average connectivity of a biological network is relatively small ($<<O(n^2)$; see Chapter 1 (Section 1.5) for review).

The patterns of convergence are at this point not well understood. Some issues were discussed in Section 2.3 and others are detailed in Appendix A. From these numerical analyses it is clear that convergence depends in a very complicated fashion on the dataset, the constraints, the choice of regressors, and the structure and parameter values of the system. Given that even the convergence features of the very well known Newton algorithm are not fully understood (Epureanu and Greenside, 1998), it is unlikely that simple theorems will reveal the convergence patterns of AR in a general manner.

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

At this stage I have deduced optimized solutions for each metabolite separately. In other words, I have not accounted for constraints among equations, such as stoichiometric precursor-product or branch point relationships. Also, it seems that similar methods should be efficacious for the estimation of Generalized Mass Action (GMA) systems (see Chapter 1 (Section 1.2.4) for review). These issues will be the subject of further study. Some of the issues and preliminary results will be discussed in Chapter 4 and Appendix B. I have also assumed that the data are error-free. This assumption was made to identify advantages and failures of the AR algorithm in a fashion as unobstructed as possible. Also, as raw data are typically smoothed before estimating parameter values, the analysis of noisy data seems to depend more on the quality of smoothing than on AR itself. The same is the case for data that do not stem from S-system models, where the quality of the estimation is driven by the accuracy of the S-system representation. Future studies will elucidate how sensitive to experimental error the algorithm is.

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

CHAPTER 3

PARAMETER ESTIMATION OF S-DISTRIBUTIONS WITH ALTERNATING REGRESSION[iv]

3.1 Introduction

Chapter 2 introduced a novel parameter estimation algorithm for S-systems called alternating regression. As an extension of the methods in the previous chapter, the present chapter applies the AR algorithm to *S-distributions* which form a family of unimodal statistical distributions motivated by S-systems (Savageau, 1982). Although S-distributions are not directly related to metabolic pathway modeling, they retain some of the mathematical properties of S-systems, and insights into estimating their parameter values may shed light on features of the AR algorithm that were obscure before.

The S-distribution was introduced in the early 1990s as a convenient univariate, unimodal four-parameter probability distribution that is capable of modeling a wide range of shapes and skewness (Voit, 1992b). Due to its rich shape flexibility and relatively simple mathematical format, the S-distribution has been shown to constitute a good general-purpose default distribution, especially for data of unknown structure. The S-distribution may also be used in lieu of the traditional distributions, because it always has the same structure and, with an appropriate choice of parameter values, rather accurately approximates many continuous central and non-central distributions, as well as a wide variety of discrete distributions (Voit, 1992b; Voit and Yu, 1994; Yu and Voit, 1996). In addition, the S-distribution allows for combinations of parameter values that do not correspond to traditional distributions and permits a spectrum of distributions with long

[iv] This chapter is adapted from: Chou, I-C., Martens, H., and Voit, E. O. (2007) Parameter estimation of S-distributions with alternating regression. *Stat. Operations Res. Transactions* (*SORT*), 31(1), 55-74.

or heavy tails and with skewness to the left or right. Thus, one might in many cases expect a better fit than is possible with traditional distributions. As a specific application of the combination of its flexibility and small number of parameters, the S-distribution is well suited for the non-trivial characterization of trends of distributions that change mean, variance, shape, and even skewness over time (Voit, 1996; Sorribas *et al.*, 2000; Voit and Sorribas, 2000).

The S-distribution is formulated as a differential equation, which renders the estimation of parameter values from data a challenge. Several methods have been suggested for this task, including nonlinear regression (Voit, 1992b; Sorribas *et al.*, 2000), a graphical method (Voit, 1992b), constrained maximum likelihood estimation (Voit, 2000b), and techniques based on quantiles (Voit and Schwacke, 2000; Hernández-Bermejo and Sorribas, 2001). Here, I propose an entirely different method called *3–way Alternating Regression* (3-AR), which was motivated by a 2-way alternating regression method used for the estimation of parameters in multivariate S-systems (see Chapter 2 for detail). The main appeal of 3-AR is its enormous speed and robustness. In this chapter, I discuss the method and apply it to several artificial and actual examples.

3.2 Methods

3.2.1 S-Distribution

The S-distribution is a four-variable statistical distribution that emphasizes the cumulative density function (*cdf*) *F*, which is formulated as a differential equation with respect to random variable *X* and reads

$$f = \frac{dF}{dX} = \alpha\left(F^{g} - F^{h}\right), \qquad F_0 = F\left(X_0\right) \in \left(0,1\right). \tag{3.1}$$

Because the probability density function (*pdf*) *f* is the derivative of *F*, the S-distribution can be seen as an algebraic function *f*(*F*). The first parameter of the distribution, X_0, characterizes the location of the distribution. The second parameter, α, is a positive real number, which determines the scale. The remaining two parameters, g and h, may be any real numbers as long as $g < h$; they determine the shape of the distribution[v]. Figure 3.1 shows two examples of S-distributions.

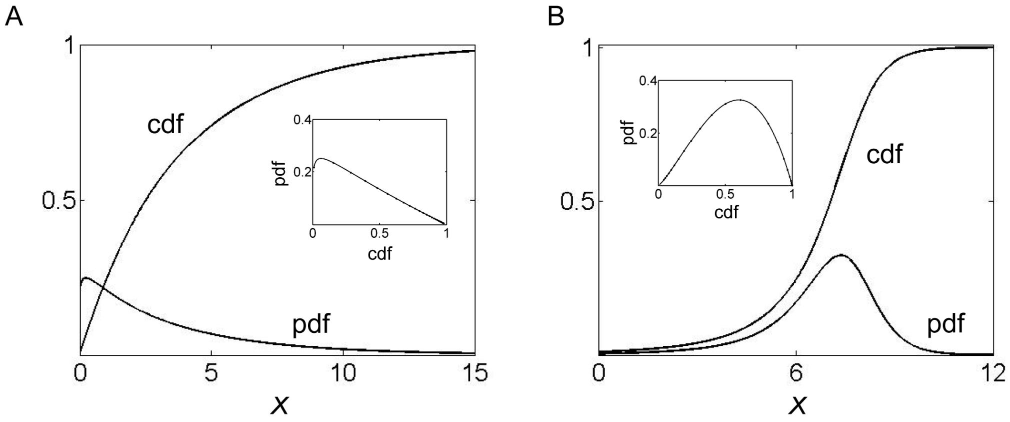


Figure 3.1. S-distributions.
Two examples show the *pdf*, *cdf*, and *f-F* plot (inserts) of S-distributions. Case A: $\alpha = 1$, $g = 0.25$, $h = 0.5$, $F_0 = 0.01$. Case B: $\alpha = 1$, $g = 1.2$, $h = 3$, $F_0 = 0.01$.

3.2.2 Alternating regression

Suppose the S-distribution is characterized through m values of the random variable, $X_1, X_2, ..., X_k, ..., X_m$, and that $F(X_k)$ and $f(X_k)$ are observed or obtainable for each k (see later sections for further discussion on the construction of *pdf*s and *cdf*s). For the purpose of parameter estimation, the original differential equation can then be analyzed in the form of m uncoupled algebraic equations as

$$\begin{gathered} f(X_1) \approx \alpha\left(F^g(X_1) - F^h(X_1)\right), \\ f(X_2) \approx \alpha\left(F^g(X_2) - F^h(X_2)\right), \\ \vdots \\ f(X_k) \approx \alpha\left(F^g(X_k) - F^h(X_k)\right), \\ \vdots \\ f(X_m) \approx \alpha\left(F^g(X_m) - F^h(X_m)\right). \end{gathered} \tag{3.2}$$

[v] Throughout the paper, random variables and *cdf*s are represented as upper-case italics, while *pdf*s are given by the corresponding lower-case italic symbols (X, F, f). An upper-case boldface variable (**L**) represents a matrix of regressor columns and a lower-case boldface variable (**y**) represents a regressand column in a linear statistical regression model.

The ≈ symbol is used because the data may only be representable in approximation by the S-distribution format. As a consequence of this decoupling step, substitution of the derivative of F with f allows us to estimate the S-distribution parameters α, g, and h in a purely algebraic system. I propose for this estimation purpose a new method called *3-way alternating regression* (3-AR).

In Chapter 2, I have shown that alternating regression (AR), applied to general S-system models and combined with methods for slope estimation and decoupling systems of differential equations, provides a fast tool for identifying parameter values from time series data. The key feature of AR is the reduction of the nonlinear inverse problem of parameter estimation into iterative steps of two phases of linear regression. In the first phase, the parameters of the β-term, β_i and h_{ij}, are set to some reasonable values. Given measurements of all X_i at m time points and estimates slope $S_i(t_k)$ at these points, the β-term becomes a number at each time point, and this number is added to both sides of the equation at each time point. As result, the left-hand side becomes a numerical value, while the right-hand side consists exclusively of the symbolic α-term. The m equations of this type are logarithmically transformed and subjected to multivariate linear regression. The resulting estimates for α_i and g_{ij} are used for the second phase of AR, where the α-term is subtracted from the slope values and the parameters of the β-term are estimated and updated. The algorithm thus switches back and forth, thereby rapidly improving estimates of all parameters (see Chapter 2 for details).

The S-distribution is obviously a special case of an S-system, with the notable feature that by definition $\alpha = \beta$. This feature is important for AR methods, because α and β are no longer independent of each other, and it turns out to be inconvenient to constrain α to be the same in both phases of the regression. Furthermore, as discussed in Chapter 2, AR tends to encounter problems if the same variable is present in both the α and β terms of the same equation. In general S-systems, this situation is rather rare. By contrast, it is the normal occurrence in S-distributions, and preliminary studies indeed confirmed that the direct application of AR was problematic. Therefore, I propose here to modify the 2-way AR approach here into a three-cycle 3-AR method specifically for S-distribution estimation. It might be useful in the future to explore 3-AR in general S-system equations that contain the same variables in both terms.

Similar to the original AR, 3-AR works by iteratively cycling between phases of linear regression. The first phase begins with guesses of the values of *g* and *h* and uses these to solve for the value of parameter α. Experience has shown that it is more expedient to start the algorithm with *g* and *h*, rather than *g* and α or *h* and α, presumably due to the fact that the typical ranges of *g* and *h* are much smaller than that of α and because *h* is per definition constrained by *g*. The second phase takes estimates of α and *h* to solve for *g*, while the third phase takes estimates of α and *g* to solve for *h* and thus improve the parameter guesses or estimates from the previous phases. The phases are iterated until a solution is found or AR terminates for other reasons. The overall flow of the method is shown in Figure 3.2, and specific steps of the 3-AR algorithm are detailed in the next section.

3.2.3 Steps of the 3-AR algorithm

{1} Define $\mathbf{L_f}$ and $\mathbf{L_F}$ as $m\times 2$ matrices of logarithms of regressors *f* and *F*, respectively:

$$\mathbf{L_f} = \begin{bmatrix} 1 & \log\left(f\left(X_1\right)\right) \\ 1 & \log\left(f\left(X_2\right)\right) \\ \vdots & \vdots \\ 1 & \log\left(f\left(X_k\right)\right) \\ \vdots & \vdots \\ 1 & \log\left(f\left(X_m\right)\right) \end{bmatrix}, \tag{3.3}$$

$$\mathbf{L_F} = \begin{bmatrix} 1 & \log\left(F\left(X_1\right)\right) \\ 1 & \log\left(F\left(X_2\right)\right) \\ \vdots & \vdots \\ 1 & \log\left(F\left(X_k\right)\right) \\ \vdots & \vdots \\ 1 & \log\left(F\left(X_m\right)\right) \end{bmatrix}. \tag{3.4}$$

L_f is used in the first phase of AR to determine α, and $\mathbf{L_F}$ is used in the second and third phases of AR to determine g and h.

{2} Select values for g and h in accordance with experience about S-distribution parameters (see (Voit, 1992b) for relationships between parameter values and distributional shape).

{3} For all X_k, $k = 1, 2,\ldots, m$, compute $F^{\hat{g}}\left(X_k\right) - F^{\hat{h}}\left(X_k\right)$, using values $F(X_k)$ from the data distribution. Here $\hat{g}$ and $\hat{h}$ denote the estimators of g and h after the 2nd iteration, while during the 1st iteration, $\hat{g}$ and $\hat{h}$ are the initial guesses for g and h, respectively. Determine the index I_α of all positive quantities $F^{\hat{g}}\left(X_k\right) - F^{\hat{h}}\left(X_k\right)$. The number of *qualified points* then becomes N_α, where N_α is the length of I_α. Quantities restricted to N_α instead of all N points are identified in the following with an additional subscript α. Theoretically $F^g\left(X_k\right)$ should always be greater than $F^h\left(X_k\right)$, because $g < h$, or at most equal, for $F = 0$ and $F = 1$. However, because of noise, this may not always be true, suggesting temporary exclusion of some data points.

{4} After logarithmic transformation and rearrangement, Eq. (3.1) can be written as $\log\left(\dfrac{f}{\alpha}\right) = \log\left(F^g - F^h\right)$. Therefore, compute the N_α-dimensional vector

$\mathbf{y_\alpha} = \log\left(F_\alpha^{\hat{g}} - F_\alpha^{\hat{h}}\right)$ for N_α points, as well as $\mathbf{L_{f_\alpha}}$, where the subscript α limits the computation to qualified points.

{5} Based on the linear regression model

$$\mathbf{y_\alpha} = \mathbf{L_{f_\alpha}}\hat{\mathbf{b}}_\alpha + \boldsymbol{\varepsilon}_\alpha, \tag{3.5}$$

estimate the regression coefficient vector $\hat{\mathbf{b}}_\alpha = \begin{bmatrix} \hat{b}_{\alpha_1} & \hat{b}_{\alpha_2} \end{bmatrix}^{\mathrm{T}}$ over the N_α qualified points, to obtain an estimate of α. In other words, this equation may be written as $\mathbf{y_\alpha} \approx \log\left(\frac{1}{\hat{\alpha}}\right) + \log\left(f_\alpha\right) + \varepsilon_\alpha$ so that $\hat{b}_{\alpha_1}$ is equivalent to $\log\left(\frac{1}{\hat{\alpha}}\right)$ and $\hat{b}_{\alpha_2}$ is the coefficient of $\log\left(f_\alpha\right)$, which is expected to converge to 1. Thus, $\hat{\mathbf{b}}_\alpha$ is estimated with any of the methods of linear regression, *e.g.*, by ordinary least squares regression (OLSR) as

$$\hat{\mathbf{b}}_\alpha = \left(\mathbf{L}_{\mathbf{f}_\alpha}^{\mathrm{T}}\mathbf{L}_{\mathbf{f}_\alpha}\right)^{-1}\mathbf{L}_{\mathbf{f}_\alpha}^{\mathrm{T}}\mathbf{y}_\alpha. \tag{3.6}$$

As an alternative to OLSR, weighted or robust estimators could be used. If $\mathbf{L}_{\mathbf{f}_\alpha}$ does not have full column rank, *i.e.*, if $\mathbf{L}_{\mathbf{f}_\alpha}^{\mathrm{T}}\mathbf{L}_{\mathbf{f}_\alpha}$ has a small eigenvalue, one could also use a small ridge regression constant κ for stabilization and compute $\hat{\mathbf{b}}_\alpha$ as

$$\hat{\mathbf{b}}_\alpha = \left(\mathbf{L}_{\mathbf{f}_\alpha}^{\mathrm{T}}\mathbf{L}_{\mathbf{f}_\alpha} + \kappa\mathbf{I}\right)^{\mathbf{-1}}\mathbf{L}_{\mathbf{f}_\alpha}^{\mathrm{T}}\mathbf{y}_\alpha. \tag{3.7}$$

{6} For the estimation of g, reformulate Eq. (3.1) as $\frac{f}{\alpha} + F^h = F^g$. Thus, using values of $f(X_k)$ and $F(X_k)$ that are directly obtained from the data (see later sections), compute $\frac{f\left(X_k\right)}{\hat{\alpha}} + F^{\hat{h}}\left(X_k\right)$ for all X_k, $k = 1, 2,\ldots, m$. Here $\hat{h}$ denotes the estimator of h after the 2nd iteration, while during the 1st iteration, $\hat{h}$ is the initial guess for h. Find the index I_g of positive quantities $\frac{f\left(X_k\right)}{\hat{\alpha}}+F^{\hat{h}}\left(X_k\right)$. The number of qualified points for this step becomes N_g, where N_g is the length of I_g.

{7} Compute the N_g-dimensional vector $\mathbf{y_g}=\log\left(\frac{f_g}{\hat{\alpha}}+F_g^{\hat{h}}\right)$ for N_g points and $\mathbf{L_{F_g}}$.

{8} Based on the linear regression model

$$\mathbf{y_g}=\mathbf{L_{F_g}}\hat{\mathbf{b}}_\mathbf{g}+\boldsymbol{\varepsilon}_\mathbf{g}, \tag{3.8}$$

and in analogy to step {5}, estimate the regression coefficient vector

$\hat{\mathbf{b}}_\mathbf{g}=\left[\hat{b}_{g_1} \quad \hat{b}_{g_2}\right]^{\mathrm{T}}$ by regression over the N_g time points as

$$\hat{\mathbf{b}}_\mathbf{g}=\left(\mathbf{L}_\mathbf{F_g}^{\mathrm{T}}\mathbf{L_{F_g}}\right)^{-1}\mathbf{L}_\mathbf{F_g}^{\mathrm{T}}\mathbf{y_g}, \tag{3.9}$$

or with an alternative regression method. The estimator $\hat{b}_{g_2}$ is the parameter of interest, $\hat{g}$; estimator $\hat{b}_{g_1}$ is expected to be zero in the model.

{9} For the estimation of h, reformulate Eq. (3.1) as $F^g-\frac{f}{\alpha}=F^h$ and compute $F^{\hat{g}}\left(X_k\right)-\frac{f\left(X_k\right)}{\hat{\alpha}}$ for all X_k, k = 1, 2,…, m, again using the values of $f(X_k)$ and $F(X_k)$. Determine the index I_h of positive quantities $F^{\hat{g}}\left(X_k\right)-\frac{f\left(X_k\right)}{\hat{\alpha}}$. The number of qualified points for this step becomes N_h, where N_h is the length of I_h.

{10} Compute the N-dimensional vector $\mathbf{y_h}=\log\left(F_h^{\hat{g}}-\frac{f_h}{\hat{\alpha}}\right)$ for N_h points and $\mathbf{L_{F_h}}$.

{11} Based on the linear regression model

$$\mathbf{y_h}=\mathbf{L_{F_h}}\hat{\mathbf{b}}_\mathbf{h}+\boldsymbol{\varepsilon}_\mathbf{h}, \tag{3.10}$$

and in analogy to steps {5} and {8}, estimate the regression coefficient vector

$\hat{\mathbf{b}}_\mathbf{h}=\left[\hat{b}_{h_1} \quad \hat{b}_{h_2}\right]^{\mathrm{T}}$ by regression over the N_h time points as

$$\hat{\mathbf{b}}_{\mathbf{h}} = \left(\mathbf{L}_{\mathbf{F_h}}^{\mathrm{T}}\mathbf{L}_{\mathbf{F_h}}\right)^{-1}\mathbf{L}_{\mathbf{F_h}}^{\mathrm{T}}\mathbf{y}_{\mathbf{h}}\,, \tag{3.11}$$

or with an alternative regression method. The estimator $\hat{b}_{h_2}$ is the parameter of interest, $\hat{h}$; estimator $\hat{b}_{h_1}$ is expected to be zero in the model.

{12} Iterate steps {3} – {11} until a solution is found or some termination criterion is satisfied.

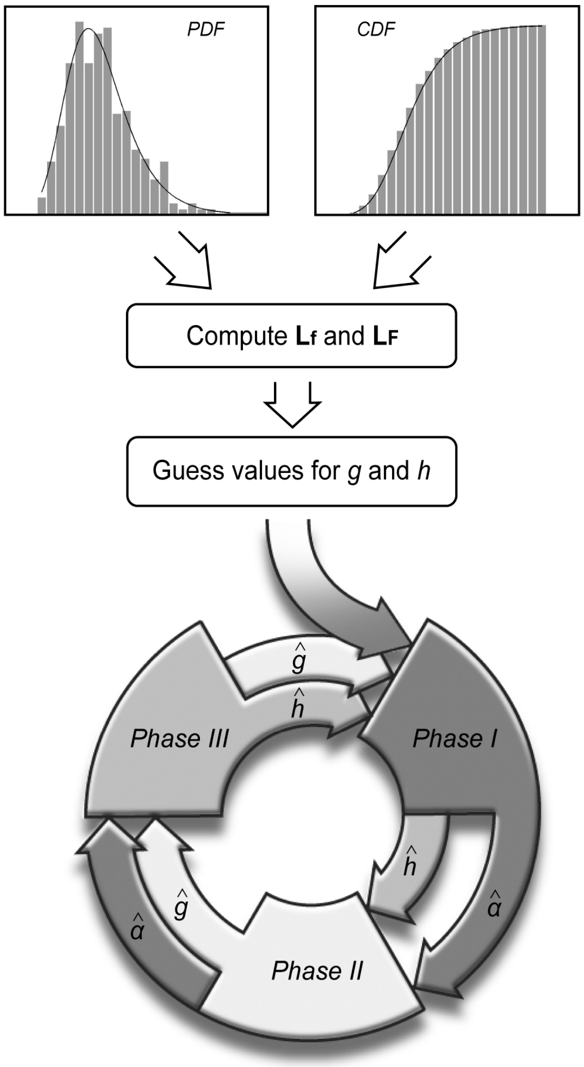


Figure 3.2. Flow of parameter estimation by 3-way alternating regression.

During each phase of 3-AR, lack-of-fit criteria are estimated and used for monitoring the iterative process and to define termination conditions. I use here specifically the logarithm of the sums of squared *y*-errors (SSE_α, SSE_g, and SSE_h) as optimization criteria for the three regression phases. Upon convergence, we also compute the residual error *SSE* of the fit and the standard deviation $S.D. = \sqrt{SSE/(N - p)}$ of the *pdf*, as well as the *cdf* and *f-F* plots, where *p* is the number of estimated parameters, which in all cases here is 3.

The location parameter X_0 is not explicit in the method, because it does not appear in the algebraic formulation of the *pdf* as a function of the *cdf*. However, it is easily estimated directly as the observed or estimated median or by optimizing the horizontal position of the distribution with parameters $\hat{\alpha}$, $\hat{g}$, and $\hat{h}$ (Voit, 2000b).

3.3 Results

I tested the 3-AR method with a large number of representative cases, including estimations based on “data” from error-free distributions, artificial noisy data obtained as random samples generated from S-distributions with known parameters, traditional statistical distributions (using Matlab®), and from actual observation data. Representative details of each case are discussed in this section.

3.3.1 Fitting the distribution without noise

In order not to confuse the features of 3-AR with possible effects of noise in the data, I begin the exploration of convergence properties by using true S-distribution *cdf*s and *pdf*s, which are evaluated directly from Eq. (3.1) at a number of values for the random variable. Specifically, I choose 50 equally spaced instances of the random variable and compute the corresponding *f* and *F* values from Eq. (3.1) to obtain the “true” *pdf* and *cdf*. Figure 3.3 shows an example of a typical convergence pattern. Starting from the (essentially arbitrary) initial guesses $g = 3$ and $h = 6$, it takes the 3-AR algorithm just 51 iterations to converge to the true solution, requiring 0.0742 seconds on a Pentium® D (~3.4GHz) machine. Since I am using noise-free data, the residual error should approach

0, which corresponds to –∞ in logarithmic coordinates. I use –9 instead as one of the termination criteria, which corresponds to a result very close to the true value, but allows for issues of machine precision and numerical inaccuracies. The low error tolerance causes the algorithm to need 51 iterations. However, as Figure 3.3 indicates, the estimates are already very close to the true optimum after just a few initial iterations. Big jumps in the beginning do not negatively affect convergence time. For instance, using the same error tolerance and initial guesses $g = 10$, $h = 10.5$ or $g = 100$ and $h = 120$, respectively, the algorithm needs 57 iterations (0.0535 second) or 63 iterations (0.0567 second) to converge to the true parameter values. Thus, somewhat different from results for general S-systems (see Chapter 2 (Section 2.3)), the speed of convergence here does not depend much on initial guesses. Also in contrast to observations with S-systems, the convergence patterns for α, g, and h are often not monotonic, and each parameter may temporarily increase or decrease during the initial iterations.

While convergence is almost always extremely fast, as in the example described above, some initial values cause 3-AR not to converge at all. In such rare cases, the value of α typically increases without bound, while g and h converge toward each other and ultimately become the same. This case corresponds to the trivial solution $\frac{f}{\alpha} \to 0 \leftarrow F^g - F^h$ in Eq. (3.1) and is easy to detect and discard.

Figure 3.4 combines results for several noise-free S-distributions and essentially exhaustive sets of initial guesses for g and h satisfying $g < h$, as required. The selected distributions are representative for different shapes and skewness, which are reflected in different categories of parameter values (*cf.* (Voit, 2000b)):

(1) $g > 0$ and $h > 0$: as exemplified in Figure 3.4A and 3.4B;

(2) $g < 0$ and $h > 0$: as exemplified in Figure 3.4C;

(3) $g < 0$ and $h < 0$.

In addition, samples from all categories must by definition satisfy the condition $g < h$.

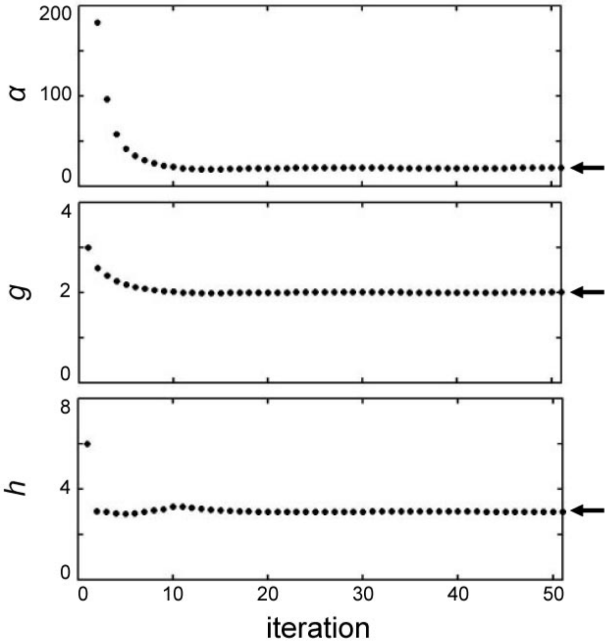


Figure 3.3. Convergence pattern of 3-AR.
For this example, 50 instances of the random variable were chosen from a parent distribution with parameters $\alpha = 20$, $g = 2$, $h = 3$, and $F_0 = 0.01$. Initial guesses were chosen as $g = 3$ and $h = 6$, but do not affect convergence much. No initial guess for α is needed in 3-AR.

The left panels in Figure 3.4 exhibit the *cdf* and *pdf* of each distribution. Inserts show the so-called *f-F* plots, where the *pdf* is plotted against the corresponding *cdf*. These plots are important because they are the basis for 3-AR and many other estimation methods for S-distributions. The right-hand panels present “heat maps” of convergence: the *x*- and *y*-axes represent the initial guesses of *h* and *g*, respectively, and the gray bar represents the logarithm (base 10) of the number of iterations needed for convergence. Once the predetermined error level is reached, 3-AR stops and the number of iterations is recorded as a measure for the speed of convergence. In each case shown here, 25 instances of the random variable were chosen and the corresponding noise-free *f* and *F* values were obtained according to the selected random variables. Black areas represent divergence to the trivial solution $\alpha \approx \infty$, $g \approx h$.

As discussed above, the convergence time for a given distribution does not vary much with different initial guesses, and the basin of convergence within each heat map is therefore almost monochrome. However, the heat maps of different distributions are quite different. For instance, the times needed to generate the heat maps in Figures 3.4A, 3.4B, and 3.4C for a total of 57,600 initial values shown are 14,957, 1,197, and 1,094 seconds on a single PC, respectively, thus yielding average convergence times of 0.26, 0.021, and 0.019 seconds per case. While reasons for the wide variations in convergence times among distributions are unclear, the convergence *patterns* are similar in all cases: 3-AR takes big steps during the first few iterations, already coming very close to the true solution, and then spends many iterations on fine-tuning. The convergence area in each case is relatively large, and it seems to be a good general strategy to choose rather large, similar initial values for g and h, such as 10 and 10.5, to avoid divergence. Of importance is that each iteration consists essentially of three linear regressions, which are very fast. Thus, even if one encounters a rare case of divergence, the choice of alternative initial settings is computationally cheap and provides for effective estimation results.

Examples with $g < 0$ and $h < 0$ or with different α values are not shown in Figure 3.4, but 3-AR performed in a similar fashion for all cases tested. Most of the estimation tasks were solved very effectively, except for cases where the difference between g and h is large, for instance, $g = 0.1$ and $h = 6$. In such cases, the algorithm sometimes converges to sets of values between the true g and h and oscillates between them. A possible reason for this behavior may be that in the 3rd phase of regression (estimation of h), the slope of the regression line in the $\mathbf{y_h} - \mathbf{L_{F_h}}$ plot (which is reflected in the high value of h) is large and greatly affected by small errors, especially when f and F values are small so that their logarithms dominate the regression. In this case, the algorithm may not converge to exactly the right solution, but the oscillation happens within a reasonable range of parameter values. If it is desirable to obtain only one g and h, instead of ranges of oscillation that bound these values, a possible solution is to exclude some of the small F values. In the cases I tested, this omission heuristically resulted in the algorithm converging to the true optimum.

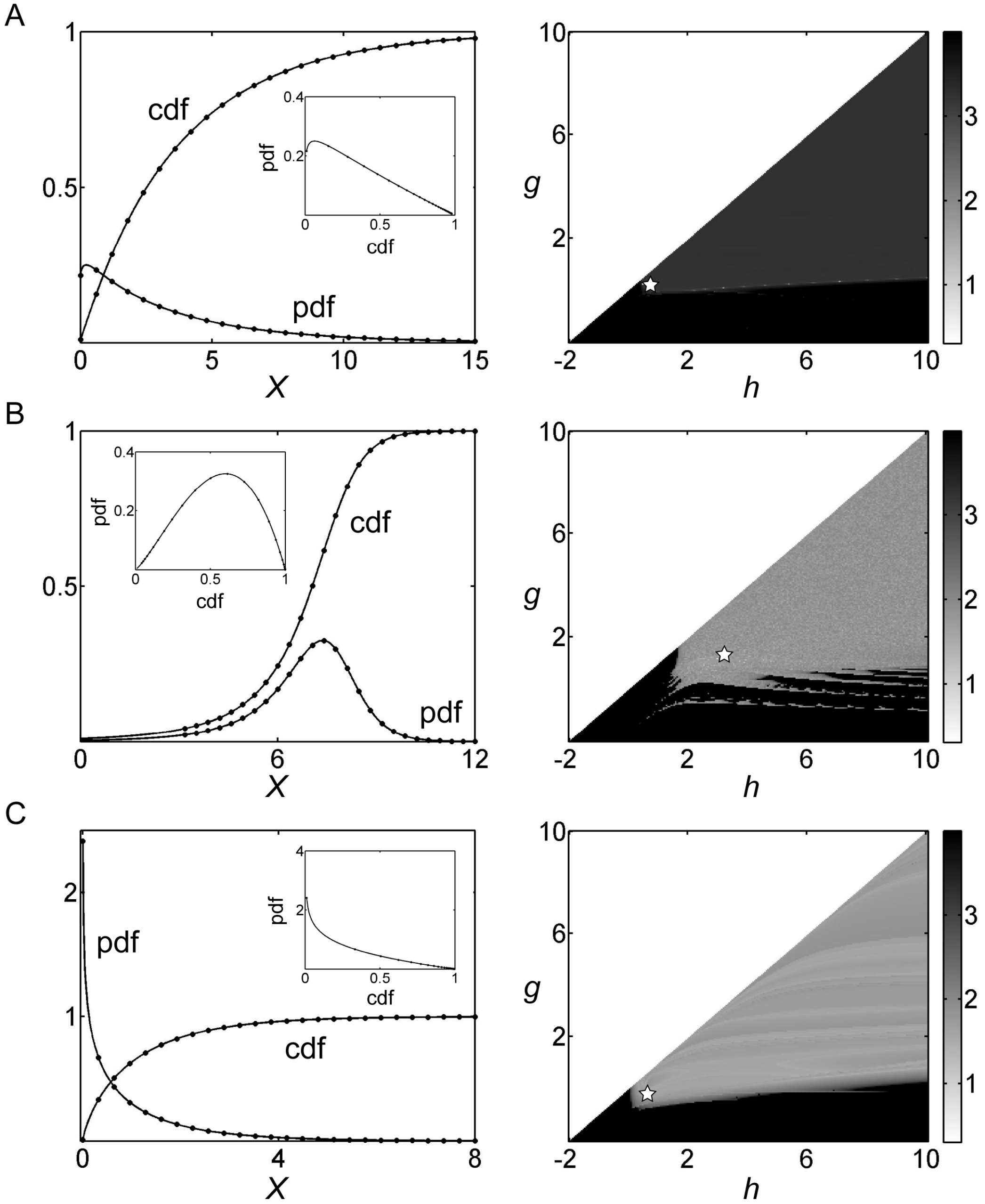


Figure 3.4. Summary of convergence patterns of 3-AR.
Panels on the left show the *pdf*, *cdf*, and *f-F* plot (insert) of each distribution. Panels on the right present heat maps of convergence as functions of starting values of *g* and *h*, with gray bar indicating the logarithm (base 10) of the number of iterations needed for convergence. Each asterisk represents the true value of *g* or *h*. Case A: $\alpha = 1$, $g = 0.25$, $h = 0.5$, $F_0 = 0.01$. Case B: $\alpha = 1$, $g = 1.2$, $h = 3$, $F_0 = 0.01$. Case C: $\alpha = 1$, $g = -0.2$, $h = 0.5$, $F_0 = 0.01$. Twenty-five instances of the random variable were chosen in each case.

