Parameter estimation in systems biology models using spline approximation

Zhan, Choujun; Yeung, Lam F.

Published in:
BMC Systems Biology

Published: 24/01/2011

Document Version:
Final Published version, also known as Publisher's PDF, Publisher's Final version or Version of Record

Published version (DOI):
10.1186/1752-0509-5-14

Publication details:
Zhan, C., & Yeung, L. F. (2011). Parameter estimation in systems biology models using spline approximation. *BMC Systems Biology*, *5*, [14]. https://doi.org/10.1186/1752-0509-5-14

BMC Systems Biology

METHODOLOGY ARTICLE Open Access

Parameter estimation in systems biology models using spline approximation

Choujun Zhan*, Lam F Yeung*

Abstract

Background: Mathematical models for revealing the dynamics and interactions properties of biological systems play an important role in computational systems biology. The inference of model parameter values from time-course data can be considered as a "reverse engineering" process and is still one of the most challenging tasks. Many parameter estimation methods have been developed but none of these methods is effective for all cases and can overwhelm all other approaches. Instead, various methods have their advantages and disadvantages. It is worth to develop parameter estimation methods which are robust against noise, efficient in computation and flexible enough to meet different constraints.

Results: Two parameter estimation methods of combining spline theory with Linear Programming (LP) and Nonlinear Programming (NLP) are developed. These methods remove the need for ODE solvers during the identification process. Our analysis shows that the augmented cost function surfaces used in the two proposed methods are smoother; which can ease the optima searching process and hence enhance the robustness and speed of the search algorithm. Moreover, the cores of our algorithms are LP and NLP based, which are flexible and consequently additional constraints can be embedded/removed easily. Eight system biology models are used for testing the proposed approaches. Our results confirm that the proposed methods are both efficient and robust.

Conclusions: The proposed approaches have general application to identify unknown parameter values of a wide range of systems biology models.

Background

In recent years, the rapid development of sophisticated experiment tools in molecular biology allows the acquisition of high qualitative time series data which can significantly improve the ability of revealing the complex dynamics and interactions of biological systems. Profiting from the rapid technological advances, more and more researchers from different disciplines can now utilize such observation data to establish mechanism-based models which can incorporate every possible detail and functioning of biological systems [1]. One common approach is to characterize the biological system with a set of Ordinary Differential Equations (ODEs) [2-7]. Generally, there are two major aspects of building an ODE model for a biological system from experimentally measured time series: (1) to determine the structure of the system through a set of suitable ODEs with unknown parameters; (2) to determine the unknown parameters of this ODE model. The identification of these unknown parameter with fixed model structure from observations is one of the central issues of computational systems biology [8]. This type of approach can be considered as a "reverse engineering process" [9-11].

The parameter estimation problem is generally formulated as an optimization problem that minimizes an objective function which represents the fitness of the model with respect to a set of experimental data [8,12-17]. Two major optimization approaches are commonly adopted; the gradient-based nonlinear optimization method and the evolutionary based method. Also, simulations had shown that the simulated annealing (SA) method can offer promising results [18]. In [19], many deterministic and stochastic global optimization (GO) methods for parameter estimation were further compared using a three-step pathway model with noise free data assumption; the best result was given by the Stochastic Ranking Evolution Strategy (SRES) method.

* Correspondence: zchoujun2@student.cityu.edu.hk; eelyeung@cityu.edu.hk
Department of Electronic Engineering, City University of Hong Kong, Hong Kong, PR China

It is worth mentioning that, due to its simplicity in implementation, evolutionary algorithms, such as genetic algorithm and their variants, are extensively utilized for identifying unknown parameters of systems biology models [1,11,20-22]. However, most of these aforementioned approaches need a numerical ODE solver to perform the numerical integration for the underlining differential equations. Studies have revealed that more than 90% of the computation time is consumed in the ODE solver during the identification process [19]. In particular, for nonlinear dynamical systems with high-parameter-dimension, one trial usually consumes tens of hours or even days [10,20,21,23]. Furthermore, the convergence property is aggravated by numerical integration failure, which is a major problem in the optimization process [11]. The computational burden can be relieved by reformulating the system involving differential equations into a system of algebraic equations [12,15,17,24], which can be classified as "decomposition approaches". These decomposition approaches are widely employed in the parameter estimation of S-systems [7,25]. The reliability of the decomposed methods depends on the accuracy of the "smooth" estimated derivatives and the states of the system. In practice, these data are subject to significant observation noise. Without proper pre-processing, the estimation faces the potential of the overfitting problem and hence the estimation can deviate badly from the "true" value [26,27]. Regularization can be considered as a mathematical pre-processing on the measured noisy data set and be used to control the trade-off between the "roughness" of the solution and the infidelity of the data [28]. Since we are dealing with a known structured bio-system, the system model itself possesses a physical inertia and can serve as physical constraints which limit the system states within a set of possible trajectories. In this paper, the over-fitting problem can be relieved by embedding the model dynamics, the mass and energy balance constraints into our constrained optimization algorithms. Owing to the nonlinearity of systems biology models, the cost function to be minimized is complex and has multiple local minima. Minimization algorithms face the high possibility of getting trapped at local optima. For these reasons, the parameter estimation problem is still a bottleneck and a challenging task of computational analysis of systems biology [1,11]. Until now, none of the parameter estimation methods is effective in all cases and can overwhelm all the other methods. Instead, various methods have their advantages and disadvantages. Consequently, it is worthy to develop acceptably "good enough" methods within a given tolerance and time frame.

For practical purpose, some essential issues should be taken into account when developing a parameter estimation method: first, the method must be "efficient" enough that a trial can be completed within a reasonable computation time; second, for biological systems, the observation data is often corrupted by high level of noise, which complicates the objective function surface and introduces unwanted additional local minima in the search space [29]. Hence, the approach should be robust subject to noise; third, it needs to be flexible enough for adding/removing physical constraints, such as model dynamics, the mass and energy balance constraints. Furthermore, the representative cost function should have less local minima so as to ease the optimization algorithm in converging to the global minima. In this paper, two parameter estimation methods of combining spline theory [28] with Linear Programming (LP) and Nonlinear Programming (NLP) are developed, respectively. These methods remove the need for an ODE solver. Our analysis exhibits that the cost function surfaces of the two proposed methods are smooth.

Moreover, the cores of our algorithms are LP and NLP based, which are very flexible and hence additional constraints can be embeded/removed easily. Eight systems biology models were used to test the proposed algorithms. Experimental results show that the proposed methods are both efficient and robust (see additional file 1 for details).

This paper is organized as follows: The preliminary problem formulation is given and the bottleneck of the problem is highlighted in the next section. Then, two parameter estimation methods surmounting those bottlenecks are presented. In section 3, two trials are given, a simple enzyme kinetic model and the mammalian G1/S transition network model, in order to illustrate the robustness and the effectiveness of these two proposed methods (more models and trial results are given in additional file 1 and 2). Finally, conclusions and discussions are given in section 4.

Methods

Parameter estimation problem of systems biology models

Biological pathway dynamics can be modelled by the following continuous ODEs:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), \theta), \qquad x(t_0) = x_0, \\ y(t) &= g(x(t)) + \eta(t), \end{aligned} \tag{1}$$

where $x \in R^n$ is the system's state vector (for example the concentrations of a process), $\theta \in R^k$ is the system's parameter vector (for instance, the reaction rates), $u(t) \in R^p$ is system's input, $y \in R^m$ denotes the measured data subject to a Gaussian white noise $\eta(t) \sim N(0, \sigma^2)$, and x_0 is the initial state. $f(\cdot)$ is a set of nonlinear transition functions describing the dynamical properties of a biological system. Here, $g(\cdot)$ represents a measurement function. If all the states can be measured, the observer $g(\cdot)$ becomes an identity matrix. Otherwise, $g(\cdot)$ usually is a

rectangular zero-one matrix with corresponding rows deleted (represent the immeasurable states) from the identity matrix I_n.

