BioRXiv, 2025, pp. 1–13

doi: https://doi.org/10.1101/2025.07.22.666125

Preprint

PREPRINT

Estimating flux dynamics from metabolite concentration time courses with `dynafluxr`

Serguei Sokol,[1,2,*] Svetlana Dubiley,[1] Pauline Rouan,[1] Cyril Charlier,[1] Guy Lippens[1] and Pierre Millard[1,2,*]

[1]TBI, CNRS, INRAE, INSA, Toulouse, France and [2]MetaboHUB, National infrastructure of metabolomics and fluxomics, Toulouse, France, Toulouse, France

[*]Corresponding authors: sokol@insa-toulouse.fr, millard@insa-toulouse.fr

Abstract

Motivation: Quantifying metabolic fluxes and their dynamics is crucial for addressing fundamental and applied questions in systems and synthetic biology, health, and biotechnology. Since metabolic fluxes are not directly observable, they must be estimated using model-based approaches. While several methods are available to quantify steady-state fluxes, approaches for quantifying flux dynamics remain limited.

Results: We present a method to quantify flux dynamics from metabolite concentration time courses. Our approach employs constrained B-splines to solve mass balance equations without requiring prior knowledge of catalytic or regulatory mechanisms, or enzymatic parameters. We provide a comprehensive mathematical framework for dynamic flux calculation and demonstrate that our method enables accurate and robust quantification of flux dynamics for both simulated and experimental datasets.

Availability and implementation: `dynafluxr` is implemented in R and available on Windows, Unix, and MacOS platforms. The source code and the documentation are freely distributed under GPL-3 license at https://github.com/MetaSys-LISBP/dynafluxr/. A companion notebook producing figures similar to those of this article is available at https://github.com/MetaSys-LISBP/DynafluxR_notebook.

Key words: fluxomics, metabolic fluxes, flux dynamics, B-spline, time-series measurements

Introduction

Quantifying metabolic fluxes is central to understand the function and regulation of living organisms, as well as to engineer efficient biological systems [11, 15, 3, 18, 26]. Steady-state (i.e. static) fluxes can be quantified by flux balance analysis (FBA) which relies on stoichiometric models and inequality constraints [7]. Static fluxes can also be calculated by ^{13}C-metabolic flux analysis (^{13}C-MFA) where underdetermination is compensated by isotope balance equations to close the stoichiometric system [1]. The practical value of these approaches is recognized and they are widely used in systems and synthetic biology.

© The Author 2025. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

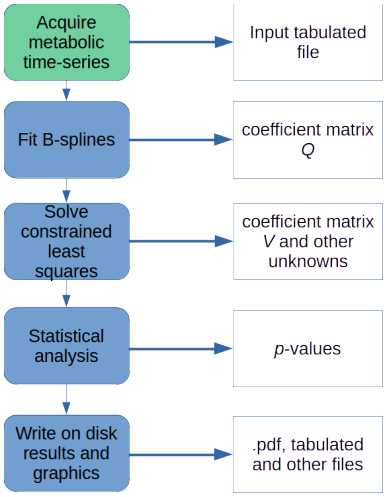

Fig. 1. Workflow of the `dynafluxr` package. The green box is the experimental step and the blue boxes represent software tasks. The order of the tasks in the workflow is indicated by thin vertical arrows. The thick horizontal arrows point to the main results of each step (white boxes). Matrices are described in the text.

However, many questions require metabolic fluxes to be monitored dynamically. This can be achieved through kinetic models of metabolism, which require, in addition to the network topology, extensive knowledge of enzyme function including detailed information on catalytic and regulatory mechanisms to determine the most appropriate rate laws. These models also depend on enzymatic parameters (such as K_m and K_I) and thermodynamic parameters (K_{eq}), which can be measured or estimated from experimental data. However, building accurate kinetic models of the dynamics of metabolic systems [14, 28] is data- and resource-intensive, and difficult to scale for large networks because of computational and experimental bottlenecks [25].

While an intermediate approach based on splines or Kalman filtering has been proposed to estimate the dynamics of growth rates and extracellular uptake or production fluxes [5], this approach cannot provide information on the dynamics of internal fluxes. This particular issue is partially resolved in dynamic MFA by extending the approach used to calculate the metabolite dynamics to internal fluxes, which are modeled using B-spline functions and nonlinear optimization [2, 13, 27]. However, this comes at the price of assuming stationary state to internal metabolite levels, with only external metabolite levels left unbalanced and free to vary over time. An approach for estimating complete flux dynamics including unbalanced internal metabolites was introduced in [8], using stoichiometric modeling and slope estimation at each time point to describe the point-wise time evolution of fluxes. The limitations of this proof-of-principle were however that no detailed mathematical framework was provided, missing data typically encountered in real-world experiments were not addressed, uncertainty propagation was overlooked, and no software implementation was proposed.

Here, we present a detailed mathematical framework to estimate flux dynamics from metabolite concentrations time-course, based only the matabolic network topology. We first describe the mathematical background of the proposed algorithm to calculate fluxes and estimate uncertainties propagation. We also highlight key considerations for implementation, which are incorporated into the accompanying software, `dynafluxr`. We demonstrate the applicability and robustness of our approach using a synthetic dataset and present a real-word use-case in which fluxes are estimated in a multi-enzymatic pathway monitored by real-time NMR spectroscopy.

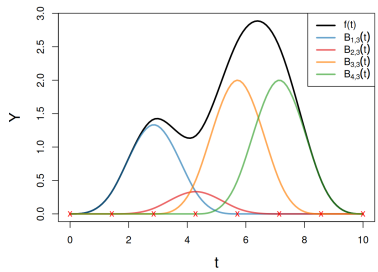

Fig. 2. Example of B-spline approximation. The black line is a function $f(t)$, represented as a sum of weighted B-splines (colored lines) defined over knotsrepresented here by red crosses.