3.3.2 Fitting distributions with noise

The preceding section discussed 3-AR for error-free samples from S-distributions. In this section I analyze finite random samples from S-distributions, which result in artificial datasets that appear noisy. To create these data, I use the quantile method, as discussed in (Voit, 2000b). Specifically, I consider the inverted *cdf* equation

$$\frac{dX}{dF} = \frac{1}{\alpha\left(F^g - F^h\right)}, \qquad F(0.5) = \text{median} \tag{3.12}$$

and draw random numbers R_i from the uniform distribution over (0,1), which are used as quantiles. Solving Eq. (3.12) numerically upwards or downwards from the median to $F = R_i$ yields in X_i the desired S-distributed random number. The S-distributed random numbers are collected and form the equivalent of an observed data sample, whose “noise” depends on the sample size.

The performance of 3-AR in fitting these artificial data is shown in Figure 3.5 with an example, where five hundred random numbers were generated from an S-distribution and categorized into 21 bins of a relative frequency histogram (Figure 3.5a). The *pdf* was constructed from the resulting histogram without smoothing and easily yielded the *cdf* (Figure 3.5b). The 3-AR algorithm converged within 47 iterations from the initial guesses $g = 10$ and $h = 10.5$ to the estimated solution. Interestingly, the fit with this solution is associated with a lower *SSE* than a fit with the parent S-distribution, from which the “data” were sampled, which confirms similar earlier observations (*e.g.*, (Sorribas *et al.*, 2000)). To assess dependence on sample size, I also tested the algorithm with smaller sample sizes, *e.g.*, $n = 100$, and 3-AR performed similarly well.

To explore the flexibility of the S-distribution, I repeated the example shown in Figure 3.5 several times with 500 points each. The results (Figure 3.6) show slightly different fits with *SSE*s around 0.0045-0.0047 (Fig. 5A), 0.0054-0.0057 (Fig. 3.5B), and 0.0096 (Fig. 3.5C), which are driven by the degree with which each random sample truly represents the underlying distribution. Within each class, the relationships between the estimates α, g, and h are similar, again confirming earlier results (Sorribas *et al.*, 2000), where classes of quasi-equivalent S-distributions with quite similar *SSE*s were produced by fixing the value of α and fitting g and h. In each class, g and h exhibit an almost linear relationship between each other and with log(α) and converge to each other when α becomes larger. Even though the parameter sets within each class are clearly different, the resulting distributions are essentially indistinguishable.

In some cases, the 3-AR algorithm does not converge to a single value. Instead, it oscillates between reasonable candidate solutions. This is probably due to noise in the data, causing 3-AR to find the best "local" fit for each phase, which however is not the best fit for other phases. This behavior is commonly seen in nonlinear algorithms. It is easy to find a suitable solution by choosing from among the candidate solutions, based on their *SSE*s.

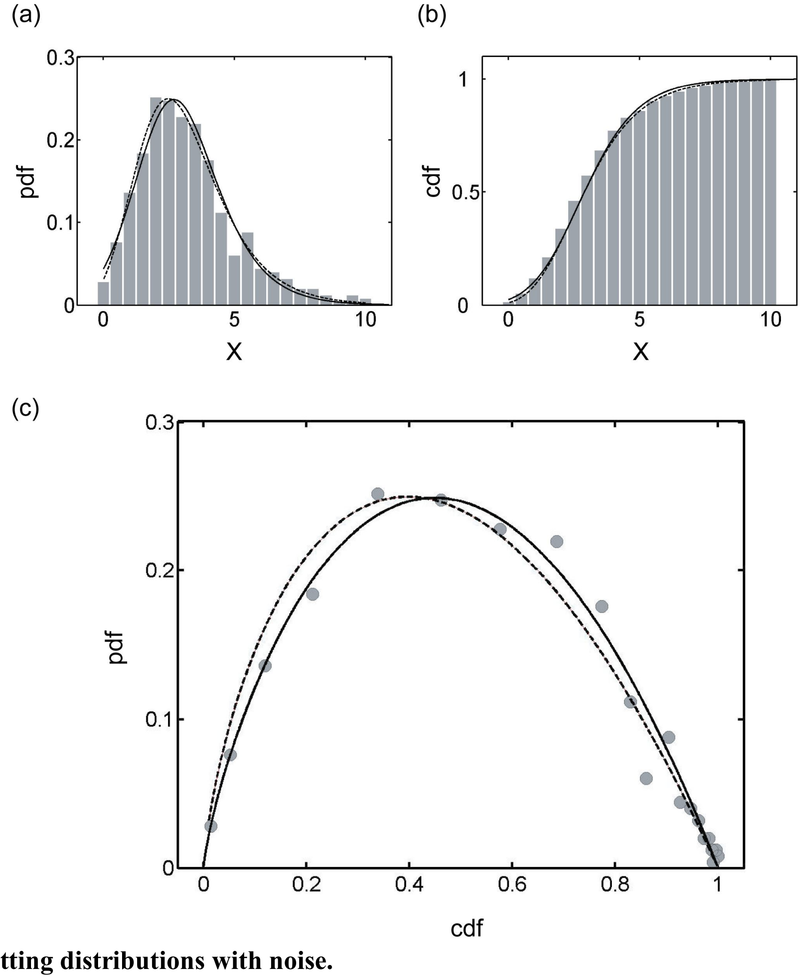


Figure 3.5. Fitting distributions with noise.
Data sampled from an S-distribution with parameter values $\alpha = 1$, $g = 0.75$, $h = 1.5$ and fits with the parent S-distribution (dashed lines) and with an S-distribution obtained with 3-AR and initial guesses $g = 10$ and $h = 10.5$ (solid lines). Optimal parameter estimates are obtained as $\alpha = 0.80$, $g = 0.78$, $h = 1.87$. (a) *pdf*s; (b) *cdf*s; (c) *f-F* plot showing the *pdf* as algebraic function of the *cdf*. *SSE* of the 3-AR optimized distribution is 0.0041 (*S.D.* = 0.0151), while *SSE* for the parent S-distribution is 0.0064 (*S.D.* = 0.0189).

A

α	g	h	SSE
1.134	0.884	1.748	0.0045
2.219	1.028	1.375	0.0046
1.005	0.848	1.805	0.0046
0.743	0.767	1.982	0.0047

B

α	g	h	SSE
0.611	0.704	2.098	0.0054
1.318	0.898	1.555	0.0055
1.830	0.959	1.429	0.0056
4.151	1.055	1.254	0.0057

C

α	g	h	SSE
1.221	0.899	1.785	0.0096
1.537	0.946	1.472	0.0096
2.403	1.004	1.360	0.0096

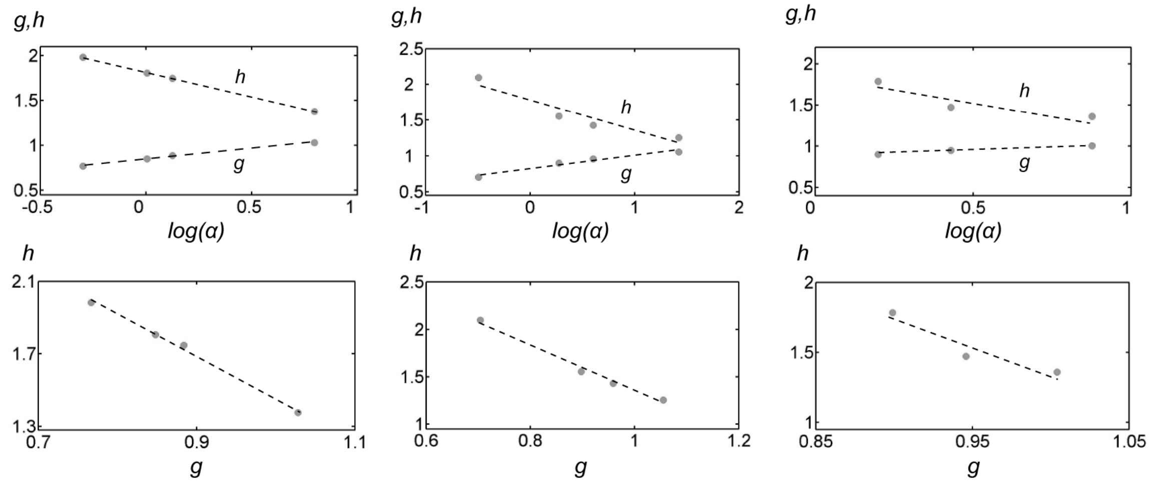


Figure 3.6. Quasi-equivalent S-distributions.
Parameters are estimated for different samples randomly generated from a given distribution ($\alpha = 1$, $g = 0.75$, $h = 1.5$). The residual errors *SSEs* are recorded and classified into three classes based on the value of *SSE*. The plots of g or h versus $\log(\alpha)$ and of g versus h are generated in each class. A: *SSE* between 0.0045 and 0.0047; B: *SSE* between 0.0054 and 0.0057; C: *SSE* equal to 0.0096.

3.3.3 Fitting traditional statistical distributions

The selection of a traditional distribution for fitting data is often difficult because the "true" parent distribution is typically not known. Testing candidate distributions one by one is cumbersome, and all-encompassing distribution families (*e.g.*, (Savageau, 1982)) often contain so many parameters that over-fitting and redundancy become complicating issues. Instead, the S-distribution may be used as an inclusive model that is capable of representing many traditional statistical distributions in sufficiently close approximation. The strategy thus becomes to fit data of unknown structure with an S-distribution and to identify which traditional distributions have similar shapes (Voit, 1992b; Voit and Yu, 1994; Yu and Voit, 1996). This section explores how well 3-AR identifies S-distributions for random samples from traditional distributions.

The S-distribution contains only two classical distributions as special cases: the exponential distribution for $g = 0$ and $h = 1$ and the logistic distribution for $g = 1$ and $h = 2$. Fitting these two distributions yield *SSE*s equal to 0 (results not shown). All other classical distributions incur some unavoidable approximation error when modeled as S-distributions. Figure 3.7 shows the results of 3-AR fitting of three examples that are not special cases, namely a noncentral *t*-distribution, an *F*-distribution, and a χ^2-distribution; the initial guesses were again chosen as $g = 10$ and $h = 10.5$. As before, 3-AR converges to a solution within a few iterations for these and many other examples. The only convergence problems occurred when fitting traditional distributions requiring $g \approx h$ (see (Voit, 1992b) for these uncommon cases). A possible reason is presumably that the S-distribution is not a very good model for such distributions.

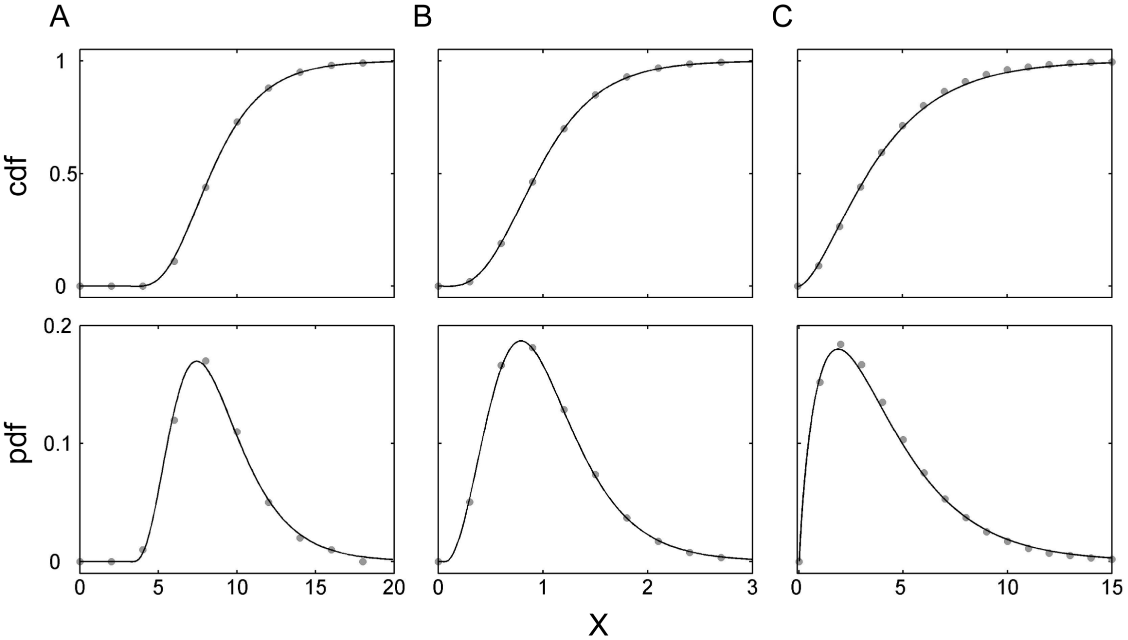


Figure 3.7. Fitting traditional distributions.
The gray dots represent data used in the regressions, while the solid curves represent the estimated S-distributions. The *SSE*s are calculated for the *f-F* plot. A: noncentral $t_{8,8}$-distribution, $SSE = 0.00007$, $S.D. = 0.0032$; B: $F_{10,100}$-distribution, $SSE = 0.00066$, $S.D. = 0.0097$; C: χ_4^2-distribution, $SSE = 0.00026$, $S.D. = 0.0045$.

3.3.4 Fitting observed data

The ultimate measure of success of any fitting algorithm is the modeling of actual data. Figure 3.8 shows the performance of 3-AR in fitting an S-distribution to weight data of males ages 20 to 29 (data from *NHANES III* (National Center for Health Statistics, 1996)). The observed distribution contains 574 males, classified into bins of 3 kg. The *pdf* and *cdf* histograms were constructed in the same fashion as in Section 3.3.2. The *SSE* of the fit is similar to the result of using a constrained maximum likelihood estimator (Voit, 2000b), although the parameter values are somewhat different, exhibiting again the flexibility and quasi-redundancy inherent in S-distributions. Visually, and judged by the *SSE*, the fit obtained here is satisfactory and obtained in less than a second.

3.4 Discussions and Conclusions

The S-distribution is a four-variable distribution that combines mathematical simplicity with superior flexibility in modeling data. A crucial prerequisite for using the distribution in practical applications is the availability of effective methods for estimating optimal parameter values from observed frequency data. Addressing this issue, I introduced here a method called *3–way alternating regression* (3-AR) that is extremely fast and robust. The 3-AR method constitutes a modification of a 2-way alternating regression method that was recently proposed for parameter estimation in S-systems, of which S-distributions are special cases.

The 3-AR method performs well in all typical scenarios, namely for estimating parameters from error-free distributions, from random samples generated from S-distributions, from traditional statistical distributions, and from actual data. The basin of convergence is rather large, and convergence speed is essentially independent of initial guesses that are selected to start the 3-AR algorithm. Therefore, even if one selects initial guesses quite far away from the true optimum, the algorithm only takes a few iterations to converge to points very close to the true solution and refines this solution with a relatively small number of further cycles. An exception is the situation where 3-AR converges to the trivial solution where α increases without bound and *g* approaches *h*. This scenario is easy to spot and the choice of another initial guess typically remedies the situation. A second exception to rapid convergence may occur if the true *g* and *h* are very different. In this rather unusual case, the algorithm sometimes converges to values between the true *g* and *h* and oscillates between them. In this case, one may select values from within the oscillation range or redo the estimation by omitting some of the very small values of the *pdf* and *cdf*.

The 3-AR fitting of data from traditional distributions works well in most cases, except for distributions that are not well approximated by S-distributions and where the relatively best fit requires $g \approx h$, as described in Section 3.3.3.

For finite random samples, the estimated solution is also obtained very quickly, but its parameters depend on the particular sample. As a consequence, the computed estimates may be rather different, even though the *SSE*s are very similar and the shapes of the resulting distributions are essentially indistinguishable. This finding is a manifestation of the shape flexibility and quasi-redundancy of S-distributions and confirms similar observations in the literature (*e.g.*, (Sorribas *et al.*, 2000)).

The 3-AR algorithm provides a strategy for parameter estimation with S-distributions that is genuinely different from all other published methods. While some issues associated with the basin of convergence should be investigated further, my results shown here provide strong indication that this algorithm is much faster than the currently available alternatives.

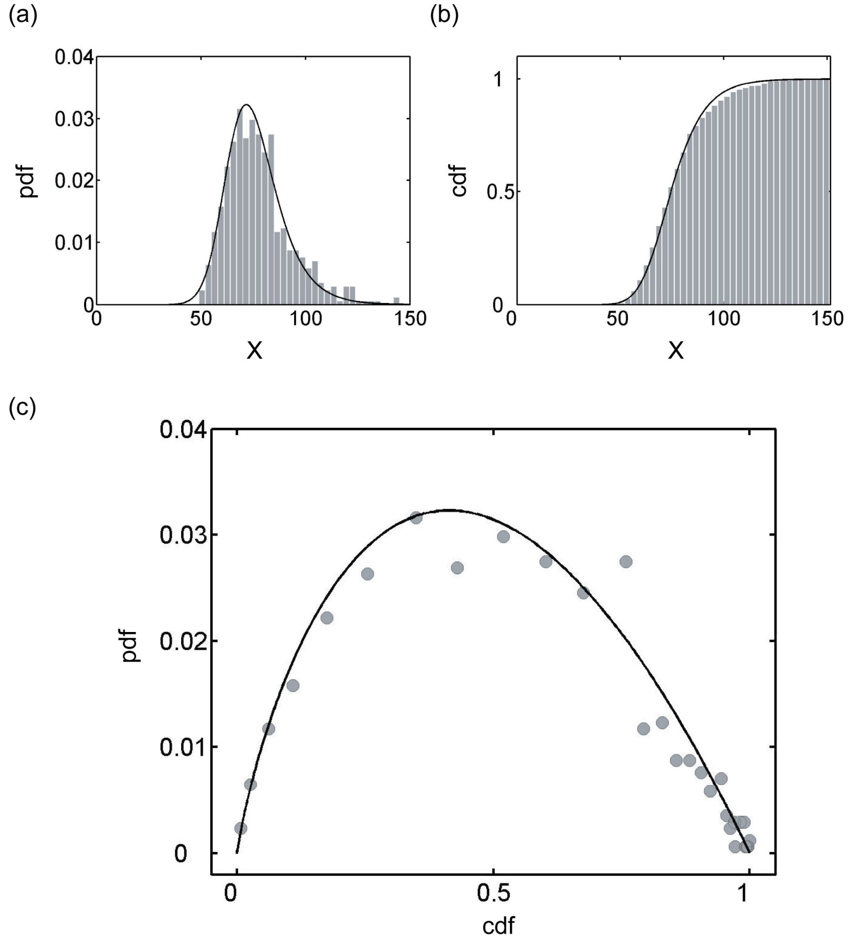


Figure 3.8. Fitting observed data.
Observed distribution (bars and dots) of weights of 574 males, ages 20-29 (National Center for Health Statistics, 1996) and S-distribution fit (lines) obtained with 3-AR and initial guesses $g = 10$, $h = 10.5$. Estimated parameter values: $\alpha = 0.270$, $g = 0.958$, $h = 1.328$, $X_{0.5} = 74.37$. (a) *pdf* (*SSE* = 0.000143, *S.D.* = 0.0023); (b) *cdf* (*SSE* =0.009629, *S.D.* = 0.0189); (c) *f-F* plot (*SSE* = 0.000187, *S.D.* = 0.0026).

An issue that seems generic to S-distributions and has been observed in other contexts is the covariance among the parameters α, g, and h (*e.g.*, (Sorribas *et al.*, 2000)). While each set of these parameters determines a unique distribution, the covariance permits distinct sets leading to solutions that are so similar that their differences are often smaller than the noise in the data. This quasi-equivalence will require future work. For instance, it might be possible to specify the theoretical uncertainty variances of the estimated parameters or analytically study the uncertainty variance by principal component analysis or linear series expansion of the model around the convergence point (α, g and h).

Quasi-equivalence also poses problems when it is necessary to determine the uncertainty in the estimated parameters, for instance in the context of significance testing. The quasi-equivalent different parameter sets, which yield essentially indistinguishable distributions, are not arbitrary, but form slightly curved, essentially one-dimensional manifolds in the parameter space, as our group and others have discussed in the literature several times. These manifolds may be similar to quasi-solution sets recently derived from Newton flow methods (see (Dedieu and Shub, 2005)). Whatever the structure of the quasi-solution sets may be, it is quite evident that equivalence tests focusing on one parameter at a time will not be useful. Instead, one will have to compare solutions globally, for instance based on Hellinger or Kullbach-Leibler distances (see (Balthis, 1998)) or on some measure of maximal distance, such as $Q_2 = \sup_X | F_1(X) - F_2(X) |$. To calculate a confidence interval for these distances, one would probably use the bootstrap. One could similarly use bootstrap methods to calculate *p*-values for the null hypothesis that two S-distributions are the same, although the bootstrap sampling for hypothesis testing would be slightly different than that used for confidence intervals. Furthermore, one could use Monte Carlo simulation methods to construct power curves for the alternative significance tests, under different true scenarios.

A related issue needing future attention will be the characterization of the intrinsic features of the 3-AR estimator, including its biasedness, consistency, and efficiency. These characterizations appear to be complex and may have to be postponed until the convergence behavior of 3-AR is more fully understood.

Finally, a future extension of 3-AR might be its generalization to the more comprehensive GS-distribution (Muiño *et al.*, 2006), which is characterized by increased flexibility in shape, in particular, for symmetric distributions, at the cost of one additional parameter. The inclusion of this additional parameter will require modifications to the 3-AR algorithm that need to be investigated in detail.

CHAPTER 4

PARAMETER OPTIMIZATION IN S-SYSTEMS WITH EIGENVECTOR OPTIMIZATION[vi]

4.1 Introduction

Chapter 2 introduced alternating regression (AR) as a fast deterministic method for parameter estimation in S-systems and showed that this method is genuinely different from traditional, much more expensive direct estimation methods. AR was shown to converge in most of the cases when the network structure is known, either by directly introducing topology constraints or by applying auxiliary structure identification algorithms (see Chapter 2 (Section 2.3.3) for details). However, AR sometimes leads to divergence when no structure information is applied. In this chapter, we propose a new method called *eigenvector optimization* (EO), which was inspired by AR and based on multiple linear regression and sequential quadratic programming (SQP) optimization. EO addresses the S-system parameter identification problem when no information about the network topology is known. In contrast to AR, EO operates initially only on one term (production or degradation), whose constant rate (α or β) and kinetic orders (g's and h's) are optimized completely before the complementary term is estimated. In many cases, the method provides alternative candidate models that fit the time series both in the decoupled and the fully integrated forms. Furthermore, the EO algorithm is extended to the optimization of network topologies with stoichiometric precursor-product constraints among equations.

[vi] This chapter is the result of a collaboration between Marco Vilela and me, therefore I will use the pronoun 'we' in this chapter. This chapter is adapted from: Vilela, M., Chou, I-C., Vinga, S., Vasconcelos, S. T. R., Voit, E. O., and Almeida, J. S. (2008) Parameter optimization in S-system models. *BMC Syst. Biol.*, 2,35.

4.2 Methods

4.2.1 Eigenvector optimization

The EO algorithm was inspired by AR method and is also based on decoupling and the substitution of differentials with estimated slopes. In contrast to AR, which estimates the parameter values by iterating between two phases of linear regression, the EO algorithm estimates one term (production or degradation term) per equation with high accuracy and then computes the other term through one step of linear regression ensuring that the new term will fall into the feasible space. Analogous to AR, EO is applied to S-system models of the format

$$\dot{X}_i = \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}, i = 1,2,...,n. \tag{4.1}$$

Suppose the S-system consists of n metabolites $X_1, ..., X_i, ..., X_n$, and for each metabolite, a time series consisting of m time points $t_1, ..., t_k, ..., t_m$ has been observed. Let $S_i(t_k)$ denote the estimated slope of metabolite i at time t_k. As shown in Chapter 2, we can reformulate the system as n sets

$$S_i(t_k) \approx \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}}(t_k) - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_k), k = 1, 2, \cdots, m. \tag{4.2}$$

Thus, the original system of n coupled differential equations can be analyzed in the form of $n \times m$ uncoupled algebraic equations. In simplified notation, we denote the production term and degradation term in Eq. (4.2) as $PT_i(t_k)$ and $DT_i(t_k)$, respectively. As the result, Eq. (4.2) is given as

$$S_i(t_k) \approx PT_i(t_k) - DT_i(t_k), k = 1, 2, \cdots, m. \tag{4.3}$$

If we move the degradation term to the left hand side, Eq. (4.3) can be rearranged as

$$S_i(t_k) + DT_i(t_k) \approx PT_i(t_k), k = 1, 2, \cdots, m. \tag{4.4}$$

Because PT_i must be positive, Eq. (4.4) can be rewritten as

$$\log\left(S_i + DT_i\right) \approx \log\left(PT_i\right), \tag{4.5}$$

where we omit the time argument for simplicity. As described in the introduction to AR, if the parameter values of the DT_i are guessed, Eq. (4.5) becomes a linear regression problem. The regression coefficient vector $\hat{\mathbf{b}}_\mathbf{i}$ contains the parameter values of PT_i and is obtained from

$$\hat{\mathbf{b}}_\mathbf{i} = \left(\mathbf{L}^\mathrm{T}\mathbf{L}\right)^{-1}\mathbf{L}^\mathrm{T}\mathbf{y}_\mathbf{i}, \tag{4.6}$$

where **L** denotes an m×(n+1) matrix of logarithms of regressors X_i, defined as

$$\mathbf{L} = \begin{bmatrix} 1 & \log\left(X_1\left(t_1\right)\right) & \cdots & \log\left(X_i\left(t_1\right)\right) & \cdots & \log\left(X_n\left(t_1\right)\right) \\ 1 & \log\left(X_1\left(t_2\right)\right) & \cdots & \log\left(X_i\left(t_2\right)\right) & \cdots & \log\left(X_n\left(t_2\right)\right) \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ 1 & \log\left(X_1\left(t_k\right)\right) & \cdots & \log\left(X_i\left(t_k\right)\right) & \cdots & \log\left(X_n\left(t_k\right)\right) \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ 1 & \log\left(X_1\left(t_m\right)\right) & \cdots & \log\left(X_i\left(t_m\right)\right) & \cdots & \log\left(X_n\left(t_m\right)\right) \end{bmatrix}, \tag{4.7}$$

and $\mathbf{y_i}$ is an m-dimensional vector $\mathbf{y}_\mathbf{i} = \log\left(S_i + DT_i\right)$. Based on the multiple linear regression model $\mathbf{y}_\mathbf{i} = \mathbf{L}\hat{\mathbf{b}}_\mathbf{i} + \boldsymbol{\varepsilon}_\mathbf{i}$, the predicted $\mathbf{y_i}$ values are

$$\hat{\mathbf{y}}_\mathbf{i} = \mathbf{L}\hat{\mathbf{b}}_\mathbf{i}. \tag{4.8}$$

Substituting $\hat{\mathbf{b}}_\mathbf{i}$ with the result in Eq. (4.6) directly yields

$$\hat{\mathbf{y}}_\mathbf{i} = \mathbf{L}\left(\mathbf{L}^\mathrm{T}\mathbf{L}\right)^{-1}\mathbf{L}^\mathrm{T}\mathbf{y}_\mathbf{i}. \tag{4.9}$$

The result will be the same if we substitute $\mathbf{y_i}$ with $\hat{\mathbf{y}}_\mathbf{i}$

$$\hat{\mathbf{y}}_\mathbf{i} = \mathbf{L}\left(\mathbf{L}^\mathrm{T}\mathbf{L}\right)^{-1}\mathbf{L}^\mathrm{T}\hat{\mathbf{y}}_\mathbf{i}. \tag{4.10}$$

Let $\mathbf{H} = \mathbf{L}\left(\mathbf{L}^\mathrm{T}\mathbf{L}\right)^{-1}\mathbf{L}^\mathrm{T}$, thus Eq. (4.10) becomes

$$\hat{\mathbf{y}}_\mathbf{i} = \mathbf{H}\hat{\mathbf{y}}_\mathbf{i}. \tag{4.11}$$

Recall that vector $\hat{\mathbf{y}}_{\mathbf{i}}$ is a function of the degradation parameters β_i and h_{ij}, which is the only set of parameter values in the equation, while information regarding the production parameters α_i and g_{ij} is embedded in matrix **H**. Specifically, $\hat{\mathbf{y}}_{\mathbf{i}}$ must be an eigenvector of the matrix **H** with an eigenvalue equaling 1. As described in Chapter 2 (Section 2.2.3.2), QR decomposition can be used to avoid augmentation of numerical error caused by floating point errors and will be described in detail in Section 4.2.2.

We used several standard algorithms to calculate the eigenvector of the matrix **H** directly, but none of them returned a satisfactory result. The presumed reason is that any vector which belongs to the eigenspace of **H** corresponding to eigenvalue 1 satisfies the Eq. (4.11). We therefore forced the eigenvector $\hat{\mathbf{y}}_{\mathbf{i}}$ to be in the form $\log(S_i + DT_i)$ and reformulated the task as a minimization problem for the logarithm of the squared residuals between the right and left side hands in Eq. (4.11) and defined this problem in matrix form with the cost function

$$F = \log\left(\left(\mathbf{H}\hat{\mathbf{y}}_{\mathbf{i}} - \hat{\mathbf{y}}_{\mathbf{i}}\right)^{\mathrm{T}}\left(\mathbf{H}\hat{\mathbf{y}}_{\mathbf{i}} - \hat{\mathbf{y}}_{\mathbf{i}}\right)\right) = \log\left(\left((\mathbf{H}-\mathbf{I})\hat{\mathbf{y}}_{\mathbf{i}}\right)^{\mathrm{T}}\left((\mathbf{H}-\mathbf{I})\hat{\mathbf{y}}_{\mathbf{i}}\right)\right). \quad (4.12)$$

The gradients of this function with respect to the degradation parameters β_i and h_{ij} can be obtained as

$$\frac{\partial F}{\partial \beta_i} = \frac{2}{\varphi}\left[(\mathbf{H}-\mathbf{I})\left(\left(\prod_{j=1}^{n} X_j^{h_{ij}}\right)\circ\left(S_i + DT_i\right)^{\circ -1}\right)\right]^{\mathrm{T}}\left[(\mathbf{H}-\mathbf{I})\log\left(S_i + DT_i\right)\right].$$

$$(4.13)$$

$$\frac{\partial F}{\partial h_{ij}} = \frac{2}{\varphi}\left[(\mathbf{H}-\mathbf{I})\left(\left(\beta_i\prod_{j=1}^{n} X_j^{h_{ij}}\circ\log\left(X_j\right)\right)\circ\left(S_i + DT_i\right)^{\circ -1}\right)\right]^{\mathrm{T}}\left[(\mathbf{H}-\mathbf{I})\log\left(S_i + DT_i\right)\right].$$

$$(4.14)$$

Here, the symbol $\circ$ represents the Hadamard product between vectors and $[\mathbf{v}]^{\circ -1} = [1/\mathbf{v}_{\mathbf{i}}]$ is the Hadamard inverse operation for a given vector (Magnus and Neudecker, 1999), and

φ is the logarithm of the argument of the right-hand side of the Eq. (4.12). The algorithm avoids infeasible solutions by satisfying the constraints

$$S_i(t_k) + \beta_i \prod_{j=1}^{n} X_j(t_k)^{h_{ij}} > 0\,, k = 1, 2, \cdots, m\,. \tag{4.15}$$

We used the **fmincon** routine in Matlab® (MathWorks) with built-in Sequential Quadratic Programming to execute the cost function constrained minimization.

After the parameters of the degradation term β_i and h_{ij} are estimated with high accuracy, the parameter values of the production term α_i and g_{ij} can be computed through one step of linear regression as in Eq. (4.6). The EO algorithm can also start with estimating the parameters of the production term, where the constraints for α_i and g_{ij} must applied as

$$\alpha_i \prod_{j=1}^{n} X_j^{g_{ij}}(t_k) - S_i(t_k) > 0, k = 1, 2, \cdots, m\,. \tag{4.16}$$

The parameter values of the degradation term are then computed with one linear regression after the production term is obtained.