The parameter estimation problem of nonlinear dynamical systems described in (1) can be formulated as a nonlinear programming problem (NLP) P_0 with differential-algebraic constraints:

$$P_0 : \min_{\hat{\theta},\hat{x}_0} \sum_{j=0}^{N-1}\sum_{i=1}^{n} w_{ij} \left\| y_i(t_j) - \hat{y}_i(t_j \,|\, \hat{\theta}) \right\|_l ,$$

$$s.t. \quad \begin{cases} \text{(i)} & \dot{\hat{x}}(t_j) = f(\hat{x}(t_j \,|\, \hat{\theta}), u(t), \hat{\theta}), \quad x(t_0) = \hat{x}_0, \\ \text{(ii)} & \hat{y}(t_j) = g(\hat{x}(t_j \,|\, \hat{\theta})), \qquad j = 1, 2, \cdots, N-1, \\ \text{(iii)} & C_{eq}(\hat{x}, \dot{\hat{x}}, \hat{\theta}) = 0, \\ \text{(iv)} & c_{ineq}(\hat{x}, \dot{\hat{x}}, \hat{\theta}) \le 0, \\ \text{(v)} & \theta_L \le \hat{\theta} \le \theta_U. \end{cases} \quad (2)$$

P_0 minimizes a cost function that measures the fitness of the model with respect to a given set of experiment data subjecting to a set of constraints, where $\hat{\theta} \in R^k$ is the set of parameters to be estimated, $||\cdot||_l$ denotes the l-norm with $l > 0$, $\hat{x}_0$ is the estimated initial condition, $\hat{x} \in R^k$ is the estimated system states ($\hat{x}(t_j \,|\, \hat{\theta})$ represents the estimated variable at time t_j with parameter $\hat{\theta}$ and initial condition $\hat{x}_0$), w_{ij} are the weighting coefficients, $\hat{y}$ is the estimated measured data. In some applications, additional constraints are introduced to impose special structural properties of a given system; they can be implemented in the form of the equality and inequality constraints C_{eq} and C_{ineq} (for instance the system performance and the mass balance constraints). Finally, θ_L and θ_U are simple structural constraints such as the parameter's upper/lower bounds (they can be part of the C_{ineq}).

For the NLP-P_0, the direct optimization methods, such as Newton type methods and many GO methods, require solving the nonlinear dynamic model (1) for $\hat{x}$ in order to compute the cost function. The common method to estimate $\dot{\hat{x}}(t_i)$ and $\hat{x}(t_i)$ is using ODE solvers, which perform the numerical integration with $\hat{\theta}$ fixed at each iteration [19]. During the process of identification, the integration has to be executed thousands, even millions of times. That is the main reason more than 90% of the time is consumed in the ODE solver [24] and the computation time spent on the P_0 can be hours even days [10,20]. Moreover, P_0 is a nonlinear optimization problem subjecting to a set of linear and non-linear differential equation constraints. Hence, P_0 is often multimodal (non-convex) and has many local minima. In a high-noise environment, the situation becomes more difficult. Consequently, P_0 requires further manipulation in order to reduce the complexity so as to relieve the computation burden and also to avoid being trapped in local minima.

Instead of using ODE solvers to estimate $x(t)$ and $\dot{x}(t)$, one can utilize spline approximation. Given L real values τ_i, called knots, with $\tau_0 \le \tau_1 \le \cdots \le \tau_{L-1}$. Using the Cox-de Boor recursion formula, the B-spline basis of degree $n_d = 0, 1, 2, \cdots, L - 2$ can be defined as follows:

$$b_{m,0}(t) \triangleq \begin{cases} 1 & \text{if} \quad \tau_m \le t < \tau_{m+1}, \quad m = 0, 1, \cdots, L-2, \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

$$b_{m,n_d}(t) \triangleq \frac{t - \tau_m}{\tau_{m+n_d} - \tau_m} b_{m,n_d-1}(t) + \frac{\tau_{m+n_d+1} - t}{\tau_{m+n_d+1} - \tau_{m+1}} b_{m+1,n_d-1}(t), \quad m = 0, 1, \cdots, L - n_d - 2. \quad (4)$$

Let $b_i(t) = [b_{1,n_d}(t), b_{2,n_d}(t), \cdots, b_{L_i - n_d - 2, n_d}(t)]^T$, a vector of length L_i - n_d - 2, be the chosen basis functions. Then, the estimated variable $\hat{x}$ can be expressed in terms of the basis function expansion [28]

$$\hat{x}_i(t) = \sum_{m=1}^{L_i - n_b - 2} p_{i,m} \cdot b_{i,n_b}(t), \quad t \in [t_0 \;\; t_N] \quad (5)$$

where $\hat{x}_i \in R$ is the estimation of the ith state of (1), $p_{i,\,m}$ is the weighting coefficient. Let $p_i = [p_{i,0}, p_{i,1}, \cdots, p_{i,L_j - n_d - 2}]^T$, (5) can be rewritten in matrix form

$$\hat{x}_i(t) = b_i^T(t) \cdot p_i, \quad i = 1, 2, \cdots, n. \quad (6)$$

Similarly, the estimated $\dot{\hat{x}}_i(t)$ can be approximated by

$$\dot{\hat{x}}_i(t) = \dot{b}_i^T(t) \cdot p_i, \quad (7)$$

where $\dot{b}_j(t) = [\dot{b}_{0,n_d}(t), \dot{b}_{0,n_d}(t), \cdots, \dot{b}_{L_j - n_d - 2, n_d}(t)]^T$ is the set of the derivatives of the basis functions. There are various types of splines suitable for this application, such as cubic spline, B-spline, uniform spline, nonuniform spline and interpolating spline. For more detail information about spline approximation theory, please refer to chapter (IX, XI, and XIV) in [28]. As B-spline is simple in formation and efficient for computation, it is adopted here. Our extensive tests have shown that uniform B-spline basis with $\frac{N}{3} \le L_i \le \frac{N}{2}$ produces good results. Hence, in this paper, unless otherwise indicated, the uniform B-spline basis with $L_i \approx \frac{N}{3}$ was used in the parameter identification process.

Next, two techniques based on spline for parameter estimation will be proposed: one is based on linear programming (LP) which is very efficient and can cover many special structured systems and the other one is based on NLP which is flexible and can cater for general system structures.

The LP Approach

In many bio-system models, $f(x, \theta)$ is autonomous system and linear in θ as follows:

$$\dot{x}(t) = \Phi(x(t))\theta, \qquad x(t_0) = x_0. \tag{8}$$

where $\Phi(x) \in R^{n \times k}$ is a matrix and its elements are a function of the state x. Systems with structure (8) covers a large set of systems biology models, such as enzyme kinetic pathway model, RKIP pathway model, IκB-NF-κB model TNFα-Mediated NF-κB-signaling pathway model, irreversible inhibition of HIV proteinase model, Laub and Loomis model [2-4,30]. In addition, these types of models are usually subject to the mass balance constraints which can be incorporated into the LP easily (It is demonstrated in the results section via the Enzyme kinetic model).