Algorithm

General principles

The main steps of the algorithm are outlined in Figure 1. The proposed approach is based on the use of constrained B-splines to fit metabolite time courses with stoichiometric constraints. As detailed in the next section, the advantages of constrained B-splines are that they respect the positivity and, if necessary, monotonicity of metabolite time courses, properties that are difficult to achieve with other basis functions such as Fourier series or wavelets. The coefficients of the fitted B-splines (which we call measured coefficients) are fitted in a linear system based on the stoichiometric matrix. The principal unknowns in this linear system are the B-spline coefficients of the reaction rates which are estimated directly by constrained least squares optimization, with inequality constraints on the positivity and monotonicity of the simulated metabolite time curses and equality constraints, e.g. in the form of zero starting values for absent metabolites at the start of experiments.

The following sections describe the advantages and drawbacks of two possible approaches to modeling time curses : the integrative approach, where the metabolite dynamics coefficients are fitted directly and the differential approach, where the fit is performed on the coefficients of the first derivatives of the metabolite time courses.

Describing metabolite time courses with constrained B-splines

Let us first recall the fundamentals of B-splines. B-splines are piecewise polynomial functions [6] that are widely used for interpolation in industry and research. B-spline basis functions $B_{k,n}$ of degree n typically ressemble a series of bell-curves (Figure 2) and can be recursively defined over a sequence of knots $\{\tau_k,\ k = 1, \ldots, K,\ \text{s.t. } \tau_{k+1} \geq \tau_k,\ \forall k\}$:

$$\begin{aligned} B_{k,0}(t) &= \begin{cases} 1 & \text{if } \tau_k \leq t < \tau_{k+1} \\ 0 & \text{otherwise} \end{cases} \\ B_{k,n}(t) &= \tfrac{t-\tau_k}{\tau_{k+1}-\tau_k} B_{k,n-1}(t) + \tfrac{\tau_{k+n+1}-t}{\tau_{k+n+1}-\tau_k} B_{k+1,n-1}(t),\ \text{for } n \geq 1. \end{aligned} \tag{1}$$

Any real-valued time-varying function can be approximated as a weighted sum of basis functions $f(t) = \sum_k q_k B_{k,n}(t)$ where $\{q_k, k = 1, \ldots, K-n-1\}$ are B-spline coefficients or weights.

It follows from the definition of B-splines (Equation 1) that $B_{k,n}(t) \geq 0,\ \forall k, n, t$. If moreover, all $q_k \geq 0$ then $f(t) \geq 0,\ \forall t$. We have used this property in the R package `bspline` [23] to achieve constrained B-spline fitting with non-negative q_k coefficients. This key feature ensures that the fitted chemical species concentrations are always non-negative.

Derivative and integration operators can also be defined based only on B-spline coefficients. If $f'(t) = \sum_k p_k B_{k,n-1}(t)$, the first derivative of $f(t)$, then the coefficients p_k and q_k are related by

$$p_k = \frac{n}{\tau_{k+n+1} - \tau_{k+1}}(q_{k+1} - q_k) \tag{2}$$

or in matrix form

$$p = Dq \tag{3}$$

where D is an appropriately sized differentiation matrix whose coefficients are given by Equation 2.

Assuming that the coefficients p_k of $f'(t)$ are known, B-spline curves can be integrated to retrieve the coefficients q_k of $f(t)$:

$$q_{k+1} = \sum_{i=1}^{k} \frac{\tau_{i+n+1} - \tau_{i+1}}{n} p_i + c \tag{4}$$

or in matrix form

$$q = Yp + c\mathbb{1} \tag{5}$$

where Y is an integration matrix of appropriate size whose coefficients are defined by Equation 4, c is an arbitrary context-dependent constant, e.g. the initial condition, $f(0)$, and $\mathbb{1}$ is a vector of ones.

Another useful feature of the `bspline` package is that the fitted curves can be constrained to be monotonous, i.e. $q_{k+1} \geq q_k, \ \forall k$. It then follows from (Equation 2) that in this case $p_k \geq 0, \ \forall k$ which means that $f'(t) \geq 0, \ \forall t$, hence $f(t)$ is monotonically increasing. The same property can be exploited for monotonically decreasing functions. These constraints can be useful for species that are known to be only consumed (e.g. monotonically decreasing substrate concentrations) or only produced (e.g. monotonically increasing concentrations of end-products).

Differential least-squares formulation

The `bspline` packagecan be used to fit individual metabolite time course $M_j(t)$ (the first step in Figure 1). This preliminary fit does not take any system level information into account since the stoichiometric matrix is not used at this stage. The goal is to capture the dynamics of the time courseswith the coefficients of their first derivative subsequently fitted by least-squares optimization involving the stoichiometric matrix. The fitted B-spline functions are constrained to be non-negative and, if appropriate, monotonic.

The continuous time evolution of a given metabolite, $M_j(t)$, can be expressed in terms of fourth order B-splines as follows:

$$M_j(t) = \sum_k Q_{kj} B_{k,4}(t) \tag{6}$$

where Q is the coefficient matrix from the preliminary fitting step. Forth order B-splines are used to ensure that reaction rates be modeled with third order B-splines, a high enough order in most cases to model complex dynamics with smooth curves and continuous second derivatives.