4.2.2 Matrix computation representation of EO algorithm

As described in Chapter 2 (Section 2.2.3.2) Eqs. (2.14-2.20), QR decomposition can be used to avoid numerical error augmentation due to floating point errors. Therefore, matrix **H** in Eq. (4.11) can be reformulated in the following steps:

$$\mathbf{L} = \mathbf{Q}\begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix}, \tag{4.17}$$

$$\mathbf{L}^{\mathrm{T}}\mathbf{L} = \begin{pmatrix} \mathbf{R}^{\mathrm{T}} & \mathbf{0} \end{pmatrix}\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{R}^{\mathrm{T}} & \mathbf{0} \end{pmatrix}\begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix} = \mathbf{R}^{\mathrm{T}}\mathbf{R}\,, \tag{4.18}$$

$$\begin{aligned}\mathbf{H} &= \mathbf{Q}\begin{pmatrix}\mathbf{R}\\ \mathbf{0}\end{pmatrix}\left(\mathbf{R}^{\mathrm{T}}\mathbf{R}\right)^{-1}\left(\mathbf{R}^{\mathrm{T}}\mathbf{0}\right)\mathbf{Q}^{\mathrm{T}}\\ &= \mathbf{Q}\begin{pmatrix}\mathbf{R}\\ \mathbf{0}\end{pmatrix}\left(\mathbf{R}^{-1}\mathbf{R}^{-\mathrm{T}}\right)\left(\mathbf{R}^{\mathrm{T}}\mathbf{0}\right)\mathbf{Q}^{\mathrm{T}}\\ &= \mathbf{Q}\begin{pmatrix}\mathbf{I}\\ \mathbf{0}\end{pmatrix}\left(\mathbf{I}\quad \mathbf{0}\right)\mathbf{Q}^{\mathrm{T}}\\ &= \mathbf{Q}\begin{pmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\mathbf{Q}^{\mathrm{T}}\end{aligned} \quad . \tag{4.19}$$

The first n+1 vectors in **Q** are the eigenvectors of **H**. Therefore, $\hat{\mathbf{y}}_{\mathbf{i}}$ is the linear combination (or the span) of $\left[\mathbf{q}_1 \quad \mathbf{q}_2 \quad \cdots \quad \mathbf{q}_{n+1}\right]$:

$$\left[\mathbf{q}_1 \quad \mathbf{q}_2 \quad \cdots \quad \mathbf{q}_{n+1}\right]\begin{bmatrix}\delta_1\\ \delta_2\\ \vdots\\ \delta_{n+1}\end{bmatrix} = \hat{\mathbf{y}}_{\mathbf{i}} . \tag{4.20}$$

The problem of eigenvector optimization in Eq. (4.11) can be formulated as a minimization problem

$$\mathbf{min}\left\|\mathbf{H}\hat{\mathbf{y}}_{\mathbf{i}} - \hat{\mathbf{y}}_{\mathbf{i}}\right\|_{\mathbf{2}} . \tag{4.21}$$

Since $\mathbf{Q}^{\mathrm{T}}$ within norm has no effect, Eq. (4.21) can be written as

$$\begin{aligned}&\mathbf{min}\left\|\mathbf{Q}^{\mathrm{T}}\left(\mathbf{H}\hat{\mathbf{y}}_{\mathbf{i}} - \hat{\mathbf{y}}_{\mathbf{i}}\right)\right\|_{\mathbf{2}}\\ &= \mathbf{min}\left\|\mathbf{Q}^{\mathrm{T}}\left(\mathbf{Q}\begin{pmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\mathbf{Q}^{\mathrm{T}}\hat{\mathbf{y}}_{\mathbf{i}}\right) - \mathbf{Q}^{\mathrm{T}}\hat{\mathbf{y}}_{\mathbf{i}}\right\|_{\mathbf{2}}\\ &= \mathbf{min}\left\|\begin{pmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\mathbf{Q}^{\mathrm{T}}\hat{\mathbf{y}}_{\mathbf{i}} - \mathbf{Q}^{\mathrm{T}}\hat{\mathbf{y}}_{\mathbf{i}}\right\|_{\mathbf{2}}\\ &= \mathbf{min}\left\|\left(\left(\begin{pmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix} - \mathbf{I}\right)\mathbf{Q}^{\mathrm{T}}\right)\hat{\mathbf{y}}_{\mathbf{i}}\right\|_{\mathbf{2}}\end{aligned} \quad . \tag{4.22}$$

Let $\mathbf{W} = \left(\begin{pmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix} - \mathbf{I}\right)\mathbf{Q}^{\mathrm{T}}$, the cost function thus becomes

$$F = \log\left(\left(\mathbf{W}\hat{\mathbf{y}}_{\mathbf{i}}\right)^{\mathrm{T}}\mathbf{W}\hat{\mathbf{y}}_{\mathbf{i}}\right), \tag{4.23}$$

where $\hat{\mathbf{y}}_\mathbf{i}$ has the format $\hat{\mathbf{y}}_\mathbf{i} = \log\left(S_i(t_k) + \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_k) \right)$ (k = 1, 2,…, m). Throughout the chapter we will mainly base our computation on the algorithm steps described in Section 4.2.1.

4.2.3 Initial parameters guesses

Like all numerical optimization algorithms, the proposed method requires initial guesses. Satisfying the constraints in Eq. (4.15), the proposed algorithm calculates initial guesses for the kinetic order h_{ij}, given a user-supplied value β_i; specifically, h_{ij} and a small buffer value ε are chosen such that

$$\beta_i \prod_{j=1}^{n} X_j^{h_{ij}} = \varepsilon - S_i^- , \tag{4.24}$$

where S_i^- represents all negative slope values from the time series of X_i. A simple linear regression step in logarithmic space thus suffices to determine admissible initial guesses for the kinetic orders h_{ij}. In this fashion, for a given $\beta_{i,}$ small values of kinetic orders h_{ij} are provided to the optimization algorithm. As a technical note, it is easier to keep a null parameter value than to bring it to zero during the optimization. If the slope vector contains no negative values, the procedure is performed without ε. A flowchart of the proposed algorithm is shown in Figure 4.1.

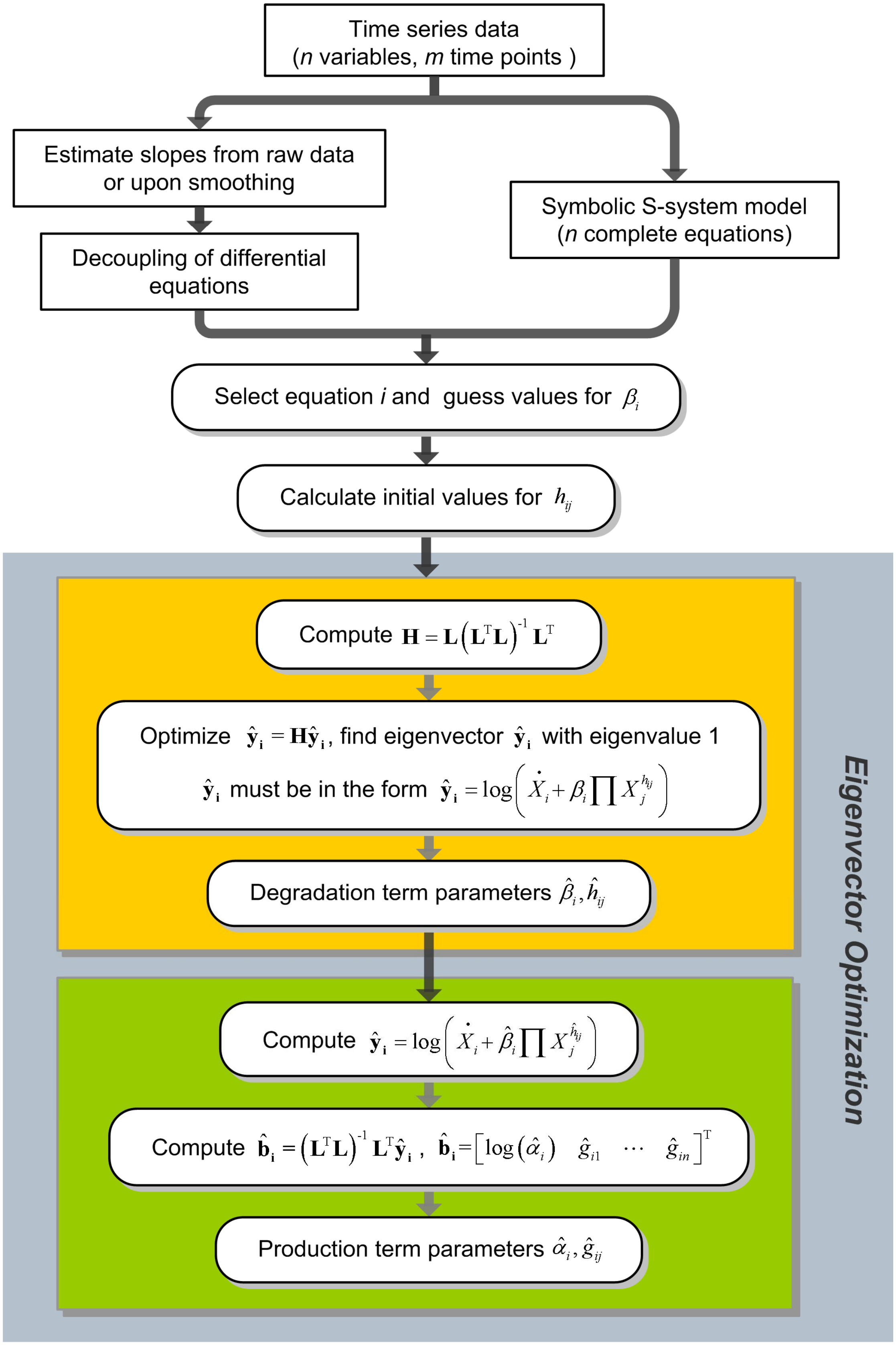

Time series data
(n variables, m time points)
Estimate slopes from raw data or upon smoothing
Decoupling of differential equations
Symbolic S-system model (n complete equations)
Select equation i and guess values for β_i
Calculate initial values for h_{ij}
Compute $\mathbf{H}=\mathbf{L}\left(\mathbf{L}^{\mathrm{T}}\mathbf{L}\right)^{-1}\mathbf{L}^{\mathrm{T}}$
Optimize $\hat{\mathbf{y}}_{\mathbf{i}}=\mathbf{H}\hat{\mathbf{y}}_{\mathbf{i}}$, find eigenvector $\hat{\mathbf{y}}_{\mathbf{i}}$ with eigenvalue 1
$\hat{\mathbf{y}}_{\mathbf{i}}$ must be in the form $\hat{\mathbf{y}}_{\mathbf{i}}=\log\left(\dot{X}_i+\beta_i\prod X_j^{h_{ij}}\right)$
Degradation term parameters $\hat{\beta}_i,\hat{h}_{ij}$
Compute $\hat{\mathbf{y}}_{\mathbf{i}}=\log\left(\dot{X}_i+\hat{\beta}_i\prod X_j^{\hat{h}_{ij}}\right)$
Compute $\hat{\mathbf{b}}_{\mathbf{i}}=\left(\mathbf{L}^{\mathrm{T}}\mathbf{L}\right)^{-1}\mathbf{L}^{\mathrm{T}}\hat{\mathbf{y}}_{\mathbf{i}}$, $\hat{\mathbf{b}}_{\mathbf{i}}=\left[\log\left(\hat{\alpha}_i\right)\quad \hat{g}_{i1}\quad \cdots\quad \hat{g}_{in}\right]^{\mathrm{T}}$
Production term parameters $\hat{\alpha}_i,\hat{g}_{ij}$
Eigenvector Optimization

Figure 4.1. Flowchart of the EO algorithm.

4.2.4 Refining solutions

Differently parameterized S-systems can exhibit quite similar temporal dynamics. This behavior is due the fact that S-systems are composed of production and degradation terms that may compensate for each other through different kinetic orders and constant rates that ultimately produce very similar time courses. As one consequence, it is quite common that optimization schemes identify non-zero values for parameters that should in truth be zero. Moreover, it is unlikely that any algorithm based on gradients will obtain parameters values exactly equal to zero. For these reasons, our algorithm automatically checks parameter values and forces kinetic orders below a quite arbitrary threshold of (0.009) to be zero; a new optimization process is initiated in which the parameter is constrained to be zero.

4.2.5 Extension to constrained topologies

To address linear pathway sections, constraints are imposed in accordance with the structure of the system when the parameter optimization is performed. For instance, for the linear system with precursor-product relationships (Figure 4.5; see Section 4.3.4 for a detailed description of the system), the optimization is performed with the degradation term of the precursor metabolite, which is forced to be equal to the production term of the product. In such a case, the Eq. (4.11) is formulated for each state variable

$$\begin{aligned} \mathbf{H}\hat{\mathbf{y}}_{\mathbf{1}}(\beta_1, h_{1j}, S_1) &= \hat{\mathbf{y}}_{\mathbf{1}}(\beta_1, h_{1j}, S_1), \\ \mathbf{H}\hat{\mathbf{y}}_{\mathbf{2}}(\alpha_2, g_{2j}, S_2) &= \hat{\mathbf{y}}_{\mathbf{2}}(\alpha_2, g_{2j}, S_2), \\ &\vdots \\ \mathbf{H}\hat{\mathbf{y}}_{\mathbf{n}}(\alpha_n, g_{nj}, S_n) &= \hat{\mathbf{y}}_{\mathbf{n}}(\alpha_n, g_{nj}, S_n). \end{aligned} \tag{4.25}$$

and the sum of the equations returns the eigenvector problem

$$\mathbf{H}\left(\sum_{i=1}^{n} \hat{\mathbf{y}}_{\mathbf{i}}\right) = \sum_{i=1}^{n} \hat{\mathbf{y}}_{\mathbf{i}} \,. \tag{4.26}$$

A cost function similar to Eq. (4.12) can be formulated using the Eq. (4.26), and the same optimization procedure is used. For instance, to force flux conservation in the example system, the following constraints are imposed on the optimization algorithm

$$\alpha_2 = \beta_1, \quad g_{2j} = h_{1j} \quad , j = 1, 2, .., n \tag{4.27}$$

to impose

$$PT_2 = DT_1, \tag{4.28}$$

and the production term of X_3 is forced to be equal the degradation term of X_2

$$PT_3 = DT_2. \tag{4.29}$$

Therefore, PT_3 can be computed as

$$\alpha_3 \prod_{j=1}^{n} X_j^{g_{3j}} = PT_2 - S_2. \tag{4.30}$$

Applying logarithms on both sides of the Eq. (4.30) and solving the equation by multiple linear regression, the final constraints are found as

$$\alpha_3 = \prod_{k=1}^{m} \left(PT_2(t_k) - S_2(t_k) \right)^{C_{1k}}, \tag{4.31}$$

and

$$g_{3j} = \sum_{k=1}^{m} C_{j+1,k} \log\left(PT_2(t_k) - S_2(t_k) \right) \ , j = 1, 2, .., n, \tag{4.32}$$

where $\mathbf{C} = \left(\mathbf{L}^{\mathrm{T}} \mathbf{L} \right)^{-1} \mathbf{L}^{\mathrm{T}}$. The constraints can be rewritten in a general form as

$$\alpha_n = \prod_{k=1}^{m} \left(PT_{n-1}(t_k) - S_{n-1}(t_k) \right)^{C_{1k}}, \tag{4.33}$$

and

$$g_{nj} = \sum_{k=1}^{m} C_{j+1,k} \log\left(PT_{n-1}(t_k) - S_{n-1}(t_k) \right) \ , j = 2, .., n. \tag{4.34}$$

Analogous optimization routines were used for other constraints.

4.3 Results

In the following sections I describe the main results related to this example using the EO algorithm. Some additional results are shown in Appendix B.

4.3.1 Synthetic time series

The EO method was tested on synthetic time series generated by reference test models of 2, 4, and 5 state variables. The 2-dimensional system (Kutalik *et al.*, 2007)

$$\begin{aligned} \dot{X}_1 &= 3X_2^{-2} - X_1^{0.5}X_2 \\ \dot{X}_2 &= X_1^{0.5}X_2 - X_2^{0.5} \end{aligned} \tag{4.35}$$

exhibits oscillatory behavior that is challenging for estimation purposes, leading to difficulties of standard algorithms in finding good solutions. The reason is that even small shifts in the oscillation phase between the dynamics of the estimated system and the true target system result in significant cumulative errors. By contrast, the 4-dimensional system (also see Chapter 3 (Section 2.3))

$$\begin{aligned} \dot{X}_1 &= 12X_3^{-0.8} - 10X_1^{0.5} \\ \dot{X}_2 &= 8X_1^{0.5} - 3X_2^{0.75} \\ \dot{X}_3 &= 3X_2^{0.75} - 5X_3^{0.5}X_4^{0.2} \\ \dot{X}_4 &= 2X_1^{0.5} - 6X_4^{0.8} \end{aligned} \tag{4.36}$$

is relatively well behaved and will be used to identify problems that are likely to emerge even for the inference of less complicated dynamic models. The 5-dimensional system (Hlavacek and Savageau, 1996)

$$\begin{aligned} \dot{X}_1 &= 5X_3X_5^{-1} - 10X_1^2 \\ \dot{X}_2 &= 10X_1^2 - 10X_2^2 \\ \dot{X}_3 &= 10X_2^{-1} - 10X_2^{-1}X_3^2 \\ \dot{X}_4 &= 8X_3^2X_5^{-1} - 10X_4^2 \\ \dot{X}_5 &= 10X_4^2 - 10X_5^2 \end{aligned} \tag{4.37}$$

describes an artificial gene regulatory network and has been used as a benchmark for S-system inference algorithms. For each test system, three different data sets, each with 100 data points, were created using different initial conditions in order to imitate different biological stimulus-response experiments (Appendix B (Section B.1.1)). These three data sets allowed us to assess the ability of the algorithm to deal with different time series dynamics. Using each data set, we performed 10 trials with the EO method for each variable (X_i) of the system. The runs differed in the random initial guess for β which was chosen from the range [0.1, 12] and the kinetic order values were initialized accordingly (see Section 4.2.3 for detail). The search space for kinetic orders was limited to a reasonable range of [-2, 3], which is consistent with collective experience in the field (see Chapter 5 in (Voit, 2000a)). In addition, no knowledge about the pathway was assumed and all parameters were considered freely variable in all three case studies.

The results demonstrate that the EO method retrieves the correct parameter values and network topology in most of the cases using noise-free time series. The procedure is computationally efficient, requiring 3 minutes to perform 40 optimizations for the 4-dimensional system (10 optimizations for each state variable corresponding to approximately 5 seconds per case), on a personal computer with a 2.00 GHz processor and 1GB RAM. Thanks to the numerical decoupling, the complexity of the algorithm is of the order $O(n \times m)$ where n is the number of state variables and m is the number of data points used in the optimization.

As an example result, the experiment with the 5-dimensional system performed on the first data set illustrates the success rate of the algorithm: the exact parameter values were found for all variables in all trails except for variable X_5 in one of the trials. Furthermore, the EO algorithm overcame the problematic identification of the kinetic orders g_{32} and h_{32} of the state variable X_3 presented by most algorithms in the literature. If a stop criterion is defined as a value of 1e-12 for the sum of the squared errors between the slopes of the optimized system and the true slopes, the time required to identify the system parameters for the 5-dimensional system is 23 sec on the machine described above. Similar results were achieved with the optimization of the 2-dimensional system.

An experiment with a 10-dimensional system was also performed and the total time consumed was 75 sec (see Appendix B (Section B.1.2)).

Issues encountered in finding the correct solutions appeared to be caused by a combination of different features of the system, such as the position of the optimal point within the feasible parameter space, an error surface with multiple local minima, as well as the particular choice of initial parameter guesses. These peculiarities of the algorithm and the problem itself lead to different parameter values, although the errors of the decoupled and integrated system are still small (typically about at the order of 1e-5).

The proposed algorithm calculates the initial guesses for the kinetic orders as close to zero as possible, given an initial β value (see Section 4.2.3 for detail). However, in a specific case study of the 2-dimentional system (Eq. (4.35)), near-zero values of the kinetic orders h_{11} and h_{12} for the constant rate β_1=1 fall into the infeasible parameter region, which complicates the parameter optimization. For instance, the smallest feasible value for h_{12} is 0.8636. The proposed algorithm overcomes this initial problem by adjusting itself and subsequently returns correct solutions when the system is rescaled in time (Voit, 1992a). This is most easily achieved by multiplying the alphas (α_1 and α_2) and betas (β_1 and β_2) with a positive factor, which increases the feasible parameter space. This step is, in fact, equivalent to multiplying the slope vector by a positive number. Thanks to the modularity of the decoupled system, this scaling can be performed separately for each state variable without affecting the kinetic order values. Only the values of the rate constants are changed, but they are easily recovered by dividing them by the positive number used for scaling. It was observed that this strategy often, but not always, enhances the algorithmic performance. It appears to improve performance most if the rate constants have small values.

The EO algorithm was not only performed with noise-free time series, but also tested in noisy data sets. Because the EO algorithm uses the decoupled, algebraic form, a signal extraction procedure was employed for the noisy data to provide smooth time series and slopes (Vilela *et al.*, 2007). The results show that the combination of smoother and the EO algorithm generate accurate dynamical responses for the cases studies used in our investigation (see Appendix B (Section B.1.3) for part of the results).

4.3.2 Error surfaces of decoupled S-systems

To explore the results of the proposed algorithm visually and to investigate patterns of convergence, we performed a grid search on the parameters of the 2-dimensional system as in Eq. (4.35). Specifically, we searched a 100×100 grid where each point represented the kinetic orders h_{11} and h_{12} over the range [-2.5, 2.0]. Correspondingly, 100 time points for X_1 and X_2 and its correspondent slopes S_1 and S_2 were generated by numerical integration of the 2-dimensional system with $X_1(t_0)$ =3 and $X_2(t_0)$ =1 as initial conditions. As described in Section 4.2.1, the time series of X_1 and X_2 were used to calculate the regression matrix **L**, and for each given initial value of the rate constant β_1 (uniformly spaced over the interval [1, 6]) and for each point of the grid, the error surface for the variable X_1 was constructed. The algorithm started with the degradation term ($DT_1 = \beta_1 X_1^{h_{11}} X_2^{h_{12}}$) for the first grid point using a given value for β_1 and the time series points for X_1 and X_2. Subsequently, the parameter vector of the production term ($\hat{\mathbf{b}}_\mathbf{1} = \begin{bmatrix} \log(\hat{\alpha}_i) & \hat{g}_{i1} & \hat{g}_{i2} & \cdots & \hat{g}_{in} \end{bmatrix}^\mathrm{T}$) was obtained from the slope vector S_1, the regression matrix **L**, and the degradation term DT_1 in Eqs. (4.5)-(4.6). Once all parameter values for variable X_1 in the production and degradation vectors were determined, the estimated slopes were calculated ($\hat{S}_1 = PT_1 - DT_1$) and the logarithm of the sum of the squared errors between these slopes and the target solutions was computed as $error = \log\left(\sum (S_1 - \hat{S}_1)^2\right)$. This process was repeated for all points on the grid such that an error surface resulted for each β_1 value. In this manner, ten surfaces were constructed using different β values; they are shown superimposed in Figure 4.2.

The first observation is that most of the search region is not feasible (unfilled *X*-*Y* space), even though there is *a priori* no hint that solutions in the open range should not converge. It turns out in retrospect that these are regions where the argument of the logarithm on left side of Eq. (4.5) is negative, due to negative slope values. Also worth noting is that for each β a similarly shaped surface ("bowl") was found, but that not all surfaces have the same minimal point (Figures 4.2 and 4.3). This information will be of critical importance in the discussion of the convergence profile of the proposed method.

The same strategy was applied to noisy time series resulting in a new set of surfaces (data not shown). Gaussian noise with 15% variance was added to the X_1 and X_2 time series and a refined Whitaker's filter (Vilela *et al.*, 2007) was used to smooth the data and estimate slopes. The error surfaces obtained using noisy data (Figure 4.4) present the same shapes as seen for the noise-free data except that the error average is higher and points to a different global minimum, which however is essentially indistinguishable in value from the local optima (see Appendix B (Section B.2) for details).

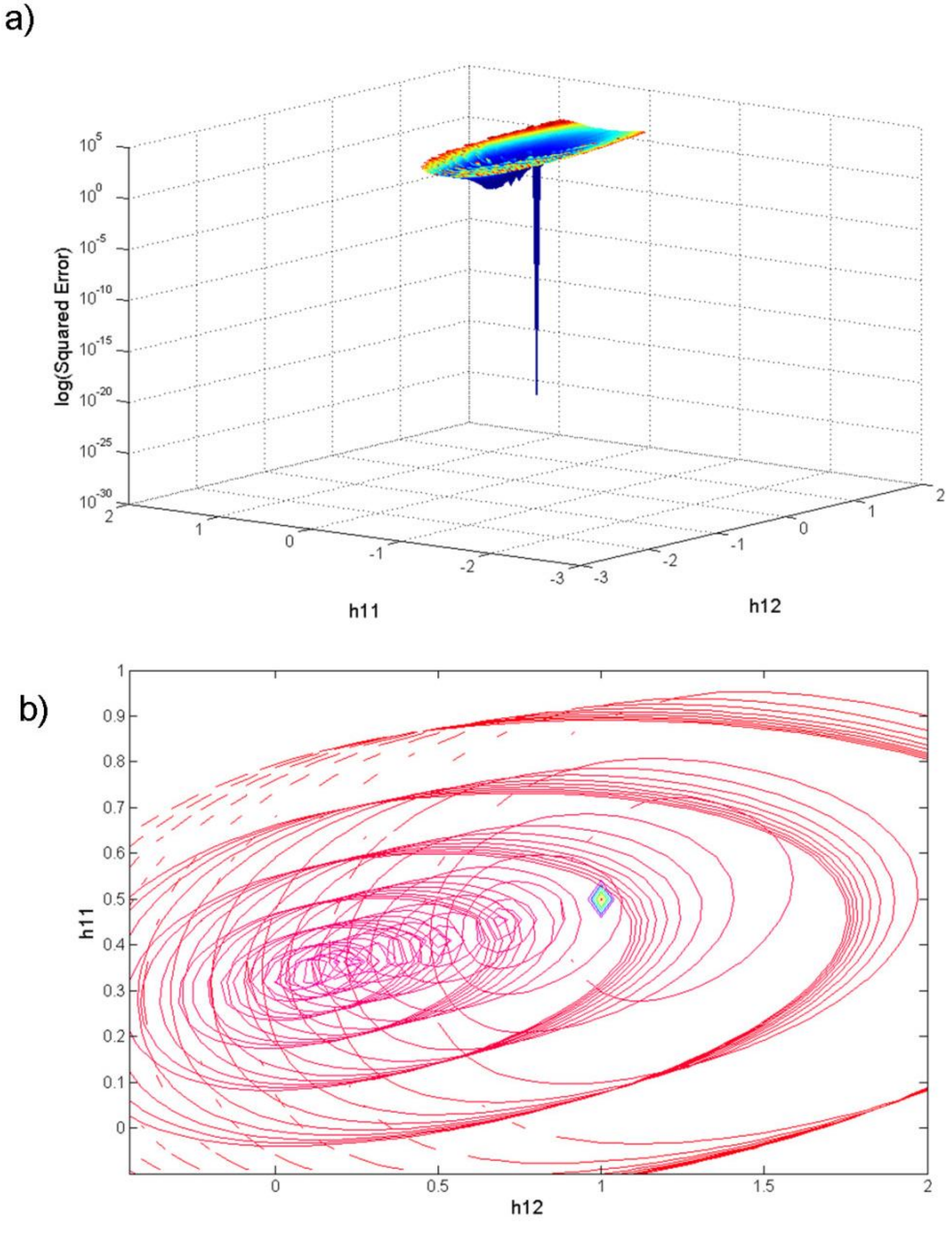


Figure 4.2. Error surfaces.
a) Ten error surfaces associated with variable X_1 of the 2-dimensional system were obtained using an exhaustive grid search covering 10 different initial guesses. b) Zooming in shows the composite contour map (level sets) of the error surfaces.

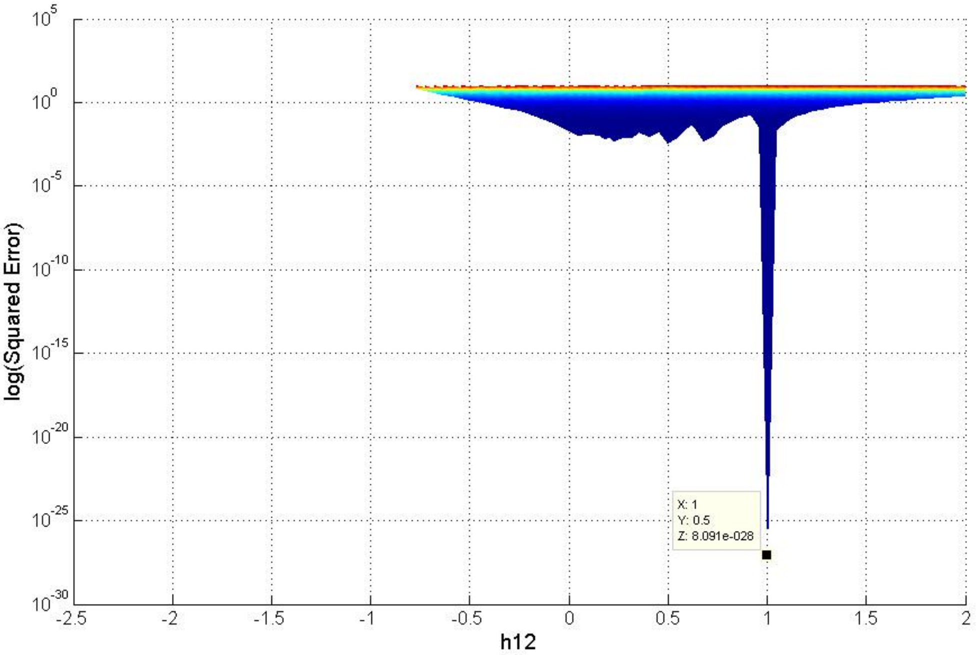


Figure 4.3. Multiple minima.
Z-*Y* projection of the error surfaces in Figure 4.2a. Different minima are found for different β values.

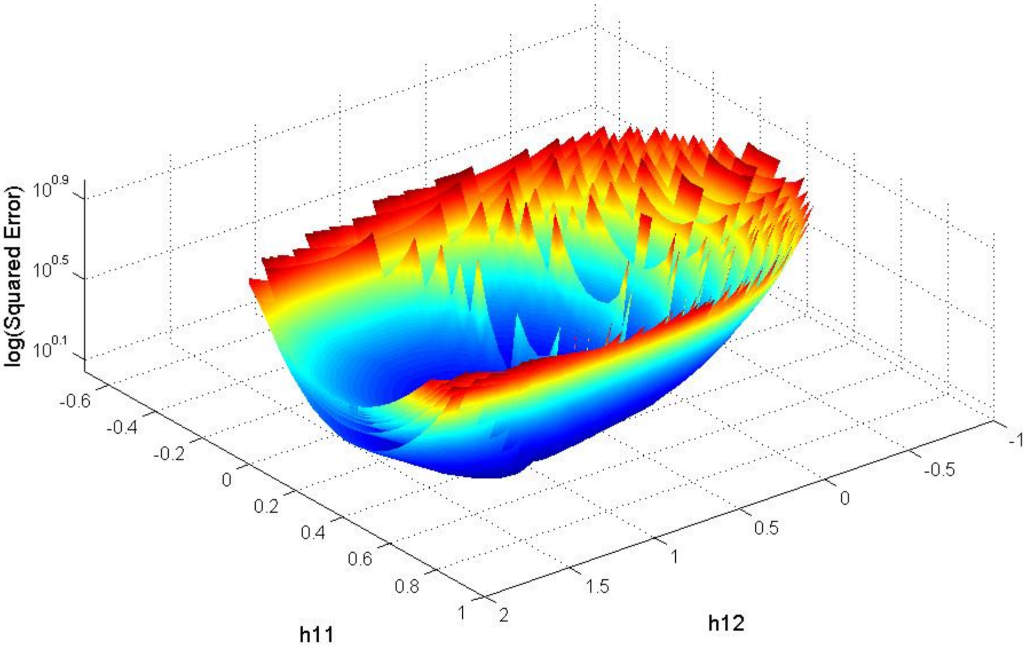


Figure 4.4. Error surfaces from noisy time series.
Ten error surfaces of the variable X_1 of the 2-dimensional system obtained from noisy time series after signal extraction and slope estimation.

4.3.3 Convergence problems

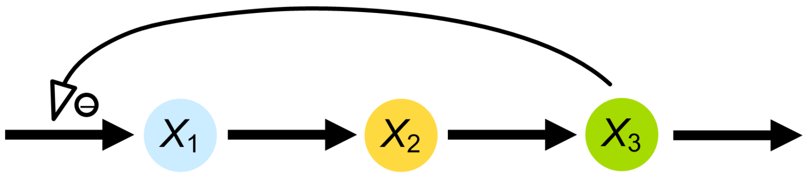
It would be unreasonable to assume that the algorithm converges to the global optimum under all imaginable conditions and initial settings: no estimation algorithm for nonlinear systems can—or should be expected to—measure up to such high a standard. For instance, if the ranges of initial guesses are changed or if the number of initial guesses is reduced, the algorithm may converge to an acceptable local minimum which, however, is not global. This is not surprising, given the complicated nature of the error surface of realistic systems and the fact that nonlinear systems often exhibit almost flat, banana-shaped or ellipsoid valleys in which the minimum is centered (Berg *et al.*, 1996; Sands and Voit, 1996; Gutenkunst *et al.*, 2007). At this point, a comprehensive picture of potential obstacles to convergence is not available.

One prominent reason for lacking or faulty convergence is that some problems are ill-posed, for instance, because of collinearity between columns of the regression matrix **L**. This situation occurs when two or more metabolites have similar dynamics or when at least one variable is essentially constant and is therefore collinear with the first column of the **L** matrix. In these and some other cases, the regression matrix **L** has a high condition number, which the proposed procedure flags. It might be possible to remedy some of these ill-posed problems with a regularization algorithm for multiple linear regression and through redesigning the algorithm with the regularized solution. It seems advisable in any event to remove model redundancies, for instance by pooling or eliminating collinear variables or merging essentially constant variables with the rate constants of the term.