For noisy data, good smoothing approximation can be achieved by minimizing the following cost function [26]

$$C_i = \frac{1}{N+1}\sum_{j=0}^{N}(y_i(t_j) - x_i(t_j))^2 + \lambda\sum_{j=0}^{N}(x_i^{(m)}(t_j))^2 \tag{9}$$

where $\hat{x}_i^{(m)}$ represents the mth derivative of $\hat{x}_i$, $m \in Z^+$ and $\lambda \geq 0$ control the trade-off between the "roughness" of the solution and hence can be used to relieve the overfitting problem. If the equality C_{eq} represents the mass balance constraint and the inequality constraint C_{ineq} represents parameter values' lower/upper bounds, the B-coefficient vector $p = [p_1; p_2; \cdots p_n]^T$ can be computed by solving the following quadratic programming sub-problem A_1 :

$$\mathrm{A}_1: \quad \min_{p}\left\{\frac{1}{N+1}\sum_{j=0}^{N}||\,y(t_j) - \hat{x}(t_j)\,||_2^2 + \lambda\sum_{j=0}^{N}||\,\hat{x}^{(m)}(t_j)\,||_2^2\right\}$$
$$s.t. \quad \begin{cases} \text{(i)} & A_{eq}\cdot\hat{x} = b_{eq}, \\ \text{(ii)} & \hat{x}_i(t_j) = b_i^T(t_j)\cdot p_i, \\ \text{(iii)} & \hat{x}_i^{(m)}(t_j) = (b_i^{(m)})^T(t_j)\cdot p_i, \\ & i, = 1, 2, \cdots n,\ j = 0, 1, 2\cdots N. \end{cases} \tag{10}$$

$A_{eq}\cdot\hat{x} = b_{eq}$ stands for the equality constraints $C_{eq}(\hat{x}, \hat{\theta}) = 0$. A_{eq} is a constant matrix, b_{eq} is a constant vector. It is found empirically that $m = 2$ and $0 \leq \lambda \leq 0.05$ produce relatively good results. Hence, the parameters $m = 2$, and $\lambda = \{0, 0.01, 0.03\}$ corresponding to a noise level {0%, 5%, 10%}, respectively, were used in this study. Then, the "smooth" estimated state $\hat{x}$ can be generated by the B-spline approximation (5).

Replace $x(t)$ by the estimated state $\hat{x}(t)$ and integrate (8) yield:

$$\tilde{x}(t_j) = \left(\int_{t=t_0}^{t_j}\Phi(\hat{x}(t))dt\right)\cdot\hat{\theta} + \hat{x}_0 = \hat{\Psi}_j\hat{\theta} + \hat{x}_0. \tag{11}$$

where $\hat{\Psi}_j = \int_{t=t_0}^{t_j}\Phi(\hat{x}(t))dt$ represents the transition matrix. Then, P_0 can be reformulated into the following optimization problem:

$$P_1: \min_{\hat{\theta}\in R^k}\sum_{j=0}^{N}\sum_{i=1}^{n} w_{ij}\,|\,\gamma_i(t_j) - \tilde{x}_i(t_j)\,|$$
$$s.t. \quad \begin{cases} \text{(i)} & \tilde{x}(\hat{\theta}, t_j) - \hat{\Psi}_j\hat{\theta} + \hat{x}_0, \\ \text{(ii)} & \theta_L \leq \hat{\theta} \leq \theta_U, \\ & i = 1, .., n; \quad j = 0, 1, 2, \cdots, N. \end{cases} \tag{12}$$

Here, $w_{ij} \geq 0$ is weighting factor. Note that the L_1-norm of a variable × is equivalent to the following relation: $|\,x\,| = \min_{\alpha\geq 0}\{\alpha\}$; $s.t.$ $-\alpha \leq x \leq \alpha$. Then P_1 can be transformed into the following augmented optimization problem by introducing the slack variables α as follows:

$$P_2: \min_{\hat{\theta},\alpha}\left\{\sum_{i,j} w_{ij}\cdot\alpha_{ij}\right\}$$
$$s.t. \quad \begin{cases} \text{(i)} & -\alpha_{ij} \leq \gamma_i(\hat{\theta}, t_j) - \tilde{x}_i(t_j), \\ \text{(ii)} & \alpha_{ij} \geq \gamma_i(\hat{\theta}, t_j) - \tilde{x}_i(t_j), \\ \text{(iii)} & \tilde{x}(\theta, t_j) = \hat{\Psi}_j\hat{\theta} + \hat{x}_0 \\ \text{(iv)} & \alpha_{ij} \geq 0. \\ \text{(v)} & \theta_L \leq \hat{\theta} \leq \theta_U, \\ & i = 1, .., n; \quad j = 0, 1, ..., N. \end{cases} \tag{13}$$

It is a Linear Programming (LP) problem with variable $\{\alpha, \theta\}$, which is a convex problem with a wealth of fast and efficient routines available [31,32].

Combine spline theory and NLP

To deal with systems biology models, of which the states and parameters are separable, the LP approach is suitable and efficient. In contrast, if the model does not belong to this category, such as the mammalian G1/S transition model and S-system model, the aforementioned approach cannot apply. Thus, a more general approach will be introduced. Recalling (6) and (7), the estimation of $\hat{x}(t_j)$ and $\dot{\hat{x}}(t_j)$ can be constructed by a

set of basis functions. We can replace $\hat{x}(t_j)$ and $\hat{\dot{x}}(t_j)$ in (i-iv) of P_0 with (6) and (7). With little change, P_0 can be reformulated as

$$\mathrm{P}_3: \min_{\hat{\theta},p} \sum_{j=0}^{N} (y(t_j) - \hat{y}(t_j \mid \hat{\theta}))^T w_j (y(t_j) - \hat{y}(t_j \mid \hat{\theta}))$$

$$s.t. \begin{cases} \text{(i)} & \hat{\dot{x}}_i(t_j) = b_i^T(t_j) \cdot p_i, \\ \text{(ii)} & \hat{\dot{x}}_i(t_j) = b_i^T(t_j) \cdot p_i, \\ \text{(iii)} & \left\| \hat{\dot{x}}(t_j) - f(\hat{x}(t_j), u(t), \hat{\theta}) \right\|_2^2 = 0, \\ \text{(iv)} & \hat{y}(t_i) = g(\hat{x}(t_j)) \\ \text{(v)} & C_{eq}(\hat{x}(t_j), \hat{\dot{x}}(t_j), \hat{\theta}) = 0, \\ \text{(vi)} & C_{ineq}(\hat{x}(t_j), \hat{\dot{x}}(t_j), \hat{\theta}) \le 0, \\ \text{(vii)} & \theta_L \le \hat{\theta} \le \theta_U, \\ & i = 1, 2, \cdots, n, \quad j = 0, 1, \cdots, N. \end{cases} \tag{14}$$

Note that the constraint-(i) of (2) has been replaced by constraints (i)-(iii) of P_3. Then, NLP-P_0-(2) with differential-algebraic constraints turns into NLP-P_3-(14) with only algebraic equation constraints. Hence, P_3 does not require ODE solvers, which eases the computation burden (as shown in Examples). In contrast to the the decomposition methods [12,15], which divide the estimation of the system states (and its derivative) and the parameter estimation into two separate steps, P_3 computes the estimated states (and its derivative) and parameter values at the same time. Note that constraint (iii) of P_3 governs the estimated state (and its derivative) so as to ensure these estimates belong to the trajectory $\hat{x}(t \mid \hat{\theta})$, which is a solution of system (1). Thus, the system model itself serves as a filter performing regularization. Hence, the overfitting problem can be relieved (see additional file 2 for details).

For a non-linear system, the Lagrangian of (2), $L(\hat{\theta}, \hat{X}_0)$, is an implicit function of $\{\hat{\theta}, \hat{x}_0\}$ [31]. However, many traditional optimization algorithms require the derivative $\partial L / \partial \hat{\theta}$ during the optimization process. As $L(\hat{\theta}, \hat{x}_0)$ is an implicit function of $\hat{\theta}$, $\partial L / \partial \hat{\theta}$ cannot be obtained directly, but has to be computed via approximation methods [16], which makes the algorithm unreliable. For P_3, the Lagrangian $L(\hat{\theta}, p)$ now consists of simple algebraic constraints. Thus, $\partial L / \partial \hat{\theta}$ and $\partial L / \partial p$ are explicit functions of $\hat{\theta}$ and p. In conclusion, many of the aforementioned difficulties can be reduced. P_3 can be solved by a number of optimization approaches; either via evolution type algorithms, such as genetic algorithm (GA), simulated annealing (SA) and etc, or via traditional NLP algorithms, such as sequential quadratic programming(SQP), sequential penalty function, the trust region approach and etc [33,34].