The j-th column $Q_{.j}$ corresponds to metabolite $M_j(t)$, thus the k-th element Q_{kj} corresponds to k-th coefficient of the basis function $B_{k,4}(t)$. The knot sequence (the same for all metabolites) is a regularly spaced time grid covering the entire measurement interval and determined by the number of internal knots (user-defined, 5 by default). This number should be set to the lowest value at which satisfactory fits are obtained. In our applications with 5 to 1000 time points, accurate fits were obtained with 0 to 5 internal knots. Using too many internal knots can lead to over-fitting and non-physical results such as excessive oscillations. Note that the fitting process is robust to the presence of a few missing data, provided that the number of available data points is higher than the number of B-spline coefficients and that each knot interval contains at least one valid data point. This allows the analysis on incomplete datasets where the dynamics of the system have only partially been measured.

Note that metabolites that are not detected experimentally, for which all data are missing, are simply excluded from the least-squares cost functions. This point is considered in greater detail later in the paper.

B-splines have already been used to fit metabolite dynamics [8, 13] but the originality of our approach, described in the following paragraphs, is to perform a constrained least-squares fitting problem directly on B-spline coefficients of reaction rates V.

Let S_{jl} be an element of the stoichiometric matrix, S, corresponding to metabolite M_j, which is either produced or consumed (depending on the sign of the product $S_{jl}v_l$) by an l-th reaction at a rate v_l. We formally distinguish the reaction rate, v_l, from the metabolic flux, $f_{jl} = S_{jl}v_l$, induced by this reaction. In this context, reaction rates indicate how often the corresponding reactions take place per unit time (dimension of 1/[Time]). In contrast, the metabolic flux, f_{jl}, represents th amount of M_j transformed by the l-th reaction per unit time (dimensions of [Concentration]/[Time]). When $|S_{jl}| = 1$, v_l and f_{jl} have the same absolute values (which explains why they are often considered synonyms) but the two quantities vahe different physical meanings. The overall metabolic flux for M_j can then be calculated from $M'_j(t) = \sum_l f_{jl}$, or over the whole concentration vector $M(t)$, using

$$M'(t) = Sv(t) \tag{7}$$

where v is a vector of unknown reaction rates. This well-known stoichiometric ordinary differential equation represents the mass conservation law at every time point t.

Modeling $v_l(t)$ as $\sum_k V_{kl}B_{k,3}(t)$ where V_{kl} are third order B-spline coefficients, the first derivative of M_j can be written

$$M'_j(t) = \sum_k P_{kj}B_{k,3}(t) \tag{8}$$

and the P_{kj} coefficients can be obtained from Q_{kj} based on (Equation 3)

$$P = DQ. \tag{9}$$

Substituting these two expressions for $M'_j(t)$ and $v_l(t)$ into Equation 7 and simplifying $B_{k,3}(t)$yields the following matrix equation for the B-spline coefficients V:

$$\boxed{P^t \approx SV^t} \tag{10}$$

where the superscript ^{t} represents the transposition operator. In this form, the system can be solved by classical linear algebra methods based on generalized inverses.

The Equation 10 has a unique solution if and only if (i) S is a "tall" matrix, i.e. the number of rows, n_r, (the number of species in the vector $M(t)$) is equal to or greater than the number of columns, n_c (the number of reactions in $v(t)$), and (ii) S is of full rank, i.e. rank$(S) = n_c$. The condition $n_r \geq n_c$ must still be satisfied after the rows of S and columns of Q and P corresponding to unmeasured metabolites are removed. Otherwise, the system is under-determined and has an infinite number of solutions. In this situation, the software provides the solution of minimum norm [9], as used for example in parsimonious FBA [20] with rational that since high fluxes correspond to higher enzyme turnover, and since producing enzymes is costly, cells have evolved to minimize overall flux.

Another situation that can arise in practice is for one or several metabolites to only have relative concentrations available, as typically occurs when metabolites are analyzed by mass spectrometry without an internal standard, or by nuclear magnetic resonance when the number of atoms contributing to the signal cannot be determined. In this case, additional scaling factors $\{\alpha_j\}$ are introduced, defined as

$$\alpha_j = \begin{cases} 1 & \text{if } M_j \text{ is measured with known scale} \\ \text{unknown} & \text{otherwise.} \end{cases} \tag{11}$$

The corresponding metabolite measurements are then scaled as

$$\mathrm{diag}(\alpha)P^t \approx SV^t \tag{12}$$

where $\mathrm{diag}(\alpha)$ is a diagonal matrix with vector α on the main diagonal. Since this scaled system is difficult to represent in the classical form, $Ax = b$, with an explicit matrix A, right hand side b and unknown vector x enclosing α and V, it is solved using the generalized minimum residual (GMRES) method [19, 24], which does not require an explicit matrix formulation.

Differential least-squares (DLS) optimization is simple, elegant and computationally efficient, but cannot include constraints on initial values, for example the values measured for a set of metabolites at the beginning of the experiment which are crucial for successful fitting in some cases. This is the main motivation for developing the integral least-squares (ILS) method described in the following section.

Integral least-squares formulation

$Sv(t)$ can be integrated over time using Equation 5 to obtain a linear system in the least-squares sense:

$$\boxed{Q^t \approx SV^tY^t + C\mathbb{1}^t} \tag{13}$$

where C is an appropriate vector of additive constants, one per metabolite. The unknowns in this equation are the matrix V and the vector C. Inequality constraints for possible monotonicity

$$(VS^t)_{.j} \geq 0 \text{ or } \leq 0 \text{ (if } M'_j(t) \geq 0 \text{ or } \leq 0), \tag{14}$$

for non-negativity

$$SV^tY^t + C\mathbb{1}^t \geq 0 \tag{15}$$

and equality constraints on initial values (too cumbersome to be detailed here) can be added to obtain the ILS system. In this case, the solution will require more advanced methods [12, 22] than classical inverses. Unknown scaling factors α can also be included, as in the DLS system. ILS is the default optimization method in `dynafluxr`, despite being more computationally intensive, because it allows the inclusion of initial values and other types of constraints.