4.3.4 Parameter estimation of constrained networks

The proposed method was extended to address the parameter identification for systems with topological constraints. This extension allows the algorithm to account for precursor-product relationships problems, which mandate that the degradation term of the precursor is equivalent to the production term of the product (Voit *et al.*, 2006a). Thus, instead of optimizing the parameters for each metabolite separately, a set of terms is optimized simultaneously, consisting of one of the parameter vectors (production or degradation vector) of each metabolite. As an illustrative, simple example, consider a linear pathway with feedback, where we have to account for constraints between the production and degradation terms of subsequent metabolites (Figure 4.5). Specifically in the example system, the efflux from X_1 is identical to the influx into X_2, and the efflux from X_2 is identical to the influx into X_3. Consequently, the degradation term of X_1 is exactly the same as the production term of X_2, and the degradation term of X_2 must be the same as the production term of X_3. The amendment of the proposed method toward simultaneous estimation readily satisfies these types of constraints.

The extended algorithm was applied to the 3-dimensional linear pathway system in Figure 4.5. The detail steps have been described in Section 4.2.5. The EO algorithm found the correct parameter set, and all 10 optimizations, in which the EO algorithm now performs a single, combined optimization for all variables simultaneously, thereby accounting for constraints, were completed in 37sec on a 2.00 GHz processor with 1GB RAM.



$$\dot{X}_1 = 12X_3^{-0.8} - 10X_1^{0.5}$$
$$\dot{X}_2 = 10X_1^{0.5} - 3X_2^{0.75}$$
$$\dot{X}_3 = 3X_2^{0.75} - 5X_3^{0.5}$$

Figure 4.5. Linear system topology.
Linear pathway with precursor-product constraints.

4.3.5 Software application

An open source Matlab® toolbox and a stand-alone compiled Graphical User Interface (GUI) application were developed as an exploratory tool (see Appendix B (Section B.3) for availability). The application was developed as a modular extension of previous work from our labs and constitutes a critical component within our long-term effort of advancing a data processing pipeline for S-system estimation from metabolomic time series (Almeida and Voit, 2003; Vilela *et al.*, 2007).

4.4 Discussion

There are many reasons why it may be desirable to reverse engineer a biological network without making assumptions about the underlying processes. The most obvious reason is that no reliable information may be available about the processes. Another situation occurs when several network topologies are *a priori* possible and the reverse approach is employed to prioritize alternative hypotheses. The eigenvector optimization (EO) proposed here is an extension of alternating regression (AR) that in many cases shows improved convergence behavior.

The EO algorithm was exhaustively tested on diverse time series (see Section 4.3 and Appendix B). In all of these tests, the convergence followed the same pattern: the error slowly decreased during the first few iterations and then suddenly dropped to a significant lower plateau, from where it gradually decreased again. This pattern repeated until one of the stop conditions (maximal number of iterations, minimal gradient value or minimal cost function value) was reached. The error drop points coincided with significant changes in the beta gradient and appear to correspond to transitions to a "bowl" with a lower error surface (*cf.* Figures 4.2 and 4.4). As shown in Figures 4.3b and 4.4, most "bowls" have different minimal points, corresponding to good, yet local minima. Because the proposed algorithm is computationally very efficient, it allows the exploration of the parameter space in a reasonable amount of time (within seconds to minutes). Such an exploration with new initial β values is recommended, if very precise solutions or alternative parameter sets are needed. Because alternative parameter combinations may correspond to different topological and regulatory structures (*e.g.*, (Voit, 2000a)), estimations with different initial values in fact constitute explorations of the structure and functionality of the biological space in which the pathway operates.

4.5 Conclusion

S-systems present a unique balance between proven biological relevance and validity on one hand, and mathematical convenience and tractability on the other. For this reason, the recent years have seen numerous methods for matching S-system models to measured biological time series data. In the relatively simpler scenario of this type, the topology and regulatory structure of the biological system is known, and the extraction of information from the data constitutes a parameter estimation task. In the more difficult situation, at least some of the structure is unknown, and in the extreme situation no information about the topology of the interactions between variables is available. In this chapter we propose a new algorithm that efficaciously identifies the correct topology of a system from time series. The only true assumptions made are that all important variables are accounted for and that the S-system model is capable of modeling the data. The first assumption is presently unavoidable, at least in the generality presented above. The second assumption has been found to be true in very many cases, as a rich body of publications on S-systems demonstrates. The EO algorithm was conceived as a critical piece of an emerging data processing "pipeline" that will eventually accept time series and other data characterizing biological pathways and more or less automatically propose topological and regulatory structures that are consistent with the input data. This algorithm will be a valuable tool for analysis and hypothesis generation in systems biology.

CHAPTER 5

INVERSE MODELING APPROACH AND PARAMETER ESTIMATION STRATEGIES[vii]

5.1 Introduction

As described in Chapter 1, many methods have been developed recently that attempt to solve parameter estimation and structure identification problems through inverse modeling using the BST formalism. Most of the methods were developed to address the main problem of optimizing parameter values against observed time series data; they used gradient base methods, regression algorithms, or evolutionary approaches. Other methods were proposed as support algorithms including, for instance, methods for avoiding the time consuming integration of differential equations, smoothing noisy data and estimating slopes, restricting the parameter search space, excluding unlikely connections within the network, or reducing the number of parameters to be estimated (see Chapter 1 (Section 1.4) for details).

Many of the published papers used a combination of several methods to solve the inverse problem. For instance, they used decoupling techniques along with various optimization algorithms, tried to reduce the number of parameters before estimating their values, or included several objective functions to constrain the solution space. The algorithms that were proposed in Chapter 2 and 4 respectively, namely alternating regression (AR) and eigenvector optimization (EO), also merge several methods for solving inverse problems in BST models. I will briefly summarize and compare the features of AR and EO methods in Section 5.2.

[vii] Some of the material in this chapter was presented at International Conference on Molecular Systems Biology 2008 (ICMSB08) in the Manila, Philippines (Chou *et al.*, 2008).

In spite of a considerable number of methods that have been proposed for inverse modeling using BST models, each method has its pros and cons and there is currently no algorithm which is perfect, or even sufficiently effective, for the majority of realistic cases. Before applying the algorithm on the real experimental data, synthetic time series data are typically used first to test the robustness and efficacy of the algorithms and examine if the inverse algorithm can correctly find the true optimum when noise does not exist. However, to some extent, it is still hard to tell from the published results which algorithms are superior to the others.

The reasons for these difficulties can be categorized into five aspects. First, different biochemical systems were used to demonstrate the usefulness of the algorithms. It is clear that different systems generate distinct synthetic time series which comprise the data matrices for subsequent computation. These matrices may be intrinsically different. For instance, the matrix may be ill-conditioned or exhibit collinearity between rows or columns which may affect the correctness and efficacy of the tested algorithms. Therefore, it is difficult to compare the methods and distinguish the influence of tested system from the algorithm itself.

Second, the numbers of time series points included for computation are different or unstated. Thus, the effect of data point inclusion on the algorithm is unclear and it can change the fitness score of the information criteria.

Third, the objective functions set up for the optimization problems are various which prevents direct comparisons among algorithms.

Fourth, the upper and lower limits or constraints of the parameter values are often different. Thus, it is hard to tell if the algorithm converges since the boundaries are relatively close to the true optimum or because of the efficiency of the algorithm.

Fifth, in addition to testing the methods using noise-free data, errors are introduced to exam if the algorithms can still find the correct parameter values. However, the way and extent of adding noise and the methods used for data smoothing often differ, which makes the comparison harder. To avoid the problems indicated above, del Rosario and co-workers (del Rosario *et al.*, 2008b) recently proposed a project called MADMan (Munich, Atlanta, DiliMAN (Philippines)), which aims to compare the published parameter estimation algorithms using BST formalisms in a systematically way, including the testing of the algorithms with the same variety of networks, uniform benchmarking bases, and standardized evaluation criteria. The goal of the benchmarking framework is to develop a strategy for choosing a set of candidate algorithms given a biochemical network and experimental data. MADMan is an ongoing project. It constitutes a huge task, which requires a lot of effort and the cooperation between different groups. Our group is and will be involved in MADMan.

The direct comparison of various optimization algorithms will ultimately be the least biased strategy to determine which algorithms are better than the others. However, speed (or lack) of convergence and unsatisfactory performance in terms of fitness, are merely some of the issues that need to be analyzed for each optimization algorithm or computational software. Other sources may contribute to the problem as well, such as data related issues, model related issues, and mathematical issues, as reviewed in Chapter 1 (Section 1.3.4). Therefore, with the same goal of developing methods for effective, robust, and scalable estimation, I have been working toward a streamlined “work-flow” strategy for estimating parameter values in models within BST. Instead of suggesting which algorithm(s) should be used, the flow diagram proposes a decision process which indicates the possibly problematic steps and suggests relevant diagnostic tools or corresponding solutions. The details of the flow diagram will be introduced in Section 5.3.

5.2 Comparison of algorithms

The details of alternating regression (AR) and eigenvector optimization (EO) have been reviewed in Chapters 2 and 4, respectively. In this section, I will summarize the features of both methods and compare their similarities and differences. In addition, their pros and cons and applicability under different conditions will be reviewed briefly.

The general algorithm flow of AR and EO is shown in Figure 5.1. For simplicity, the flow chart shows the steps of parameter estimation of the i^{th} equation in the model. High-throughput technologies enable measurements of biological components, such as metabolites, at a series of time points *in vivo* after defined stimuli from the same organism. The time series contain the data from n variables (metabolites), and for each variable there are m measurements (Step ①). The slope at each time point of the time series is measured (or estimated) directly or upon smoothing, if the time series data are more or less noisy (Step ②). After the slopes are deduced, the differentials are substituted with slopes, which replaces the n original differential equations with n sets of m algebraic equation (Step ③). At the same time, a symbolic S-system model is derived, where all variables are involved and fully connected with each other (Step ④). So far the steps are identical for both AR and EO methods. As described in Chapter 2, the AR algorithm works better when the network topology is known. Therefore, given a concept map of a network whose structure and regulation are fully or partially known, a symbolic S-system model can be generated by directly translating the network structure into equations by hand or with the aid of supporting network identification techniques (Step ⑤). The symbolic model is fitted to the time series data by means of the AR algorithm, which reduces the nonlinear estimation problem into iterative steps of linear regression, starting with guesses for all β_i and h_{ij} values for each set of algebraic equations (Step ⑥). These guesses are used to obtain the parameters of the α_i-term by multivariate linear regression. The resulting estimates for α_i and all g_{ij} are used for the next iteration and the parameters of the β_i-term are estimated. The method thus switches back and forth, thereby improving estimates of all parameters (Step ⑦).

Different from AR, EO method does not necessary require the knowledge of network topology, and thus the full symbolic S-system model is used in data fitting. In analogy with the AR method, the initial guess of β_i is needed (Step ⑧), however, the initial values of h_{ij} are computed according to β_i and other constraints (Step ⑨). Given the initial values of β_i and h_{ij}, the EO algorithm optimizes the β_i-term using a distinct objective function which involves finding an eigenvector of the matrix in which the information of the α_i-term parameters. Unlike the AR algorithm which iteratively switches back and forth between two phases of linear regression, the EO method estimates the β_i-term completely and uses the result to estimate the α_i-term parameters in just one step of linear regression (Step ⑩).

Both AR and EO algorithms are designed for the S-system format and demonstrate good convergence speed compared to other traditional optimization methods. They both incorporate decoupling techniques to avoid the integration of differential equations. As the result, data smoothing and slope estimating are required for both methods. Several smoothers and slope estimation techniques have been reviewed in Chapter 1 (Section 1.4.2) and need no further discussion here. Furthermore, the parameters of each equation are estimated separately after decoupling. Therefore, the development of methods to account for constraints among equations, such as stoichiometric precursor-product or branch point relationships, is needed for both methods. The EO method was shown to be able to find the correct parameter values by simultaneously optimizing the objective functions of all equations in a simple linear pathway (see Chapter 4 (Section 4.2.5) for detail).

It is clear that for both AR and EO, the data matrix **L** is an essential constituent component in either the linear regression or the computation of the **H** matrix. Therefore, the characteristics and quality of the matrix must be expected to be crucial in the parameter estimation process. I have shown with test cases that an ill-conditioned matrix may cause problems for both algorithms. It might be possible to remedy some of these ill-posed problems, such as the collinearity, by pooling variables or merging essentially constant variables with the rate constants of the term. Another problem with the data matrix may be caused by time series data with very small values (~zero), since the **L** matrix contains the logarithmic values of the measurements. The problem may be alleviated to some extent by time domain subdivision or some weighting schemes.

In addition to the differences described in the previous paragraph regarding the flow diagram of the two methods, the convergence patterns are quite different in AR and EO. As shown in Chapter 4 (Figure 4.4) with a cascaded attractor, the EO method finds the true optimum as long as the initial guesses are within the range of the attractor that contains the global minimum. If the initial guesses are outside that attractor, the algorithm will lead to other local minima, even though the error is still small and the fitting is visually good. This is possibly so because the EO method does not include the information of network topology in the symbolic model initially and thus keeps all the parameter freely adjustable. As the result, the parameters in the production term and the degradation term are compensating each other, which may generate perfect fitting with small error, but the model may not have much meaning and have little predictive power.

The problem can be partially alleviated by pruning parameters when the values are smaller than a threshold or using other pruning methods (see Chapter 1 (Section 1.4.4) for review). In contrast to EO, Figure 2.4 in Chapter 2 shows quite a different convergence attractor for the AR algorithm. Testing on the synthetic time series, if AR converges, it converges to the global optimum no matter how far away the initial guesses are, as long as the initial values are within the basin of attraction. In other word, the AR method breaks the boundaries of the cascaded attractor as shown in Chapter 4 Figure 4.4 and gradually approaches the true optimum. However, for some cases when AR does not converge, the convergence patterns and basins are complex and need further investigation.

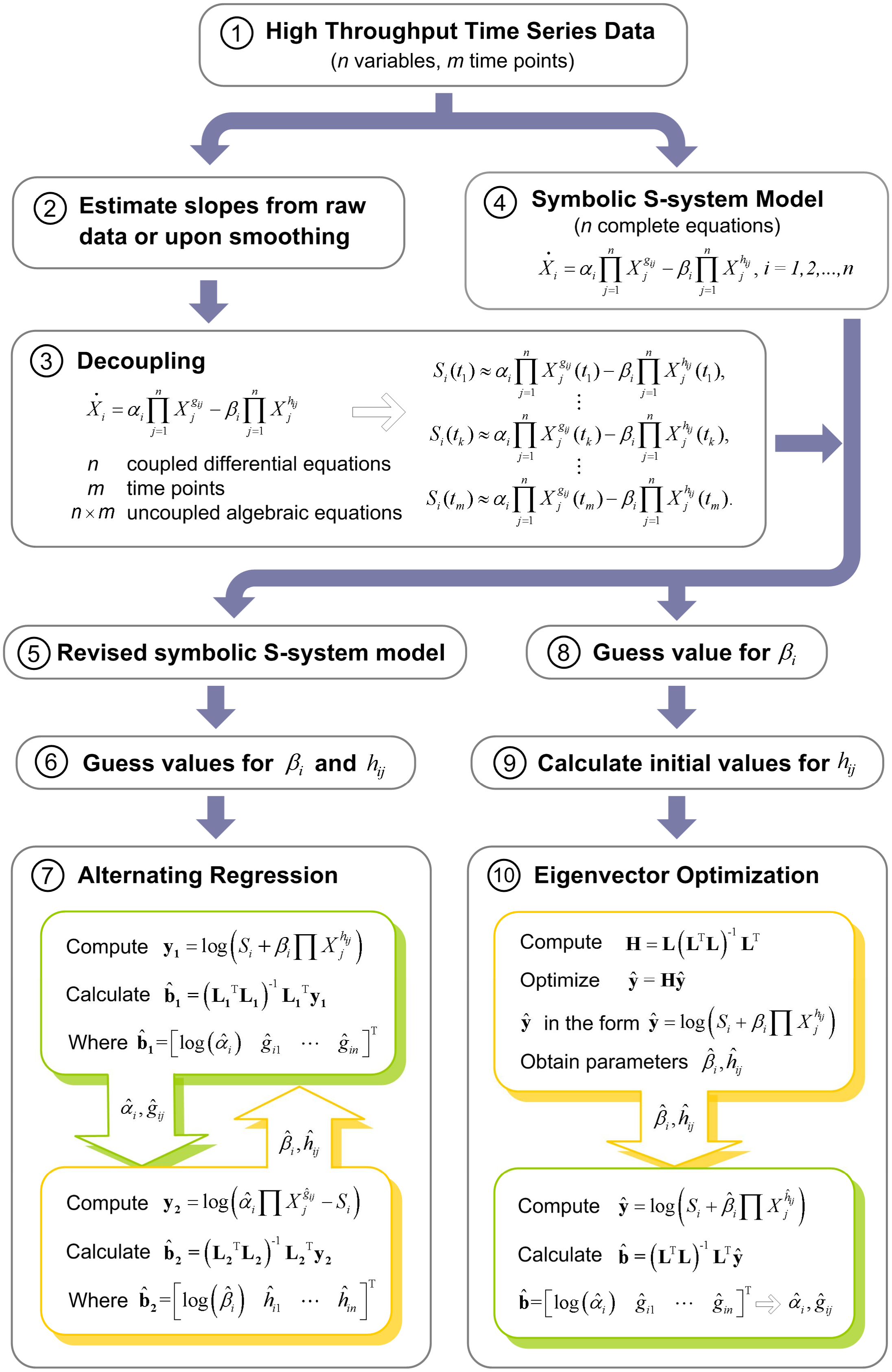

1 High Throughput Time Series Data
(n variables, m time points)
2 Estimate slopes from raw data or upon smoothing
4 Symbolic S-system Model
(n complete equations)
$\dot{X}_i = \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}, i = 1, 2, ..., n$
3 Decoupling
$\dot{X}_i = \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}$
n coupled differential equations
m time points
$n \times m$ uncoupled algebraic equations
$S_i(t_1) \approx \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}}(t_1) - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_1),$
$\vdots$
$S_i(t_k) \approx \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}}(t_k) - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_k),$
$\vdots$
$S_i(t_m) \approx \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}}(t_m) - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}(t_m).$
5 Revised symbolic S-system model
8 Guess value for β_i
6 Guess values for β_i and h_{ij}
9 Calculate initial values for h_{ij}
7 Alternating Regression
Compute $\mathbf{y}_1 = \log\left(S_i + \beta_i \prod X_j^{h_{ij}}\right)$
Calculate $\hat{\mathbf{b}}_1 = \left(\mathbf{L}_1{}^{\mathrm{T}}\mathbf{L}_1\right)^{-1} \mathbf{L}_1{}^{\mathrm{T}}\mathbf{y}_1$
Where $\hat{\mathbf{b}}_1 = \left[\log(\hat{\alpha}_i) \quad \hat{g}_{i1} \quad \cdots \quad \hat{g}_{in}\right]^{\mathrm{T}}$
$\hat{\alpha}_i, \hat{g}_{ij}$
$\hat{\beta}_i, \hat{h}_{ij}$
Compute $\mathbf{y}_2 = \log\left(\hat{\alpha}_i \prod X_j^{\hat{g}_{ij}} - S_i\right)$
Calculate $\hat{\mathbf{b}}_2 = \left(\mathbf{L}_2{}^{\mathrm{T}}\mathbf{L}_2\right)^{-1} \mathbf{L}_2{}^{\mathrm{T}}\mathbf{y}_2$
Where $\hat{\mathbf{b}}_2 = \left[\log(\hat{\beta}_i) \quad \hat{h}_{i1} \quad \cdots \quad \hat{h}_{in}\right]^{\mathrm{T}}$
10 Eigenvector Optimization
Compute $\mathbf{H} = \mathbf{L}\left(\mathbf{L}^{\mathrm{T}}\mathbf{L}\right)^{-1}\mathbf{L}^{\mathrm{T}}$
Optimize $\hat{\mathbf{y}} = \mathbf{H}\hat{\mathbf{y}}$
$\hat{\mathbf{y}}$ in the form $\hat{\mathbf{y}} = \log\left(S_i + \beta_i \prod X_j^{h_{ij}}\right)$
Obtain parameters $\hat{\beta}_i, \hat{h}_{ij}$
$\hat{\beta}_i, \hat{h}_{ij}$
Compute $\hat{\mathbf{y}} = \log\left(S_i + \hat{\beta}_i \prod X_j^{\hat{h}_{ij}}\right)$
Calculate $\hat{\mathbf{b}} = \left(\mathbf{L}^{\mathrm{T}}\mathbf{L}\right)^{-1}\mathbf{L}^{\mathrm{T}}\hat{\mathbf{y}}$
$\hat{\mathbf{b}} = \left[\log(\hat{\alpha}_i) \quad \hat{g}_{i1} \quad \cdots \quad \hat{g}_{in}\right]^{\mathrm{T}} \Rightarrow \hat{\alpha}_i, \hat{g}_{ij}$

Figure 5.1. Flow chart of alternating regression (AR) and eigenvector optimization (EO) algorithms.

In summary, AR and EO methods are both fast. The AR method works well when the network topology is known and when the connectivity is sparse. The EO method can be used when the network structure is barely known. However, the results need to be evaluated carefully since the model might not have much meaning because of compensation between terms. Good smoother and slope estimating algorithms are needed for both AR and EO methods. Both algorithms are negatively affected by ill-posed problems and small numerical values in the data matrix. Data matrix preprocessing is needed when such conditions exist.

5.3 Toward a Streamlined "Work-Flow"

The comparison of AR and EO in the previous sections clearly demonstrates that each algorithm has its pros and cons and that there are conditions and situations where one works well and the other not so. While the MADMan project is attempting to clarify the applicability of methods under a wide range of conditions, I propose in this section a streamlined "work-flow" strategy for estimating parameter values in models within BST using a more general approach instead of naming the specific winning algorithm, which so far does not exist. The work-flow diagram consists of a decision process based on possible problems that are often encountered. These include issues related to the time series data, model of choice, computational efficiency, and mathematical redundancy during the inverse modeling process. The work-flow also suggests relevant diagnostic tools or corresponding solutions. One can safely anticipate that there is no unique recipe for solving the inverse problem in absolute generality. In many cases, a mixture of various methods, consisting of a main optimization algorithm and other supporting methods, augmented by diagnostic techniques along with some assumptions or educated guesses, will be required to estimate all parameter values of a system with realistic size. Before I go into the detail of the flow diagram, the goal of this approach will be first discussed in the next Section (5.3.1).

5.3.1 Goal of work-flow strategy

The ultimate goal of inverse modeling is to find a mathematical model that can describe the biological phenomenon and predict situations that had not been used for model identification or data fitting with correctness, robustness, and also, efficiency. These standards may not be fulfilled at the same time, or only partially satisfied with some compromise. For instance, the algorithm that finds the optimal solution may cost more computational time, whereas some of the fast algorithms may only be able to find coarse solutions.

The decision of the algorithms to be used is relatively easy when testing with synthetic time series since the "correctness" is easy to assess by checking the fitness and comparing the estimates with the true model parameters. However, in reality, the "correct" model is not known and the goodness of fit cannot always guarantee the reliability and applicability of the model. A model with the "smallest" fitting error is mathematically the "best" model in terms of goodness of fit. However, it does not necessarily imply that the model is the best model to describe the biological system. In many actual cases, the "best model" cannot be extrapolated toward untested conditions when no extra constraints are introduced, and the model tends to have over-fitting problems (see Chapter 1 (Section 1.5) for review). Furthermore, the solution that fits the observed time series quite well is not necessarily determined uniquely. Other solutions may exist which yield fits with similar quality and all solutions should be considered as candidate models. Therefore, instead of aiming to find one model with as small a fitting error as possible using a costly algorithm, the goal of inverse modeling strategy I propose here is to use a combination of approaches, starting with fast algorithms, to find a set of coarse candidate models that are all consistent with the data. The candidate set of parameters scattered in the search space is helpful to explore the discrepancies between models and data and to propose the possible causal relationship among the network components. These coarse models can be used to test stability, sensitivity, logarithmic gains, or other diagnostic tools to study the features of the models (Voit, 2000a; Goel *et al.*, 2006). These features show whether the coarse model has a chance to be correct and has predictive power, because lack of stability or high sensitivity are often unrealistic in biological systems. Furthermore, the model can be used to do various simulations, which are cheap to execute and usually quickly reveal some of the potential problems of the model and its assumptions. These models can then be experimentally validated and used for guiding further experimental designs.

5.3.2 Flow diagram of inverse modeling strategy

The proposed flow diagram of inverse modeling is shown in Figure 5.2. Given global time series data (Step ①), the data matrix is processed by specific diagnostic tools. For instance, if the variable traces have similar dynamics or are essentially constant, the traces are (approximately) collinear with each other. The calculation of the condition number or correlation coefficient can point out the possible collinearity in the data matrix (Step ②). If the time traces are collinear, one may remove the model redundancy by pooling collinear variables or merge constant variables with the rate constant (Step ③). If there is no collinearity, a symbolic mathematical model of the system can be derived based on the model of choice, without numerical specification of parameter values (Step ④). It has been shown that S-system and GMA representations in BST are good candidates for this propose. After setting up the full model, if the network topology is known, a revised symbolic model can be formulated directly based on the network diagram (Step ⑤). If there are ubiquitous metabolites in the system, partial modeling techniques may be applied, which further refine the symbolic model. Since fast optimization methods are recommend for the initial stage, most of the algorithms require the decoupling technique, which converts the differential to algebraic equations. The decoupling step involves the measurement of slopes directly or upon smoothing (Step ⑥). Once the symbolic model is decoupled, the parameters of each equation can be estimated by some fast optimization algorithms (Step ⑦). Alternating regression (AR) is shown to be one of the algorithms that works quite well in most of the cases under this condition. If AR converges, a coarse model is generated for further analysis and evaluation. If the initial guesses lead to inadmissible areas or lack of convergence, the fast speed enables the algorithm to start with different set of initial guesses. However, if the topology is not known or only partially known, algorithms or techniques for finding the network connectivity are applied, such as prior linearization of the system dynamics or sorting of parameter combinations by their empirical likelihood of inclusion in an equation (Step ⑧; see Chapter 1 (Section 1.5) for detail). Another choice under the condition when the network topology is not known is to choose an optimization algorithm where the topological information is not necessary required, such as eigenvector optimization (EO) with decoupling (Steps ⑨ and ⑩). These algorithms are usually embedded with pruning methods which eliminate unlikely connections between network components and reduce the number of parameters during the process of estimation. If the fast algorithms are not able to yield acceptable fittings, some other, more expensive algorithms such as genetic algorithm or evolutionary approaches are applied (Step ⑪). The estimation results from the previous algorithms can be used as initial guesses for the subsequent algorithms, although this approach may not always be effective, if the initial fitting is far from acceptable. However, if the candidate estimates which are obtained very quickly during a coarse parameter estimation using fast methods are more or less acceptable, the solution can then be refined toward a very good local or even global minimum afterwards. A significant consequence and advantage of the combined approach is that the result often consists of multiple parameter sets that are all consistent with the data and that can lead to hypotheses offering guidance for further theoretical and experimental investigation (Step ⑫). It may also be useful to resample the data with jackknife or bootstrap methods (Voit, 2000a) and to redo the analysis in order to explore possible alternative solutions.

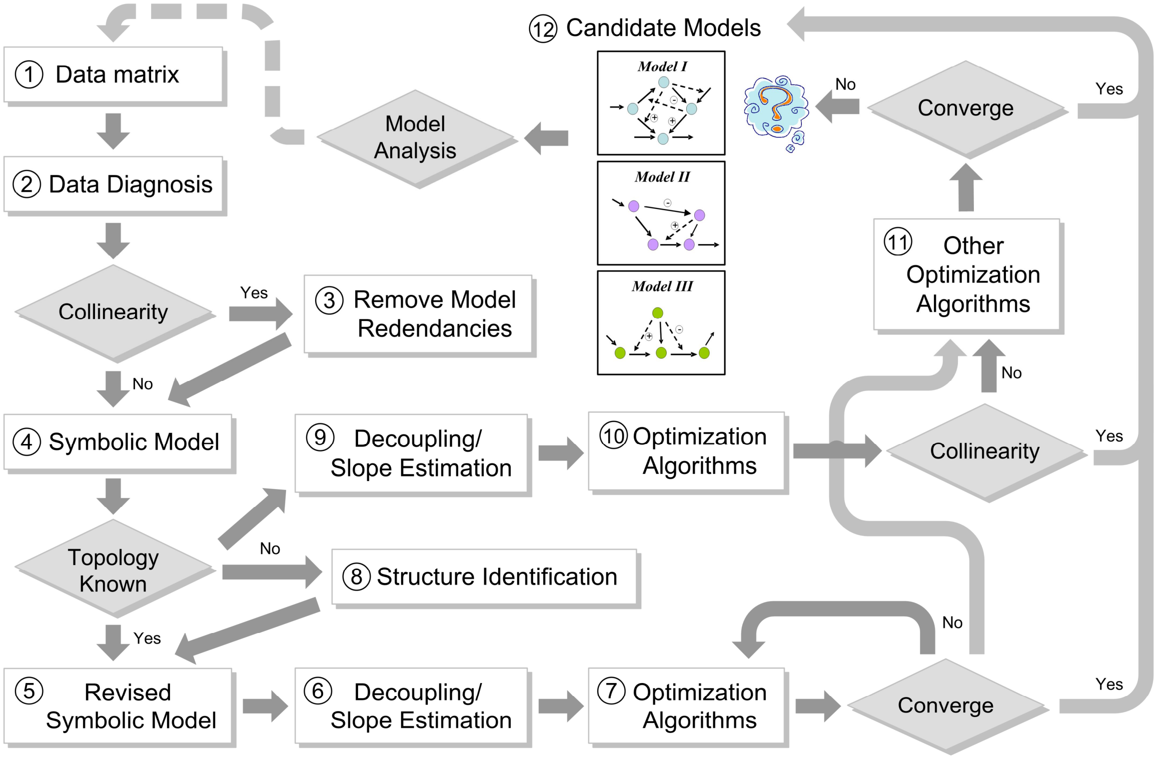


Figure 5.2. Flow diagram of inverse modeling strategy.
See text for detail description.

Once the candidate solutions are obtained, the question becomes if there are any guidelines that one can use for judging between candidate solutions. There are several scenarios one can anticipate regarding the candidate models. If the resulting solutions, either found by different optimization methods or obtained using the re-sampling scheme, are clustered together in the parameter space, it means the solutions are similar and the networks they interpret are essentially the same or very close. In contrast, if there are several distinctly different solutions with essentially the same residual error, it is difficult to decide which one is the best model. One of our recent results showed that a single data set allowed multiple distinctly different numerical solutions, especially if constraints on kinetic orders were set loosely. This was not unexpected because even one-variable S-systems are flexible enough to permit different parameter sets generating very similar graphs (*e.g*. Chapter 3 Figure 3.5). Without additional information, each such solution is a valid solution since it fits the data essentially equally well. By requiring many data sets and experimentally testing the same pathway under different conditions, the problem can often be alleviated. Using several data sets clearly constrains the flexibility of the underlying model considerably. However, one has to ask how often such complete data are available.

In spite of the many options outlined before, it is still possible that even a combination strategy cannot find an acceptable fit. Problem areas in this context and suggested future work will be discussed in Chapter 6.

CHAPTER 6

CONLUSIONS AND FUTURE WORK[viii]

6.1 Summary and Conclusions

Cells function and survive by orchestrating the expression of genes and their downstream products at the organizational levels of genes, proteins and metabolites. Metabolites, the end products of gene expression, are ultimately the causative agents for physiological responses and responsible for much of the functionality of the organism. Therefore, a comprehensive understanding of how metabolism works provides much insight into how cells and organisms operate.

Metabolic pathways consist of series of biochemical reactions that enzymatically convert metabolites into other metabolites. Several pathways collectively comprise a metabolic network. Typically the pathways are not only complicated themselves, but they are also highly interrelated since some of their metabolites are coupled with each other through reactions and regulatory interactions. The metabolites can either directly regulate other components in their own or in other pathways at the metabolic level, or affect the expression of genes or modification of proteins per signaling, which further increases the complexity of their roles. Hence, it is seldom possible to analyze or predict the behavior and dynamics of metabolism intuitively, and it is instead necessary to involve mathematical modeling as a means for assessing the functioning and regulation of metabolic networks.

The typical approach to mathematical model construction of metabolic pathways consists of five phases, namely: (1) collection of information and development of hypotheses associated with network structure and regulation; (2) selection of a suitable

[viii] Some of the material are adapted from: Goel, G., Chou, I-C., Voit, E. O. (submitted) System estimation from metabolic time series data.

mathematical modeling framework (Chapter 1 (Section 1.2)); (3) estimation of parameter values (Chapter 1 (Sections 1.3 and 1.4)); (4) model diagnostics; and (5) model application. Among these phases, the most challenging task continues to be the estimation of parameter.

After the symbolic model is constructed, based on the network structure diagram and the choice of a model format, the numerical kinetic model is obtained by estimating the values of all parameters. Traditionally, the parameter estimation strategies have been following a "forward" or "bottom-up" approach, which uses "local" descriptions of each step within the metabolic pathway and merges these into one comprehensive model. Another established approach uses steady-state data based on experiments that measure the responses of several metabolites after a small perturbation around the normal steady state. However, these approaches often do not yield an integrated model that is consistent with biological observations, because either input information is missing or uncertain, or the individually modeled pieces do not lead to a functioning model of the entire system (see Chapter 1 (Sections 1.3.1 and 1.3.2) for details).