Results

Two biological system models, a simple enzyme kinetic model and the mammalian G1/S transition network model, are chosen as benchmarks for evaluating the performance of P_2 and P_3 respectively.

Enzyme kinetic model

Consider the well-known simplified enzyme kinetic model. E is the concentration of an enzyme that combines with a substrate S to form an enzyme-substrate complex ES with a rate constant k_1. The complex ES holds two possible out comes in the next step. It can be dissociated into E and S with a rate constant k_2, or it can further proceed to form a product P with a rate constant k_3. It is assumed that none of the products reverts to the initial substrate. These relations can be represented by the following set of ODEs.

$$\begin{aligned} \frac{dS(t)}{dt} &= -k_1 \cdot E(t) \cdot S(t) + k_2 \cdot ES(t), \\ \frac{dE(t)}{dt} &= -k_1 \cdot E(t) \cdot S(t) + (k_2 + k_3) \cdot ES(t), \\ \frac{dES(t)}{dt} &= k_1 \cdot E(t) \cdot S(t) - (k_2 + k_3) \cdot ES(t) \\ \frac{dP(t)}{dt} &= k_3 \cdot ES(t), \end{aligned} \tag{15}$$

where k_1, k_2, k_3, are the system unknown parameters. Let $x_1(t)$, $x_2(t)$, $x_3(t)$and $x_4(t)$ represent $S(t)$,$E(t)$, $ES(t)$ and $P(t)$ respectively. Then, the above equation can be rewritten into

$$\begin{aligned} \dot{x}_1 &= -k_1 x_1 x_2 + k_2 x_3, \\ \dot{x}_2 &= -k_1 x_1 x_2 + k_2 x_3 + k_3 x_3, \\ \dot{x}_3 &= k_1 x_1 x_2 - k_2 x_3 - k_3 x_3, \\ \dot{x}_4 &= k_3 x_3, \\ \Rightarrow \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} &= \begin{bmatrix} -x_1 x_2 & x_3 & 0 \\ -x_1 x_2 & x_3 & x_3 \\ x_1 x_2 & -x_3 & -x_3 \\ 0 & 0 & x_3 \end{bmatrix} \cdot \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} \end{aligned} \tag{16}$$

Then, the mass balance constraint becomes:

$$\begin{cases} x_2(t) + x_3(t) = x_2(t_0) + x_3(t_0) \\ x_1(t) + x_3(t) + x_4(t) = x_1(t_0) + x_3(t_0) + x_4(t_0) \end{cases} \tag{17}$$

Or in matrix form, we have

$$A_{eq} \cdot x = \mathrm{b}_{eq} \tag{18}$$

where $A_{eq} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}$ and $b_{eq} = \begin{bmatrix} x_2(t_0) + x_3(t_0) \\ x_1(t_0) + x_3(t_0) + x_4(t_0) \end{bmatrix}$.

According to (16), we have

$$\Phi(x(t)) = \begin{bmatrix} -x_1x_2 & x_3 & 0 \\ -x_1x_2 & x_3 & x_3 \\ x_1x_2 & -x_3 & -x_3 \\ 0 & 0 & x_3 \end{bmatrix}, \theta = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix},$$

$$\text{and} \quad \hat{\Psi}(\hat{x}(t_i)) = \int_{t=t_0}^{t_i} \begin{bmatrix} -\hat{x}_1\hat{x}_2 & \hat{x}_3 & 0 \\ -\hat{x}_1\hat{x}_2 & \hat{x}_3 & \hat{x}_3 \\ \hat{x}_1\hat{x}_2 & -\hat{x}_3 & -\hat{x}_3 \\ 0 & 0 & \hat{x}_3 \end{bmatrix} dt. \tag{19}$$

An artificial data set with four time courses was created. A total of 40 sampling points were assigned on each time courses. The observation data was perturbed by a zero mean Gaussian white noise $\eta(t) \sim \mathrm{N}(0, \sigma^2)$ in order to simulate the observation error. The estimated state $\hat{x}$ was computed by solving the quadratic programming sub-problem A_1. The unknown parameter values were estimated using P_2. The searching region of the parameter values was $[0, +\infty)$. All the computations were performed on a Pentium Dual Core computer (2.13 GHz ×2) with 2 GB RAM. The algorithm was implemented with Matlab-7 using the interior point algorithm. To quantify the fitness of the estimated model, the following relative squared error (RSE) measure J is employed:

$$J = \frac{1}{N \cdot n} \sum_{i=1}^{n} \sum_{j=0}^{N} \left(\frac{\hat{x}_i(t_j) - x_i(t_j)}{x_i(t_j)} \right)^2, \tag{20}$$

where $\hat{x}_i(t_j)$ is the estimated time-course at time t_j of a state variable x_i, and $x_i(t_j)$ represents the "true" time-courses without noise at time t_j. Note that smaller RSE J reflects better estimation. In order to obtain a statistical result on the quality of the estimation, 5,000 trials were performed. At each trial an estimated $\hat{\theta}$ is computed using P_2. Then, the mean estimation and standard deviation were deduced. The computation was very efficient and only took a few seconds for one estimation trial.

Table 1 shows the statistical results of the estimation. It reveals that all the system parameter values are estimated successfully with a relative error of around 0.01% in noise free condition. When the system is subjected to a 10% observation noise level, all the mean estimated parameter values are within a relative tolerance, better than 2%. The RSE between the time-courses, produced by inferred model, and the given time-series data, averaged smaller than 1% even subject to 10% observation noise level condition. For this reason, we have confidence that the proposed method is robust within ±10% noise ratio (more complicated models and trials on P_2 can be found in part I of additional file 1). Figure 1 shows the responses of one trial.

The mammalian G1/S transition network model

Next, the mammalian G1/S transition network model, which includes a set of proteins and regulatory gene network, is used to test P_3. In the mammalian G1/S transition network, pRB and AP-1 are the tumor suppressor from the family of pocket proteins and the family of transcription factors that mediate mitogenic signals, E2F1 is the transcription factor targeting genes that regulate cell cycle progression, Cyclin D/cdk4,6, cyclin E/cdk2, complexes characterizing the G1- and S- phases. There are various positive and negative feedback loops in the network controlling the G1/S transition. The positive feedback regulation of E2F1 and a double activator-inhibitor module can lead to bistability. The double activator-inhibitor module of the antagonistic plays E2F/DP on pRB make up the key unit of this phase transition. The graph representation of the mammalian G1/S transition network model can be found in additional file 1 and more details can refer to Swat et al. [5]. Definition of Variables for G1/S Transition Model is shown in Table 2. The corresponding ODE model is as follows

Table 1 Statistical results of parameter estimation of enzyme kinetic model

	Nominal Value	Mean estimation ± standard deviation		
		Noise level: 0%	Noise level: 5%	Noise level: 10%
k_1	0.18	0.1796 ± 33.5081e-6	0.1794 ± 0.0008	0.1808 ± 0.0025
k_2	0.20	0.1993 ± 108.4891e-6	0.1968 ± 0.0031	0.1963 ± 0.0106
k_3	0.23	0.2300 ± 1.5946e-6	0.2326 ± 0.0001	0.2345 ± 0.0004
J		6.8620e-8 ± 7.7337e-8	1.0996e-4 ± 9.3270e-5	3.0328e-4 ± 3.2160e-4