Optimization by ILS or DLS corresponds to the second blue box in Figure 1.

Sensitivity analysis

The package `bspline` provides covariance matrix estimates of the uncertainties of the fitted B-spline coefficients for each metabolite. The covariance matrix for the j-th metabolite is represented as the product of a basis covariance matrix, Z^q, and a corresponding factor s_j^2:

$$\mathrm{cov}(Q_{.j}) = Z^q s_j^2. \tag{16}$$

The same basis matrix Z^q is used for all fitted metabolites. The vector of s_j^2 elements is referred as s^2 hereafter. Note that elements on the same row in Q are not correlated as they correspond to independently measured metabolites.

To calculate the covariance matrix for V, solution of the DLS optimization problem, we first calculate the covariance of $P_{.j}$:

$$\mathrm{cov}(P_{.j}) = D \cdot \mathrm{cov}(Q_{.j}) \cdot D^t = s_j^2 D Z^q D^t = s_j^2 Z^p \tag{17}$$

where D is differentiation matrix introduced in equation 3 and Z^p is a shorthand expression for DZ^qD^t.

The covariance of any two elements of V can then be calculated using

$$< V_{ij} V_{kl} >= Z^p_{ik} (S^\dagger \cdot \mathrm{diag}(s^2) \cdot (S^\dagger)^t)_{jl} \tag{18}$$

where $\mathrm{diag}(s^2)$ is a diagonal matrix with the elements of vector s^2 on the main diagonal. The derivation of this formula is given in SupplementaryFile1.

The square root of $< V_{kl} V_{kl} >$ is the standard deviation (SD) of the k-th B-spline coefficient for the l-th reaction rate. This formula can be used to trace B-spline curves with smooth confidence intervals.

For ILS, the formula for estimating uncertainties is too complex to be presented here but the resulting method is implemented in `dynafluxr` package (see source code for details).

Goodness-of-fit evaluation

Goodness of fits in `dynafluxr` are estimated using χ^2-tests. The statistic for the j-th species :

$$R_j = \sum_i (\hat{M}_j(t_i) - M_{ij})^2 / \sigma_j^2 \tag{19}$$

is the sum of squared residuals divided by the variance σ_j^2 estimated from the variance of differences between fitted and measured values: $\mathrm{var}(M_j^*(t_i) - M_{ij})$. Note that M_{ij} are measurements and $M^*(t)$ are preliminary fitted B-splines (the superscript * means that corresponding fit was performed on a grid with an optimal number of knots estimated automatically in `dynafluxr`), while $\hat{M}_j(t_i)$ in Equation 19 is the value of splines obtained (i.e. estimated) by integrating $Sv(t)$. If the uncertainties in M_{ij} are normally distributed (a common assumption) then R_j follows a χ^2 distribution with $n_m - n_k$ degrees of freedom (where n_m is the number of measurements and n_k is the number of B-spline coefficients estimated for a given metabolite). The test's p-value is then $P[\chi^2(n_m - n_k) \geq R_j]$. Since the null-hypothesis of this test is that R_j follows $\chi^2(n_m - n_k)$, small p-values (typically defined as $p \leq 0.05$) indicate poor quality fit.

Note that these χ^2 tests are performed globally but also separately for each species, allowing user to differentiate between well-fitted and poorly fitted metabolites.

Another point to keep in mind is that R_j can be nonzero even in the absence of uncertainty, e.g. for synthetic data. This is because polynomials such as B-splines cannot perfectly fit other types of functions such as exponential and sigmoid functions.

Goodness-of-fit testing corresponds to the third blue box in Figure 1.

Implementation

The algorithms described in the previous sections are implemented in the R [17] packages `bspline` and `dynafluxr`, accessible from CRAN [23, 21]. The `dynafluxr` package can be used as a command line tool, programmatically, as a classical R package or as a local Shiny [4] application with a graphical user interface on a dynamic web page. The input data (network topology and measured metabolite concentrations) can be provided as tab-separated values (TSVs). The results are written in TSV files and PDF plots are generated for visual inspection. Results can also be returned as R objects for further manipulation in R (see the example `notebook.Rmd` at https://github.com/MetaSys-LISBP/DynafluxR_notebook). In this article, we have used `dynafluxr` v1.0.1 and `bspline` v2.5.0.