Recent advancements in modern biological high-throughput techniques enable us to tackle the parameter estimation task using a distinctly different option, namely the "top-down" or "inverse" approach. These tools are able to generate time series data from the same organism, under the same experimental condition, and sometimes even *in vivo*. Therefore, in contrast to the "local" data obtained from traditional experiments, the clear advantage of using "global" data is that the collected information is more likely to represent the "true" behavior of the system in a comprehensive manner. However, this information about the structure and regulation of the biological system described by these data is mostly implicit, and there are several challenging issues of extracting it from the time series data. These challenges of inverse modeling are both on the biological and the computational sides. They can be generally categorized in four problem areas (also see Chapter 1 (Section 1.3.4)):

1. *Data related issues*: Typical biological datasets usually contain noise and measurement errors, and are seldom complete. Usual scenarios of missing data points include that the data are sparsely missing, that data collection is lacking at certain time points, that entire time series are missing, or that the existence of relevant metabolites was not known and that the corresponding time series are therefore missing. Sometimes the particular experimental conditions at the time of observation are uncertain which further complicates the situation. Other potential problems in the dataset are that the data matrix is ill-conditioned, which may be caused by collinearity among time series data, or that the time profiles are essentially constant or otherwise non-informative.
2. *Model related issues*: All mathematical models are more or less crude abstractions of reality rather than based on deep theory regarding the underlying mechanisms, as it is the case in physics (see Chapter 1 (Section 1.3.4) for reasons). There are some criteria for choosing a modeling framework, such as the ability to capture the dynamics of the time profile, mathematical simplicity and tractability, and interpretability of results within the biological realm. However, many mathematical formulations could be potential candidates for the optimal data representation. Some of the modeling frameworks and their pros and cons have been discussed in Chapter (Section 1.2). The selection of the model is supported by the criteria described above and, to some degree, personal preference.
3. *Computational issues*: The computational issues associated with parameter estimation are very challenging and have been the focus of many recently published papers regarding inverse modeling. The describing biological models potentially contain many components, and the systems are usually nonlinear and formulated as a set of differential equations. Therefore, the typical computational problems include computational efficiency, slow algorithmic progress toward the error minimum, lacking convergence or convergence to local minima, and

substantial time requirements for integration of the differential equations. Other challenges are reviewed in Chapter 1 (Section 1.3.4).

4. *Mathematical issues*: A further source of problems comes from issues of mathematical redundancy in the models. These redundancies include that different sets of parameter values, which fit the experiment data exactly equally well, are mathematically or numerically equivalent (Voit, 1992a) or that non-equivalent solutions exhibit similar residual errors. These mathematical redundancies may occur within or between the flux descriptions. The former is due to numerical compensation, for instance, between a rate constant and the kinetic orders within a single flux of a power-law model, while the latter is a consequence of compensation between the production fluxes and degradation fluxes.

To address the challenges outlined above, many algorithms and mathematical tools have been developed in recent years. The main tasks of these algorithms include: development and selection of suitable mathematical models for metabolic networks; development of strategies for the pre-handling and diagnosis of input time series data; development of optimization algorithms for extracting information from biological data sets; and creation of diagnostic tools to avoid mathematical compensation. The current achievements have been briefly reviewed in Chapter 1 (Section 1.3.5). They include the selection of S-system and GMA models within the BST framework as a promising representation for biological systems modeling; employment of a decoupling and smoothing strategy to alleviate the problem of missing data points or time series; and most intensively, the development of computational solutions to deal with the parameter estimation problem itself. These computational solutions typically require a combination of techniques that include methods to attack the main challenge of parameter value optimization, as well as other supporting algorithms.

The main optimization methods can generally be grouped into: gradient-based methods, stochastic search algorithms, and other techniques that do not belong to the first two groups. The essential part of solving the parameter optimization problem is to decide on an objective function and to minimize its error. Most of the objective functions are coupled with pruning strategies to omit unlikely parameters, especially when the topology of the system is unknown or only partially known. Many articles have been published recently regarding the computational methods for the inverse problem using BST, and the details are reviewed in Chapter 1 (Section 1.4.5). In addition to the main methods, supporting algorithms include methods for circumventing the time consuming integration of differential equations (Chapter 1 (Section 1.4.1)), smoothing overly noisy data and estimating slopes of time series (Chapter 1 (Section 1.4.2)), reducing the complexity of the inference task (Chapter 1 (Section 1.4.3)), and reducing the parameter search space (Chapter 1 (Section 1.4.4)).

As described several times throughout this dissertation, one should keep in mind that there is no clear boundary between parameter estimation and structure identification, although generally the latter task is much more difficult than the former task. Structure identification becomes a problem of parameter estimation if the parameter values can easily be translated into a specific biological role within the topology of the system. Conversely, a good structure prediction reduces the complexity of parameter estimation. Some of the most relevant structure identification methods are introduced in Chapter 1 (Section 1.5), namely methods based on the Jacobian matrix, direct observations, correlation-based approaches, simple-to-general and general-to-specific modeling, and time series data analysis using the framework BST.

In spite of the considerable amount of methods that have been proposed regarding the inverse modeling problem in the past ten years, every method has its pros and cons, and so far none of them has risen to the top as the perfect solution that can be declared as the clear general winner in terms of efficiency, robustness and reliability, for the majority of realistic cases. Even in terms of the published examples and results it is difficult to judge which algorithms are superior to the others and under what conditions (Chapter 5

(Section 5.1)). The MADMan (Munich, Atlanta, DiliMAN (Philippines)) project recently proposed by del Rosario and co-workers aims to compare these algorithms using BST formalisms in a systematic fashion. The ultimate goal is to develop a rational strategy for selecting candidate algorithms with the highest probability of success, given specific biochemical networks and experimental data. MADMan is still in its infancy and will demand concerted effort from the different groups involved.

In this dissertation I proposed two novel algorithms for improved inverse modeling within BST, namely alternating regression (AR) and eigenvector optimization (EO) methods. The AR method (Chapter 2) is specific to S-systems within BST and, combined with methods for decoupling systems of differential equations, provides a fast new tool for identifying parameter values from time series data that is genuinely different from all existing methods. The key feature of AR is that it dissects the complex nonlinear parameter estimation task into iterative steps of linear regression by utilizing the fact that power-law functions are linear in logarithmic space. I showed with several artificial examples that the method works well in many applications. In cases where convergence is an issue, it is feasible to dedicate some computational effort to identifying suitable start values and search settings, because the method is fast in comparison to conventional methods so that the search with different initial values is easily recouped. Specifically, I showed with an example from the literature that AR is three to five orders of magnitudes faster than direct structure identification methods for systems of nonlinear differential equations. The AR method is beneficial for the identification of system structure in S-system modeling as well. The convergence patterns of AR are complex and will require further investigation.

As an extension of using the AR method for parameter estimation in S-systems, I applied the AR algorithm to statistical S-distribution families which are motivated by growth functions represented as S-systems. Although S-distributions are not directly related to my main topic in metabolic pathway modeling, they shed additional light on some issues of convergence because they preserve some of the properties of general S-system models, and it turned out that their parameter values can be estimated efficiently with a modified AR algorithm. Specifically, I proposed a novel *3-way Alternating Regression* (3-AR) method (Chapter 3) as an effective strategy for the estimation of parameter values in S-distributions from frequency data. The 3-AR algorithm is very fast and performs well for artificial, error-free and noisy datasets, as well as for random samples generated from traditional statistical distributions and for observed raw data. In rare cases where the algorithm does not immediately converge, its enormous speed renders it feasible to select several initial guesses and search settings as an effective countermeasure.

Another method our group proposed is called *eigenvector optimization* (EO) (Chapter 4), which is inspired by AR and based on a matrix formed from multiple regression equations of the decoupled S-systems. In contrast to AR, EO operates initially only on one term (production or degradation), whose constant rate and kinetic orders are optimized completely by sequential quadratic programming (SQP) optimization, before the complementary term is estimated. The method is called eigenvector optimization because the objective function is based on the reformulation of simple multiple linear regression to a problem of finding the eigenvector with eigenvalue 1 of the estimation matrix. We demonstrated with several synthetic time series that the algorithm can be expected to converge in most cases. Furthermore, the EO algorithm is easily extended to the optimization of network topologies with stoichiometric precursor-product constraints among equations. These constraints rejoin the system in cases where it had been fragmented by decoupling. EO addresses specifically the S-system parameter identification problem when no information about the network topology is known. However, the algorithm tends to have problems when the data matrices are ill-posed.

A detailed comparison of AR and EO is presented in Chapter 5 (Section 5.2). Summarizing this comparison, both AR and EO are designed for S-system models and incorporated with decoupling techniques to avoid the integration of differential equations. Because of decoupling, good smoother and slope estimating techniques are needed. Both AR and EO converge fast compared to other traditional optimization methods. The AR algorithm works best when the topology of the system is known and when the components are sparsely connected. It also works in some cases where the topology is unknown. However, in this case, the EO method typically works better. The results of both, AR and EO, need to be evaluated carefully since the resulting model might fit the data but not have much biological meaning because of compensation between terms. Since both algorithms are computed with the data matrix, they are negatively affected by ill-posed problems and small numerical values in the data set.

Like many other published algorithms, AR and EO use a combination of several methods that include the core algorithm and other supporting techniques to solve the inverse problem. Each of these algorithms has its pros and cons, and there are conditions and situations where one works well and the other not so. The development of a "super" algorithm, which solves all inverse problems, has so far not succeeded, and it might be that a single algorithm, which is ideal with respect to correctness, robustness, and efficiency, does not even exist for all purposes (Chapter 5 (Section 5.3)).

Hence, a more feasible strategy might be to understand in more depth the characteristics of the best existing algorithms and to propose a decision tree or operational "work-flow" that takes the specific problems of a metabolic system and the given data into account and suggests the best solution for the given situation. While the MADMan project is attempting to characterize the specific properties of all published algorithms, the work-flow I proposed in Chapter 5 (Section 5.3) is a rather general approach that is independent of specific algorithms. The goal of this work-flow is to efficiently find a set of coarse candidate models that are sufficiently consistent with the data, instead of targeting one "optimal" solution. Each coarse model can be tested using diagnostic tools and various simulations to show whether it has a chance to be correct and has predictive power. The alternative candidate models can then be validated experimentally and used for guiding further experimental designs.

6.2 Future work

As mentioned in Chapter 1 (Section 1.3.4) and in the previous section, the challenges of inverse modeling can be classified into data related issues, model related issues, computational issues, and mathematical issues. Many recently published articles have acknowledged and discussed various computational issues in great detail and some have addressed data and model related issues. However, there has been little discussion of model validity and quality beyond residual errors, the conditions under which the models can be obtained, and diagnostic tools for non-convergence or for situations where models cannot even be obtained with any degree of reliability.

These situations can be generally addressed in two ways. First, when the algorithms are able to find a set of candidate models, it is possible that none of these models is valid and that diagnoses and simulation results show that none of the models has predictive ability. Other problems are lack of model fit for data not used in the estimation and model failure in extrapolations. Second, when the algorithms are not even able to produce acceptable fits, the failure is usually imputed to the computational algorithms themselves. However, attention should be devoted to investigating other possible sources of problems that result from the data and/or the system under investigation. Even though the outcomes look different, their causes are not exclusively independent and both are the consequences of different sources of problems. In Chapter 5 (Section 5.3) I proposed a work-flow strategy that suggests that the input data matrix should be diagnosed and handled before the main parameter estimation steps. However, there are still other issues that should be addressed to improve the validity of the estimated model.

To address these issues further, the four challenges associated with inverse modeling should be examined again in the following categories:

1. *Data related issues*: Even though good smoothing techniques can solve part of the problems of missing data points or time series, effective diagnostic tools of checking the consistency within data are still needed. One special property in modeling metabolic networks is that the mass of metabolites is conserved during the reaction. Therefore, by accounting for material flows entering and leaving each metabolite pool, one may be able to identify flows which might have been unknown or difficult to measure in the experiment. Furthermore, methods for assessing whether residual errors are due to idiosyncrasies or noise in the data are needed.
2. *Model related issues*: Traditionally, when a mathematical framework is chosen for modeling, the fluxes in the metabolic pathway are represented using the same basis functions, for instance, a Michaelis-Menten or power-law representation. However, it is possible that not all fluxes are appropriately modeled by the same format; an example is the glucose uptake step in *Lactococcus*, which we discussed in our recent work (Goel *et al.*, submitted). Furthermore, most of the mathematical formalisms are local approximations around an operating point. If the metabolite concentrations do not fall within the valid range of approximation, the model can not properly represent the dynamics. This phenomenon typically becomes important when a single model is used for more than one set of time series, each of which represents different experimental conditions. However, good criteria for determining the appropriateness of the chosen mathematical representations are still lacking.
3. *Computational issues*: Current model fitting is based on time series of the main components in the biological system, such as the concentrations of metabolites in the pathway. However, rates of material flows are usually unavailable. If

available, they typically refer to input and output fluxes but not to the intermediate fluxes. Therefore, if one could determine all fluxes, together with the time series of each variable in the system, the estimation of parameter values would become more reliable.

4. *Mathematical issues*: As mentioned several times, mathematical redundancies in the model may occur within or between fluxes and equations. The compensation between fluxes can be avoided if each of the true fluxes is obtained as described in the previous paragraph. However, solutions for numerical compensation within a single flux are still needed in order to generate reliable extrapolations. The removal of compensation within flux seems to require data covering relatively wide ranges of variation, multiple datasets or additional information about some of the parameter values.

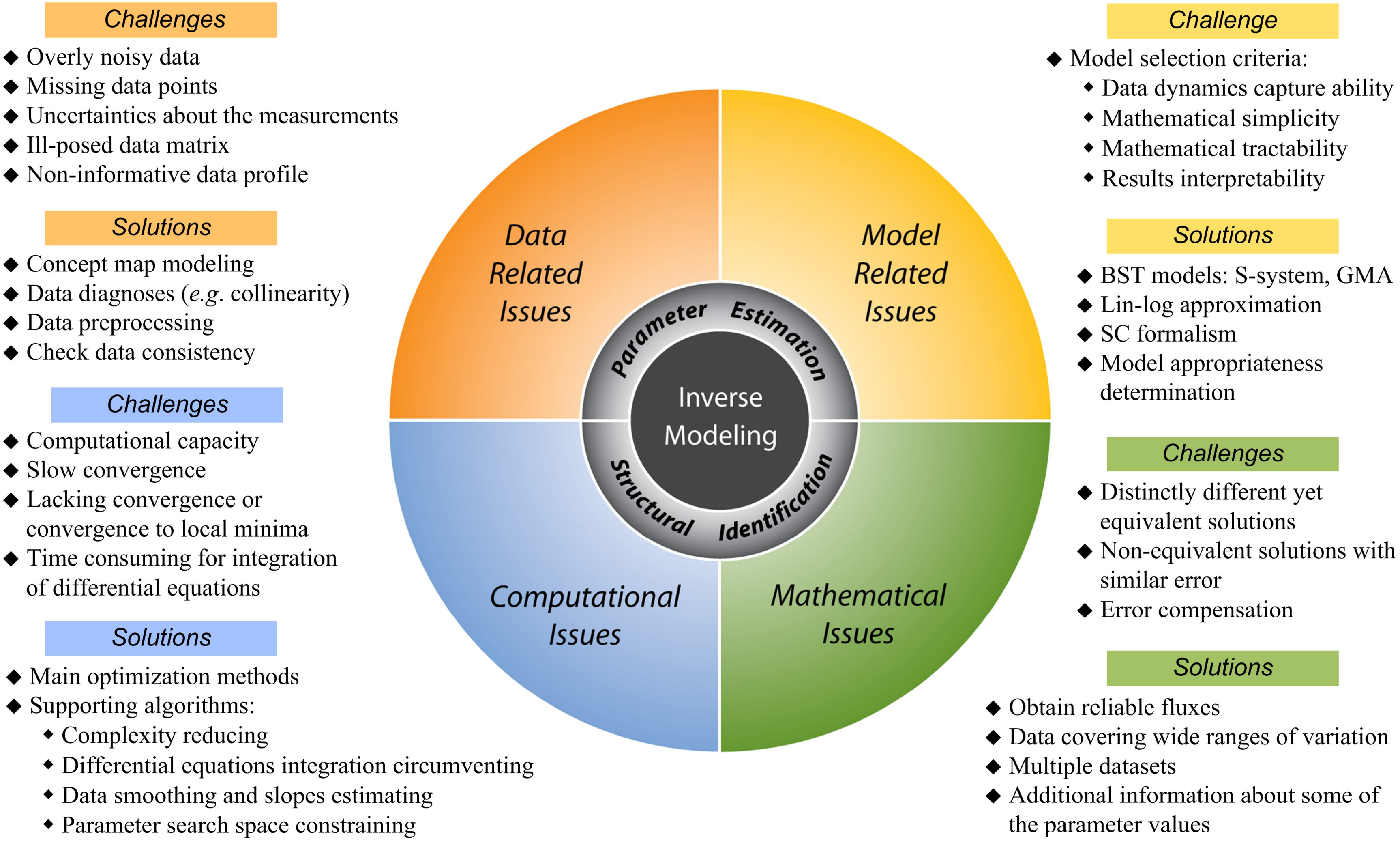
Figure 6.1 summarizes the typical challenges and their corresponding tasks based on the problem areas, including those mentioned in the previous chapters and in this section.

Our group recently proposed a novel approach to metabolic systems estimation, called *Dynamic Flux Estimation* (DFE), which resolves several of the issues mentioned above (Goel *et al.*, submitted). This approach consists of two distinct phases. The first phase consists of an entirely model-free and essentially assumption-free data analysis and quickly reveals inconsistencies within the time series, and between data and the alleged system topology. The consistency check within the data leads to numerical representations of fluxes as functions of the variables affecting them. The second, model-based phase addresses the mathematical formulation of the processes in the biological system. Different from currently available methods, this phase allows quantitative diagnostics of whether—or to what degree—the assumed mathematical formulations are appropriate or in need of improvement. The two-phased approach thus permits rigorous, quantitative diagnoses of the data, the model structure, the assumptions made in the choice of flux representations, and the causes of residual errors.

Our preliminary results suggest that the proposed approach is more effective and robust than alternatives that are presently available. Its combined model-free and model-based analyses reduce compensation of error between equations and between flux terms and promise significantly improved extrapolability toward new data or experimental conditions. Its diagnostic tools pinpoint causes of inadequate fits between model and data and suggest either changes in assumptions related to model choice or the use of data as un-modeled "off-line data."

The main drawback of DFE is the requirement of rather comprehensive time series data, which however can be obtained in many cases with already existing experimental methods. Also, while DFE significantly reduces error compensation between equations and between fluxes, it still admits error compensation among the parameters within a given flux, independent of what representation is chosen. Issues needing further development are related to missing data, missing flux information, underdetermined stoichiometric matrices, and ill-characterized systems topologies.

Finally, one should emphasize the need for obtaining reliable solutions within short periods of time. In some cases, only a single estimation of the system may be needed, and it may be acceptable if this estimation takes a few hours. However, once the field moves to "estimation on the fly," solutions must be obtained within a few minutes or, preferably, within seconds. The need for fast solutions becomes especially pertinent if biologists and modelers together engage in concept map modeling Chapter 1 (Section 1.3.5), which permits the conversion of hypothesized network diagrams into numerical mathematical models. Because this method is based on the biologist's intuition and hypotheses, many iterations between hypothesis formulation and diagram-to-model conversion are needed, thus demanding fast solutions that might not be absolutely precise but allow the interactive exploration of complex biological systems.



Challenges

- Overly noisy data
- Missing data points
- Uncertainties about the measurements
- Ill-posed data matrix
- Non-informative data profile

Solutions

- Concept map modeling
- Data diagnoses (*e.g.* collinearity)
- Data preprocessing
- Check data consistency

Challenges

- Computational capacity
- Slow convergence
- Lacking convergence or convergence to local minima
- Time consuming for integration of differential equations

Solutions

- Main optimization methods
- Supporting algorithms:
  - Complexity reducing
  - Differential equations integration circumventing
  - Data smoothing and slopes estimating
  - Parameter search space constraining

Challenge

- Model selection criteria:
  - Data dynamics capture ability
  - Mathematical simplicity
  - Mathematical tractability
  - Results interpretability

Solutions

- BST models: S-system, GMA
- Lin-log approximation
- SC formalism
- Model appropriateness determination

Challenges

- Distinctly different yet equivalent solutions
- Non-equivalent solutions with similar error
- Error compensation

Solutions

- Obtain reliable fluxes
- Data covering wide ranges of variation
- Multiple datasets
- Additional information about some of the parameter values

Figure 6.1. Challenging areas and corresponding solutions of inverse modeling strategy.

APPENDIX A

ADDITIONAL DOCUMENTATION OF PARAMETER ESTIMATION USING ALTERNATING REGRESSION IN S-SYSTEMS

A.1 Further documentation of patterns of convergence

Accuracy and speed of solution

The following tables (Tables A.1 and A.2) correspond to Table 2.2 in Chapter 2, but use different error thresholds. In all cases, convergence depends on a number of factors, such as the selection of data sets. If one chooses data sets 1, 2, and 5, for example, then AR converges quickly to the right solution for metabolite X_4. Slightly modifying constraints after each phase of AR is another strategy to improve the likelihood of convergence. For example, in the case of metabolite X_2, one could relax the true constraints from [g_{21} 0 0 0] [0 h_{22} 0 0] to more generalized combinations like [g_{21} 0 0 g_{24}] [0 h_{22} h_{23} 0], [g_{21} 0 g_{23} g_{24}] [0 h_{22} 0 0], or [g_{21} g_{22} 0 g_{24}] [0 h_{22} 0 0], where 0 indicates exclusion of the corresponding variable. In all these cases AR converges quickly to the right solution. In other words, even if one doesn't constrain some parameters to zero that should truly be 0, AR automatically forces them to approach zero. It appears that the relaxing of constraints gives the AR algorithm more space to find the optimal solution. Using different combinations of regressors can also help. Again, in the case of metabolite X_4, if one uses X_1 and X_3 to fit the model in the first phase of AR and then use metabolites X_1, X_3, and X_4 to fit the model in the second phase of AR, the algorithm successfully converges to the correct solution. This trial and error approach may appear somewhat *ad hoc*, but exploring several combinations in troublesome cases is still considerably faster than any competing algorithm that I am aware of.

Table A.1. Estimated parameter values of the S-system model of the pathway in Figure 2.2 using *log*(*SSE*)<-20 as termination criterion.
[a] Regressor: A: all variables used as regressors and subsequently constrained; B: use of "union" variables as regressors (see Chapter); C: fully informed selection of regressors (see Chapter 2). [b] time (secs) needed to converge to the solution with *log*(*SSE*)<-20. [c] *: convergence to the true solution; **: convergence to different solution; ***: no convergence. [d] time after running 1,000,000 iterations.

	Regressor[a]	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	*log*(*SSE*)	Time[b]	Note[c]
X_1	A	12.00	0.00	0.00	-0.80	-0.00	10.00	0.50	-0.00	0.00	0.00	-19.18	0.97	*
	B	12.00	-0.00	0	-0.80	0	10.00	0.50	0	0.00	0	-20.00	5.48	*
	C	12.00	0	0	-0.80	0	10.00	0.50	0	0	0	-19.94	270.97	*
X_2	A	44.50	-0.00	-0.02	-0.04	0.11	31.48	0.04	0.14	0.05	-0.13	0.51	1062.83[d]	**
	B	8.00	0.50	0.00	0	0	3.00	-0.00	0.75	0	0	-20.01	1.95	*
	C	8.00	0.50	0	0	0	3.00	0	0.75	0	0	-20.00	103.39	*
X_3	A	3.00	0.00	0.75	-0.00	-0.00	5.00	-0.00	0.00	0.50	0.20	-19.79	0.05	*
	B	7.29	0	0.37	-0.00	-0.00	8.76	0	-0.00	0.19	0.04	-4.04	1111.63[d]	**
	C	3.00	0	0.75	0	0	5.00	0	0	0.50	0.20	-20.00	589.97	*
X_4	A	96.80	0.01	0.01	-0.00	0.00	100.00	-0.00	-0.01	0.00	0.02	-3.83	3.50	***
	B	98.29	0.01	0	0	0.00	100	-0.00	0	0	0.01	-5.85	340.34	***
	C	2.00	0.50	0	0	0	6.00	0	0	0	0.80	-19.97	289.09	*

Table A.2. Estimated parameter values of the S-system model of the pathway in Figure 2.2 using *log*(*SSE*)<-4 as termination criterion.
[a] Regressor: A: all variables used as regressors and subsequently constrained; B: use of "union" variables as regressors (see Chapter 2); C: fully informed selection of regressors (see Chapter 2). [b] time (secs) needed to converge to the solution with *log*(*SSE*)<-4. [c] *: convergence to the true solution; **: convergence to different solution; ***: no convergence. [d] time after running 1,000,000 iterations. [e] false positive.

	Regressor[a]	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	*log(SSE)*	Time[b]	Note[c]
X_1	A	12.04	0.00	0.00	-0.79	-0.00	10.07	0.50	-0.00	0.00	0.00	-3.84	0.44	*
	B	13.81	-0.00	0	-0.60	0	12.16	0.38	0	0.00	0	-4.00	0.94	*
	C	12.29	0	0	-0.83	0	10.08	0.51	0	0	0	-3.92	0.06	*
X_2	A	44.50	-0.00	-0.02	-0.04	0.11	31.48	0.03	0.14	0.05	-0.13	0.51	1073.05[d]	**
	B	8.47	0.46	0.00	0	0	3.42	-0.00	0.69	0	0	-4.00	0.58	*
	C	8.46	0.46	0	0	0	3.42	0	0.69	0	0	-4.00	59.91	*
X_3	A	3.00	0.00	0.75	-0.00	-0.00	5.00	-0.00	0.00	0.50	0.20	-9.44	0.06	*
	B	3.81	0	0.63	-0.00	-0.00	5.60	0	-0.00	0.39	0.13	-4.65	0.03	*
	C	2.80	0	0.83	0	0	5.56	0	0	0.63	0.31	-4.01	0.20	*
X_4	A	96.80	0.01	0.01	-0.00	0.00	100.00	-0.00	-0.00	0.00	0.02	-3.83	4.52	***
	B	10.08	0.06	0	0	0.00	11.98	-0.00	0	0	0.12	-3.98	1.72	*[e]
	C	2.24	0.40	0	0	0	5.60	0	0	0	0.66	-3.97	29.42	*

Density of sampling points

Instead of using time series with 50 sampling points, I applied AR to data sets with only 10 points, consisting of the same starting and end times, but larger time intervals. The results (Table A.3) demonstrate that the density of time points in this case does not affect the efficacy of AR if the data are noise free. In addition to increasing the intervals between data points, I also reduced the time series from 50 observations to the first 25 points. The results (Table A.4) show that AR still converges in most cases to the true solution.

Noisy data and data from non-S-system models

As is typical with demonstrations of new algorithms in this field, it is beneficial at first to concentrate on error-free data in order to investigate how well the algorithm works under ideal conditions. In cases of noise-corrupted (artificial or real) data, one typically smoothes the data with methods like the three-point method, some smoother like the Whitaker filter, or an artificial neural network (see discussion in (Voit and Almeida, 2004)). If the raw data are smoothed before application of the proposed (or other) algorithm(s), the question of the effects of noise in truth become questions of the power, reliability, and efficiency of the chosen smoother. Similarly, if the data represent a model that is not optimally modeled with an S-system, the issue is not so much the proposed search algorithm as the quality of the S-system representation. I will analyze these issues elsewhere in greater detail.

Table A.3. Estimated parameter values of the S-system model of the pathway in Figure 2.2 using *log*(*SSE*)<-7 as termination criterion with 10 sampling points.

[a] Regressor: A: all variables used as regressors and subsequently constrained; B: use of "union" variables as regressors (see Chapter 2); C: fully informed selection of regressors (see Chapter 2). [b] time (secs) needed to converge to the solution with *log*(*SSE*)<-7. [c] *: convergence to the true solution; **: convergence to different solution; ***: no convergence. [d] time after running 1,000,000 iterations.

	Regressor[a]	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	*log(SSE)*	Time[b]	Note[c]
	A	11.99	0.00	0.00	-0.80	-0.00	9.99	0.50	-0.00	0.00	0.00	-4.66	0.45	*
X_1	B	12.07	0.00	0	-0.79	0	10.10	0.49	0	0.00	0	-7.00	1.80	*
	C	12.07	0	0	-0.79	0	10.10	0.49	0	0	0	-6.99	14.72	*
	A	50.56	-0.20	-0.06	-0.22	0.25	27.54	0.098	0.11	0.26	-0.30	0.72	544.03	**
X_2	B	8.02	0.50	-0.00	0	0	3.02	-0.00	5	0	0	-7.00	0.76	*
	C	8.02	0.50	0	0	0	3.02	0	0.75	0	0	-6.98	27.00	*
	A	3.00	-0.00	0.75	-0.00	0.00	5.00	-0.00	-0.00	0.50	0.20	-12.84	0.02	*
X_3	B	3.07	0	0.74	-0.00	-0.00	5.06	0	-0.00	0.49	0.19	-6.81	0.49	*
	C	3.04	0	0.75	0	0	5.08	0	0	0.50	0.20	-7.00	0.20	*
	A	96.11	0.02	0.00	0.00	0.00	100.00	-0.00	-0.00	-0.00	0.03	-3.41	2.86	***
X_4	B	98.28	0.01	0	0	0.00	100.00	-0.00	0	0	0.01	-6.40	87.09	***
	C	2.01	0.49	0	0	0	5.97	0	0	0	0.79	-6.98	34.00	*

Table A.4. Estimated parameter values of the S-system model of the pathway in Figure 2.2 using *log*(*SSE*) < -7 as termination criterion with the first 25 points.

[a] Regressor: A: all variables used as regressors and subsequently constrained; B: use of "union" variables as regressors (see Chapter 2); C: fully informed selection of regressors (see Chapter 2). [b] time (secs) needed to converge to the solution with *log*(*SSE*)<-7. [c] *: convergence to the true solution; **: convergence to different solution; ***: no convergence. [d] time after running 1,000,000 iterations.

	Regressor[a]	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	log(SSE)	Time[b]	Note[c]
	A	100.05	-0.00	-0.01	-0.05	0.02	96.72	0.05	0.01	-0.00	-0.02	-1.44	0.55	***
X_1	B	12.03	-0.00	0	-0.79	0	10.05	0.50	0	0.00	0	-6.93	3.73	*
	C	12.03	0	0	-0.79	0	10.05	0.50	0	0	0	-6.92	21.53	*
	A	73.18	4.92	-4.89	6.94	0.80	1.27	0.53	0.47	1.28	-0.37	1.96	0.03	***
X_2	B	8.0	0.50	0.00	0	0	3.01	-0.00	0.75	0	0	-7.01	0.72	*
	C	8.01	0.50	0	0	0	3.01	0	0.75	0	0	-7.00	36.49	*
	A	3.00	0.00	0.75	-0.00	-0.00	5.00	-0.00	0.00	0.50	0.20	-7.06	0.08	*
X_3	B	3.02	0	0.75	-0.00	-0.00	5.00	0	-0.00	0.49	0.120	-6.3	0.19	*
	C	3.03	0	0.74	0	0	5.00	0	0	0.49	0.19	-7.00	30.53	*
	A	97.33	0.01	0.00	-0.00	0.00	100.00	-0.00	-0.00	0.00	0.02	-3.87	6.38	***
X_4	B	98.49	0.00	0	0	0.00	100.00	-0.00	0	0	0.01	-6.19	196.98	***
	C	2.01	0.50	0	0	0	5.97	0	0	0	0.79	-6.94	71.56	*

A.2 Numerical characterization of AR's basin of attraction for different datasets

Any analytical characterization of the convergence of a nonlinear search algorithm for dynamical models is a very demanding task. Even for the Newton algorithm, which has been used and analyzed by generations of researchers in mathematics, computer science, and various application fields, convergence can be extremely complex and essentially impossible to predict. As an example, Epureanu and Greenside (Epureanu and Greenside, 1998), as well as numerous original papers and websites, review the basins of attraction for this algorithm, which even in really simple cases of algebraic functions can consist of very complicated fractals. The same is true for every other nonlinear search algorithm, including Levenberg-Marquardt, genetic algorithms, and simulated annealing, where it is close to impossible to predict with reliability whether a search will succeed in finding the true solution.

Given this complexity and the long history of the Newton algorithm and other search algorithms, it is not likely that one will be able to develop crisp and general theorems characterizing the convergence behavior of our new algorithm. Indeed, it seems not possible with present mathematical means to characterize the convergence features of our proposed algorithm in generality. As the next best alternative, I have therefore chosen to pursue the topic with a comprehensive computational analysis (comprising with over 1,000,000 alternative regressions) of two examples (Kikuchi *et al.*, 2003; Voit and Almeida, 2004), which have become something like unofficial case studies for comparisons of algorithms in the field. In addition to the discussions in the Chapter 2, I describe here the effects of using different datasets from the same system, which are characterized by different initial values of the dependent variables. It was recently shown (Schwacke and Voit, 2005) with "time-dependent sensitivities" how initial values affect the dynamics of trajectories. The analysis here illuminates a related issue, but from a different angle.

In order to demonstrate the effects of initial conditions on convergence, I investigated in great detail dataset 1 of system in Figure 2, with initial conditions $X_1(t_0)$ = Int1 = 1.4, $X_2(t_0)$ = Int2 = 2.7, $X_3(t_0)$ = Int3 = 1.2, and $X_4(t_0)$ = Int4 = 0.4. To allow for a two-dimensional representation, I fixed Int3 and Int4 and changed Int1 and Int2 (essentially exhaustively) in different combinations. Figures A.1, A.3 and A.5 represent the 2-D "dataset convergence maps" of using all variables as regressors, "union" variables as regressors, and variables that are known to appear in each term as regressors, respectively (as described in Chapter 2). Each map consists of about 160,000 alternating regression analyses, where each dot represents a dataset. The color of the dot codes for the number of iterations needed to converge to the right solution, starting with the same initial guesses of β_i and h_{ij} that I used as example in the paper. The color scales are the same in three figures.