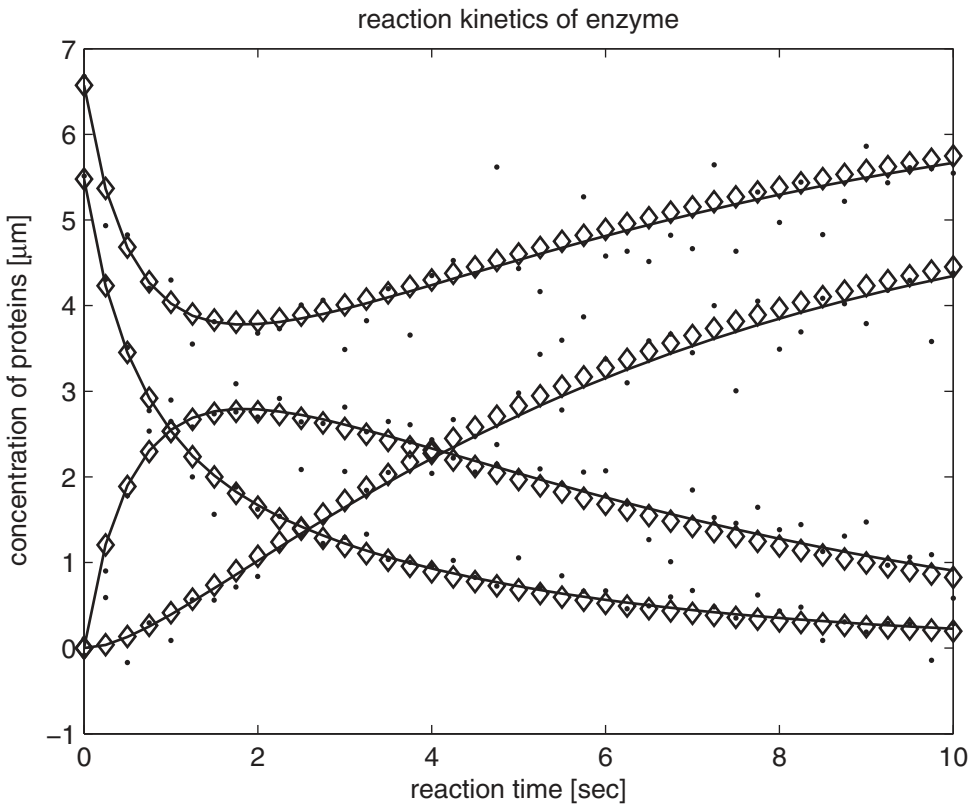

Figure 1 The dynamic profiles of a trial with observation data subject to 10% random noises. Solid lines represent the "true" time-series data without noise, dots represent the measured time-series data with added artificial noise, and diamonds represent the estimated time-series data produced by the model.

$$
\begin{aligned}
\dot{x}_1 &= k_1 \frac{x_2}{K_{m1}+x_2}\frac{J_{11}}{J_{11}+x_1}\frac{J_{61}}{J_{61}+x_6} - k_{16}x_1x_4 + k_{61}x_6 - \phi_1 x_1,\\
\dot{x}_2 &= k_p + k_2 \frac{a_2^2 x_2^2}{K_{m2}^2+x_2^2}\frac{J_{12}}{J_{12}+x_1}\frac{J_{62}}{J_{62}+x_6} - \phi_2 x_2,\\
\dot{x}_3 &= k_3x_5 + k_{23}x_2\frac{J_{13}}{J_{13}+x_1}\frac{J_{63}}{J_{63}+x_6} + k_{43}x_4 - k_{34}x_3\frac{x_4}{K_{m4}+x_4}\\
&\quad - \phi_3 x_3,\\
\dot{x}_4 &= k_{34}x_3\frac{x_4}{K_{m4}+x_4} - k_{43}x_4 - \phi_4 x_4,\\
\dot{x}_5 &= F_m + k_{25}x_2\frac{J_{15}}{J_{15}+x_1}\frac{J_{65}}{J_{65}+x_6} - \phi_5 x_5,\\
\dot{x}_6 &= k_{16}x_1x_4 - k_{61}x_6 - k_{67}x_6x_9 + k_{76}x_7 - \phi_6 x_6,\\
\dot{x}_7 &= k_{67}x_6x_9 - k_{76}x_7 - \phi_7 x_7,\\
\dot{x}_8 &= k_{28}x_2\frac{J_{18}}{J_{18}+x_1}\frac{J_{68}}{J_{68}+x_6} + k_{98}x_9 - k_{89}x_8\frac{x_9}{K_{m9}+x_9} - \phi_8 x_8,\\
\dot{x}_9 &= k_{89}x_8\frac{x_9}{K_{m9}+x_9} - k_{98}x_9 - \phi_9 x_9,
\end{aligned}
\tag{21}
$$

Table 2 Definition of Variables for G1/S Transition Model

Symbol	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
Acronym	*pRB*	*E2F1*	$CycD_i$	$CycD_a$	*AP - 1*	pRB_p	pRB_pp	$CycE_i$	$CycE_a$

where x is the set of state variables. There are totally 9 states and 39 parameters. The nominal parameter values are shown in Table 3.

Here, P_3 was solved by the Stochastic Raking Evolution Strategy (SRES) algorithm [35]. The searching region of the parameters was [0, 50θ]. SRES uses stochastic ranking as the constraint handling technique, which adjusts the balance between the objective and penalty functions automatically during the evolutionary search. The observation data include 4 sets of time course, which consists of 40 sample points. For trials with noise free data, the algorithm converged in 8 ~ 9 hours after 250,000 ~ 300,000 iterations. The estimated parameter values, as shown in Table 3 are almost identical to the nominal parameter values. However, for k_{23}, k_{25} and J_{15}, the estimated values are far from the nominal values, but the RSE measure is almost zero, which possibly implies that the system is insensitive with the changes of k_{23}, k_{25} and J_{15}. This phenomenon reveals that the G1/S transition model either has some parameters that are insensitive to the chosen observation, or they are non-identifiable parameters [36,37]. It is worth

Table 3 Results of parameter estimation of the mammalian G1/S transition network model

Parameters	Nominal value	Estimated parameters			
		Noise level 0%	Noise level: 2.5%	Noise level: 5%	Noise level: 10%
k_1	1	0.9957	1.0150	1.1105	1.6037
k_2	1.6	1.5989	1.4138	1.5187	1.0315
k_3	0.05	0.0500	0.0528	0.0392	0.0381
k_{16}	0.4	0.4002	0.4440	0.3959	0.9331
k_{34}	0.04	0.0400	0.0414	0.0337	0.0215
k_{43}	0.01	0.0100	0.0142	0.0090	1.45e-10
k_{61}	0.3	0.2985	0.3432	0.2847	0.8185
k_{67}	0.7	0.6999	0.4535	1.3974	1.3108
k_{76}	0.1	0.0999	0.0457	0.2446	0.1845
k_{23}	0.3	0.1219	0.4134	0.6132	0.5579
k_{25}	0.9	0.1785	0.7063	0.8291	0.7874
k_{28}	0.06	0.0601	0.0669	0.0222	0.0198
k_{39}	0.07	0.0700	0.0549	0.0520	0.0334
k_{96}	0.01	0.0100	0.0441	0.0002	4.55e-14
a	0.04	0.0400	0.1257	0.1260	0.1265
J_{11}	0.5	0.4992	0.5612	0.4252	0.6523
J_{12}	5	5.0025	4.8940	4.6892	5.4021
J_{15}	0.001	0.0051	0.0011	0.0010	0.0011
J_{18}	0.6	0.5990	0.7253	0.8014	1.1290
J_{61}	5	5.2581	4.1474	6.4585	7.2003
J_{62}	8	8.0088	29.734	39.403	41.408
J_{65}	6	5.9222	8.7804	9.3474	7.8076
J_{68}	7	6.9916	31.979	25.125	36.795
J_{13}	0.002	0.0050	0.0013	0.0016	2.61e-14
J_{63}	2	1.9740	1.4726	0.4203	19.871
K_{m1}	0.5	0.4905	0.5267	0.5601	0.0410
K_{m2}	4	3.9985	4.0482	4.1061	3.8495
K_{m4}	0.3	0.2999	0.2838	0.2735	0.2338
K_{m9}	0.005	0.0054	3.69e-5	2.03e-5	3.88e-6
K_p	0.05	0.0499	0.0452	0.0496	0.0311
φ_1	0.005	0.0044	0.0057	0.0041	0.0073
φ_2	0.1	0.0999	0.0920	0.0983	0.0693
φ_3	0.023	0.0230	0.0261	0.0164	0.0152
φ_4	0.03	0.0300	0.0279	0.0253	0.0218
φ_5	0.01	0.0100	0.0098	0.0101	0.0101
φ_6	0.06	0.0606	0.0627	0.0608	0.1518
φ_7	0.04	0.0401	0.0436	0.0404	0.0788
φ_8	0.06	0.0600	0.1546	0.0024	0.0260
φ_9	0.05	0.0500	0.0025	0.0439	0.0276
J		7.5399e-6e	0.0005	0.0009	0.0025