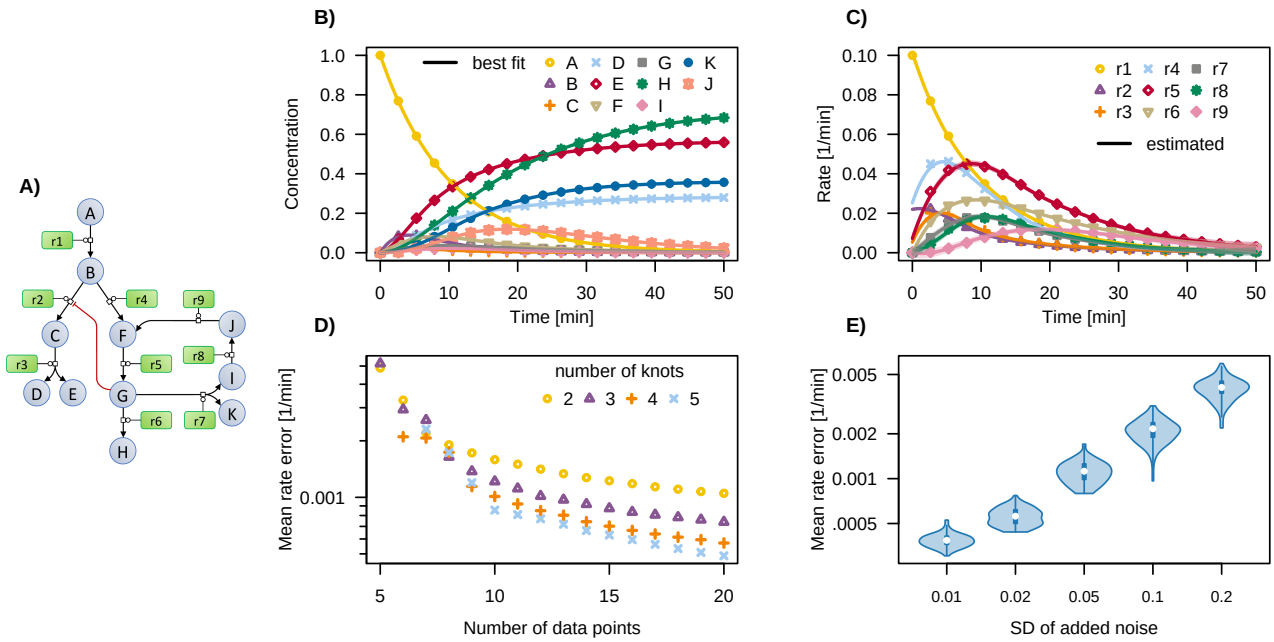

Fig. 3. Validation of `dynafluxr` with simulated data. (A) Topology of the metabolic network used for simulations. (B,C) Simulated and estimated (B) metabolite and (C) flux dynamics with simulated data as points and `dynafluxr` best fits shown as lines. (D,E) Mean error on reaction rates (D) as a function of the number of time points and knots used for flux calculation and (E) as a function of the added noise level.

As for other CRAN packages, a user manual and step-by-step description of a use-case (human readable text explaining R code excerpts with embedded results) are provided. The source code of these packages is available under open source GPLv3 licence from the following GitHub repositories https://github.com/MathsCell/bspline and https://github.com/MetaSys-LISBP/dynafluxr.

Validation using a synthetic dataset

The ability of the proposed method to accurately retrieve reaction rates was first tested using simulated data. We built a kinetic model of the network shown in Figure 3A, consisting of eleven metabolites (A to K) and nine reactions (r1 to r9). Reaction r1 was regulated by the law of mass action, while the other reactions were modeled using Michaelis-Menten kinetics, with reaction r2 regulated by metabolite G through competitive inhibition. Reaction rates and metabolite dynamics were simulated using the software COPASI [10] at 20 time points equally spaced over 25 min. This model has been uploaded to the Biomodels database (under identifier MODEL2502210001). The stoichiometric model as well as the simulated metabolite time courses can be found in Supplemental Data as files `network_toy.txt` and `data_toy.tsv` respectively. The resulting fluxes and metabolite concentrations show a broad range of dynamic profiles. For instance, some metabolites are continuously depleted (e.g., A), some only accumulate (e.g., H and E), and others increase in concentration at first before decreasing (e.g., G and F).

We used `dynafluxr` to estimate flux dynamics from the simulated metabolite dynamics, using only the network stoichiometry and default parameters. The calculations took less than 1 s to perform on a laptop computer. All metabolite dynamics were accurately captured, with a mean error in metabolite concentrations of $2.9 \cdot 10^{-4}$ (Figure 3B). All reaction rates were also estimated with high accuracy (Figure 3C), with a mean error of just $4.9 \cdot 10^{-4}$, very low relative to the flux values. These results confirm that under ideal conditions (when measurements are noise-free, all metabolites are measurable, and the network topology is correctly defined) the proposed method does indeed provide accurate flux estimates.

To assess the robustness of our method with respect to the number of measured time points and the number of knots, we calculated fluxes from the same simulated dataset with 5 to 20 equally spaced time points, using 2 to 5 knots. Figure 3D shows

that the mean reaction rate error was high when less than 10 time points were included. In situation such as this, estimates with fewer knots are more accurate because of instabilities when the number of knots is similar to the number of measurements. When at least 10 time points were included however, the error stabilized at $\leq 10^{-3}$ regardless of the number of knots used. In these cases, using more knots slightly improves the accuracy of the predicted fluxes, providing more flexibility to accurately fit the metabolite dynamics. These results indicate that above a certain data threshold (at least 7-10 time points in this example), our approach is robust with respect to the number of data points and knots used. Note that since the optimal number of knots and measurements depend both on the metabolic network and on the metabolite dynamics, optimal fitting parameters should be determined on a case-by-case basis.

Flux calculations were also repeated with increasing levels of Gaussian noise, ranging from 1 % to 20 % of the mean concentration of each metabolite. All datasets were successfully fitted, and, as expected, the mean error increased gradually with the noise level (Figure 3E). In this example, the errors on reaction rates remained low for noise SDs below 10 %. Although the precise effects of noise depend on the studied metabolic network and metabolite dynamics, these results confirm that our method is robust to noisy datasets.