The main result is that the "convergence maps" in Figures A.1, A.3 and A.5 are very complicated. They do not seem to be fractal as in the Newton case, but in some sense even more complicated by not revealing obvious patterns. Striped areas represent domains in the space where the logarithm of some slope minus one power-law term is not defined, as described in detail in the paper. In a nutshell, using a dataset from within these areas, and again starting with the initial guess of β_i and h_{ij}, the expressions in steps {5} and {9} of the algorithm become negative, thereby disallowing the necessary logarithmic transformation. Shaded areas represent no-convergence areas. When using datasets from within these areas, the value of α_i (or β_i) typically increases continuously and without bound while some or all g_{ij} (or h_{ij}) gradually approach zero; in some other cases g_{ij} and the corresponding h_{ij} increase (or decrease) in a parallel manner. These situations seem to indicate low information content of the dataset.

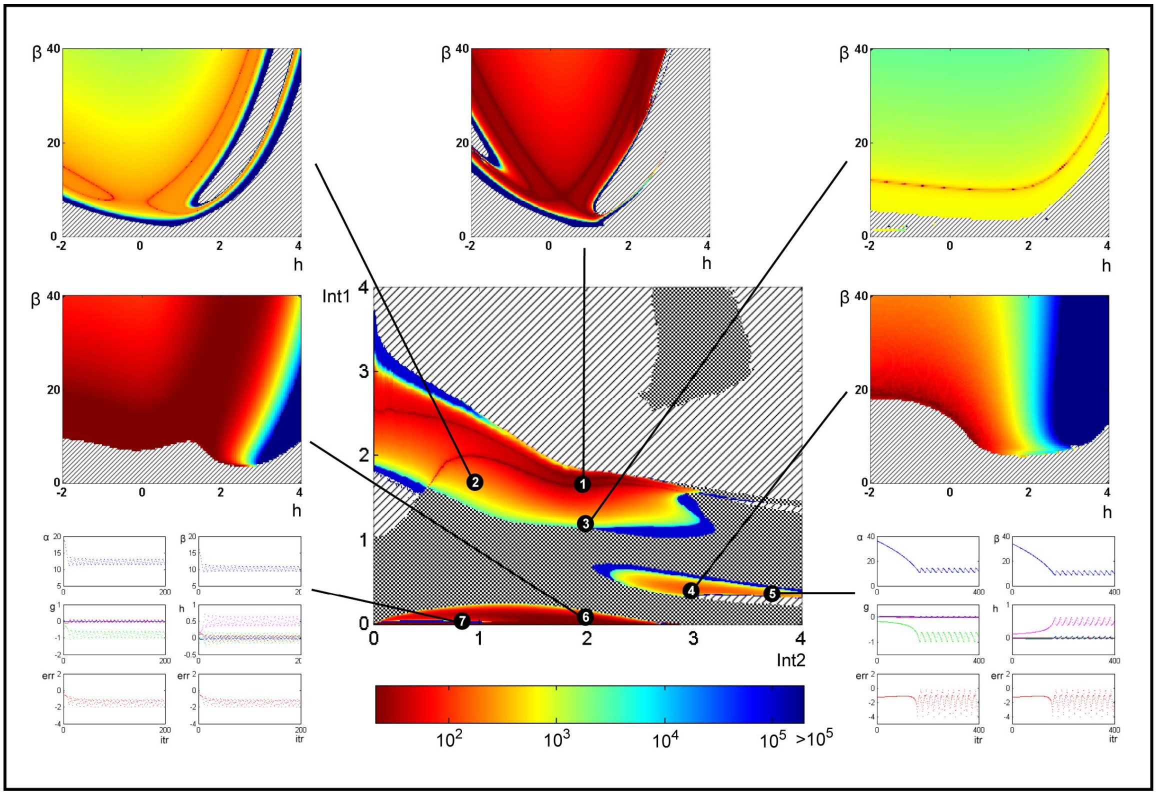


Figure A.1. Pattern of convergence: Use of all variables as regressors.
See Chapter 2 for general explanations.

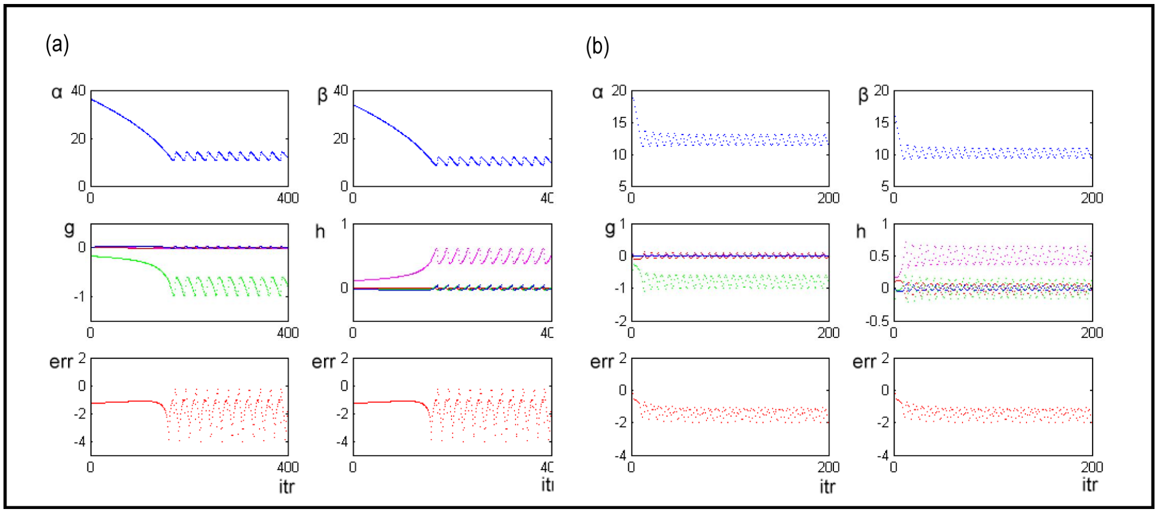


Figure A.2. Close views of Figure A.1.
(a) Close-up of Figure A.1 ❺; (b) Close-up of Figure A.1 ❼.

The satellite figures around the central plots in Figures A.1, A.3, and A.5 represent the convergence maps of particular datasets. These plots show the effects of changing the start guesses (β_i and h_{ij}) used in the alternating regression, given the particular dataset indicated by a number. Figure A.2 represents close-ups of figures identified as ❺ and ❼ in Figure A.1. In these cases, the parameter values are oscillating near the true solutions. Such two-cycle oscillations are not unusual in iterative searches. The changes in parameter values within the non-convergence (shaded) areas are similar in Figures A.1, A.3, and A.5. One representative example is shown in Figure A.3 ❼.

Figure A.1 has the largest "problem" areas. However, outside these areas convergence is very fast. Intriguingly, the problem areas are substantially reduced in size when one uses fewer variables as regressors (*i.e.*, if the degrees of freedom are decreased). For instance, Figure A.5 does not even have a "no-convergence" area. Interestingly, and not yet fully explained, the convergence speed in these cases is usually much slower than in Figures A.1 and A.3.

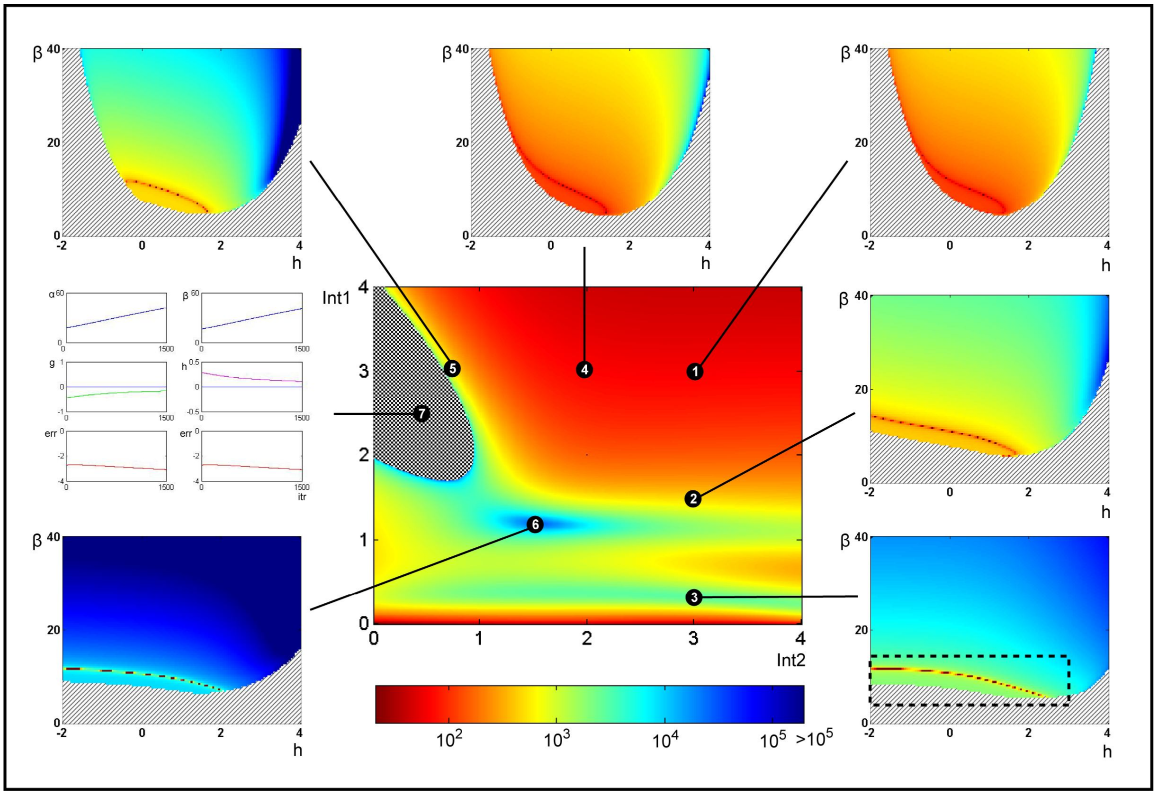


Figure A.3. Pattern of convergence: Use of the "union" of variables as regressors.
See Chapter 2 for general explanations.

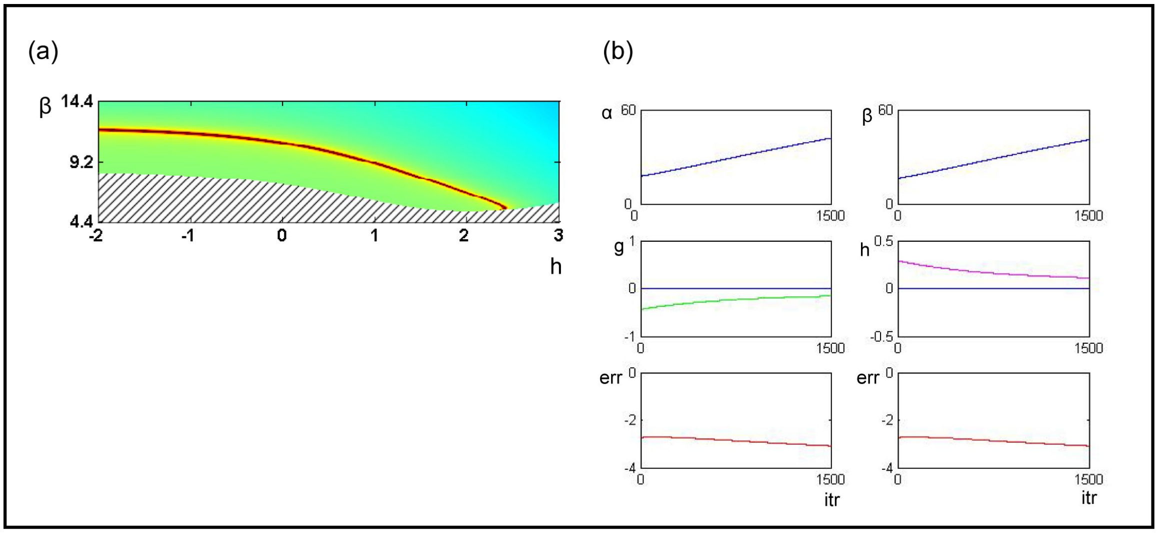


Figure A.4. Close views of Figure A.3.
(a) Close-up of Figure A.3 ❸; (b) Close-up of Figure A.3 ❼.

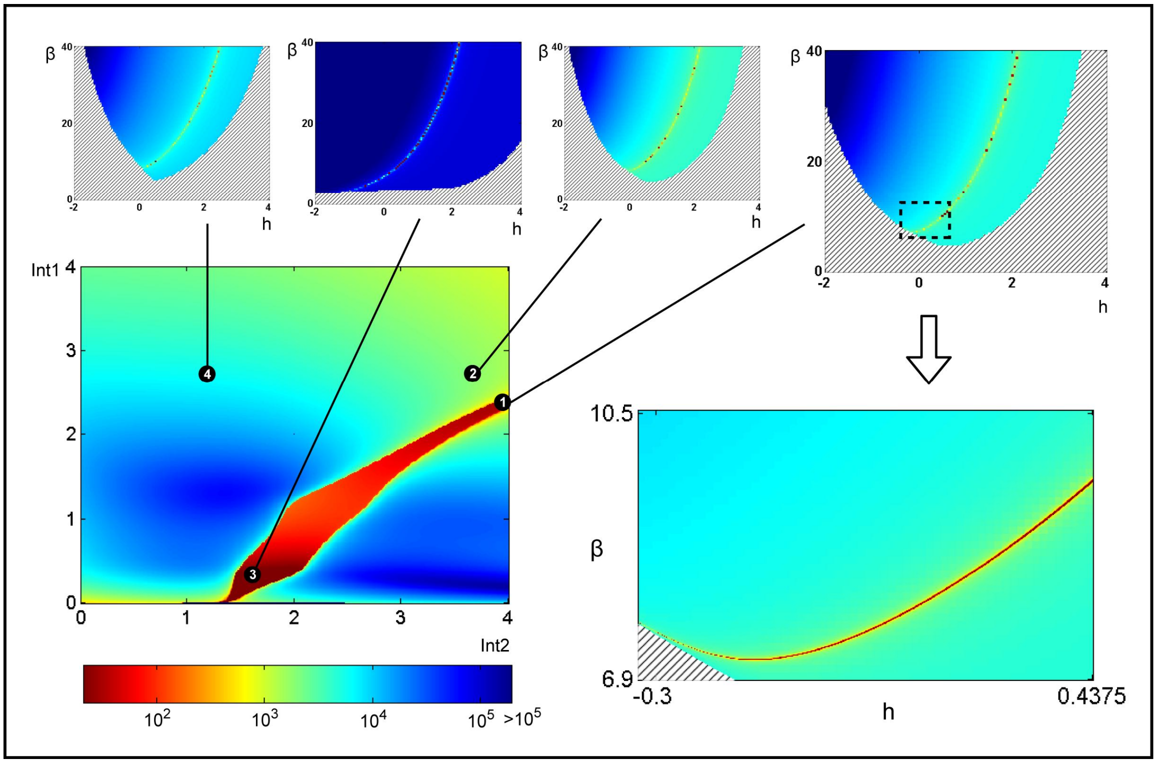


Figure A.5. Pattern of convergence: Use of variables that are known to appear in each term as regressors.
See Chapter 2 for general explanations.

The graphs in Figures A.1, A.3, and A.5 provide strong indication that it will be very difficult to determine the convergence areas analytically, especially when more variables then necessary are used as regressors. Even for some points very close to each other, their convergence properties could be entirely different (see Figure A.5 (a)): One point may lead to convergence to the right solution, while “neighbors” may not converge or lead to negative arguments in the logarithmic transformations in steps {5} or {9} in Chapter 2 (Section 2.2.3.1). In contrast to highly symmetric fractal pictures often associated with the Newton method, the basin of attraction here is complicated and does not suggest an intuitive pattern.

So far I only varied the values of Int1 and Int2, in order to facilitate a graphical representation. The question then becomes how changing Int1, Int2, Int3, and Int4 simultaneously would affect convergence. Because of the complexity of the situation one can only show select results of how the initial conditions affect the number of iterations needed when one uses as regressors those variables that are known to appear in each term.

Figure A.6 shows some results of partial least square regression (PLSR), elucidating this situation. In this case, Int1, Int2 and Int3 are statistically significant (striped areas) in predicting the number of iterations needed. Int1 and Int2 contribute negatively to convergence speed, while Int3 significantly increases the number of iterations needed; changing Int4 has no significant effect. These results have to be considered with caution, because they are highly dependent not only on the initial values, but also on the error threshold and other factors. A more comprehensive study, including all contributing factors will be needed.

Finally, I designed a multiple-level full-factorial experiment to identify which design variable (initial condition) influences response (number of iterations needed) significantly and which not. Table A.5 shows that seven effects are found to be significant, four of them are confounded interactions. Again, these results have to be considered with caution. A more comprehensive study, including all contributing factors will be needed.

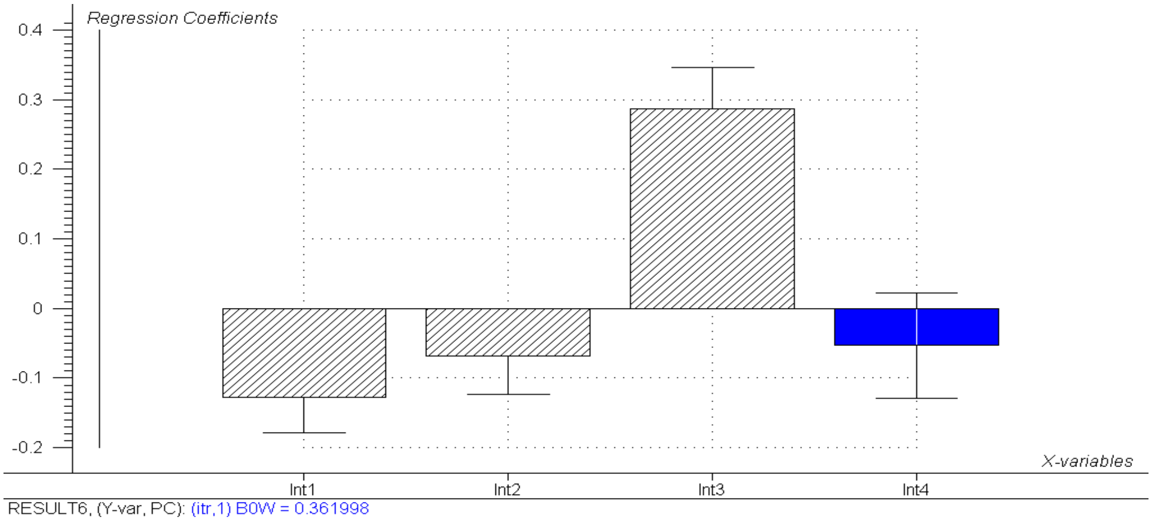


Figure A.6. Results of the partial least squares (PLS) analysis.
The result shows the influence of initial values of variables X_1-X_4 on convergence speed. *Unscrambler*® software was used.

Table A.5. Results of an ANOVA characterizing the influence of initial values of variables X_1-X_4 on convergence speed. *Unscrambler*® software was used.

	SS	DF	MS	F-ratio	p-value
Summary					
Model	4.820e+12	170	2.836e+10	6.444	0.0000
Error	4.950e+12	1125	4.400e+09		
Adjusted Total	9.771e+12	1295	7.545e+09		
Variable					
Int1	5.840e+10	5	1.168e+10	2.654	0.0215
Int2	3.924e+11	5	7.847e+10	17.834	0.0000
Int3	1.854e+12	5	3.708e+11	84.260	0.0000
Int4	3.248e+10	5	6.497e+09	1.476	0.1947
(Int1)(Int2)	6.184e+11	25	2.474e+10	5.622	0.0000
(Int1)(Int3)	2.746e+11	25	1.098e+10	2.496	0.0001
(Int1)(Int4)	1.396e+11	25	5.585e+09	1.269	0.1695
(Int2)(Int3)	1.072e+12	25	4.287e+10	9.742	0.0000
(Int2)(Int4)	1.436e+11	25	5.744e+09	1.305	0.1441
(Int3)(Int4)	2.355e+11	25	9.419e+09	2.140	0.0009

Summarizing all results of Chapter 2 and Appendix A, it is very difficult to determine precise conditions of convergence, especially if a system has a high degree of freedom.

APPENDIX B

ADDITIONAL DOCUMENTATION OF PARAMETER ESTIMATION USING EIGENVECTOR OPTIMIZATION IN S-SYSTEMS

B.1 Numeric experiments

In this appendix, we present some results obtained using the EO algorithm. Two sets of experiments were performed with the systems presented in the main manuscript, namely the 4-dimensional system (Eq. (4.36)) and the 5-dimensional system (Eq. (4.37)). In order to test the robustness of the algorithm, we also performed experiments in a 10-dimensional system

$$\begin{aligned}
\dot{X}_1 &= 1 - X_1 \\
\dot{X}_2 &= 1 - X_2 \\
\dot{X}_3 &= X_1^{0.5} X_2^{0.4} - X_3 \\
\dot{X}_4 &= X_3^{0.3} - X_4 \\
\dot{X}_5 &= X_3^{-0.2} - X_5 \\
\dot{X}_6 &= X_{10}^{0.2} - X_6 \\
\dot{X}_7 &= X_5^{0.5} - X_7 \\
\dot{X}_8 &= X_6^{0.7} X_{10}^{0.3} - X_8 \\
\dot{X}_9 &= X_7^{0.6} - X_9 \\
\dot{X}_{10} &= X_9^{-0.2} - X_{10}
\end{aligned} \quad . \qquad \text{(B.1)}$$

B.1.1 Initial conditions

To perform the experiments, different initial conditions for the system variables were chosen (Table B.1 and B.2) to generate time series by numerical integration. For each of these conditions, 10 runs were performed for each system's variables. In all result tables, the sums of squared errors in relation with the decoupled and numerically integrated system are presented as Error1 and Error2 respectively. All data sets were generated with the software PLAS (Ferreira, 2000; Voit, 2000a).

Table B.1. Initial conditions for integration of the 4-dimensional system

Dataset	$X_1(t_0)$	$X_2(t_0)$	$X_3(t_0)$	$X_4(t_0)$
1	1.0	1.0	1.0	1.0
2	1.0	3.0	1.3	1.3
3	1.5	0.5	0.5	1.5

Table B.2. Initial conditions for integration of the 5-dimensional system

Dataset	$X_1(t_0)$	$X_2(t_0)$	$X_3(t_0)$	$X_4(t_0)$	$X_5(t_0)$
1	0.10	0.70	0.70	0.16	0.18
2	0.70	0.12	0.14	0.16	0.18
3	0.70	0.70	0.14	0.16	0.70

B.1.2 Noise-free datasets

4-Dimensional system results – noise-free time series

Tables B.3-C.6 show the parameters found with the proposed algorithm for the 4-dimensional system (Eq. (4.36)) using the first set of initial values of the Table B.1. The time series used in this case study for all datasets were obtained by numerical integration of the 4-dimensional system in the interval [0,10] with 0.1 sampling interval.

Table B.3. Result of the 10 runs for the variable X_1 of the 4-dimensional system with beta initial guesses randomly distributed in the range [1, 12].

Run	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	Error1	Error2
1	12.00	0.00	0.00	-0.80	0.00	10.00	0.50	0.00	0.00	0.00	1.75483E-20	1.33873E-05
2	12.00	0.00	0.00	-0.80	0.00	10.00	0.50	0.00	0.00	0.00	1.71974E-19	1.21796E-05
3	12.00	0.00	0.00	-0.80	0.00	10.00	0.50	0.00	0.00	0.00	5.68827E-19	1.21338E-05
4	12.00	0.00	0.00	-0.80	0.00	10.00	0.50	0.00	0.00	0.00	1.71382E-19	1.31119E-05
5	12.00	0.00	0.00	-0.80	0.00	10.00	0.50	0.00	0.00	0.00	9.24261E-19	1.46669E-05
6	12.00	0.00	0.00	-0.80	0.00	10.00	0.50	0.00	0.00	0.00	4.38633E-19	1.22694E-05
7	12.00	0.00	0.00	-0.80	0.00	10.00	0.50	0.00	0.00	0.00	5.29743E-19	1.3364E-05
8	12.00	0.00	0.00	-0.80	0.00	10.00	0.50	0.00	0.00	0.00	3.03162E-20	1.32749E-05
9	12.00	0.00	0.00	-0.80	0.00	10.00	0.50	0.00	0.00	0.00	6.26499E-19	1.3495E-05
10	12.00	0.00	0.00	-0.80	0.00	10.00	0.50	0.00	0.00	0.00	1.62522E-20	1.21132E-05

Table B.4. Result of the 10 runs for the variable X_2 of the 4-dimensional system with beta initial guesses randomly distributed in the range [1, 12].

Run	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	Error1	Error2
1	8.00	0.50	0.00	0.00	0.00	3.00	0.00	0.75	0.00	0.00	4.73294E-20	9.0079E-05
2	8.00	0.50	0.00	0.00	0.00	3.00	0.00	0.75	0.00	0.00	1.94596E-19	8.7003E-05
3	8.00	0.50	0.00	0.00	0.00	3.00	0.00	0.75	0.00	0.00	2.42293E-18	8.6766E-05
4	14.10	0.35	0.19	-0.03	0.02	9.10	0.11	0.53	-0.03	0.02	3.74444E-05	0.00010221
5	16.18	0.33	0.21	-0.02	0.02	11.18	0.13	0.50	-0.03	0.02	4.24392E-05	0.00010673
6	8.00	0.50	0.00	0.00	0.00	3.00	0.00	0.75	0.00	0.00	2.44827E-20	8.7397E-05
7	16.47	0.33	0.22	-0.02	0.02	11.47	0.14	0.50	-0.02	0.02	4.62284E-05	0.00010031
8	13.40	0.36	0.18	-0.01	0.01	8.40	0.12	0.54	-0.01	0.02	3.76855E-05	9.9757E-05
9	15.86	0.33	0.21	-0.02	0.02	10.86	0.13	0.51	-0.02	0.02	4.29122E-05	0.00010555
10	8.00	0.50	0.00	0.00	0.00	3.00	0.00	0.75	0.00	0.00	5.2642E-19	8.6676E-05

Table B.5. Result of the 10 runs for the variable X_3 of the 4-dimensional system with beta initial guesses randomly distributed in the range [1, 12].

Run	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	Error1	Error2
1	3.00	0.00	0.75	0.00	0.00	5.00	0.00	0.00	0.50	0.20	2.43701E-18	7.6974E-05
2	8.63	0.02	0.53	0.19	0.07	10.63	0.03	0.22	0.40	0.15	2.20633E-06	7.4771E-05
3	10.00	0.02	0.51	0.21	0.07	12.00	0.03	0.24	0.39	0.14	2.44862E-06	7.467E-05
4	9.99	0.02	0.51	0.21	0.07	11.99	0.03	0.24	0.39	0.14	2.42734E-06	8.1619E-05
5	3.00	0.00	0.75	0.00	0.00	5.00	0.00	0.00	0.50	0.20	1.06429E-18	8.7564E-05
6	7.74	0.02	0.54	0.18	0.06	9.74	0.03	0.21	0.41	0.15	2.06481E-06	7.5389E-05
7	8.17	0.02	0.53	0.18	0.06	10.17	0.03	0.21	0.40	0.15	2.1591E-06	8.2248E-05
8	7.28	0.02	0.55	0.17	0.06	9.28	0.02	0.20	0.41	0.15	1.92141E-06	8.0877E-05
9	10.00	0.02	0.51	0.21	0.07	12.00	0.03	0.24	0.39	0.14	2.47521E-06	8.7686E-05
10	10.00	0.02	0.51	0.21	0.07	12.00	0.03	0.24	0.39	0.14	2.41783E-06	7.4469E-05

Table B.6. Result of the 10 runs for the variable X_4 of the 4-dimensional system with beta initial guesses randomly distributed in the range [1, 12].

Run	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	Error1	Error2
1	8.00	0.36	-0.04	0.06	0.26	12.00	0.18	-0.05	0.06	0.56	8.19133E-08	1.3953E-06
2	8.00	0.36	-0.04	0.06	0.26	12.00	0.18	-0.05	0.06	0.56	8.10149E-08	1.4304E-06
3	8.00	0.36	-0.04	0.06	0.26	12.00	0.18	-0.05	0.06	0.56	8.16504E-08	1.4308E-06
4	8.00	0.36	-0.04	0.06	0.26	12.00	0.18	-0.04	0.06	0.56	7.26348E-08	1.4368E-06
5	2.00	0.50	0.00	0.00	0.00	6.00	0.00	0.00	0.00	0.80	1.14483E-19	1.3818E-06
6	8.00	0.36	-0.04	0.06	0.26	12.00	0.18	-0.05	0.06	0.56	7.88685E-08	1.4308E-06
7	2.00	0.50	0.00	0.00	0.00	6.00	0.00	0.00	0.00	0.80	1.97709E-19	1.3985E-06
8	7.65	0.37	-0.04	0.06	0.26	11.65	0.18	-0.04	0.06	0.56	8.21441E-08	1.4234E-06
9	7.83	0.36	-0.04	0.06	0.26	11.83	0.18	-0.04	0.06	0.56	7.9069E-08	1.4524E-06
10	8.00	0.36	-0.04	0.06	0.26	12.00	0.18	-0.05	0.06	0.56	8.28791E-08	1.4294E-06

5-Dimensional system results – noise-free time series

Tables B.7-B.10 show the parameters found with the proposed algorithm for the 5-dimensional system (Eq. (4.37)) using the first set of initial values of the Table B.2. The time series used in this case study were obtained by numerical integration of the 5-dimensional system in the interval [0,5] with 0.1 sampling interval.

Table B.7. Result of the 10 runs for the variable X_1 of the 5-dimensional system with beta initial guesses uniformly distributed in the range [1, 10].

Run	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	g_{i5}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	h_{i5}	Error1	Error2
1	5.00	0.00	0.00	1.00	0.00	-1.00	10.00	2.00	0.00	0.00	0.00	0.00	1.22343E-21	7.99402E-21
2	5.00	0.00	0.00	1.00	0.00	-1.00	10.00	2.00	0.00	0.00	0.00	0.00	1.32077E-21	4.59591E-21
3	5.00	0.00	0.00	1.00	0.00	-1.00	10.00	2.00	0.00	0.00	0.00	0.00	1.32077E-21	3.6282E-21
4	5.00	0.00	0.00	1.00	0.00	-1.00	10.00	2.00	0.00	0.00	0.00	0.00	1.32077E-21	4.03837E-21
5	5.00	0.00	0.00	1.00	0.00	-1.00	10.00	2.00	0.00	0.00	0.00	0.00	1.32077E-21	2.72773E-21
6	5.00	0.00	0.00	1.00	0.00	-1.00	10.00	2.00	0.00	0.00	0.00	0.00	5.15785E-22	4.19796E-21
7	5.00	0.00	0.00	1.00	0.00	-1.00	10.00	2.00	0.00	0.00	0.00	0.00	1.05374E-20	5.19914E-21
8	5.00	0.00	0.00	1.00	0.00	-1.00	10.00	2.00	0.00	0.00	0.00	0.00	2.22619E-21	1.58118E-20
9	5.00	0.00	0.00	1.00	0.00	-1.00	10.00	2.00	0.00	0.00	0.00	0.00	6.3629E-22	3.98716E-07
10	5.00	0.00	0.00	1.00	0.00	-1.00	10.00	2.00	0.00	0.00	0.00	0.00	1.83537E-21	5.99782E-21

Table B.8. Result of the 10 runs for the variable X_2 of the 5-dimensional system with beta initial guesses uniformly distributed in the range [1, 10].

Run	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	g_{i5}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	h_{i5}	Error1	Error2
1	10.00	2.00	0.00	0.00	0.00	0.00	10.00	0.00	2.00	0.00	0.00	0.00	1.58004E-18	1.26958E-20
2	10.00	2.00	0.00	0.00	0.00	0.00	10.00	0.00	2.00	0.00	0.00	0.00	6.21989E-19	5.93212E-21
3	10.00	2.00	0.00	0.00	0.00	0.00	10.00	0.00	2.00	0.00	0.00	0.00	2.40218E-18	1.14226E-20
4	10.00	2.00	0.00	0.00	0.00	0.00	10.00	0.00	2.00	0.00	0.00	0.00	3.48423E-18	1.03853E-20
5	10.00	2.00	0.00	0.00	0.00	0.00	10.00	0.00	2.00	0.00	0.00	0.00	1.31001E-18	5.43993E-21
6	10.00	2.00	0.00	0.00	0.00	0.00	10.00	0.00	2.00	0.00	0.00	0.00	1.31001E-18	9.8609E-21
7	10.00	2.00	0.00	0.00	0.00	0.00	10.00	0.00	2.00	0.00	0.00	0.00	1.31001E-18	7.29945E-21
8	10.00	2.00	0.00	0.00	0.00	0.00	10.00	0.00	2.00	0.00	0.00	0.00	6.02476E-19	8.94865E-21
9	10.00	2.00	0.00	0.00	0.00	0.00	10.00	0.00	2.00	0.00	0.00	0.00	3.64807E-18	1.20049E-07
10	10.00	2.00	0.00	0.00	0.00	0.00	10.00	0.00	2.00	0.00	0.00	0.00	4.91918E-18	3.01878E-20

Table B.9. Result of the 10 runs for the variable X_3 of the 5-dimensional system with beta initial guesses uniformly distributed in the range [1, 10].