mentioning that the this large computational effort is the consequence of the very tight convergence criteria, an almost equal good result can be reached within 200,000 generations in about 6.5 hours with the RSE measure J is smaller than 1%. Figure 2(a) shows the "true" time-series data without noise and computed dynamic time-series data from one identified model. When 10% random noises are added, the convergence time increased and the relative estimation errors between estimated parameters and nominal parameters increased with the increase of noise. However, the time-series produced by the estimated model is very similar to the original data, namely the RSE J is still small. This phenomenon may imply that there is no need to estimated every parameters accurately to achieve a model with equivalent dynamical properties with a good degree of accuracy. As the simulation time is long, performing thousands of simulations as the first method in order to evaluate the mean and variance of estimated parameters is impractical. Thus, due to the lack of space, results of just a few selected trial are shown in Table 3 (more trial results can be found in additional file 1).

Trials were performed using Matlab-7. The main reason to use Matlab is that it is a convenient environment to visualize all the information arising from the optimization runs of the solver, evaluate new algorithms and modify existing algorithms. In contrast to the convenience, it is worth mentioning that Matlab programs usually are one order of magnitude (10 times or more) slower than equivalent compiled Fortan or C codes [19]. This is the major drawbacks of carrying programs out with Matlab. However, even in this situation, the performance of the proposed methods is acceptable.

For fair comparison, we also used the SRES algorithm to solve the same parameter estimation problem in the same searching region, but using NLP-P_0 with differential algebraic constraints as cost function. In this condition, after running 1 day, the algorithm failed to produce a set of parameters that can produce reasonable simulation result. We further reduced the searching region to [0, 3θ] and used noise free data, but the estimation result was still not good and the RSE J is larger than 10.

Here, we use the G1/S model to show the differences of the cost function surfaces between NLP-P_0 and NLP-P_3: in this case, the cost function of P_0 is a highly irregular and complicated manifold with multiple local minima; the augmented cost function adopted in problem P_3 is a much "smoother" function and hence it is easier for the NLP algorithm to converge to the solution. In order to simplify the analysis for exposition purpose, we only vary parameters k_1 and k_2 over the range $k_1 \in [0; 2]$ and $k_2 \in [0, 3.2]$ and fix all other parameters at their nominal values. Figure 3(a) displays the cost function surface of P_0 , while Figure 3(b) exhibits the same data as Figure 3(a) on the expanded scale and Figure 3(c) is the corresponding contour plots. Figure 3(a) shows that the cost function surface of is a ridge, which drops suddenly from 10^9 to 0. However, Figure 3(b) reveals the cost function surface of P_0 are actually

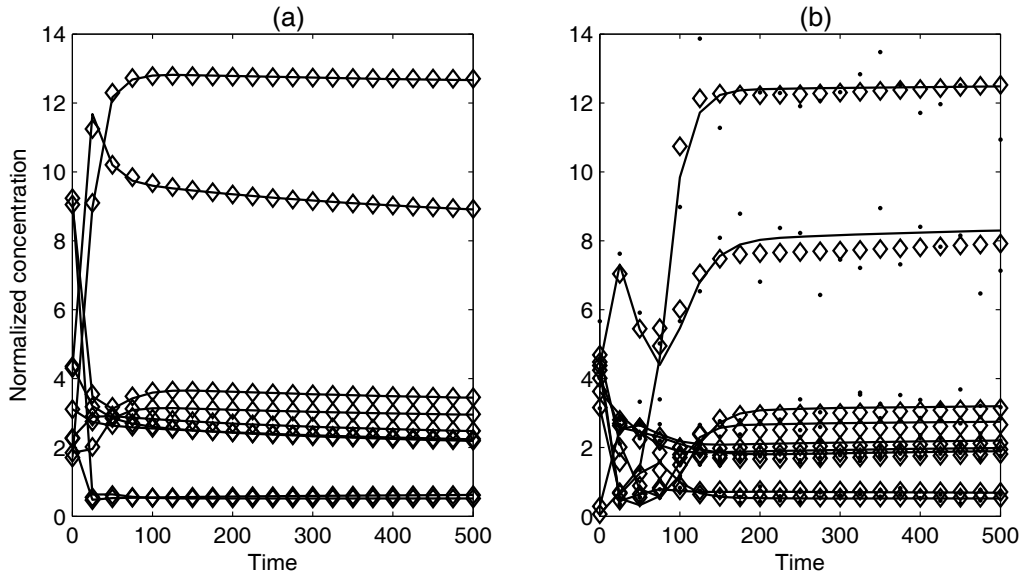

Figure 2 The dynamic profiles of two trials. Solid lines represent the "true" time-series data without noise, dots represent the measured time-series data with added artificial noise, and diamonds represent the estimated time-series data produced by the model: (a) noise free condition (b) 10% random noise condition.

banana-shaped valley around the nominal value of the fixed parameters, this unfavorable profile can slow down the convergence rate of the algorithm. Furthermore, there are many local minima in the banana-shaped valley. Some algorithms, such as simulated annealing, genetic algorithm, have been proposed to overcome these problem. However, these algorithms are all computationally demanding. In conclusion, these cost function features make the problem P_0 a severe challenge to every optimization algorithm.

With the same condition, Figure 4(a) displays the cost function surface of P_3, while Figure 4(b) shows the corresponding contour line. Compared with the cost function surface of P_0, the cost function surface of P_3 is

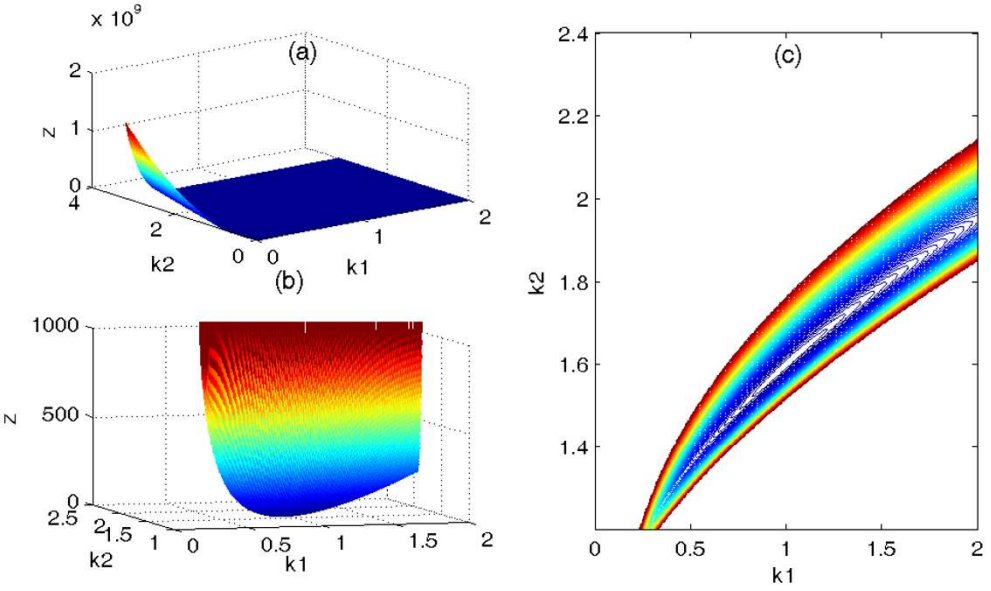

Figure 3 Cost function surface and contours. (Color online) (a) Cost function surface of the P_0 as parameters k_1 and k_2 are varied; (b) displays the same data as (a) on the expanded scale; (c) corresponding contours near the nominal parameter value.

