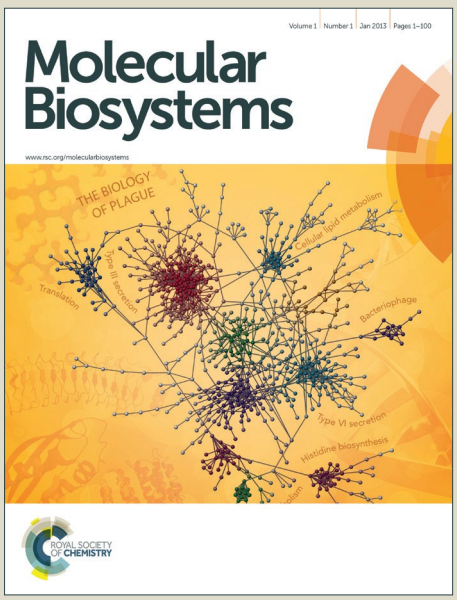
Molecular BioSystems

Accepted Manuscript



ROYAL SOCIETY OF CHEMISTRY

www.rsc.org/molecularbiosystems

Improved metabolite profile smoothing for flux estimation

Robert A. Dromms[1] and Mark P. Styczynski[1]

[1]School of Chemical & Biomolecular Engineering, Georgia Institute of Technology, 311 Ferst Drive, Atlanta, GA 30332-0100

Address for correspondence:

Mark P. Styczynski

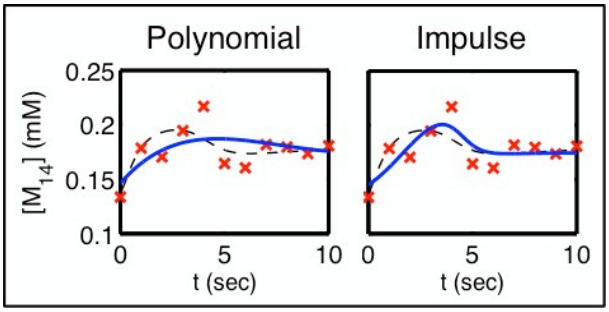
School of Chemical & Biomolecular Engineering, Georgia Institute of Technology, 311 Ferst Drive, Atlanta, GA 30332-0100. Tel: (404) 894-2825 Email: mark.styczynski@chbe.gatech.edu

Abstract (250 words max):

As genome-scale metabolic models become more sophisticated and dynamic, one significant challenge in using these models is to effectively integrate increasingly prevalent systems-scale metabolite profiling data into them. One common data processing step when integrating metabolite data is to smooth experimental time course measurements: the smoothed profiles can be used to estimate metabolite accumulation (derivatives), and thus the flux distribution of the metabolic model. However, this smoothing step is susceptible to the (often significant) noise in experimental measurements, limiting the accuracy of downstream model predictions. Here, we present several improvements to current approaches for smoothing metabolite time course data using defined functions. First, we use a biologically-inspired mathematical model function taken from transcriptional profiling and clustering literature that captures the dynamics of many biologically relevant transient processes. We demonstrate that it is competitive with, and often superior to, previously described fitting schemas, and may serve as an effective single option for data smoothing in metabolic flux applications. We also implement a resampling-based approach to buffer out sensitivity to specific data sets and allow for more accurate fitting of noisy data. We found that this method, as well as the addition of parameter space constraints, yielded improved estimates of concentrations and derivatives (fluxes) in previously described fitting functions. These methods have the potential to improve the accuracy of existing and future dynamic metabolic models by allowing for the more effective integration of metabolite profiling data.

Table of Contents entry

We develop several methods to improve the estimation of metabolite concentrations and accumulation fluxes from noisy time-course data, including use of a sigmoidal impulse function and a resampling-based approach.



Introduction

Genome-scale metabolic modeling is an area of research with the potential for significant impact on many biomedical and biotechnological applications. Such models have been used to identify drug targets that specifically inhibit cancer proliferation[1], to identify genomic manipulations that can facilitate production of valuable chemicals[2], and to uncover and characterize metabolic pathways even in well-understood models[3]. This approach entails using metabolic reconstructions that include all of the cataloged metabolic reactions in an organism (i.e., genome-scale reconstructions) in a defined mathematical modeling framework.

Effectively modeling biological systems at the genome scale calls for measurements and data also at the genome scale. Metabolomics is the systems-scale measurement of the small molecule intermediates in metabolism (the metabolites), a field that has experienced rapid growth in the past decade. Modern analytical technology enables the characterization of metabolic profiles in cells with increasingly fine resolution; this provides relevant information to begin to replace steady state assumptions on a genome-wide scale. However, to date, very few genome-scale metabolic models have attempted to integrate metabolite profiling information, in contrast to the prominent use of transcriptomic, fluxomic, and proteomic data in such models[4-8]. In the few cases where metabolomics data have been integrated into these models, the application of the data has typically been in setting thermodynamic constraints and estimating free energies rather than in more direct applications[9, 10].

The primary reason for this omission is that most metabolic models using genome-scale metabolic reconstructions assume the cell or organism to be at a steady state, typically to simplify the model framework and associated computational complexity. While models exploiting such an assumption have shown great utility, their validity and potential for extrapolation have an intrinsic limit: while the steady state assumption may be true over short time periods, it ultimately is violated once varying forms of metabolic regulation begin to exert their influence.

The use of detailed ordinary differential equation (ODE) models would allow for the capture of dynamic behaviors and regulation, but application of ODE models on a genome-wide scale is not currently feasible due to (among other issues) the many unknown reaction rate and thermodynamic parameters[11-13], each of which would require extensive effort to be ascertained experimentally. As such, significant recent effort has focused on softening the steady state assumption in genome-scale metabolic modeling without requiring a full ODE model of the entire metabolic system[5, 6, 14]. These efforts hold great promise for future biotechnological applications, and they are the motivation for the work presented here.