Run	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	g_{i5}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	h_{i5}	Error1	Error2
1	10.00	0.00	-1.00	0.00	0.00	0.00	10.00	0.00	-1.00	2.00	0.00	0.00	1.28105E-20	2.16667E-22
2	10.00	0.00	-1.00	0.00	0.00	0.00	10.00	0.00	-1.00	2.00	0.00	0.00	1.28105E-20	7.2618E-23
3	10.00	0.00	-1.00	0.00	0.00	0.00	10.00	0.00	-1.00	2.00	0.00	0.00	1.28105E-20	2.00922E-22
4	10.00	0.00	-1.00	0.00	0.00	0.00	10.00	0.00	-1.00	2.00	0.00	0.00	1.28105E-20	2.0588E-22
5	10.00	0.00	-1.00	0.00	0.00	0.00	10.00	0.00	-1.00	2.00	0.00	0.00	1.28105E-20	1.62388E-22
6	10.00	0.00	-1.00	0.00	0.00	0.00	10.00	0.00	-1.00	2.00	0.00	0.00	1.28105E-20	1.64324E-22
7	10.00	0.00	-1.00	0.00	0.00	0.00	10.00	0.00	-1.00	2.00	0.00	0.00	1.03057E-19	1.04365E-22
8	10.00	0.00	-1.00	0.00	0.00	0.00	10.00	0.00	-1.00	2.00	0.00	0.00	3.29855E-21	3.31941E-22
9	10.00	0.00	-1.00	0.00	0.00	0.00	10.00	0.00	-1.00	2.00	0.00	0.00	1.82851E-21	1.44575E-09
10	10.00	0.00	-1.00	0.00	0.00	0.00	10.00	0.00	-1.00	2.00	0.00	0.00	2.01525E-19	2.13752E-21

Table B.10. Result of the 10 runs for the variable X_4 of the 5-dimensional system with beta initial guesses uniformly distributed in the range [1, 10].

Run	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	g_{i5}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	h_{i5}	Error1	Error2
1	8.00	0.00	0.00	2.00	0.00	-1.00	10.00	0.00	0.00	0.00	2.00	0.00	2.9075E-21	2.58337E-21
2	8.00	0.00	0.00	2.00	0.00	-1.00	10.00	0.00	0.00	0.00	2.00	0.00	8.9608E-22	4.95432E-21
3	8.00	0.00	0.00	2.00	0.00	-1.00	10.00	0.00	0.00	0.00	2.00	0.00	8.1267E-22	4.23046E-21
4	8.00	0.00	0.00	2.00	0.00	-1.00	10.00	0.00	0.00	0.00	2.00	0.00	8.1267E-22	4.83447E-21
5	8.00	0.00	0.00	2.00	0.00	-1.00	10.00	0.00	0.00	0.00	2.00	0.00	8.1267E-22	5.07495E-21
6	8.00	0.00	0.00	2.00	0.00	-1.00	10.00	0.00	0.00	0.00	2.00	0.00	8.1267E-22	6.43373E-21
7	8.00	0.00	0.00	2.00	0.00	-1.00	10.00	0.00	0.00	0.00	2.00	0.00	7.7966E-22	2.41294E-21
8	8.00	0.00	0.00	2.00	0.00	-1.00	10.00	0.00	0.00	0.00	2.00	0.00	5.8343E-20	2.24091E-20
9	8.00	0.00	0.00	2.00	0.00	-1.00	10.00	0.00	0.00	0.00	2.00	0.00	1.5997E-21	6.49168E-07
10	8.00	0.00	0.00	2.00	0.00	-1.00	10.00	0.00	0.00	0.00	2.00	0.00	3.5062E-21	8.42201E-21

Table B.11. Result of the 10 runs for the variable X_5 of the 5-dimensional system with beta initial guesses uniformly distributed in the range [1, 10].

Run	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	g_{i5}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	h_{i5}	Error1	Error2
1	10.00	0.00	0.00	0.00	2.00	0.00	10.00	0.00	0.00	0.00	0.00	2.00	1.0044E-18	1.64228E-21
2	10.00	0.00	0.00	0.00	2.00	0.00	10.00	0.00	0.00	0.00	0.00	2.00	1.0044E-18	2.32214E-21
3	10.00	0.00	0.00	0.00	2.00	0.00	10.00	0.00	0.00	0.00	0.00	2.00	1.0044E-18	2.16712E-21
4	10.00	0.00	0.00	0.00	2.00	0.00	10.00	0.00	0.00	0.00	0.00	2.00	1.0044E-18	2.7372E-21
5	10.00	0.00	0.00	0.00	2.00	0.00	10.00	0.00	0.00	0.00	0.00	2.00	5.1333E-19	2.84682E-21
6	10.00	0.00	0.00	0.00	2.00	0.00	10.00	0.00	0.00	0.00	0.00	2.00	6.2861E-19	2.81858E-21
7	10.00	0.00	0.00	0.00	2.00	0.00	10.00	0.00	0.00	0.00	0.00	2.00	3.4288E-19	1.1659E-21
8	10.00	0.00	0.00	0.00	2.00	0.00	10.00	0.00	0.00	0.00	0.00	2.00	3.9076E-20	1.14064E-20
9	9.15	-0.60	0.07	-0.29	2.59	0.04	10.96	0.56	-0.23	-1.00	-0.51	2.00	0.00011205	3.68627E-07
10	10.00	0.00	0.00	0.00	2.00	0.00	10.00	0.00	0.00	0.00	0.00	2.00	6.7383E-23	3.01361E-21

10-Dimensional system results – noise-free data

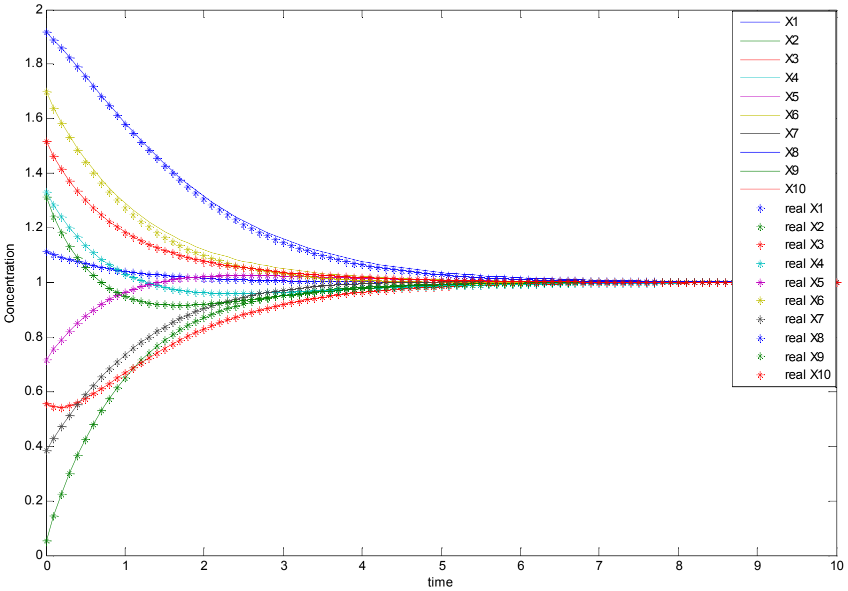


Figure B.1. Simulation result of the 10-dimensional model obtained using EO algorithm. The dots represent the synthesis time series and the lines represent the fitting.

B.1.3 Noisy time series

4-Dimensional system results

Tables B.12-B.15 show the parameters found with the proposed algorithm for the 4-dimensional system (Eq. (4.36)) using noisy time series.

Table B.12. Result of the 10 runs for the variable X_1 of the 4-dimensional system with beta initial guesses randomly distributed in the range [1, 12].

Run	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	Error1	Error2
1	10.38	0.53	1.07	-0.61	-0.62	9.49	0.61	1.11	-0.53	-0.67	0.4762953	0.3589214
2	10.04	0.53	1.07	-0.60	-0.62	9.15	0.61	1.10	-0.52	-0.66	0.4754111	0.3621649
3	7.49	0.53	1.05	-0.43	-0.44	6.50	0.66	1.10	-0.32	-0.51	0.4120744	0.212672
4	7.27	0.52	1.02	-0.44	-0.43	6.27	0.65	1.08	-0.31	-0.50	0.4005951	0.2729454
5	6.79	0.55	0.97	-0.49	-0.56	5.83	0.69	1.03	-0.36	-0.63	0.4618373	0.321666
6	3.80	0.59	0.61	-0.10	-0.32	2.57	0.96	0.76	0.28	-0.49	0.4082135	0.1781068
7	10.97	0.52	1.08	-0.62	-0.63	10.08	0.60	1.12	-0.55	-0.67	0.477732	0.328733
8	9.85	0.53	1.06	-0.60	-0.62	8.95	0.61	1.10	-0.52	-0.66	0.474871	0.244898
9	10.02	0.53	1.07	-0.60	-0.62	9.12	0.61	1.10	-0.52	-0.66	0.475339	0.348363
10	9.24	0.53	1.05	-0.58	-0.61	8.33	0.62	1.09	-0.49	-0.66	0.473011	0.325423

Table B.13. Result of the 10 runs for the variable X_2 of the 4-dimensional system with beta initial guesses randomly distributed in the range [1, 12].

Run	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	Error1	Error2
1	16.87	0.09	0.19	0.00	-0.12	11.77	0.12	0.32	0.17	-0.20	0.4738779	2.5140256
2	16.03	0.09	0.18	-0.01	-0.12	10.95	0.12	0.33	0.17	-0.21	0.4708193	3.0556242
3	7.68	-0.16	0.39	-0.09	0.28	2.35	0.00	0.78	0.55	0.00	0.5107566	2.1002638
4	12.04	0.06	0.20	-0.03	-0.09	6.96	0.12	0.39	0.24	-0.23	0.4966603	1.6639319
5	9.20	0.07	0.25	-0.10	-0.04	4.27	0.13	0.54	0.24	-0.22	0.4698792	2.4678547
6	7.67	-0.06	0.42	-0.11	0.06	2.83	0.11	0.74	0.34	-0.23	0.5198797	1.1053894
7	10.05	0.13	0.25	-0.03	-0.08	5.12	0.19	0.51	0.25	-0.24	0.481542	2.284232
8	8.07	-0.13	0.55	-0.09	0.08	3.30	0.06	0.82	0.33	-0.20	0.5677575	1.7613932
9	17.05	0.09	0.19	0.00	-0.12	11.95	0.12	0.32	0.17	-0.20	0.4746868	2.6943604
10	13.51	0.16	0.16	0.03	-0.17	8.47	0.21	0.33	0.23	-0.29	0.4833689	1.7781254

Table B.14. Result of the 10 runs for the variable X_3 of the 4-dimensional system with beta initial guesses randomly distributed in the range [1, 12].

Run	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	Error1	Error2
1	5.84	0.40	0.50	0.33	-0.60	5.50	0.39	0.41	0.32	-0.66	0.2020232	4.0742276
2	3.45	0.44	0.44	0.39	-0.57	3.09	0.41	0.28	0.36	-0.69	0.2055951	5.705558
3	8.55	0.37	0.56	0.34	-0.62	8.21	0.36	0.51	0.33	-0.66	0.2160534	2.6206851
4	4.23	0.10	0.70	0.37	-0.18	3.96	0.07	0.53	0.39	-0.26	0.2222057	2.2206794
5	4.67	0.41	0.48	0.35	-0.59	4.32	0.40	0.37	0.34	-0.67	0.2030091	4.7700925
6	9.33	0.37	0.56	0.33	-0.62	8.99	0.36	0.51	0.33	-0.66	0.2156485	0.730108
7	4.84	0.41	0.48	0.35	-0.59	4.49	0.39	0.37	0.33	-0.67	0.2028097	4.8496191
8	10.35	0.36	0.57	0.33	-0.62	10.01	0.36	0.52	0.33	-0.65	0.2152312	3.1622848
9	4.46	0.06	0.63	0.10	-0.22	4.21	0.00	0.44	0.06	-0.31	0.2010062	4.7682708
10	4.58	0.02	0.85	0.01	-0.42	4.29	0.00	0.74	0.00	-0.49	0.2204152	2.301361

Table B.15. Result of the 10 runs for the variable X_4 of the 4-dimensional system with beta initial guesses randomly distributed in the range [1, 12].

Run	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	Error1	Error2
1	5.48	0.85	0.47	-0.95	-0.71	6.56	0.72	0.50	-1.00	-0.56	0.0291203	0.4771414
2	5.49	0.86	0.49	-0.95	-0.69	6.57	0.73	0.53	-1.00	-0.55	0.0291989	0.4827924
3	10.74	0.76	0.63	-0.97	-0.54	11.81	0.69	0.65	-1.00	-0.46	0.027442	0.4236371
4	6.56	0.81	0.59	-0.95	-0.58	7.63	0.69	0.62	-1.00	-0.46	0.0275361	0.4594413
5	9.68	0.50	0.48	-0.96	-0.43	10.92	0.42	0.50	-1.00	-0.33	0.0158422	0.44427
6	10.16	0.44	0.84	-0.89	-0.34	11.30	0.36	0.86	-0.92	-0.26	0.0269737	0.3826479
7	5.81	0.82	0.56	-0.95	-0.62	6.89	0.70	0.60	-1.00	-0.48	0.0278796	0.4574748
8	6.35	0.81	0.58	-0.95	-0.59	7.42	0.69	0.62	-1.00	-0.46	0.0274093	0.4284966
9	8.65	0.79	0.61	-0.96	-0.56	9.72	0.70	0.63	-1.00	-0.46	0.0274775	0.4720405
10	7.91	0.79	0.60	-0.96	-0.56	8.98	0.70	0.63	-1.00	-0.46	0.0273709	0.4441718

B.2 Error surfaces

In order to visually explore the results of the proposed algorithm and clarify the pattern of convergence, several error surfaces are presented in this section, all resulting from experiments with the 2- and 4-dimensional systems (Eqs. (4.35) and (4.36)). The surfaces were built with the same procedure described in Chapter 4 (Section 4.3.2).

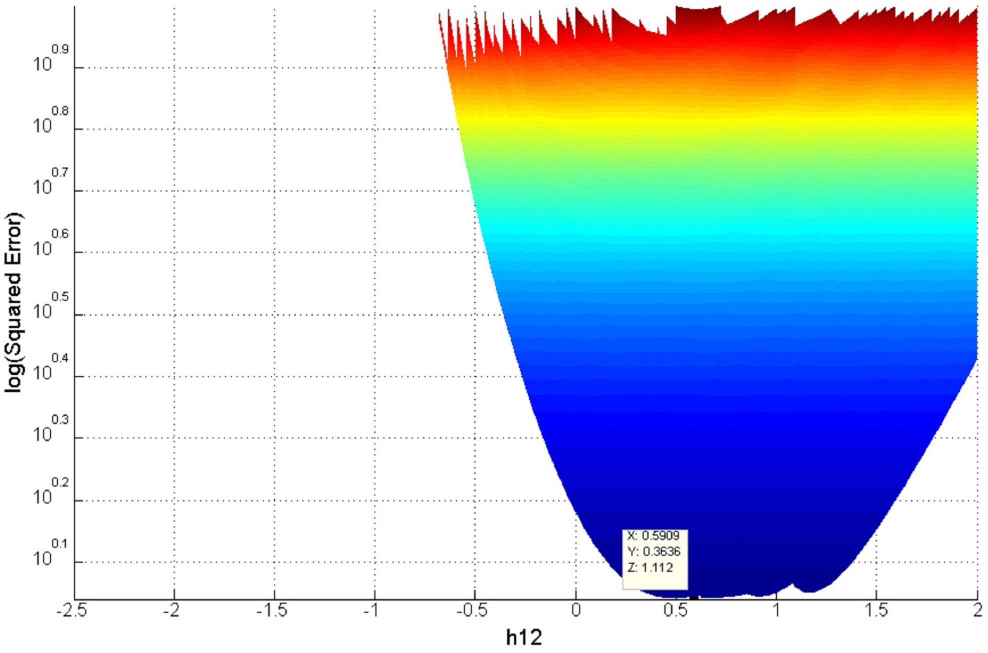


Figure B.2. Z-Y projection of the error surfaces shown in the Figure 4.4 of Chapter 4 obtained with noisy time series.
The optimal point (labeled) is not conserved from the noise-free error surfaces, but it is essentially indistinguishable from local minimum.

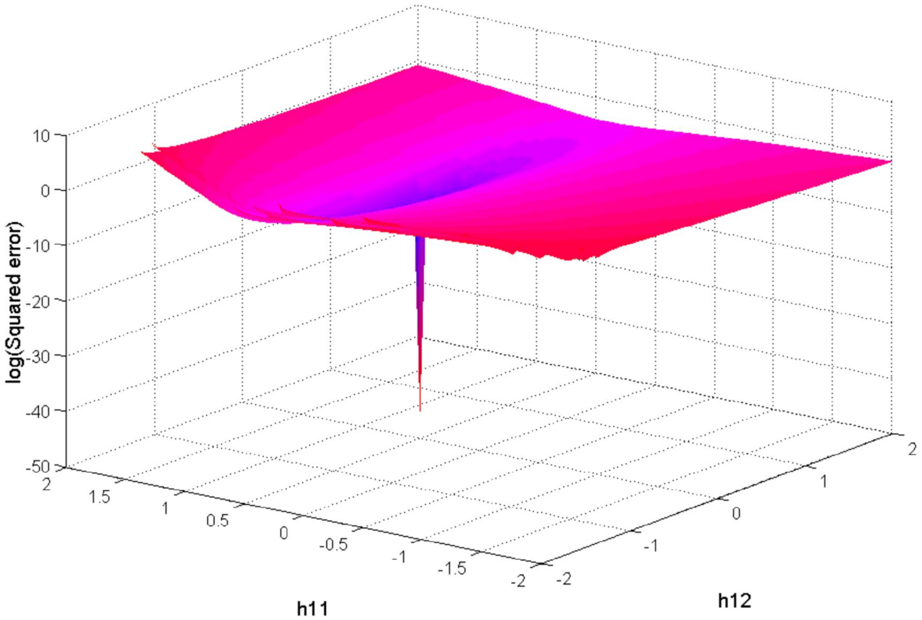


Figure B.3. Error surfaces (for β_1=10 and β_1=12) of the state variable X_1 of the 4-dimensional system. Only the kinetic orders h_{11} and h_{12} were screened (h_{13} and h_{14} were set to zero).

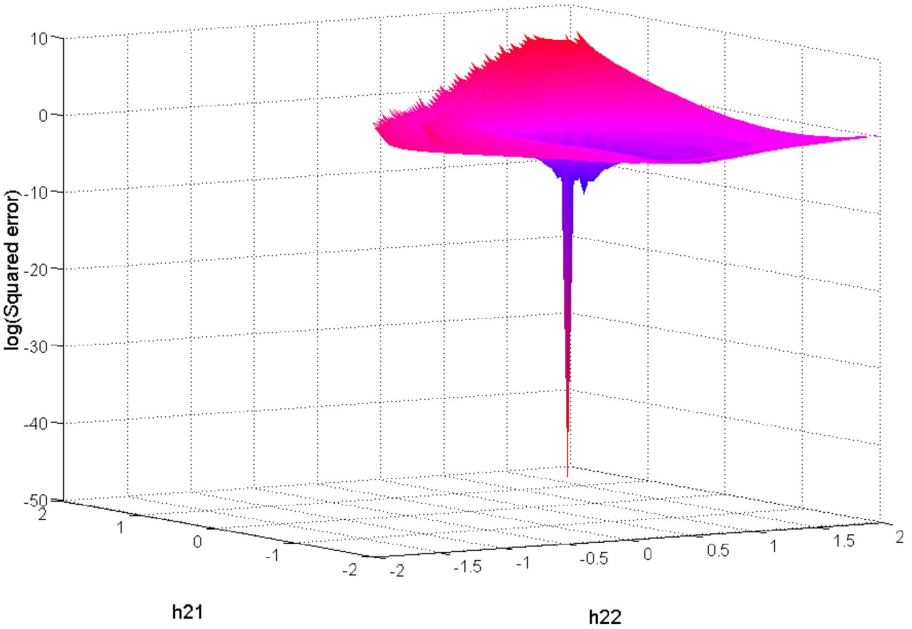


Figure B.4. Error surfaces (for β_2=2 and β_2=3) of the state variable X_2 of the 4-dimensional system. Only the kinetic orders h_{21} and h_{22} were screened (h_{23} and h_{24} were set to zero).

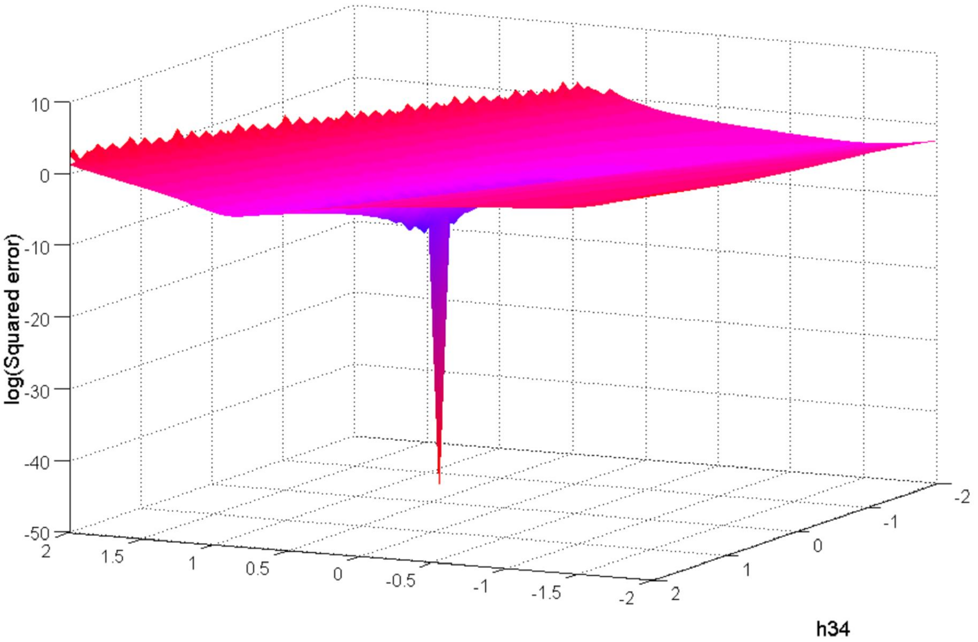


Figure B.5. Error surfaces (for β_3=5 and β_3=7) of the state variable X_3 of the 4-dimensional system. Only the kinetic orders h_{33} and h_{34} were screened (h_{31} and h_{32} were set to zero).

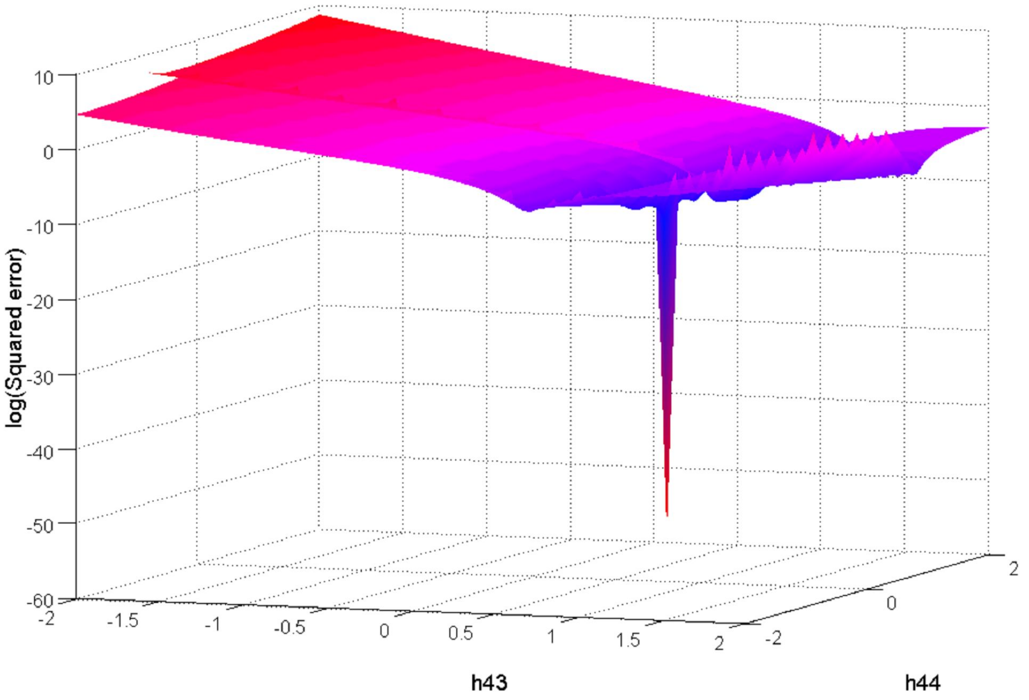
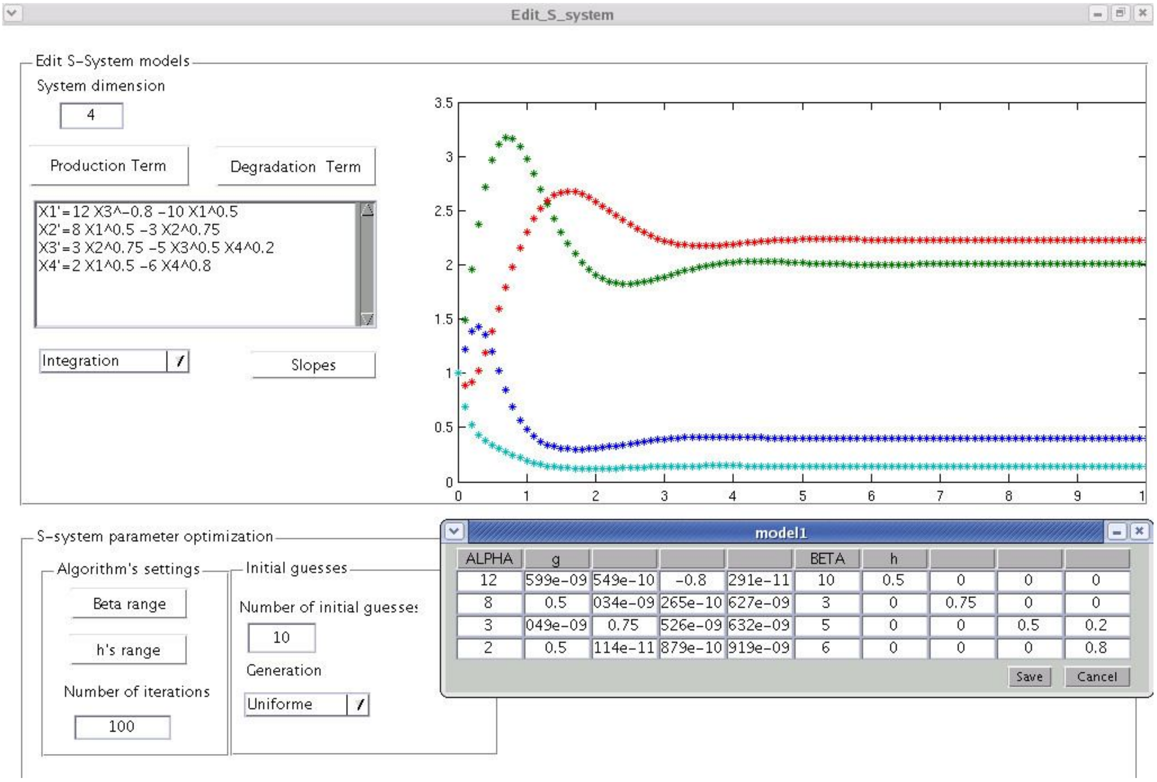


Figure B.6. Error surfaces (for β_4=4 and β_4=6) of the state variable X_4 of the 4-dimensional system. Only the kinetic orders h_{43} and h_{44} were screened (h_{41} and h_{42} were set to zero).

B.3 Software availability

The implementation of the algorithm described in this report is made publicly (GNU GPL) available with open source as Matlab® m-code (MathWorks Inc.) at http://code.google.com/p/s-system-inference/. For the convenience of those without a MathWorks license we have also compiled the code as a stand-alone application made publicly available at the same site, or as a module ("Signal Extraction Toolbox") of the code distribution infrastructure of the Bioinformatics Station resource http://bioinformaticstation.org. A snapshot of the Graphical User Interface (GUI) is shown in Figure B.7. All computational results and graphics described in this report can be reproduced using this application.



System dimension: 4

```
X1'=12 X3^-0.8 -10 X1^0.5
X2'=8 X1^0.5 -3 X2^0.75
X3'=3 X2^0.75 -5 X3^0.5 X4^0.2
X4'=2 X1^0.5 -6 X4^0.8
```

Integration

Number of iterations: 100

Number of initial guesses: 10

Generation: Uniforme

ALPHA	g				BETA	h			
12	599e-09	549e-10	-0.8	291e-11	10	0.5	0	0	0
8	0.5	034e-09	265e-10	627e-09	3	0	0.75	0	0
3	049e-09	0.75	526e-09	632e-09	5	0	0	0.5	0.2
2	0.5	114e-11	879e-10	919e-09	6	0	0	0	0.8

Figure B.7. Software application.
Snapshot of the graphical user interface provided as a free stand-alone application.

REFERENCES

Akaike, H. (1974) New Look at Statistical-Model Identification. *IEEE Transactions on Automatic Control*, AC19**,** 716-723.

Almeida, J. S. (2002) Predictive non-linear modeling of complex data by artificial neural networks. *Curr. Opin. Biotechnol.*, 13**,** 72-76.

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

Alvarez-Vasquez, F., Sims, K. J., Hannun, Y. A., and Voit, E. O. (2004) Integration of kinetic information on yeast sphingolipid metabolism in dynamical pathway models. *J. Theor. Biol.*, 226**,** 265-291.

Alvarez-Vasquez, F., Sims, K. J., Voit, E. O., and Hannun, Y. A. (2007) Coordination of the dynamics of yeast sphingolipid metabolism during the diauxic shift. *Theor. Biol. Med. Model.*, 4**,** 42.

Alvarez-Vasquez, F., Sims, K. J., Cowart, L. A., Okamoto, Y., Voit, E. O., and Hannun, Y. A. (2005) Simulation and validation of modelled sphingolipid metabolism in Saccharomyces cerevisiae. *Nature*, 433**,** 425-430.

Alves, R., Herrero, E., and Sorribas, A. (2004) Predictive reconstruction of the mitochondrial iron-sulfur cluster assembly metabolism: I. The role of the protein pair ferredoxin-ferredoxin reductase (Yah1-Axh1). *Proteins: Structure Function and Bioinformatics*, 56**,** 354-366.

Arkin, A. and Ross, J. (1995) Statistical construction of chemical-reaction mechanisms from measured time-series. *J. Phys. Chem.*, 99**,** 970-979.

Arkin, A., Shen, P. D., and Ross, J. (1997) A test case of correlation metric construction of a reaction pathway from measurements. *Science*, 277**,** 1275-1279.

Atkinson, M. R., Savageau, M. A., Myers, J. T., and Ninfa, A. J. (2003) Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in Escherichia coli. *Cell*, 113**,** 597-607.

Balthis, W. L. (1998) Application of Hierarchical Monte Carlo Simulation to the Estimation of Human Exposure to Mercury via Consumption of King Mackerel (Scomberomorus cavalla), Medical University of South Carolina.

Barabási, A.-L., Albert, R., Jeong, H., and Bianconi, G. (2000) Power-law distribution of the World Wide Web. *Science*, 287**,** 2115.

Berg, P. H., Voit, E. O., and White, R. L. (1996) A pharmacodynamic model for the action of the antibiotic imipenem on Pseudomonas aeruginosa populations in vitro. *Bull. Math. Biol.*, 58**,** 923-938.

Bono, H., Ogata, H., Goto, S., and Kanehisa, M. (1998) Reconstruction of amino acid biosynthesis pathways from the complete genome sequence. *Genome Res.*, 8**,** 203-210.

Burden, R. L. and Faires, J. D. (1993) *Numerical Analysis*. 5th ed. PWS Publishing Co, Boston, MA.

Cascante, M., Curto, R., and 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.

Chevalier, T., Schreiber, I., and Ross, J. (1993) Toward a Systematic Determination of Complex Reaction Mechanisms. *J. Phys. Chem.*, 97**,** 6776-6787.

Cho, D. Y., Cho, K. H., and Zhang, B. T. (2006) Identification of biochemical networks by S-tree based genetic programming. *Bioinformatics*, 22**,** 1631-1640.

Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T. G., Gabrielian, A. E., Landsman, D., Lockhart, D. J., and Davis, R. W. (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell*, 2**,** 65-73.

Chou, I-C., Vilela, M., Almeida, J. S., and Voit, E. O. (2008) Computational identification of dynamic biological networks: Inverse modeling approach and parameter estimation strategies. *International Conference on Molecular Systems Biology 2008 (ICMSB08)*, Manila, Philippines.

Crampin, E. J., Schnell, S., and McSharry, P. E. (2004a) Mathematical and computational techniques to deduce complex biochemical reaction mechanisms. *Prog. Biophys. Mol. Biol.*, 86**,** 77-112.

Crampin, E. J., McSharry, P. E., and Schnell, S. (2004b) Extracting biochemical reaction kinetics from time series data. In, *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 329-336.

Curto, R., Sorribas, A., and 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.

Curto, R., Voit, E. O., and Cascante, M. (1998a) Analysis of abnormalities in purine metabolism leading to gout and to neurological dysfunctions in man. *Biochem. J.*, 329 (Pt 3)**,** 477-487.