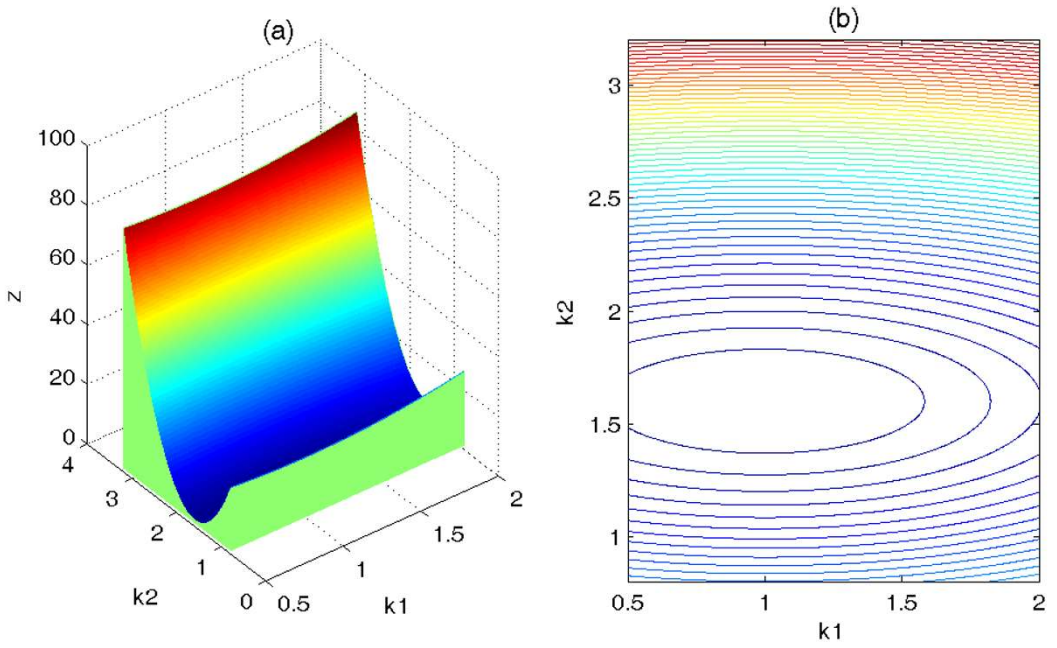

Figure 4 Cost function surface and contours. (Color online) Cost function surface of the P_3 as parameters k_1 and k_2 are varied; (b) corresponding contours of the cost function.

bowl-shaped, which is smoother. Similar results has also been observed when other combination of parameters served as variables. Obviously, if all 39 parameters vary at the same time, the surface of the cost function will be more "uneven" and more complicated. However, in this case, from the previous observations, the cost function surface of P_3 is smoother than the cost function surface of P_0.

Furthermore, P_3 only involves algebraic equations as objective function and constraints. These properties make the NLP-P_3 easier to solve.

Discussion and Conclusion

In this paper, two parameter estimation methods based on spline theory are proposed. One aims at a narrower class of systems which is linear in parameters; however, it can cover many commonly found biological systems. The benefit is that the estimation problem can be transformed in an LP sub-algorithm which are fast and robust. Additional linear constraints can be embedded relative easily. For general systems, the problem is solved by an NLP with algebraic constraints, which is more computationally demanding.

A simple enzyme kinetic model and the mammalian G1/S transition network model were used as benchmarks to evaluate the performance of the two proposed methods. We illustrate the usefulness with more examples in additional files 1 but these do not remotely cover all the conditions.

During the simulation of the mammalian G1/S transition network model, we found that the estimated parameter set $\Phi_A \equiv \{k_1, k_2, k_p, J_{11}, J_{12}, K_{m1}, K_{m2}, \varphi_1, \varphi_2\}$ were well within the respective nominal values. While the set $\Phi_B \equiv \{J_{61}, J_{62}, J_{63}, J_{65}, J_{68}\}$ were far from their nominal values. However, the time-series produced by the estimated model were very similar to the original data. This phenomena reveals that some parameter values are insensitive in the searching region. Interestingly, we find that the "sensitive" or "easily identified" parameters set Φ_A are also the parameters of the double-activator-inhibitor module of the antagonistic players E2F/DP and pRB, which makes up the core unit of the G1/S transition model [5]. This phenomenon may imply that the parameter values of the core module are sensitive and easy to identify. In contrast, the parameters set Φ_B seems to be insensitive, which may reflect that pRG_p (x_6) is not a key element of the total system. However, to identify which parameter values or variables are important, a sensitivity analysis is needed [38], which is another important topic in systems biology and deserves a more detailed study. This sensitivity analysis is a preprocess for isolating those states and parameters which are sensitive in order to reduce the dimension of the system model and to improve the numerical stability for the core estimation problem.

For most biological systems, the ODE models are often high-dimensional and nonlinear. The problem of system parameter estimation is computationally

expensive and can easily be trapped in local minima. We find that under noisy conditions, it is almost impossible to accurately estimate every parameter of the sloppy biological system model. However, in practice, a model with equivalent dynamical properties with a good degree of accuracy can be constructed based on dominant sensitive parameters and system states. The following are some of practical observations:

1. High quality experiment data is essential for identifying accurate biology systems. When the experiment data is corrupted with high level noise, it needs more experimental data. If an insufficient amount of time-series data is given as observed profiles, the high degree-freedom of systems biology models ensures that many candidate solutions will be found.
2. Perform a sensitivity analysis and identifiability analysis before the identification phase [36-38].
3. For systems models with insensitive or non-identifiable parameters, the search may lead to a solution where some parameters can have large deviation, but still produce satisfactory system responses. This problem can be partly relieved by introducing auxiliary information (additional constraints such as shrinking the searching region) of the model into the algorithm. However, it remains difficult to be solved completely by improving parameter estimation strategy. It indicates that researchers should focus on predictions rather than on accurately estimating every parameter.

Although the proposed algorithms are fast and robust, there is certainly room for improvement: for method 1, it is not general enough to catch every case; for method 2, the price for the simplicity and generality is at the expansion of the optimization variable dimension. Under high noise condition, method 2 is still not robust enough. At the moment, the testing is based on Matlab which is much slower than native codes produced by C, Fortran, etc, however the conversion is straight forward. Currently, many high-speed computation engines are available that make use of parallelism, for instance multi-cluster engines, array-processing engines etc. Hence, one possible way is developed algorithm on these high-speed computation engines environment. Another possible way is developing hybrid algorithms to incorporate elements from evolution algorithms such as GA, SA and PSO. In this paper, we have considered the parameter estimation problem with known structure. However, it is easy to expand our method to structure identification by introducing an additional penalty term to the objective function [39].

Additional material

Additional file 1: In this additional file, we tested the proposed methods on seven systems biology models were used to test: TNFα -Mediated NF-κB-Signaling Pathway Model, RKIP Regulated ERK Pathway model and the model of irreversible inhibition of HIV proteinase; Yeast fermentation pathway Model, large-scale target genetic network model, a three step pathway model and the mammalian G1/S transition network model.

Additional file 2: In this additional file, we use E2F/DP dimmer model to illustrate the differences between the three different type methods mentioned in the paper: (i) the direct optimization method; (ii) decomposition methods; (iii) methods Combine spline theory and NLP.