Use of metabolomics data is a promising approach for bridging the gap between the steady state assumption and the dynamic intracellular reality. This data can be used to estimate the accumulation or depletion “fluxes” of certain metabolites in a system, which can then be used in place of the steady state assumption so common in genome-scale metabolic modeling. This approach has been described and implemented in multiple prior works[15-19]. The most common approach to estimating these accumulation

fluxes from metabolite data is to first smooth the data or fit it to a specific mathematical function, and then use the resulting data or function to determine the flux of that metabolite at any given time (potentially between measured time points). The accuracy of these estimates has an obvious impact on the accuracy of the overall model, but effective estimation of these fluxes is a non-trivial problem given the noise inherent to measurement of metabolite levels and the limitations of the current methods for flux estimation[15].

One of the more thorough treatments of the problem of flux estimation from metabolite data for metabolic modeling was included in work by Ishii *et al.*[18] While the main focus of that work was on developing a broader metabolic model, data smoothing and flux estimation were integral parts of the data processing for the algorithm. They fit a variety of polynomial and rational functions to simulated metabolite data and, on a metabolite-wise basis, selected as the representative function the one that minimizes the fitting error (accounting for the number of free parameters to minimize over-fitting). Of note is that none of the candidate fitting functions are derived from or selected based on biological insight. Additionally, as we show later, the fitting of an arbitrary dataset can yield unphysical results. Splines, another common alternative, are sensitive to noise and outliers—this is particularly problematic when the derivative of the concentration (the accumulation flux) is the important quantity being estimated.

Here, we present two approaches for improving the estimation of accumulation fluxes from metabolite time series data. First, we investigate the use of a biologically reasonable and biologically-inspired sigmoidal impulse function[20, 21] as an effective and

perhaps generalizable alternative to the fitting functions previously used. This functional form emulates behavior observed in known biological systems, and our work represents the first time that it has been applied in the context of metabolic modeling. Second, we investigate whether a resampling-based approach to smoothing and fitting data might yield more accurate concentration profile fits and derivative (flux) predictions than the previously used approach. In the course of these investigations, we also identified the importance of enforcing constraints on fitting equation parameter values to prevent the selection of unphysical solutions. Each of these approaches improves the accuracy of flux estimation from metabolite time series data, providing more reliable results to be integrated into the larger metabolic modeling framework with reasonable computational expense.

Methods

Fitting functions

Eight functions, shown in Table 1, were considered as candidates to best fit the time series metabolite data. The first seven were used by Ishii *et al.*[18]. Four of these were polynomials, of order two to five. The other three were rational functions, composed of a first, second, or third order polynomial numerator and a first or second order polynomial denominator. The eighth function was the sigmoidal impulse, which was first presented in the context of filtering and clustering gene expression profiles[20, 21]; it is here applied for the first time in the context of metabolic models. Unlike the other functions, it has a biologically relevant interpretation: a two-phase transition from one

steady state to a (potentially new) steady state through an intermediate state. Its parameters directly correspond to features of this trajectory, representing: transition time delays; the initial, intermediate state, and steady-state metabolite levels; and the sharpness of the transitions

Synthetic Reference Data

We tested our new methods using two different ODE models of central carbon metabolism taken from the literature, which were used to generate noise-free "gold standard" synthetic reference data for our analyses. These models were selected because their dynamics are believed to reasonably represent *in vivo* metabolic dynamics; the fact that they are not genome-scale does not detract from their relevance as a model system, as the data smoothing/fitting step of flux estimation is independent of the scale of the model.

The first model simulates central carbon metabolism in *E. coli*[11]. While the model includes 18 metabolites, only the 17 metabolites with substantial dynamics were included in our analysis. (As implemented, metabolite 1 was a fixed value.) The second model simulates central carbon metabolism in *S. cerevisiae*[22], comprising 22 metabolites (21 of which had substantial dynamics, and were included in our analysis—changes in metabolite 17 were several orders of magnitude smaller than the concentration). While this model was initially presented in the context of stable concentration oscillations, the initial conditions we used for our simulations do not

produce oscillatory behaviors. To validate our implementation of the model, we used it to reproduce Fig. 6 from Hynne et al. (See Fig. S1)[22].

We obtained curated SBML code for both models from the BioModels Database, and solved systems of ODEs using the LSODA method in the Time Course module of Copasi 4.14, Build 89, with the default tolerances and parameters[23, 24]. For each model, we solved the system of ODEs using the initial conditions specified in Table S1, derived from those previously reported[18], to simulate a perturbation in glucose concentration. As previously described[18], we used a perturbation from 0.0556 mM to 1.67 mM for "Extracellular Glucose" in the *E. coli* model, and a perturbation from 2.5 mM to 5.0 mM for "Mixed flow glucose" in the *S. cerevisiae* model. For the *E. coli* model, we fixed the concentrations of ATP, ADP, AMP, NAD(H), and NADP(H) at their initial values, as was done previously. The resulting gold-standard data contained concentrations at intervals of 0.01 seconds for the *E. coli* model and 0.0025 minutes and for the *S. cerevisiae* model.

To generate data for parameter estimation, simulated time points were sampled at 1 second intervals from 0 seconds to 20 seconds for the *E. coli* model, and at 0.25 minute intervals from 0 minutes to 15 minutes for the *S. cerevisiae* model. The selection of different sampling rates was to be consistent with the approach taken by Ishii *et al*. for the *E. coli* model, but to account for the different time scales of the dynamics in the two mathematical models as observed in the BioModels implementations while still keeping the number of samples used for each respective model the same as that used by Ishii *et al*. By keeping the number of samples the same as in previous work for each

respective model, our fitting results would be most directly comparable. We used a first-order centered finite difference approximation on the ODE output to estimate the derivatives in the synthetic reference data for each metabolite, C_i.

Synthetic Noisy Data

We generated sets of noisy metabolite time courses from this synthetic reference data. For each metabolite C_i, we generated a noisy time course by adding noise at each sampled time point, t_k, to the true value at that timepoint, $C_i(t_k)$, by drawing 5 simulated measurements from a normal distribution, $N_{i,k} \sim (C_i(t_k), CoV \cdot C_i(t_k))$, and then taking the mean of those 5 measurements, called $D_i(t_k)$. We refer to each individual noisy time course as $D_{i,m}$. This approach paralleled the common experimental approach of taking biological replicate measurements and then collapsing them into one value for analyses. Here, we set the Coefficient of Variation (CoV) to 0.15, a reasonable value for many mass spectrometry-based metabolite profiling approaches. The same noisy values were used for all functions, allowing for direct comparison of the performance of each function. In total, 500 noisy time courses were generated for each metabolite in each model for the Direct Fit Method (described below), while an additional 50 time courses were used as the base data for the Resampling Method (described below).