Application to a multi-enzymatic system

We further evaluated `dynafluxr` by calculating fluxes in a multi-enzymatic system reproducing the first three reactions of glycolysis (Figure 4A). As described in [16], the experiment involved adding glucose (5 mM) at $t = 0$ to a buffer solution containing ATP (10 mM) and the first three glycolytic enzymes: glucokinase (HXK), glucose-6-phosphate isomerase (PGI), and phosphofructokinase (PFK). The concentration time courses of glucose (GLC), glucose-6-phosphate (G6P), fructose-6-phosphate (F6P), and fructose-1,6-bisphosphate (FBP) were monitored by real-time ^{1}H-NMR. Data were recorded as a pseudo-2D spectrum with 1,024 equally spaced time points from 10 to 723.8 min after the glucose pulse. The integrals of the signals were obtained with MultiNMRFit [16]. Absolute concentrations of GLC and G6P were determined by normalization against the internal standard. For F6P and FBP, only relative concentrations could be determined because of assignment ambiguity. The stoichiometric model as well as the measured data can be found in Supplemental Data as files `network_glyco.txt` and `data_glyco.tsv` respectively.

Experimental metabolite time courses are shown in Figure 4B. For GLC and G6P, absolute concentrations were fitted directly, while for F6P and FBP, scaling factors were used because only relative concentrations were available. Reaction rates (Figure 4C) were estimated in less than 1 s, and the complete dataset was satisfactorily fitted with 95 % confidence. The HXK flux was maximal at the start of the experiment, when the GLC concentration was highest, and decreased over time as substrate levels declined. In contrast, fluxes through PGI and PFK initially increased as G6P and F6P concentrations rose, before decreasing as GLC was completely converted into FBP. The estimated rates allowed us to isolate the time-resolved contributions of individual reactions to metabolite dynamics (see Figure 4D for the contribution of HXK and PGI to G6P kinetics), providing detailed insights into system dynamics. `Dynafluxr` also validated the self-consistency of the dataset by confirming the carbon balance (Figure 4E).

The number of available measurements for the considered network meant that the system of equations was overdetermined. To test the robustness of our approach to missing data, we removed FBP measurements from the dataset and used dynafluxr to (i) calculate fluxes using the reduced dataset and (ii) predict the concentration time course of FBP. The flux profiles with reduced dataset remained similar to those obtained with the complete dataset (Figure 4F), and the predicted FBP dynamics closely matched experimental data (Figure 4G). These results highlight the robustness of `dynafluxr` in handling missing data and illustrate how metabolite profiles can be predicted. This feature may be useful to deduce spectral assignments by comparing predictions to the profiles of putative signals that cannot be assigned experimentally.

Finally, we compared our method with the current state-of-the-art in kinetic modeling approaches. A kinetic model of the three reactions was constructed, assuming Michaelis-Menten kinetics, with HXK and PFK as irreversible reactions and PGI as reversible. The model has been deposited in the Biomodels database (under identifier MODEL2502210002). The 18 parameters

