Computers and Chemical Engineering xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

A constrained wavelet smoother for pathway identification tasks in systems biology

Sepideh Dolatshahi, Brani Vidakovic, Eberhard O. Voit*

The Wallace H. Coulter Department of Biomedical Engineering at Georgia Tech and Emory University, 313 Ferst Drive, Suite 4103, Atlanta, GA 30332-0535, USA

ARTICLE INFO

Article history:
Received 8 May 2014
Accepted 20 July 2014
Available online xxx

Keywords:
Biochemical systems theory (BST)
Dynamic flux estimation (DFE)
Metabolic pathway analysis
Parameter estimation
Smoothing
Wavelets

ABSTRACT

Metabolic time series data are being generated with increasing frequency, because they contain enormous information about the pathway from which the metabolites derive. This information is not directly evident, though, and must be extracted with advanced computational means. One typical step of this extraction is the estimation of slopes of the time courses from the data. Since the data are almost always noisy, and the noise is typically amplified in the slopes, this step can become a critical bottleneck. Several smoothers have been proposed in the literature for this purpose, but they all face the potential problem that smoothed time series data no longer correspond to a system that conserves mass throughout the measurement time period. To counteract this issue, we are proposing here a smoother that is based on wavelets and, through an iterative process, converges to a mass-conserving, smooth representation of the metabolic data. The degree of smoothness is user defined. We demonstrate the method with some didactic examples and with the analysis of actual measurements characterizing the glycolytic pathway in the dairy bacterium *Lactococcus lactis*. MATLAB code for the constrained smoother is available as a Supplement.

1. Introduction – motivation

Data characterizing genomic, proteomic or metabolic processes in the form of time series measurements contain very valuable, yet implicit information about the structure and dynamics of biological systems. In an effort to gain deeper insight into these systems, numerous recent articles have been addressing the extraction of this information and its integration into functional models, which may subsequently be utilized for explanation, prediction, manipulation, and optimization. The majority of methods for the extraction of information from time series employ techniques for minimizing the discrepancy between the measured data, *i.e.*, the time profiles, and the assumed model, which typically consists of a system of non-linear ordinary differential equations that are to be parameterized (Chou and Voit, 2009). The currently available estimation techniques include different regression, simulated annealing, or evolutionary optimization approaches, such as genetic algorithms, as well as various support algorithms for preprocessing the data. The latter algorithms are intended to reduce noise and smooth the time courses, which naturally are often quite ragged.

Dynamic flux estimation (DFE) (Goel et al., 2008) is a methodological framework for extracting information from time series measurements. It is distinct from all other methods, as it does not presume knowledge of an appropriate underlying model. DFE combines the tenets of stoichiometric (Gavalas, 1968; Heinrich and Schuster, 1996; Stephanopoulos et al., 1998) and flux balance analysis (FBA; *e.g.*, (Palsson, 2006)), which are genuinely static, with dynamic aspects of ODE modeling. DFE consists of two phases. The first phase is model-free and essentially assumption-free and includes steps of data preprocessing, time course smoothing, and slope estimation, and ultimately yields point-wise inferences of dynamic time series profiles for all fluxes in the system. Expressed differently, the first phase results in a numerical representation of flux values throughout the time period of the experiment. One must note, however, that this numerical representation does not directly reveal optimal, or even appropriate explicit functional forms that capture the dynamics of the fluxes in the system. The second phase addresses this issue. It consists of the mathematical characterization of the numerical flux profiles, based on an assumed format. Fig. 1 summarizes the phased approach of DFE and the expected outcomes of each step.

* Corresponding author. Tel.: +1 404 385 5057; fax: +1 404 894 4243.
E-mail addresses: sepid.dsh@gmail.com (S. Dolatshahi), brani@bme.gatech.edu (B. Vidakovic), eberhard.voit@bme.gatech.edu (E.O. Voit).

http://dx.doi.org/10.1016/j.compchemeng.2014.07.019
0098-1354/© 2014 Elsevier Ltd. All rights reserved.

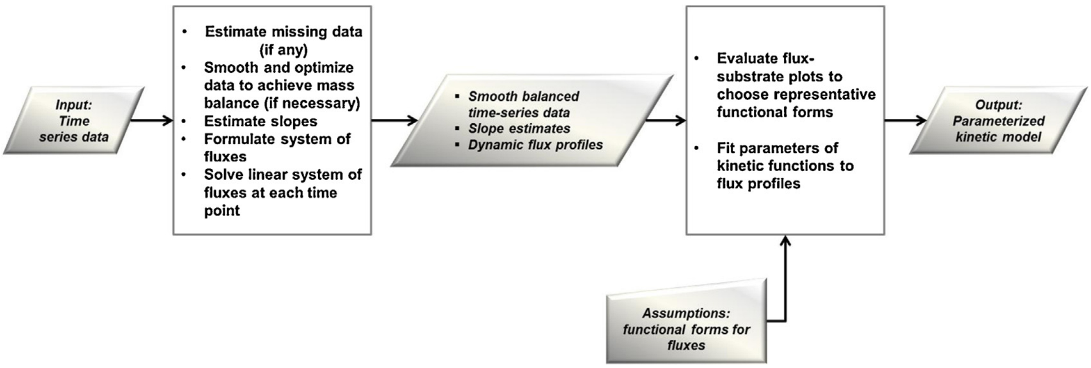
Input: Time series data
• Estimate missing data (if any)
• Smooth and optimize data to achieve mass balance (if necessary)
• Estimate slopes
• Formulate system of fluxes
• Solve linear system of fluxes at each time point
▪ Smooth balanced time-series data
▪ Slope estimates
▪ Dynamic flux profiles
• Evaluate flux-substrate plots to choose representative functional forms
• Fit parameters of kinetic functions to flux profiles
Output: Parameterized kinetic model
Assumptions: functional forms for fluxes

Fig. 1. Phases and steps of the dynamic flux estimation (DFE) technique utilized for metabolic time-series data.