Direct Fit Method

We refer to a basic nonlinear least squares fitting of parameters as the “Direct Fit” method for the purposes of this work. In this approach, we directly fitted each

function of interest to each noisy time course, $D_{i,m}$, to produce the smoothed time course estimate, $f_{i,j,m}$. Best-fit parameters for a given function were selected by minimizing the root-mean-square-displacement (RMSD) of the function to the data, defined as

$$RMSD_{i,j,m} = \sqrt{\sum_{k} \frac{\left(D_{i,m}(t_k) - f_{i,j,m}(t_k)\right)^2}{n - p_j}}$$

where i represents a specific metabolite, j represents a function being fitted, k represents an individual time point, m represents the use of a specific noisy data set, n is the number of sampled time points in the time course $D_{i,m}$, and p_j is the number of parameters being fit for function f_j. The denominator reflects a penalty on the number of parameters for a function, to help guard against over-fitting when comparing different functions[25].

Polynomials were fit using the built-in `polyfit()` function in MATLAB. Rational functions and the impulse function were fitted using `fmincon()` in MATLAB to allow for bounds on the parameter space, as described in the Supplementary Methods (found in Supplementary File 1). To improve the likelihood of finding globally optimal parameter sets for the rational and impulse functions, we selected optimal parameters from 20 solver runs seeded with different sets of initial conditions (see Supplementary Methods).

Resampling Method

In an approach we refer to as the "Resampling Method", we took advantage of the stabilizing effect of calculating the median of fits to multiple noisy datasets to produce more robust estimates of metabolite concentrations and derivatives.

Starting with the noisy time courses that model experimental data (described above), we generated resampled time courses by repeating the procedure used to produce the original noisy time courses, but using a noisy time course $D_{i,m}$ as input rather than the true metabolite concentration C_i. We again used a fixed CoV of 15% for this procedure; however, in practice, a dataset-specific and/or metabolite-specific CoV could be estimated and use in place of the fixed CoV. We generated 250 such resampled noisy time courses, $R_{i,m,w}$, for each initial noisy time course $D_{i,m}$.

We used the Direct Fit Method as described above to generate a nominal parameter solution from each base noisy time course $D_{i,m}$. Then, for each resampled time course $R_{i,m,w}$ derived from that noisy time course, we fit the function of interest (once) using the parameter solution from the Direct Fit Method as the initial guess. Parameter fitting was performed as described above.

We then used the resample-derived parameters to calculate concentration and derivative trajectories for each resampled time course $R_{i,m,w}$, and calculated the median value across all resampled time courses at the time points of interest (either the original or interpolated time points, as described below). The output of the Resampling Method was this list of concentration and derivative medians.

Performance Calculations

The performance of each fitting function using each method (direct and resampling) on both concentration and derivative predictions was quantified for each metabolite and for each base noisy time course, $D_{i,m}$. Concentration accuracy is useful for assessing the effectiveness of smoothing, while derivative accuracy is more relevant for downstream applications in estimating flux distributions[17]. Accuracy for each noisy time course $D_{i,m}$ was calculated using an adjusted RMSD between the synthetic reference data, C_i, and the predicted value for a given function, parameter set, and noisy data set, $f_{i,j,m}$. Specifically, we calculate accuracy as

$$RMSD_{i,j,m} = \frac{\sqrt{\sum_k \left(C_i(t_k) - f_{i,j,m}(t_k)\right)^2}}{n_I \cdot S \cdot \mu}$$

where

$$S = \sqrt{\frac{\sum_k \left(f_{i,j,m}(t_k)\right)^2}{n}}$$

$$\mu = \frac{n - p_j}{n}$$

and n_I is the number of time points used in assessing predictive accuracy, S is a scaling factor facilitating comparison and visualization by controlling for differences in the magnitude of different metabolites, and μ is a penalty factor scaling with the number of parameters in a function and the number of data points used to fit the function. For calculating derivative accuracy, the derivative values $f'_{i,j,m}(t_k)$ and $C'_i(t_k)$ are substituted in place of $f_{i,j,m}(t_k)$ and $C_i(t_k)$.

For these performance calculations, we more densely sampled metabolite concentration and derivative time courses to provide a more accurate representation of

interpolation performance, relevant to the general case of dynamic genome-scale metabolic modeling. For each model, results were sampled at time steps a factor of ten smaller than those used for the fitting data, resulting in n_I = 201 interpolated points for the *E. coli* model and n_I = 601 interpolated points for the *S. cerevisiae* model (these sets included the original sampled time points).

We ranked the functions' performance and averaged these ranks to provide a quantitative overall comparison of each function. We ranked the performance of each function for each noisy time course ($D_{i,m}$) of each metabolite and averaged the ranks for each function across all of these time courses. In both cases, a harmonic mean was used to average ranks, emphasizing the relative importance of comparing functions that perform strongly in some cases; in this way, the difference between rank 1 and rank 2 was weighted more heavily than the difference between, for example, rank 4 and rank 5.

This averaged rank approach was used to compare performance of fitting functions for the Direct Fit method only and for the Resampling Method only, as well as to compare performance between these two methods for all of the different fitting functions.

The MATLAB codes used to generate gold standard datasets, fit parameter values, calculate metrics, and plot metrics, are collectively available in Supplementary File 2.

Results

Two small-scale ODE metabolic models describing *E. coli* and *S. cerevisiae* metabolism were used to generate synthetic reference data for the assessment of new methods for concentration and flux inference from metabolite data. Using this synthetic reference data as a basis, noisy time courses were generated to represent the noisy data that typically result from metabolite profiling experiments. Eight different functions, including four polynomials, three rational functions, and one impulse model function (as described in the Methods section and in Table 1), were used as candidate fitting functions for these noisy metabolite time course data. Two different approaches were used to fit metabolite concentration curves to the noisy synthetic datasets generated from the original ODE models.