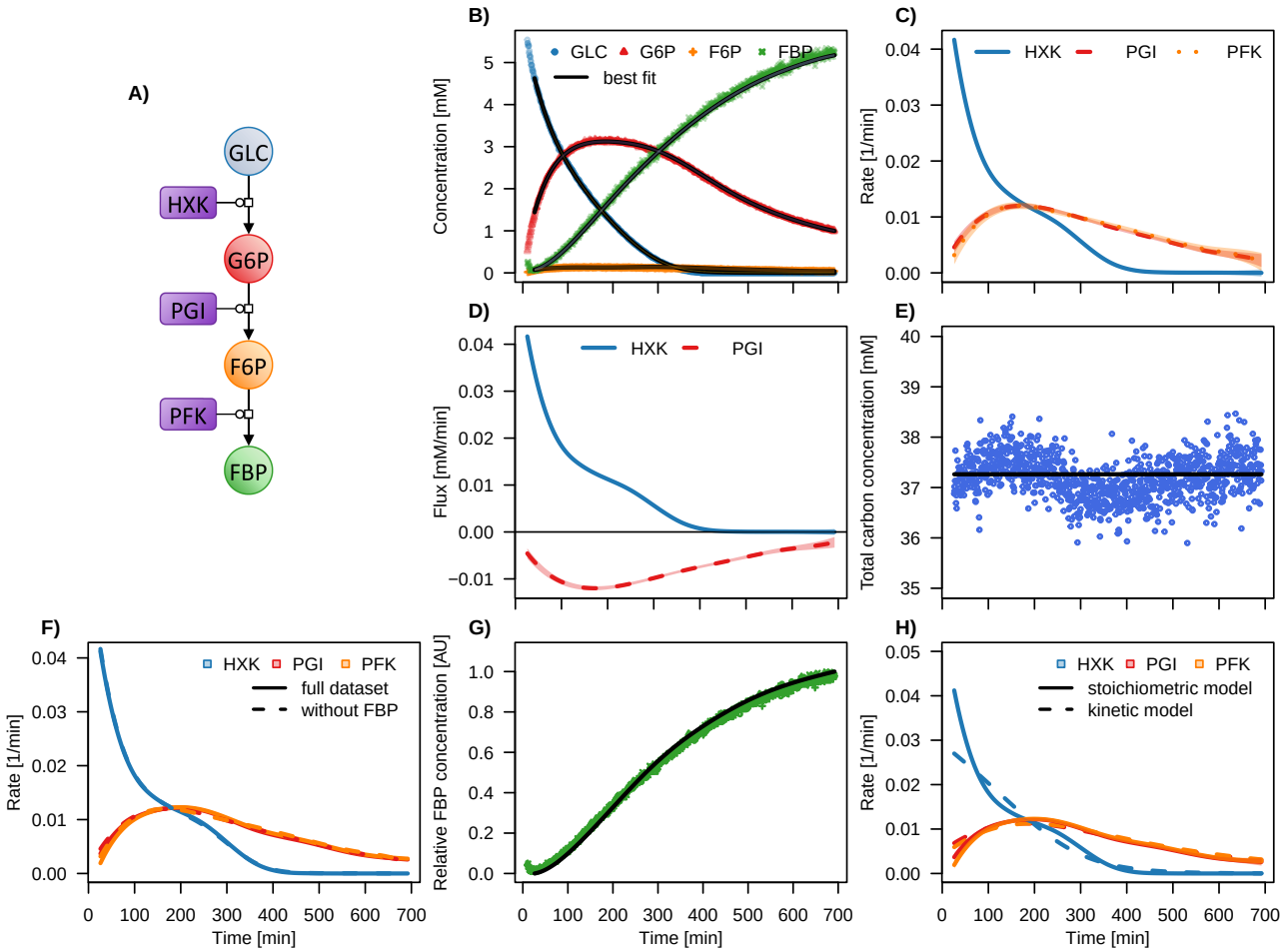

Fig. 4. Flux analysis in a multi-enzymatic reaction system. (A) Topology of the metabolic network used for simulations. (B) Measured and estimated metabolite dynamics, with experimental data shown as points and `dynafluxr` best fits as lines. (C) Reaction rates calculated with `dynafluxr`. (D) Contribution of HXK and PGI fluxes to G6P dynamics. (E) Experimentally measured (circles) and estimated (line) carbon balance. (F) Comparison of reaction rates estimated from the complete metabolite dataset (solid lines) and after excluding FBP (dotted lines). (G) The predicted FBP concentration, whose evolution was not used for flux calculations, is shown as a black line and experimental data are shown as dots. (H) Comparison of reaction rates estimated by `dynafluxr` and by COPASI (using a kinetic model). The colored bands around the fitted curves in parts (B), (C) and (D) correspond to ± 2 standard deviations (they are in some cases indistinguishable from the curves themselves).

(10 kinetic constants, the initial concentrations of six metabolites, and two scaling factors for F6P and FBP) were estimated using COPASI by fitting the complete dataset with particle swarm optimization (5,000 iterations, swarm size of 50). The calculations were completed within a few minutes. The fluxes simulated using the best-fitted COPASI parameters were similar to those obtained using `dynafluxr` (Figure 4H). While the results of this test are not generalizable to all experimental scenarios, they nevertheless highlight the efficiency and simplicity of `dynafluxr`, which is less demanding in computational resources and assumptions than kinetic modeling. `Dynafluxr` also provides confidence intervals of the predicted fluxes, which are difficult to calculate for kinetic models.

Conclusions

Current flux calculation approaches are based either on stoichiometric models, which are easy to construct and computationally cheap to analyze but inherently static, or on kinetic models, which are dynamic by definition but challenging to construct and parameterize. We have developed a method that combines the strengths of these two approaches to estimate flux dynamics from metabolite concentration time courses.

This method is intrinsically dynamic, but relies on minimal biological assumptions (only the network topology) and does not require information on rate laws, catalytic or regulatory mechanisms, or kinetic parameters. Moreover, our method combines a comprehensive mathematical framework with many practical features: computational efficiency resulting from the linear formulation of the problem; amenability to uncertain or missing data; enforced non-negativity and monotonicity of metabolite dynamics; and integration of absolute and relative concentrations through the use of scaling factors.

The precision of the estimated fluxes depends on the condition of the stoichiometric matrix S and on the accuracy, precision and frequency of the metabolite concentration measurements. The results presented here for simulated data show the method is accurate and robust against measurement noise. It was also successfully applied to real-world data from real-time NMR experiments on a multi-enzymatic system. Satisfactory fits were obtained for both datasets, highlighting the simplicity and efficiency of the proposed approach.

The accompanying open-source software, `dynafluxr`, enables fast and reliable estimation of flux dynamics. Beyond addressing fundamental questions in metabolic systems biology, this information is useful in healthcare (e.g., to investigate metabolic diseases) and in biotechnology (e.g., to design more efficient metabolic systems). `Dynafluxr` can complement the current metabolic flux analysis toolbox and help build a reliable framework to better understand and control metabolic fluxes. It opens the way for real-time metabolic flux analysis and can support methodological developments, as illustrated here for data-driven metabolite annotation. Finally, since the proposed method is generic, it can directly be applied to other biological, chemical, geological, or physical systems.

Competing interests

None declared.

Author contributions statement

S.S. conceived the mathematical algorithm, programmed software, analyzed results, and wrote the manuscript;
P.M. conceived and supervised the whole project, analyzed NMR data and results, tested software, developed and programmed kinetic models, and wrote the manuscript;
S.D., P.R., C.C. and G.L conceived and conducted NMR experiments, analyzed the data, and reviewed the manuscript.
PM and SS contributed equally to this work.

Acknowledgments

We thank the MetaToul-Fluxomet platform for access to NMR facilities. This work was funded by the FUNCEMM (ANR-23-CE44-0038) and HEPATOTWIN (INRAE Metaprogram DIGIT-BIO) projects, and was also supported by MetaboHub (MetaboHUB-ANR-11-INBS-0010).