Curto, R., Voit, E. O., Sorribas, A., and Cascante, M. (1997) Validation and steady-state analysis of a power-law model of purine metabolism in man. *Biochem. J.*, 324 (Pt 3)**,** 761-775.

Curto, R., Voit, E. O., Sorribas, A., and Cascante, M. (1998b) Mathematical models of purine metabolism in man. *Math. Biosci.*, 151**,** 1-49.

Daisuke, T. and Horton, P. (2006) Inference of scale-free networks from gene expression time series. *J. Bioinform. Comput. Biol.*, 4**,** 503-514.

de Boor, C. (1978) *A practical guide to splines*. *Applied mathematical sciences: 27*, Springer-Verlag, New York, xxiv, 392 pp.

de Boor, C., Höllig, K., and Riemenschneider, S. D. (1993) *Box splines*. *Applied mathematical sciences; v. 98*, Springer-Verlag, New York; Hong Kong, xvii, 200 pp.

Dedieu, J.-B. and Shub, M. (2005) Newton flow and interior point methods in linear programming. *Int. J. Bifurcat. Chaos*, 15**,** 827-840.

del Rosario, R. C., Mendoza, E., and Voit, E. O. (2008a) Challenges in lin-log modelling of glycolysis in Lactococcus lactis. *IET Syst. Biol.*, 2**,** 136.

del Rosario, R. C. H., Echavez, M. T., de Paz, M. T., Zuñiga, P. C., Bargo, M. C. R., Talaue, C. O., Arellano, C., Pasia, J. M., Naval, P. C., Voit, E. O., and Mendoza, E. (2008b) MADMan: a benchmarking framework for parameter estimation in biochemical systems theory models. *International Conference on Molecular Systems Biology 2008 (ICMSB08)*, Manila, Philippines, 10-13.

Díaz-Sierra, R., Lozano, J. B., and Fairén, V. (1999) Deduction of chemical mechanisms from the linear response around steady state. *J. Phys. Chem.*, 103**,** 337-343.

Eberhart, R. and Kennedy, J. (1995) A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science (MHS'95)*, 39-43.

Edwards, J. S. and Palsson, B. Ø. (2000) The Escherichia coli MG1655 in silico metabolic genotype: Its definition, characteristics, and capabilities. *Proc. Natl. Acad. Sci. U S A*, 97**,** 5528-5533.

Eilers, P. H. C. (2003) A perfect smoother. *Analytical Chemistry*, 75**,** 3631-3636.

Epureanu, B. I. and Greenside, H. S. (1998) Fractal basins of attraction associated with a damped Newton's method. *SIAM Rev*, 40**,** 102-109.

Fell, D. A. (1997) *Understanding the Control of Metabolism*. Portland Press, London.

Ferreira, A. (2000) Version 1.2. http://www.dqb.fc.ul.pt/docentes/aferreira/plas.html.

Ferreira, A. E., Ponces Freire, A. M., and Voit, E. O. (2003) A quantitative model of the generation of N(epsilon)-(carboxymethyl)lysine in the Maillard reaction between collagen and glucose. *Biochem. J.*, 376**,** 109-121.

Forster, J., Famili, I., Fu, P., Palsson, B. Ø., and Nielsen, J. (2003) Genome-scale reconstruction of the Saccharomyces cerevisiae metabolic network. *Genome Res.*, 13**,** 244-253.

Gavalas, G. R. (1968) *Nonlinear Differential Equations of Chemically Reacting Systems*. Springer-Verlag, Berlin.

Goel, G. (2008) *Reconstructing Biochemical Systems: Systems Modeling and Analysis Tools for Decoding Biological Designs*. VDM Verlag Dr. Müller, Saarbrücken, Germany.

Goel, G., Chou, I-C., and Voit, E. O. (2006) Biological systems modeling and analysis: A biomolecular technique of the twenty-first century. *J. Biomol. Tech.*, 17**,** 252-269.

Goel, G., Chou, I-C., and Voit, E. O. (submitted) System estimation from metabolic time series data.

Gombert, A. K. and Nielsen, J. (2000) Mathematical modelling of metabolism. *Curr. Opin. Biotechnol.*, 11**,** 180-186.

Gonzalez, O. R., Kuper, C., Jung, K., Naval, P. C., Jr., and Mendoza, E. (2007) Parameter estimation using Simulated Annealing for S-system models of biochemical networks. *Bioinformatics*, 23**,** 480-486.

Green, P. J. and Silverman, B. W. (1994) *Nonparametric regression and generalized linear models: A roughness penalty approach*. 1st ed. *Monographs on statistics and applied probability; 58.*, Chapman & Hall, London; New York, ix, 182 pp.

Gutenkunst, R. N., Waterfall, J. J., Casey, F. P., Brown, K. S., Myers, C. R., and Sethna, J. P. (2007) Universally sloppy parameter sensitivities in systems biology models. *PLoS Comput. Biol.*, 3**,** 1871-1878.

Hatzimanikatis, V. and Bailey, J. E. (1996) MCA has more to say. *J. Theor. Biol.*, 182**,** 233-242.

Hatzimanikatis, V., Floudas, C. A., and Bailey, J. E. (1996a) Analysis and design of metabolic reaction networks via mixed-integer linear optimization. *AIChE Journal*, 42**,** 1277-1292.

Hatzimanikatis, V., Floudas, C. A., and Bailey, J. E. (1996b) Optimization of regulatory architectures in metabolic reaction networks. *Biotechnol. Bioeng.*, 52**,** 485-500.

Heinrich, R. and Rapoport, T. A. (1974) A linear steady-state treatment of enzymatic chains. General properties, control and effector strength. *Eur. J. Biochem.*, 42**,** 89-95.

Heinrich, T. and Schuster, S. (1996) *The Regulation of Cellular Systems*. Chapman and Hall, New York.

Hendry, D. F. and Krolzig, H. M. (2003) New developments in automatic general-to-specific modelling. In Stigum, B. P. Ed., *Econometrics and the Philosophy of Economics*, Princeton University Press.

Hernández-Bermejo, B. and Sorribas, A. (2001) Analytical quantile solution for the S-distribution, random number generation and statistical data modeling. *Biometr. J.*, 43**,** 1007-1025.

Hlavacek, W. S. and Savageau, M. A. (1996) Rules for coupled expression of regulator and effector genes in inducible circuits. *J. Mol. Biol.*, 255**,** 121-139.

Ho, S. Y., Hsieh, C. H., Yu, F. C., and Huang, H. L. (2007) An intelligent two-stage evolutionary algorithm for dynamic pathway identification from gene expression profiles. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 4**,** 648-660.

Huang, W. H., Yuh, C. H., and Wang, F. S. (2006) Reverse engineering for embryonic gene regulatory network in zebrafish via evolutionary optimization with data collocation. *7th International Conference on Systems Biology*, Yokohama, Japan.

Hynne, F., Danø, S., and Sorensen, P. G. (2001) Full-scale model of glycolysis in Saccharomyces cerevisiae. *Biophys. Chem.*, 94**,** 121-163.

Imade, H., Mizuguchi, N., Ono, I., Ono, N., and Okamoto, M. (2005) "Gridifying" an evolutionary algorithm for inference of genetic networks using the improved GOGA framework and its performance evaluation on OBI grid. In Konagaya, A. and Satou, K. (eds), *Grid Computing in Life Science: First International Workshop on Life Science Grid, LSGRID 2004 Kanazawa, Japan, May 31-June 1, 2004*, Springer 171-186.

Ingalls, B. P. (2004) Autonomously oscillating biochemical systems: Parametric sensitivity of extrema and period. *Syst. Biol. (Stevenage)*, 1**,** 62-70.

Jeong, H., Tombor, B., Albert, R., Oltval, Z. N., and Barabási, A.-L. (2000) The large-scale organization of metabolic networks. *Nature*, 407**,** 651-654.

Judd, K. and Mees, A. (1995) On selecting models for nonlinear time-series. *Physica D*, 82**,** 426-444.

Kacser, H. and Burns, J. A. (1973) The control of flux. *Symp. Soc. Exp. Biol.*, 27**,** 65-104.

Kacser, H. and Burns, J. A. (1979) Molecular democracy: who shares the controls? *Biochem. Soc. Trans.*, 7**,** 1149-1160.

Karjalainen, E. J. (1989) The spectrum reconstruction problem: Use of alternating regression for unexpected spectral components in two-dimensional spectroscopies. *Chemom. Intell. Lab. Syst.*, 7**,** 31-38.

Kauffman, K. J., Prakash, P., and Edwards, J. S. (2003) Advances in flux balance analysis. *Curr. Opin. Biotechnol.*, 14**,** 491-496.

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

Kim, K.-Y., Cho, D.-Y., and Zhang, B.-T. (2006) Multi-stage evolutionary algorithms for efficient identification of gene regulatory networks. *EvoWorkshops 2006*, Springer, 45-56.

Kimura, S., Hatakeyama, M., and Konagaya, A. (2004) Inference of S-system models of genetic networks from noisy time-series data. *Chem-Bio Informatics Journal*, 4**,** 1-14.

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

Kitayama, T., Kinoshita, A., Sugimoto, M., Nakayama, Y., and Tomita, M. (2006) A simplified method for power-law modelling of metabolic pathways from time-course data and steady-state flux profiles. *Theor. Biol. Med. Model.*, 3**,** 24.

Koza, J. R., Mydlowec, W., Lanza, G., Yu, J., and Keane, M. A. (2001) Reverse engineering of metabolic pathways from observed data using genetic programming. *Pac. Symp. Biocomput.***,** 434-445.

Kuper, C. and Jung, K. (2005) CadC-mediated activation of the cadBA promoter in Escherichia coli. *J Mol. Microbiol. Biotechnol.*, 10**,** 26-39.

Kutalik, Z., Tucker, W., and Moulton, V. (2007) S-system parameter estimation for noisy metabolic profiles using newton-flow analysis. *IET Syst. Biol.*, 1**,** 174-180.

Lall, R. and Voit, E. O. (2005) Parameter estimation in modulated, unbranched reaction chains within biochemical systems. *Comput. Biol. Chem.*, 29**,** 309-318.

Liao, J. C., Boscolo, R., Yang, Y. L., Tran, L. M., Sabatti, C., and Roychowdhury, V. P. (2003) Network component analysis: reconstruction of regulatory signals in biological systems. *Proc. Natl. Acad. Sci. U S A*, 100**,** 15522-15527.

Liu, P. K. and Wang, F. S. (2008) Inference of biochemical network models in S-system using multiobjective optimization approach. *Bioinformatics*, 24**,** 1085-1092.

Magnus, J. R. and Neudecker, H. (1999) *Matrix Differential Calculus with Applications in Statistics and Econometrics*. *Wiley series in probability and mathematical statistics*, Wiley & Sons Ltd., Chichester, England; New York.

Mahadevan, R., Edwards, J. S., and Doyle, F. J. (2002) Dynamic flux balance analysis of diauxic growth in Escherichia coli. *Biophysical Journal*, 83**,** 1331-1340.

Maki, Y., Tominaga, D., Okamoto, M., Watanabe, S., and Eguchi, Y. (2001) Development of a system for the inference of large scale genetic networks. *Pac. Symp. Biocomput.***,** 446-458.

Maki, Y., Ueda, T., Okamoto, M., Uematsu, N., Inamura, Y., and Eguchi, Y. (2002) Inference of genetic network using the expression profile time course data of mouse P19 cells. *Genome Inform.*, 13**,** 382-383.

Mao, F., Wu, H., Dam, P., Chou, I-C., Voit, E. O., and Xu, Y. (2008) Prediction of biological pathways through data mining and information fusion. In Xu, Y. and Gogarten, J. P. (eds), *Computational Methods for Understanding Bacterial and Archaeal Genomes*, Imperial College Press.

Marino, S. and Voit, E. O. (2006) An automated procedure for the extraction of metabolic network information from time series data. *J. Bioinform. Comput. Biol.*, 4**,** 665-691.

Martens, H. and Naes, T. (1989) *Multivariate Calibration*. John Wiley & Son Ltd., Chichester, UK, 419 pp.

Matsubara, Y., Kikuchi, S., Sugimoto, M., and Tomita, M. (2006) Parameter estimation for stiff equations of biosystems using radial basis function networks. *BMC Bioinformatics*, 7**,** 230.

Mendes, P. and Kell, D. B. (1996) On the analysis of the inverse problem of metabolic pathways using artificial neural networks. *Biosystems*, 38**,** 15-28.

Mendes, P. and Kell, D. (1998) Non-linear optimization of biochemical pathways: Applications to metabolic engineering and parameter estimation. *Bioinformatics*, 14**,** 869-883.

Michaelis, L. and Menten, M. L. (1913) Die kinetik der invertinwirkung. *Biochemische Zeitschrift*, 49**,** 333-369.

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002) Network motifs: Simple building blocks of complex networks. *Science*, 298**,** 824-827.

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

Morishita, R., Imade, H., Ono, I., Ono, N., and Okamoto, M. (2003) Finding multiple solutions based on an evolutionary algorithm for inference of genetic networks by S-system. *The 2003 Congress on Evolutionary Computation 2003 (CEC2003)*, 615-622.

Muiño, J. M., Voit, E. O., and Sorribas, A. (2006) GS-distributions: A new family of distributions for continuous unimodal variables. *Comp. Stat. Data Anal.*, 50**,** 2769-2798.

Nakatsui, M., Ueda, T., and Okamoto, M. (2003) Integrated system for inference of gene expression network. *Genome Inform.*, 14**,** 282-283.

Naval, P. C., Sison, L. G., and Mendoza, E. (2006) Metabolic network parameter inference using particle swarm optimization. *International Conference on Molecular Systems Biology 2006 (ICMSB06)*, Munich, Germany.

Noman, N. and Iba, H. (2005a) Inference of gene regulatory networks using s-system and differential evolution. *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, Washington DC, USA, 439-46.

Noman, N. and Iba, H. (2005b) Reverse engineering genetic networks using evolutionary computation. *Genome Inform.*, 16**,** 205-214.

Noman, N. and Iba, H. (2005c) Enhancing differential evolution performance with local search for high dimensional function optimization. *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, Washington DC, USA, 967-974.

Noman, N. and Iba, H. (2006) Inference of genetic networks using S-system: information criteria for model selection. *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO'06)*, Seattle, Washington, USA, ACM Press, 263-270.

Noman, N. and Iba, H. (2007) Inferring gene regulatory networks using differential evolution with local search heuristics. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 4**,** 634-647.

Okamoto, M. (2008) System analysis of acetone-butanol-ethanol fermentation based on time-sliced metabolic flux analysis. *Symposium on Cellular Systems Biology*, National Chung Cheng University, Taiwan.

Okamoto, M., Nonaka, T., Ochiai, S., and Tominaga, D. (1998) Nonlinear numerical optimization with use of a hybrid Genetic Algorithm incorporating the Modified Powell method. *Applied Mathematics and Computation*, 91**,** 63-72.

Oliveira, M. R., Branco, J. A., Croux, C., and Filzmoser, P. (2004) Robust redundancy analysis by alternating regression. In Hubert, M., Pison, G., Struyf, A., and Van Aelst, S. (eds), *Theory and Applications of Recent Robust Methods*, Birkhauser, 235-246.

Ono, I., Seike, Y., Morishita, R., Ono, N., Nakatsui, M., and Okamoto, M. (2004) An evolutionary algorithm taking account of mutual interactions among substances for inference of genetic networks. *Congress on Evolutionary Computation 2004 (CEC2004)* 2060-7.

Palsson, B. Ø. (2006) *Systems Biology: Properties of Reconstructed Networks*. Cambridge University Press, New York.

Park, L. J., Park, C. H., Park, C., and Lee, T. (1997) Application of genetic algorithms to parameter estimation of bioprocesses. *Med. Biol. Eng. Comput.*, 35**,** 47-49.

Podani, J., Oltvai, Z. N., Jeong, H., Tombor, B., Barabasi, A. L., and Szathmary, E. (2001) Comparable system-level organization of Archaea and Eukaryotes. *Nat. Genet.*, 29**,** 54-56.

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

Rank, E. (2003) Application of Bayesian trained RBF networks to nonlinear time-series modeling. *Signal Process*, 83**,** 1393-1410.

Runarsson, T. P. and Yao, X. (2000) Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 4**,** 284-294.

Sakamoto, E. and Iba, H. (2001) Inferring a system of differential equations for a gene regulatorynetwork by using genetic programming. *Proceedings of the 2001 Congress on Evolutionary Computation (CEC2001)*, Seoul, South Korea, IEEE Press, 720-726.

Samoilov, M., Arkin, A., and Ross, J. (2001) On the deduction of chemical reaction pathways from measurements of time series of concentrations. *Chaos*, 11**,** 108-114.

Sands, P. J. and Voit, E. O. (1996) Flux-based estimation of parameters in S-systems. *Ecol. Modeling*, 93**,** 75-88.

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. (1976) *Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology*. Addison-Wesley Pub. Co. Advanced Book Program, Reading, Mass, xvii, 379 pp.

Savageau, M. A. (1982) A suprasystem of probability distributions. *Biometr. J.*, 24**,** 323-330.

Savageau, M. A. (1995) Enzyme kinetics in vitro and in vivo: Michaelis-Menten revisited. In Bittar, E. E. Ed., *Principles of Medical Biology*, JAI Press Inc.

Savageau, M. A. (1998) Development of fractal kinetic theory for enzyme-catalysed reactions and implications for the design of biochemical pathways. *Biosystems*, 47**,** 9-36.

Savageau, M. A. (2001) Design principles for elementary gene circuits: Elements, methods, and examples. *Chaos*, 11**,** 142-159.

Schulz, A. R. (1994) *Enzyme Kinetics: From Diastase to Multi-enzyme Systems*. Cambridge University Press, Cambridge; New York.

Schwacke, J. H. and Voit, E. O. (2005) Computation and analysis of time-dependent sensitivities in Generalized Mass Action systems. *J. Theor. Biol.*, 236**,** 21-38.

Seatzu, C. (2000) A fitting based method for parameter estimation in S-Systems. *Dynam. Systems Appl.*, 9**,** 77-98.

Selkov, E., Maltsev, N., Olsen, G. J., Overbeek, R., and Whitman, W. B. (1997) A reconstruction of the metabolism of Methanococcus jannaschii from sequence data. *Gene*, 197**,** GC11-26.

Shin, A. and Iba, H. (2003) Construction of genetic network using evolutionary algorithm and combined fitness function. *Genome Inform.*, 14**,** 94-103.

Shiraishi, F. and Savageau, M. A. (1992) The tricarboxylic-acid cycle in Dictyostelium discoideum. 1. Formulation of alternative kinetic representations. *Journal of Biological Chemistry*, 267**,** 22912-22918.

Sorribas, A. and Cascante, M. (1994) Structure identifiability in metabolic pathways: parameter estimation in models based on the power-law formalism. *Biochem. J.*, 298 (Pt 2)**,** 303-311.

Sorribas, A., Curto, R., and 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., Lozano, J. B., and Fairén, V. (1998) Deriving chemical and biochemical model networks from experimental measurements. *Recent Res. Devel. Phys. Chem.*, 2**,** 553-573.

Sorribas, A., March, J., and Voit, E. O. (2000) Estimating age-related trends in cross-sectional studies using S-distributions. *Stat. Med.*, 19**,** 697-713.

Sorribas, A., Hernandez-Bermejo, B., Vilaprinyo, E., and Alves, R. (2007) Cooperativity and saturation in biochemical networks: a saturable formalism using Taylor series approximations. *Biotechnol. Bioeng.*, 97**,** 1259-1277.

Spieth, C., Worzischek, R., and Streichert, F. (2006) Comparing evolutionary algorithms on the problem of network inference. *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO'06)*, Seattle, Washington, USA, ACM Press, 305-306.

Spieth, C., Streichert, F., Speer, N., and Zell, A. (2004a) Optimizing topology and parameters of gene regulatory network models from time-series experiments. In, *Genetic and Evolutionary Computation-GECCO 2004 (LNCS)*, Springer, 461-470.

Spieth, C., Streichert, F., Speer, N., and Zell, A. (2004b) A memetic inference method for gene regulatory networks based on S-Systems. *Congress on Evolutionary Computation 2004 (CEC2004)*, 152-157.

Spieth, C., Streichert, F., Supper, J., Speer, N., and Zell, A. (2005) Feedback memetic algorithms for modeling gene regulatory networks. *IEEE Symposium on*

Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), IEEE Press, 61-67.

Srividhya, J., Crampin, E. J., McSharry, P. E., and Schnell, S. (2007) Reconstructing biochemical pathways from time course data. *Proteomics*, 7**,** 828-838.

Stelling, J., Klamt, S., Bettenbrock, K., Schuster, S., and Gilles, E. D. (2002) Metabolic network structure determines key aspects of functionality and regulation. *Nature*, 420**,** 190-193.

Stephanopoulos, G., Aristidou, A. A., and Nielsen, J. (1998) *Metabolic Engineering: Principles and Methodologies*. Academic Press, San Diego, CA.

Sugimoto, M., Kikuchi, S., and Tomita, M. (2005) Reverse engineering of biochemical equations from time-course data by means of genetic programming. *Biosystems*, 80**,** 155-164.

Sutton, M. D., Smith, B. T., Godoy, V. G., and Walker, G. C. (2000) The SOS response: recent insights into umuDC-dependent mutagenesis and DNA damage tolerance. *Annu. Rev. Genet.*, 34**,** 479-497.

Teixeira, A. P., Santos, S. S., Carinhas, N., Oliveira, R., and Alves, P. M. (2008) Combining metabolic flux analysis tools and 13C NMR to estimate intracellular fluxes of cultured astrocytes. *Neurochem. Int.*, 52**,** 478-486.

Teusink, B., Passarge, J., Reijenga, C. A., Esgalhado, E., van der Weijden, C. C., Schepper, M., Walsh, M. C., Bakker, B. M., van Dam, K., Westerhoff, H. V., and Snoep, J. L. (2000) Can yeast glycolysis be understood in terms of in vitro kinetics of the constituent enzymes? Testing biochemistry. *Eur. J. Biochem.*, 267**,** 5313-5329.

Thomas, R., Mehrotra, S., Papoutsakis, E. T., and Hatzimanikatis, V. (2004) A model-based optimization framework for the inference on gene regulatory networks from DNA array data. *Bioinformatics*, 20**,** 3221-3235.

Tominaga, D., Koga, N., and Okamoto, M. (2000) Efficient numerical optimization algorithm based on genetic algorithm for inverse problem. *Proceedings of the Genetic and Evolutionary Computation Conference*, 251–258.

Torralba, A. S., Yu, K., Shen, P., Oefner, P. J., and Ross, J. (2003) Experimental test of a method for determining causal connectivities of species in reactions. *Proc. Natl. Acad. Sci. U S A*, 100**,** 1494-1498.

Torres, N. V. (1994) Modeling approach to control of carbohydrate-metabolism during citric-acid accumulation by Aspergillus-niger. 1. Model definition and stability of the steady-state. *Biotechnol. Bioeng.*, 44**,** 104-111.

Torres, N. V. and Voit, E. O. (2002) *Pathway Analysis and Optimization in Metabolic Engineering*. Cambridge University Press, Cambridge, U.K.

Torres, N. V., Voit, E. O., and Alcón, C. H. (1996) Optimization of nonlinear biotechnological processes with linear programming. Application to citric acid production in Aspergillus niger. *Biotechnol. Bioeng.*, 49**,** 247-258.

Tran, L. M., Brynildsen, M. P., Kao, K. C., Suen, J. K., and Liao, J. C. (2005) gNCA: A framework for determining transcription factor activity based on transcriptome: identifiability and numerical implementation. *Metab. Eng.*, 7**,** 128-141.

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

Tucker, W. and Moulton, V. (2006) Parameter reconstruction for biochemical networks using interval analysis. *Reliable Computing*, 12**,** 1-14.

Tucker, W., Kutalik, Z., and Moulton, V. (2007) Estimating parameters for generalized mass action models using constraint propagation. *Math. Biosci.*, 208**,** 607-620.

Ueda, T., Koga, N., and Okamoto, M. (2001) Efficient numerical optimization technique based on real-coded genetic algorithm. *Genome Inform.*, 12**,** 451-453.

Ueda, T., Ono, I., and Okamoto, M. (2002) Development of system identification technique based on real-coded genetic algorithm. *Genome Inform.*, 13**,** 386-387.

Vallino, J. J. and Stephanopoulos, G. (1993) Metabolic flux distributions in Corynebacterium glutamicum during growth and lysine overproduction. *Biotechnol. Bioeng.*, 41**,** 633-646.

Vance, W., Arkin, A., and Ross, J. (2002) Determination of causal connectivities of species in reaction networks. *Proc. Natl. Acad. Sci. U S A*, 99**,** 5816-5821.

Veflingstad, S. R., Almeida, J., and Voit, E. O. (2004) Priming nonlinear searches for pathway identification. *Theor. Biol. Med. Model.*, 1**,** 8.

Veflingstad, S. R., Dam, P., Xu, Y., and Voit, E. O. (2008) Microbial pathway models. In Xu, Y. and Gogarten, J. P. (eds), *Computational Methods for Understanding Bacterial and Archaeal Genomes*, Imperial College Press.

Vera, J., de Atauri, P., Cascante, M., and Torres, N. V. (2003) Multicriteria optimization of biochemical systems by linear programming: application to production of ethanol by Saccharomyces cerevisiae. *Biotechnol. Bioeng.*, 83**,** 335-343.

Vera, J., Balsa-Canto, E., Wellstead, P., Banga, J. R., and Wolkenhauer, O. (2007) Power-law models of signal transduction pathways. *Cellular Signalling*, 19**,** 1531-1541.

Vilela, M., Chou, I. C., Vinga, S., Vasconcelos, A. T., Voit, E. O., and Almeida, J. S. (2008) Parameter optimization in S-system models. *BMC Syst. Biol.*, 2**,** 35.

Vilela, M., Borges, C. C., Vinga, S., Vasconcelos, A. T., Santos, H., Voit, E. O., and Almeida, J. S. (2007) Automated smoother for the numerical decoupling of dynamics models. *BMC Bioinformatics*, 8**,** 305.

Visser, D. and Heijnen, J. J. (2002) The mathematics of metabolic control analysis revisited. *Metab. Eng.*, 4**,** 114-123.

Voit, E., Neves, A. R., and Santos, H. (2006a) The intricate side of systems biology. *Proc. Natl. Acad. Sci. U S A*, 103**,** 9452-9457.

Voit, E. O. (1992a) Symmetries of S-systems. *Math. Biosci.*, 109**,** 19-37.

Voit, E. O. (1992b) The S-distribution. A tool for approximation and classification of univariate, unimodal probability distributions. *Biometr. J.*, 34**,** 855-878.

Voit, E. O. (1992c) Optimization in integrated biochemical systems. *Biotechnol. Bioeng.*, 40**,** 572-582.

Voit, E. O. (1993) S-system modeling of complex systems with chaotic input. *Environmetrics*, 4**,** 153-186.

Voit, E. O. (1996) Dynamic trends in distributions. *Biometr. J.*, 38**,** 587-603.

Voit, E. O. (2000a) *Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists*. Cambridge University Press, Cambridge, UK, xii + 530 pp.

Voit, E. O. (2000b) A maximum likelihood estimator for the shape parameters of S-distributions. *Biometr. J.*, 42**,** 471-479.

Voit, E. O. (2003) Biochemical and genomic regulation of the trehalose cycle in yeast: review of observations and canonical model analysis. *J. Theor. Biol.*, 223**,** 55-78.

Voit, E. O. (2004) The Dawn of a New Era of Metabolic Systems Analysis. *Drug Discovery Today BioSilico*, 2**,** 182-189.

Voit, E. O. and Savageau, M. A. (1982a) Power-law approach to modeling biological systems; II. Application to ethanol production. *J. Ferment. Technol.*, 60**,** 229-232.

Voit, E. O. and Savageau, M. A. (1982b) Power-law approach to modeling biological systems; III. Methods of analysis. *J. Ferment. Technol.*, 60**,** 233-241.

Voit, E. O. and Yu, S. (1994) The S-distribution: Approximation of discrete distributions. *Biometr. J.*, 36**,** 205-219.

Voit, E. O. and Sorribas, A. (2000) Computer modeling of dynamically changing distributions of random variables. *Math. Comput. Modelling*, 31**,** 217-225.

Voit, E. O. and Schwacke, L. H. (2000) Random number generation from right-skewed, symmetric, and left-skewed distributions. *Risk Anal.*, 20**,** 59-71.

Voit, E. O. and Almeida, J. S. (2003) Dynamic profiling and canonical modeling: Powerful partners in metabolic pathway identification. In Goodacre, R. and Harrigan, G. G. (eds), *Metabolite Profiling: Its Role in Biomarker Discovery and Gene Function Analysis*, Kluwer Academic Publishing.

Voit, E. O. and Almeida, J. (2004) Decoupling dynamical systems for pathway identification from metabolic profiles. *Bioinformatics*, 20**,** 1670-1681.

Voit, E. O. and Schwacke, J. H. (2007) Understanding through modeling. In Konopka, A. K. Ed., *Systems Biology: Principles, Methods, and Concepts*, CRC Press/Taylor & Francis Books, 27-82.

Voit, E. O., Marino, S., and Lall, R. (2005) Challenges for the identification of biological systems from in vivo time series data. *In Silico Biol.*, 5**,** 83-92.

Voit, E. O., Almeida, J., Marino, S., Lall, R., Goel, G., Neves, A. R., and Santos, H. (2006b) Regulation of glycolysis in Lactococcus lactis: an unfinished systems biological case study. *IEE Proceedings Systems Biology*, 153**,** 286-298.

Wagner, A. and Fell, D. A. (2001) The small world inside large metabolic networks. *Proc. Biol. Sci.*, 268**,** 1803-1810.

Wanders, R. J. A., Vanroermund, C. W. T., and Meijer, A. J. (1984) Analysis of the control of citrulline synthesis in isolated rat-liver mitochondria. *Eur. J. Biochem.*, 142**,** 247-254.

Wang, F.-S., Ko, C.-L., and Voit, E. O. (2007) Kinetic modeling using S-systems and lin-log approaches. *Biochem. Eng. J.*, 33**,** 238-347.

Wang, F. S., Su, T. L., and Jang, H. J. (2001) Hybrid differential evolution for problems of kinetic parameter estimation and dynamic optimization of an ethanol fermentation process. *Industrial & Engineering Chemistry Research*, 40**,** 2876-2885.

Whittaker, E. T. (1923) On a new method of graduation. *Proceedings of the Edinburgh Mathematical Society*, 41**,** 63-75.

Xiu, Z. L., Chang, Z. Y., and Zeng, A. P. (2002) Nonlinear dynamics of regulation of bacterial trp operon: Model analysis of integrated effects of repression, feedback inhibition, and attenuation. *Biotechnol. Prog.*, 18**,** 686-693.

Yang, C., Hua, Q., and Shimizu, K. (2002) Quantitative analysis of intracellular metabolic fluxes using GC-MS and two-dimensional NMR spectroscopy. *J. Biosci. Bioeng.*, 93**,** 78-87.

Yeung, M. K., Tegner, J., and Collins, J. J. (2002) Reverse engineering gene networks using singular value decomposition and robust regression. *Proc. Natl. Acad. Sci. U S A*, 99**,** 6163-6168.

Yu, S. S. and Voit, E. O. (1996) A graphical classification of survival distributions. In Jewell, N. P., Kimber, A. C., Lee, M.-L. T., and Whitmore, G. A. (eds), *Lifetime Data: Models in Reliability and Survival Analysis*, Kluwer Academic Publishers, 385-392.

Zuñiga, P. C., Pasia, J., Adorna, H., del Rosario, R. C. H., and Naval, P. (2008) An ant colony optimization algorithm for parameter estimation and network inference problems in S-system models. *International Conference on Molecular Systems Biology 2008 (ICMSB08)*, Manila, Philippines, 105-106.

VITA

I-CHUN CHOU

I-Chun Chou was born in Taipei, Taiwan. She received a B.S. in Pharmacy and an M.S. degree in Clinical Pharmacy from National Taiwan University, Taipei. Accompanying her classroom work she completed a series of internships in clinical pharmacy at National Taiwan University Hospital and wrote a Master's thesis about β_2-adrenergic and IgE receptor polymorphisms in the susceptibility and therapeutic responses in childhood asthma in Taiwan. During this period she began to realize that medical or pharmaceutical problems should be approached in a more systematically way than was traditionally common. After graduation, she entered the Center for Drug Evaluation (CDE) in Taipei as a project manager. During her work at CDE she developed a comprehensive perspective of the processes governing the pharmaceutical industry, from drug development and trial evaluation to regulatory review and post marketing surveillance. Her experiences during this time gradually led to the idea of applying computational techniques to pharmaceutical research. Therefore, in 2001 she decided to enroll in a Master's program in System and Information Science at Syracuse University in Syracuse, NY. Entering a new field of endeavor in a different culture was initially a challenge but turned out to be very fruitful as she experienced the potential of informatics as well as open-mindedness toward people from diverse backgrounds and the value of teamwork and cooperation. After receiving her second M.S. degree, she entered the Georgia Institute of Technology in 2003 to pursue a doctorate in Bioinformatics with the goal of learning how to apply computational methods to biomedical questions. She joined Dr. Voit's lab in the Spring of 2005. During her doctoral studies, she published several papers on novel algorithms for parameter estimation and structure identification in complex biological systems. She also presented her work at various conferences worldwide, including two International Conferences on Molecular Systems Biology (ICMSB 2006 and 2008), the International Georgia Tech-ORNL Bioinformatics Conference 2007, Symposium on Cellular Systems Biology (SCSB 2008), and the annual conference on Data Mining of the Society of Industrial and Applied Mathematics (SIAM 2008).