The Direct Fit Method, described in the Methods section, was a standard fitting of functions to given experimental data. The approach used to assess the effectiveness of the Direct Fit Method for each of the candidate fitting functions is outlined in Fig. 1. Briefly, after multiple noisy time courses were generated from the synthetic reference data, each candidate function was fitted to each of the noisy time courses. Each of these fits was then assessed for their performance at recapitulating and interpolating the original data; these assessments were performed on both the fitted concentrations and the derivative values that resulted from those fitted concentrations.

The Resampling Method, also described in the Methods section, involved fitting multiple noisy datasets generated from a single experimental (or noisy synthetic) dataset. By taking the median of these multiple fits, susceptibility to noise and outliers in the original experimental data was reduced, providing more robust estimates of

metabolite concentrations and derivatives. The approach used to assess the effectiveness of the Resampling Method for each of the candidate fitting functions is outlined in Fig. 2. Briefly, multiple "base" noisy time courses were generated from the original model to represent experimental measurements; these were fitted using the Direct Fit Method for comparison. In parallel, additional noisy time course profiles were generated ("resampled") from each of these base noisy time courses and subsequently fitted using the methods described for the Direct Fit Method—yielding a fitted concentration for each resampled noisy time course for a given base noisy time course. For each base noisy time course, the median per time point of the fitted profiles (or profile derivatives) for the resampled noisy time courses was then used to determine the overall fitted profile. This profile, along with the Direct Fit Method profile, was compared to the original synthetic reference data to assess prediction accuracy.

Parameter constraints improved the behavior of fitted results

Fig. 3 provides representative examples of performance for different candidate fitting functions using the Direct Fit Method and the *E. coli* model. Polynomial functions provided computationally efficient data smoothing with little susceptibility to noise, but had limited abilities to qualitatively capture the dynamics present in the *E. coli* model. For certain sets of noisy data, the rational functions or the impulse function returned unphysical or unreasonable results. This result highlighted a shortcoming in the basic implementation of the rational functions and prompted the development of additional

322 constraints for use in the optimization step of fitting the rational functions and the
323 impulse function.

324 We observed that for approximately 29% of noisy datasets, the R_{22} rational
325 function produced asymptotic behavior, as shown in Fig. 3D. The frequency of
326 asymptote occurrence varied significantly across the different metabolites in the model,
327 as shown in Fig. S3A. The source of these asymptotes was selection of “optimal”
328 parameters such that the polynomial in the denominator of R_{22} had a root over the time
329 range of the data. Technically, such parameter selections would be optimal based on
330 the RMSD objective function, since the RMSD only considers the ability of the function
331 to match the data provided for fitting. However, such selections lead to clearly
332 unphysical profiles at interpolated points that would confound any efforts to use such
333 fitted functions in genome-scale metabolic simulations. Accordingly, we constrained the
334 RMSD optimization for all rational functions (as described in detail in the Supplementary
335 Methods, Fig. S3, and Table S4) such that parameters could not be selected that would
336 cause a zero in the denominator over the time range of the data. Fig. 3E shows the
337 trajectory of R_{22} after adding additional constraints to the allowed parameter values in
338 rational functions. However, this solution does not protect against near-asymptotic
339 behavior in R_{22}, where the denominator approaches but does not reach zero; Fig. 3F
340 depicts such a case using a different set of noisy data for the same metabolite.
341 Nonetheless, the results in Fig. 3E demonstrate significant improvement upon the
342 results from Fig. 3D with no parameter constraints.

The impulse function exhibited a similar phenomenon, insofar as it yielded results that were technically correct based on the RMSD optimization function but were physically unreasonable. As depicted in Fig. 3B, the impulse function sometimes produced sharp shifts in concentration, which translated to sharp spikes in the derivative trajectory. In addition, we noticed that our parameter-fitting solver was prone to getting stuck in local minima when the resulting time delay parameters were outside the time span of the data. These observations led us to implement an additional parameter constraint strategy described in more detail in the Supplementary Methods.

Briefly, one fixed constraint and two new adjustable optimization parameters were created that were used to constrain the possible parameter space. Since any arbitrary dataset would not provide evidence for a sigmoidal shift outside of the time range of the data, we constrained the possible sigmoidal response times to only be within the time range of the data. We then defined two parameters, h_f and b_f, to further constrain the parameter space based on the data. Since an arbitrary dataset would not provide evidence for initial steady state, intermediate state, and final steady state levels far outside of the range of the measured metabolite concentrations, the deviation of function values above the maximum and below the minimum measured values was constrained to be no more than h_f times the range of the metabolite data (with an additional non-negativity constraint). Since an arbitrary dataset would not provide evidence for concentration changes at a higher frequency than that of the sampling frequency, sharp transitions between time points are unlikely to be realistic. Thus, the steepness of the sigmoidal shift was constrained to be less than a value proportional to

the range of the data divided by the time difference between data points, with b_f as the proportionality constant.

Using h_f=0.1 and b_f=0.5 resulted in more realistic profiles like those shown in Fig. 3C. Importantly, in addition to the direct physical interpretation of these, the results of the parameter fitting are not highly sensitive to small changes in h_f and b_f (see Fig. S4), and as a result the values of h_f and b_f that we used were generalizable to both model systems even though they were selected only based on their performance for the *E. coli* model.

The impulse model consistently fits metabolite data with low error

To quantitatively assess the effectiveness of the candidate fitting functions using the Direct Fit Method in the *E. coli* model, we generated 500 noisy time course data sets for each of the 17 metabolites. The parameters resulting from fitting each noisy time course were used to calculate concentration and derivative trajectories, with the corresponding performance accuracy calculated and averaged as described in the Methods section. The results of these calculations are summarized in Table 2, which presents the averaged ranks for each function and each metric. Fig. 4A and 4B provide a detailed quantitative comparison of each fitting function. The impulse function, I, showed the best rank averages for accuracy in both concentration and derivatives, and was almost always the best-performing function across all of the metabolites.