References

1. Maciek R. Antoniewicz. Methods and advances in metabolic flux analysis: a mini-review. *J. Ind. Microbiol. Biotechnol.*, 42:317–325.
2. Maciek R Antoniewicz. Dynamic metabolic flux analysis—tools for probing transient states of metabolic networks. *Current opinion in biotechnology*, 24(6):973–978, 2013.
3. Sara Castaño-Cerezo, Alexandre Chamas, Hanna Kulyk, Christian Treitz, Floriant Bellvert, Andreas Tholey, Virginie Galéote, Carole Camarasa, Stéphanie Heux, Luis F Garcia-Alles, et al. Combining systems and synthetic biology for in vivo enzymology. *The EMBO Journal*, 43(21):5169–5185, 2024.
4. Winston Chang, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert, and Barbara Borges. *shiny: Web Application Framework for R*, 2025. R package version 1.10.0.9000, https://github.com/rstudio/shiny.
5. Eugenio Cinquemani, Valérie Laroute, Muriel Cocaign-Bousquet, Hidde de Jong, and Delphine Ropers. Estimation of time-varying growth, uptake and excretion rates from dynamic metabolomics data. *Bioinformatics*, 33(14):i301–i310, July 2017.
6. Carl De Boor. On calculating with b-splines. *Journal of Approximation theory*, 6(1):50–62, 1972.
7. Erwin P. Gianchandani, Arvind K. Chavali, and Jason A. Papin. The application of flux balance analysis in systems biology. *WIREs Systems Biology and Medicine*, 2:372–382.
8. Gautam Goel, I-Chun Chou, and Eberhard O Voit. System estimation from metabolic time-series data. *Bioinformatics*, 24(21):2505–2511, 2008.
9. Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.
10. Stefan Hoops, Sven Sahle, Ralph Gauges, Christine Lee, Jürgen Pahle, Natalia Simus, Mudita Singhal, Liang Xu, Pedro Mendes, and Ursula Kummer. Copasi—a complex pathway simulator. *Bioinformatics*, 22(24):3067–3074, 2006.
11. Sheng Hui, Jonathan M Ghergurovich, Raphael J Morscher, Cholsoon Jang, Xin Teng, Wenyun Lu, Lourdes A Esparza, Tannishtha Reya, Le Zhan, Jessie Yanxiang Guo, et al. Glucose feeds the tca cycle via circulating lactate. *Nature*, 551(7678):115–118, 2017.
12. Charles L. Lawson and Richard J. Hanson. *Solving least squares problem*. Prentice-Hall series in automatic computation. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974.
13. Verónica S Martínez, Maria Buchsteiner, Peter Gray, Lars K Nielsen, and Lake-Ee Quek. Dynamic metabolic flux analysis using b-splines to study the effects of temperature shift on cho cell metabolism. *Metabolic Engineering Communications*, 2:46–57, 2015.
14. Pierre Millard, Brice Enjalbert, Sandrine Uttenweiler-Joseph, Jean-Charles Portais, and Fabien Létisse. Control and regulation of acetate overflow in escherichia coli. *Elife*, 10:e63661, 2021.
15. Pierre Millard, Thomas Gosselin-Monplaisir, Sandrine Uttenweiler-Joseph, and Brice Enjalbert. Acetate is a beneficial nutrient for e. coli at low glycolytic flux. *The EMBO Journal*, 42(15):e113079, 2023.
16. Pierre Millard, Loïc Le Grégam, Svetlana Dubiley, Thomas Gosselin-Monplaisir, Guy Lippens, and Cyril Charlier. Multinmrfit: A software to fit 1d and pseudo-2d nmr spectra. *bioRxiv*, pages 2024–12, 2024.
17. R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing.
18. Mohsin Rahim, Tomasz K Bednarski, Clinton M Hasenour, Deveena R Banerjee, Irina Trenary, and Jamey D Young. Simultaneous in vivo multi-organ fluxomics reveals divergent metabolic adaptations in liver, heart, and skeletal muscle during obesity. *Cell Reports*, 44(5), 2025.
19. Youcef Saad and Martin H Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.

20. Barbara Schnitzer, Linnea Österberg, and Marija Cvijovic. The choice of the objective function in flux balance analysis is crucial for predicting replicative lifespans in yeast. *PloS one*, 17(10):e0276112, 2022.
21. Serguei Sokol. dynafluxr: Retrieve reaction rate dynamics from time-series measurements of metabolites. R package version 1.0.1.
22. Serguei Sokol. *nlsic: Non Linear Least Squares with Inequality Constraints*. INRAE/INSA/CNRS. R package version 1.0.2.
23. Serguei Sokol. *bspline: B-Spline Interpolation and Regression*. INRAE/INSA/CNRS, 2023. R package version 2.2.
24. Serguei Sokol. *gmresls: Solve Least Squares with GMRES(k)*, October 2025. R package version 0.2.3.
25. Jonathan Strutz, Jacob Martin, Jennifer Greene, Linda Broadbelt, and Keith Tyo. Metabolic kinetic modeling provides insight into complex biological questions, but hurdles remain. *Current opinion in biotechnology*, 59:24–30, 2019.
26. Johan H Van Heerden, Meike T Wortel, Frank J Bruggeman, Joseph J Heijnen, Yves JM Bollen, Robert Planqué, Josephus Hulshof, Tom G O'Toole, S Aljoscha Wahl, and Bas Teusink. Lost in transition: start-up of glycolysis yields subpopulations of nongrowing cells. *Science*, 343(6174):1245114, 2014.
27. Dominique Vercammen, Filip Logist, and Jan Van Impe. Dynamic estimation of specific fluxes in metabolic networks using non-linear dynamic optimization. *BMC systems biology*, 8:1–22, 2014.
28. Jie Yuan, Christopher D Doucette, William U Fowler, Xiao-Jiang Feng, Matthew Piazza, Herschel A Rabitz, Ned S Wingreen, and Joshua D Rabinowitz. Metabolomics-driven quantitative analysis of ammonia assimilation in e. coli. *Molecular systems biology*, 5(1):302, 2009.