Acknowledgements
The work is supported by CERG Grant 9041393 (CityU 123808). We would like to express our appreciation for Dr. Robin S. Bradbeer and Dr. K. T. Ko for their fruitful suggestions.

Authors' contributions
All authors contributed to the development of the theory and analysis of the results. CJ designed, analyzed and implemented the algorithm, performed the experiments and wrote the main body of manuscript. LY contributed to the development of the method and writing the manuscript and providing the critical review of the manuscript. All authors have read and approved the final version of the manuscript.

Received: 18 March 2010 Accepted: 24 January 2011
Published: 24 January 2011

References

1. Chang WC, Li CW, Chen BS: **Quantitative inference of dynamic regulatory pathways via microarray data.** *BMC Bioinformatics* 2005, **6**:1-19.
2. Cho KH, Shin SY, Kim HW, Wolkenhauer O, McFerran B, Kolch W: **Mathematical modeling of the influence of RKIP on the ERK signaling pathway.** *Lecture Notes in Computer Science* 2003, **2602**:127-131.
3. Cho KH, Shin SY, Lee HW, Wolkenhauer O: **Investigations into the analysis and modeling of the TNFα -Mediated NF-κB-Signaling pathway.** *Genome Res* 2003, **13(11)**:2413-2422.
4. Hoffmann A, Levchenko A, Scott ML, Baltimore D: **The IkB-NF-kB Signalling module: temporal control and selective gene activation.** *Science* 2002, **298(5596)**:1241-1245.
5. Swat M, Kel A, Herzel H: **Bifurcation analysis of the regulatory modules of the mammalian G1/S transition.** *Bioinformatics* 2004, **20(10)**:1506-1511.
6. Tyson JJ, Chen K, Novak B: **Network dynamics and cell physiology.** *Nature Reviews Molecular Cell Biology* 2001, **2**:908-916.
7. Voit EO: *Computational analysis of biochemical systems, a practical guide for biochemists and molecular biologists* Cambridge: Cambridge University Press; 2000.
8. Goel G, Chou IC, Voit EO: **System estimation from metabolic time-series data.** *Bioinformatics* 2008, **24(21)**:2505-2511.
9. Ashyraliyev M, Fomekong-Nanfack Y, Kaandorp JA, Blom JG: **Systems biology: parameter estimation for biochemical models.** *FEBS J* 2009, **276(4)**:886-902.
10. Kikuchi S, Tominaga D, Arita M, Takahashi K, Tomita M: **Dynamic modeling of genetic networks using genetic algorithm and S-system.** *Bioinformatics* 2003, **19(5)**:643-650.
11. Tsai KY, Wang FS: **Evolutionary optimization with data, collocation for reverse engineering of biological networks.** *Bioinformatics* 2005, **21(7)**:1180-1188.
12. Chou IC, Martens H, Voit EO: **Parameter estimation in biochemical systems models with alternating regression.** *Theor Biol Med Model* 2006, **19**:3-25.
13. Gadkar KG, Gunawan R, Doyle F Jr: **Iterative approach to model identification of biological networks.** *BMC Bioinformatics* 2005, **6**:155.

14. Gonzalez OR, Küer C, Jung K, Naval PCJ, Mendoza E: **Parameter estimation using Simulated Annealing for S-system models of biochemical networks.** *Bioinformatics* 2007, **23(4)**:480-486.
15. Lall R, Voit EO: **Parameter estimation in modulated, unbranched reaction chains within biochemical systems.** *Comput Biol Chem* 2005, **29(5)**:309-318.
16. Peifer M, Timmer J: **Parameter estimation in ordinary differential equations for biochemical processes using the method of multiple shooting.** *IET Syst Biol* 2007, **1(2)**:78-88.
17. Veflingstad SR, Almeida J, Voit EO: **Priming nonlinear searches for pathway identification.** *Theor Biol Med Model* 2004, **1**:8.
18. Mendes P, Kell D: **Non-linear optimization of biochemical pathways: application to metabolic engineering and parameter estimation.** *Bioinformatics* 1998, **14(10)**:869-883.
19. Moles CG, Mendes P, Banga JR: **Parameter estimation in biochemical pathways: a comparison of global optimization methods.** *Genome Res* 2003, **13(11)**:2467-2474.
20. Cho DY, Cho KH, Zhang BT: **Identification of biochemical networks by S-tree based genetic programming.** *Bioinformatics* 2006, **22(13)**:1631-1640.
21. Koh G, Teong HF, Cléent MV, Hsu D, Thiagarajan PS: **A decompositional approach to parameter estimation in pathway modeling: a case study of the Akt and MAPK pathways and their crosstalk.** *Bioinformatics* 2006, **22(24)**:271-280.
22. Kimura S, Ide K, Kashihara A, Kano M, Hatakeyama M, Masui R, Nakagawa N, Yokoyama S, Kuramitsu S, A K: **Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm.** *Bioinformatics* 2005, **21(7)**:1154-1163.
23. Polisetty PK, Voit EO, Gatzke EP: **Identification of metabolic system parameters using global optimization methods.** *Theor Biol Med Model* 2006, **27(3)**:4.
24. Voit EO, Almeida J: **Decoupling dynamical systems for pathway identification from metabolic profiles.** *Bioinformatics* 2004, **20(11)**:1670-1681.
25. Savageau MA: *Biochemical Systems Analysis: a study of Function and Design in Molecular Biology* Reading, MA: Addison-Wesley, Reading; 1976.
26. Craven P, Wahba G: **Smoothing noisy data with spline functions estimating the correct degree of smoothing by the method of generalized cross validation.** *Numer Math* 1979, **31**:377-403.
27. Tetko IV, Livingstone DJ, Luik AI: **Neural network studies. 1. Comparison of overfitting and overtraining.** *J Chem Inf Comput Sci* 1995, **35**:826-833.
28. Boor D: *A practical guide to splines* New York: Springer; 1987.
29. Voss HU, Timmer J, Kurths J: **Nonlinear dynamical system identification from uncertain and indirect measurements.** *International Journal of Bifurcation and Chaos* 2004, **14(6)**:1905-1933.
30. Kuzmic P: **Program DYNAFIT for the analysis of enzyme kinetic data: application to HIV proteinase.** *Ana Biochem* 1996, **237**:260-273.
31. Boyd SP, Vandenberghe L: *Convex Optimization* Cambridge, U.K.: Cambridge University Press; 2003.
32. Winston WL: *Operations research: applications and algorithms* Belmont, CA: Duxbury Press; 1994.
33. Fletcher RE: **Numerical experiments with an exact penalty function method.** *Nonlinear Programming* 1981, **4**:99-129.
34. Powell MJD: **A Fast Algorithm for Nonlinearly Constrained Optimization Calculations.** *Numerical Analysis* 1978, **27**:630.
35. Runarsson TP, Yao X: **Stochastic ranking for constrained evolutionary optimization.** *IEEE Trans Evol Comput* 2000, **4**:284-294.
36. Raue A, Kreutz C, Maiwald T, J B, Schilling M, Klingmuller U, Timmer J: **Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood.** *Bioinformatics* 2009, **25(15)**:1923-1929.
37. Hengl S, Kreutz C, Timmer J, Maiwald T: **Data-based identifiability analysis of non-linear dynamical models.** *Bioinformatics* 2007, **23(19)**:2612-2618.
38. Yue H, Brown M, Knowles J, Wang H, Broomhead DS, Kell DB: **Insights into the behaviour of systems biology models from dynamic sensitivity and identifiability analysis: a case study of an NF-kB signalling pathway.** *Mol Biosyst* 2006, **2(12)**:640-649.
39. Liu PK, Wang FS: **Inference of biochemical network models in S-system using multiobjective optimization approach.** *Bioinformatics* 2008, **24(8)**:1085-1092.

doi:10.1186/1752-0509-5-14
Cite this article as: Zhan and Yeung: Parameter estimation in systems biology models using spline approximation. *BMC Systems Biology* 2011 **5**:14.