The notable exceptions to the superior performance of the impulse function were on Metabolites 12 and 18. Fig. 5 summarizes the performance of the impulse function

and an average fitting function, P_4, for Metabolite 12, with representative fitted profiles in Fig. 5A and 5B, and a direct comparison between the performance of P_4 and I in Fig. 5C. P_4 consistently performed better than I. However, as is clear from Fig. 5A and 5B, the total change in metabolite level was smaller than the expected range of variability of experimental measurements. Given the sparsity of samples, this metabolite's profile is likely essentially unidentifiable, and so the performance of the different functions is likely based only on general trends of the functional forms near the ends of the time range, rather than any reliably accurate fitting.

The Resampling Method can improve fitting and predictions in the E. coli *model*

To quantitatively assess the performance of the Resampling Method in the *E. coli* model, we generated 50 noisy time courses from the synthetic reference data for each of the 17 metabolites, and for each noisy time course, an additional 250 resampled noisy time courses. For each noisy and resampled time course, each function was fitted as described in the Methods, and the resulting Direct Fit or Resampling Method trajectories used to calculate the performance metrics. The overall results are shown in Table 3. Results jointly ranking the performance of functions across both the Direct Fit Method and the Resampling Method are shown in Table 4. The Resampling Method had the greatest impact on the ranking of the rational function R_{22}, resulting in it being similar in accuracy and consistency to the impulse function, I. This consistently good

performance is also evident in Fig. 4C and 4D, which provide a detailed quantitative comparison of each fitting function.

The impacts of the Resampling Method varied across the different types of functions; representative graphs are presented in Fig. 6, with a complete summary provided in Table 4. Polynomial functions showed little to no change in results from using the Resampling Method, while rational functions show moderate to noticeable benefit. The impulse function benefited in some cases as well. Across all functions, use of the Resampling Method only infrequently caused decreased performance, and typically with very small changes relative to the magnitude of the error.

S. cerevisiae *model results show similar trends*

We then quantitatively assessed the performance of all candidate fitting functions using both the Direct Fit Method and the Resampling Method in the *S. cerevisiae* model. We generated 500 noisy time courses for each of the 21 metabolites for use in the Direct Fit method. For use in the Resampling Method we generated 50 base noisy time courses for each of the 21 metabolites, along with an additional 250 resampled noisy time courses for each base noisy time course. Parameters were fit for each method as described in the Methods section. Tables 5 and 6 present the average ranks for the Direct Method and Resampling Method, both separately and combined, respectively. Fig. 7 provides a detailed quantitative comparison of each fitting function. For this model, the R_{22} rational function and the impulse function, I, were usually among the

best-performing fitting functions, with R_{22} performing best for concentrations and I performing best for derivatives.

Discussion

The goal of this work was to improve the prediction of concentration and derivative time-course profiles derived from experimentally measured (or synthetic, noisy) metabolite data. Two small-scale model metabolic systems were used as the basis for assessing the performance of new methods to calculate and interpolate concentration and flux values based on metabolite data. These two models have different time scales and dynamics, which provided a broader assessment of the potential utility of our approaches. These models were also used in previous work on estimating flux distributions from metabolite data[18], which allowed for direct comparison. Integrating these systems numerically provided an exact reference dataset to which we could compare fitted results. However, real metabolite concentration data contain significant variability, so we only used noisy synthetic data derived from this reference data to test the effectiveness of our approaches. In this way, we were able to generate data of defined quality and arbitrary quantity with known underlying dynamics; this allowed us to precisely and rigorously determine the performance of each approach under study.

The approach of Ishii *et al*. was to fit all of the functions to the time course in question and select the function with the lowest fitting error, once accounting for the number of fitted parameters[18]. While this is certainly a viable approach that can be

extended to include the sigmoidal impulse model, here we have also investigated whether this single, biologically reasonable function can be used instead of selecting the best-fitting function from a list of arbitrary candidates. We consider the relative benefits of each function type below.

Polynomials are consistent but inaccurate

The polynomial functions are computationally inexpensive to fit, use few parameters (ranging from three to six), and are widely used for smoothing noisy data. They are consistent and well-behaved, exhibiting very little sensitivity to noise. (As described in Supplementary Methods and Tables S2 and S3, robustness of smoothed profiles to noise was also assessed, but was found to closely depend on the number of parameters used in a function and essentially represented a tradeoff between consistency and accuracy of fitting.) As demonstrated by their ranks in Tables 2, 3, and 5, they can do a reasonable job in estimating concentrations and at times even in estimating derivatives (ranking as low as 2.5 but often closer to 3.5 or 4). However, they are ill-suited to capturing dynamics that include a terminal steady state, particularly since their functional form requires them to be monotonically increasing or decreasing at the ends of the time range; this also makes them a poor choice for even limited extrapolation.

Resampling improves rational function accuracy

The rational functions (using three to five parameters) can exhibit a wider range of behaviors than the polynomials with the same number of parameters, and it has been reported that for many metabolite time courses, they yield better performance than the polynomials[18]. Our parameter restriction strategy was largely effective in addressing their potential to fit best with parameters that produce asymptotic behavior, though there are still lingering issues with near-asymptotes that yield spurious behavior and even negative concentrations for the R_{22} function (see Fig. 2F). However, as shown in Table 3, this effect is largely ameliorated by the use of the Resampling Method to filter out asymptotic trajectories, making R_{22} one of the more effective functions we studied.

The impulse function is a generally effective single fitting function model

The last function, the sigmoidal impulse, is the product of two sigmoidal logistic functions[20, 21]. As previously stated, it recapitulates the dynamics of a common biological process: a two-phase transition from one steady state to a (potentially new) steady state through an intermediate state. Its parameters directly correspond to features of this trajectory: the h parameters represent the initial, intermediate, and steady-state metabolite levels; the τ parameters represent the timing of the on and off transitions (accumulation and depletion driven by processes such as synthesis and degradation) in response to a perturbation; and the β parameters represent how rapidly those transition processes occur. In contrast with the work done by Chechik *et al.*, we allowed the β parameters to vary independently to reflect the fact that the on and off