While the performance of DFE can be excellent, it has the significant drawback that a direct application of the method requires a stoichiometric matrix that has full rank. This is seldom the case, because most pathway systems contain more fluxes than metabolites. Several auxiliary methods have been proposed that “fill the rank” with additional information ((Chou and Voit, 2012; Iwata et al., 2013; Voit et al., 2009); see also (Voit, 2013)) but none of them presents a perfect solution.

A secondary issue of a more computational nature is the need for using slopes of the time courses in the model-free phase. On the one hand, the use of slopes is very advantageous, because parameter values may be estimated without the integration of differential equations (Varah, 1982; Voit and Savageau, 1982a,b). Indeed, the integration of a system of differential equations is computationally expensive and prone to a host of technical challenges, associated with complicated error surfaces that can contain numerous local minima (Voit and Almeida, 2004). On the other hand, the slopes are rather sensitive to noise in the time courses, which renders it necessary to smooth and balance the data. Smoothing reduces noise, while balancing assures that there is no gain or loss of mass over time in a closed system.

Numerous methods have been proposed for smoothing time course data. They include splines, moving average algorithms, finite difference approximations, and various types of non-linear programming (Eilers, 2003; Vilela et al., 2007; Whittaker, 1923). These methods are time consuming and need to be performed interactively, or at least in a closely supervised manner. Furthermore, this type of smoothing process can lead to secondary issues. Especially important for the purposes of metabolic pathway analysis is the potential problem that the overall mass in a system may no longer be constant if the data are smoothed. To address these issues, we propose here an automated smoothing technique that takes as input any given data set and estimates and removes noise while at the same time satisfying the required mass balance within the system. The proposed approach is iterative and called constrained iterative wavelet-based smoother (CIWS).

2. Background and data

2.1. Multiresolution analysis using wavelets

The proposed smoothing technique is built upon the notion of multiresolution analysis (MRA) from wavelet theory which we will briefly explain here.

Wavelets are becoming a standard data analysis tool that is excellent for tasks of data compression as well as for denoising and smoothing. One of their advantages is that they are flexible as well as local, which means that they do not ignore desirable functional details. The reason is that the resolution in MRA can be adapted to the situation at hand.

Mathematically speaking, wavelets are orthogonal basis functions which span the space of all square-integrable functions ($L^2(\boldsymbol{R})$). Thus, any element in $L^2(\boldsymbol{R})$ may be represented as a possibly infinite linear combination of these basis functions. An important property of this linear representation is that it may be partitioned into orthogonal subspaces $W_j = span[\psi_{j,k}(x)]$, each of which captures a certain level of “detail” information. The key concept of orthogonal MRA is to partition a given function $f(x)$ into its components $f^{(j)}(x) \in W_j$. Here, the space W_j consists of functions with lower resolution than the ones in W_{j+1} which means that if some arbitrary function $g(x)$ is in W_j, then $g(2x)$ is in W_{j+1} (Strang, 1989).

For example, in the traditional wavelet representation

$$f(x) = \sum\nolimits_{k \in z} C_{J_0,k}\phi_{J_0,k}(x) + \sum\nolimits_{j \ge j_0, k \in Z} d_{j,k}\psi_{j,k}(x)$$

the second sum contains the terms which capture the higher levels of detail (*i.e.*, $U_{j \ge j_0} W_j$, which is the union of all levels of detail greater than or equal to J_o). Choosing the appropriate coarsest resolution J_0 gives rise to different transforms. We can also just approximate

$$f(x) \cong \sum\nolimits_{k \in Z} c_{J_0,k}\phi_{J_0,k}(x).$$

The choice of J_0 provides us with the flexibility of selecting the desired level of detail, which is traded against the desired level of smoothness. In the above representation of $f(x)$, the functions

$$\phi_{J_0,k}(x) = 2^{j/2}\phi(2^j x - k)$$

and $\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k)$ are scaling and wavelet functions, which correspond to commonly called “smooth” and “detail” coefficients, respectively; j is the dilation/scale index, and k indicates shift or position (Vidakovic, 1999).

In wavelet decomposition, as mentioned before, the wavelet coefficients represent details, and if these are small, they can actually be removed without affecting the general trend of the data. In fact, wavelet transformations are known to be parsimonious in that they can be well described by a relatively small number of “energetic” wavelet coefficients.

Wavelet thresholding is the process of removing the wavelet coefficients that are smaller in magnitude than some threshold λ. The resulting signal, after the inverse wavelet transformation, is expected to have its noise removed or at least reduced. The characteristics of the data determine the magnitude of noise, and it is therefore useful to specify the threshold value magnitude of noise, and it is therefore useful to specify the threshold λ based on the variability of the data at hand. Different thresholding policies and threshold values are discussed in Section 4 in more detail.

All wavelet computations were performed in *WaveLab*, a MATLAB wavelet toolbox available from the website of Stanford University (Buckheit and Donoho, 1995). Sample MATLAB codes