493 transitions can represent different biological processes (e.g., glucose uptake versus
494 metabolism), which one would reasonably expect to exhibit distinct dynamics[20].
495 While potentially exhibiting undesirable behaviors with unrestricted parameter
496 values, our parameter bounding strategies for avoiding broad local minima and overly
497 sharp curves were effective at preventing these undesirable behaviors (Fig. 3B and 3C).
498 Of particular note is that these parameters themselves typically exhibited broad local
499 optima in performance (Fig. S4), meaning that the fitting method was not very sensitive
500 to the specific values selected; additionally, the default parameters we selected for the
501 *E. coli* model generalized well to a completely separate model, meaning that while they
502 are technically adjustable parameters, they did not add significant risk of over-fitting to
503 the parameter selection process.
504 Using the Direct Fit Method for the *E. coli* model, the impulse function performed
505 consistently better than other functions (see Table 2) across all metabolites except for
506 two: metabolites 12 and 18. For these metabolites, the actual dynamic range of
507 metabolite concentrations in the synthetic reference data was substantially less than the
508 range of the random noise used to construct the noisy time courses (see Fig. 5). We
509 cannot realistically expect to recover the underlying concentration in this case without
510 either much more dense or much more accurate sampling. We suspect that the better
511 performance of the polynomials was due in part to their tendency to swing upwards or
512 downwards near the edges of the data, which captured the early time dynamics of each
513 of these metabolites well; we note that the other high-performing fitting function, R_{22}, did
514 poorly on these metabolites as well. The Resampling Method substantially improved the

performance of R_{22} and slightly improved the performance of the impulse function on these metabolites (Fig. 4), leading to qualitative behavior where the derivative effectively fluctuated around zero. Given the lack of statistically significant change over the time course of these metabolites, we argue that this is the behavior we should not only expect, but actually be seeking given the essentially unidentifiable change in metabolite levels.

The Resampling Method generally improves on Direct Fit Method results

In general, the resampling method ranged from negligibly detrimental to highly beneficial. In a few cases, a very minor loss of performance was observed. Consistently, resampling provided no benefit to polynomials (Fig. 6A); this is to be expected, since the polynomial functions are already insensitive to small changes in the data. The R_{11} and R_{31} rational functions saw minor improvements in general, while the impulse function saw improvements in cases where it performed most poorly (Fig 5C). The Resampling Method had the biggest effect on R_{22}; in the *E. coli* model, it moved from one of the worst performers to one of the overall best (Fig. 4, Table 4). Generally speaking, then, the Resampling Method seems to be an effective way to improve accuracy at only a mild computational cost.

The Resampling Method appears to have an effect similar to parameter regularization by avoiding over-fitting due to noisy data[26]. However, we note that the Resampling Method returns a median of multiple fits, rather than a single parameter set. As a result, concentration and derivative values derived from this method need not

strictly adhere to the functional form of the smoothing function; this flexibility can allow better approximation of the underlying data in cases where the form of the particular function happens to be biased against the correct behavior.

S. cerevisiae *model results generally recapitulate* E. coli *model results*

The *S. cerevisiae* model generally recapitulated results from the *E. coli* model, demonstrating the potential generalizability of the Resampling Method and the impulse function (including the parameters used to restrict the fitting search space for the impulse function). For both the Direct Fit and Resampling Methods, the impulse function performed fairly well. One feature that distinguished the *S. cerevisiae* model from the *E. coli* model was the wider range of time scales present in the model's dynamics. Several metabolites (1-4,8-10,18-20) reached steady-state in several minutes, while others (12,13,14) took tens of minutes, and as a result did not reach steady-state during the time interval of the data. As the impulse function assumes long-term steady-state behavior for the time course, it did not perform as strongly for the Direct Fit Method for these metabolites. However, the Resampling Method did provide some improvement for these metabolites.

Selection of fitting functions should be driven by applications

In this work we considered the problem of data smoothing specifically in the context of genome-scale metabolic modeling. Two key factors in this application have driven our assessment of function and method performance. First, we expect that we

may need to provide flux values at points other than those for which experimental measurements are available (for instance, if a genome-scale model entails something akin to a Runge-Kutta numerical integration). This means that function accuracy should be assessed not only at the sampled points, but in between them as well. Without the inclusion of such interpolated values, some differences can be seen in apparent effectiveness; for example, previous work indicated that polynomials were more frequently optimal for the *S. cerevisiae* model[18], but in terms of practical applications they are usually inferior to R_{22} and the impulse function. Second, the main application of the metabolite concentration smoothing is for the estimation of metabolite fluxes; this means that while recapitulating the concentration profile is important, the more directly applicable metric is how accurate the derivative profile is. This distinction is most relevant for the *S. cerevisiae* model, where R_{22} more accurately recapitulates concentrations, but the impulse model more accurately recapitulates the derivatives that will be used in downstream analyses.

Single functions and biologically-inspired functions can be effective fitting models

While previous work selected the best-fitting of an essentially arbitrary set of functions for each individual metabolite based on the experimental data, we suggest that this may be a suboptimal approach. First, this increases the likelihood for over-fitting; it is difficult to estimate the number of effective parameters that are introduced to the system by allowing for the variable selection of seven different models, but it suffices

to say that the number of effective parameters is likely greater than the number of explicit parameters in the highest-order polynomial. As such, restricting the fitting to one function may be desirable from an information content perspective; both the R_{22} and impulse functions seem like reasonable, viable candidates for universal fitting functions. In fact, once the assessment metrics are based on a criterion more reasonable for the application (i.e., inclusion of interpolated points), there are few if any cases where the polynomials would be a desirable option. Second, there is inherent value in using biologically-inspired fitting functions. These functions, by design, recapitulate behaviors previously observed in biological systems; biasing the fit towards these results integrates prior knowledge that may help ensure that the model is closer to the underlying biology. Even though there are more parameters in these functions than the polynomials, the space of characteristic curves that can be fit is more restrictive and more relevant to expected biology, partially mitigating concerns about over-fitting due to excess parameters. In this sense, the impulse function may be the most desirable choice; either way, applying the Resampling Method ensures that the smoothing and fitting is improved over previous approaches.

Limitations

There are a few limitations to our analyses that bear noting. First, the number of variable parameters in the impulse function places a lower limit on the number of samples needed to fit the function well, which could stretch the experimental feasibility of acquiring a sufficient number of samples. However, our analyses have been

consistent with previous work in terms of the number of samples used, and considering the possibility of using multiple biological replicates and multiple experiments to fit the same data, obtaining one or two dozen samples is often reasonable for a metabolomics experiment. Second, the impulse model assumes a steady state is reached at the end of the experiment, which may not be valid for all datasets. However, this concern is partially mitigated by the fact that many experiments would actually be continued until something more closely resembling a steady state is reached, minimizing the number of times significant non-zero derivatives were present at the end of the time range. There is also an obvious computational cost to fitting non-linearizable functions (as opposed to polynomials) and to applying the Resampling Method; however, since the data smoothing task is ultimately performed just once, not many times, we believe that the improvement in results is worth this computational cost, which is itself reasonable and does not require parallelization or even particularly long runtimes. Finally, we have not analyzed the ultimate downstream impacts in the genome-scale metabolic modeling application of the improvements we have made to assess their magnitude. Based on the tendency of functions like polynomials to have nonzero derivatives at the end of the time range and the importance of being able to capture a steady state in a metabolic model, we expect that these improvements may be important, but will be to some extent model-specific and is thus beyond the scope of this work. Either way, it is often generally accepted that optimization of each intervening analysis or data processing step is desirable for complex modeling schema.

Conclusions

In this work, we have demonstrated two improvements to standard approaches to smooth metabolite concentration data for application to genome-scale metabolic modeling, including a Resampling Method to minimize susceptibility to experimental noise and the establishment of a single, biologically-inspired fitting function that performs well in almost all cases. In the course of this work, we also identified additional constraints that should be applied to existing data smoothing fitting functions to increase their robustness and activity. Taken together, these contributions have provided consistent and substantial improvements in existing methods to smooth and fit metabolite data for downstream applications, whether via a new fitting function or improvements made to existing fitting functions. We have shown these results to be generalizable across multiple models of metabolism, suggesting the potential for general utility of these improved methods to improve the accuracy of flux distributions calculated from the derivatives of their time courses.

Acknowledgements

RAD participated in the design of the study, carried out the computational experiments, and helped to draft the manuscript. MPS conceived of the study, participated in its design and coordination, and helped to draft the manuscript. All authors read and approved the final manuscript. RAD was supported by NSF IGERT award # DGE 0965945, and MPS and RAD were supported by NSF award # 1254382. We would also like to thank McKenzie Smith and Amy Su for their feedback on the manuscript draft.

647

Table 1. Fitting functions evaluated in this work.

Name	Formula
P_2	$C(t) = p_1 \cdot t^2 + p_2 \cdot t + p_3$
P_2	$C(t) = p_1 \cdot t^3 + p_2 \cdot t^2 + p_3 \cdot t + p_4$
P_4	$C(t) = p_1 \cdot t^4 + p_2 \cdot t^2 + p_3 \cdot t^2 + p_4 \cdot t + p_5$
P_5	$C(t) = p_1 \cdot t^5 + p_2 \cdot t^4 + p_3 \cdot t^3 + p_4 \cdot t^2 + p_5 \cdot t + p_6$
R_{11}	$C(t) = \frac{p_1 \cdot t + p_2}{t + p_3}$
R_{22}	$C(t) = \frac{p_1 \cdot t^2 + p_2 \cdot t + p_3}{t^2 + p_4 \cdot t + p_5}$
R_{31}	$C(t) = \frac{p_1 \cdot t^3 + p_2 \cdot t^2 + p_3 \cdot t + p_4}{t + p_5}$
I	$C(t) = \frac{1}{h_1} \cdot s(t, \tau_1, h_0, \beta_1) \cdot s(t, \tau_2, h_2, \beta_2)$ $s(t, \tau, h, \beta) = h + \frac{(h_1 - h)}{1 + e^{-4\beta(t-\tau)}}$

Fig. 1. Schematic of the Direct Fit Method.
Synthetic gold standard data are generated by simulating a system of ODEs over the time interval of interest. From the synthetic data, noisy time courses are generated by adding Gaussian noise with a 15% coefficient of variation to the synthetic data, to simulate experimental sources of variation in measurements. Multiple such noisy time courses are generated. A smoothing function is fit directly to a noisy time course, and the resulting fit (or its derivative) is compared against the synthetic data to determine how closely they match. The performance of each function can then be compared based on their performance relative to the initial synthetic data.

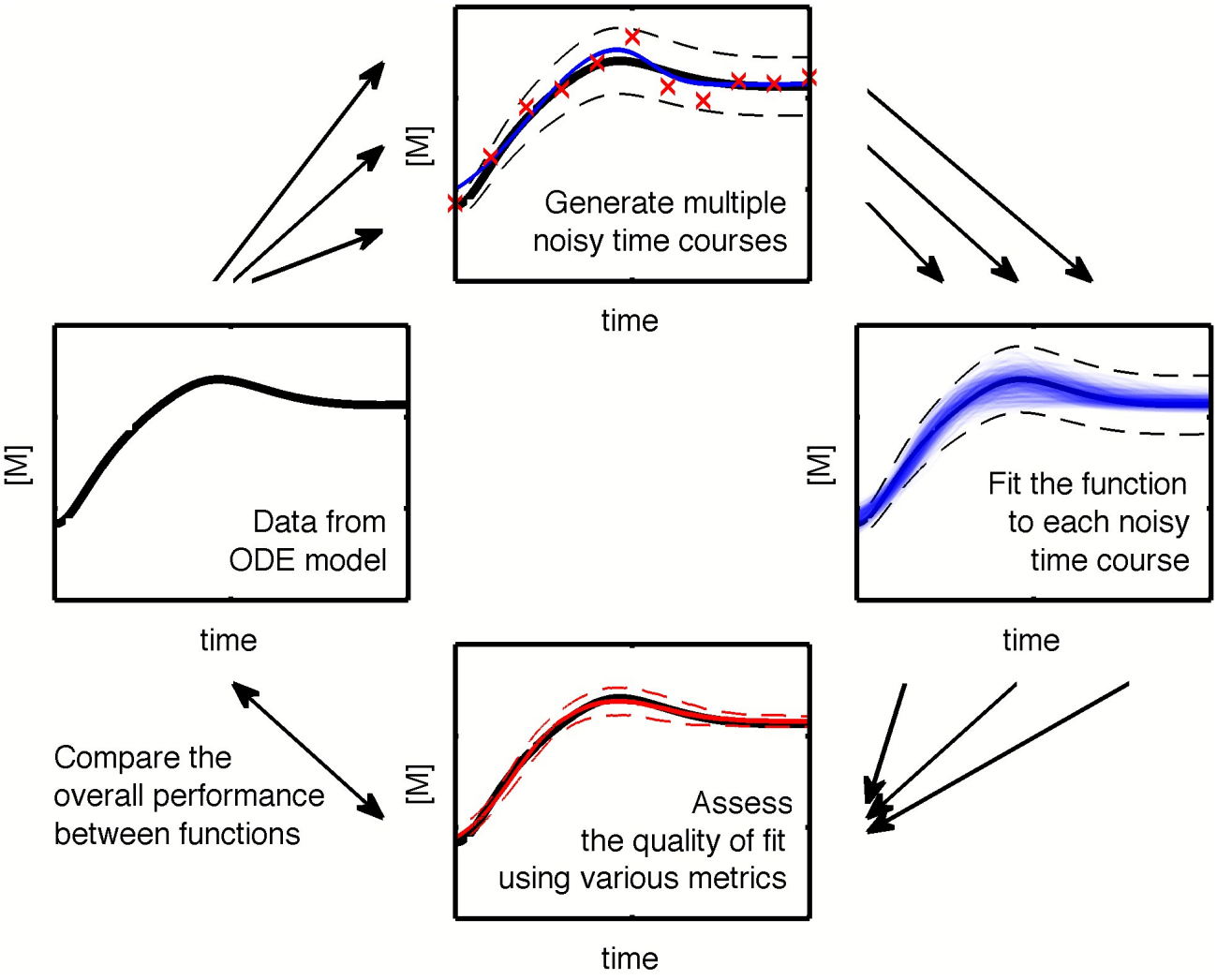


Fig. 2. Schematic of the Resampling Method.
As in the Direct Fit method, synthetic data and base noisy time courses are generated from a system of ODEs. In the Resampling Method, each base noisy time course is then used to generate a set of "Resampled" time courses, by using the same process used to generate the base noisy time courses from the synthetic data, only now with the base noisy time course as the input. The function of interest is fit to each of these resampled time courses, and the median of these functions (or their derivatives) is used to generate the resulting smoothed time course corresponding to the specific base noisy time course. As in the Direct Fit method, these median profiles can be assessed to determine accuracy and performance of the function.

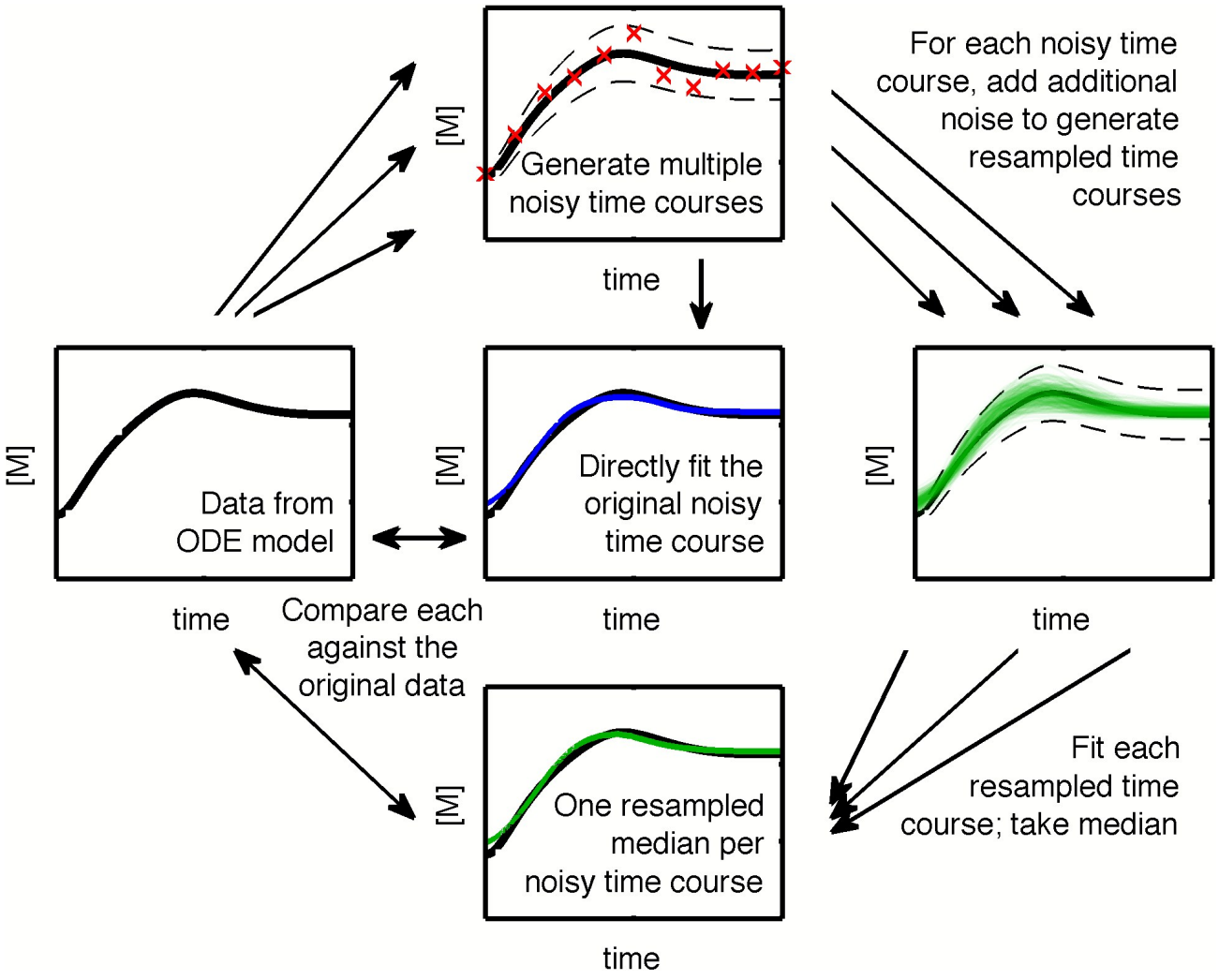
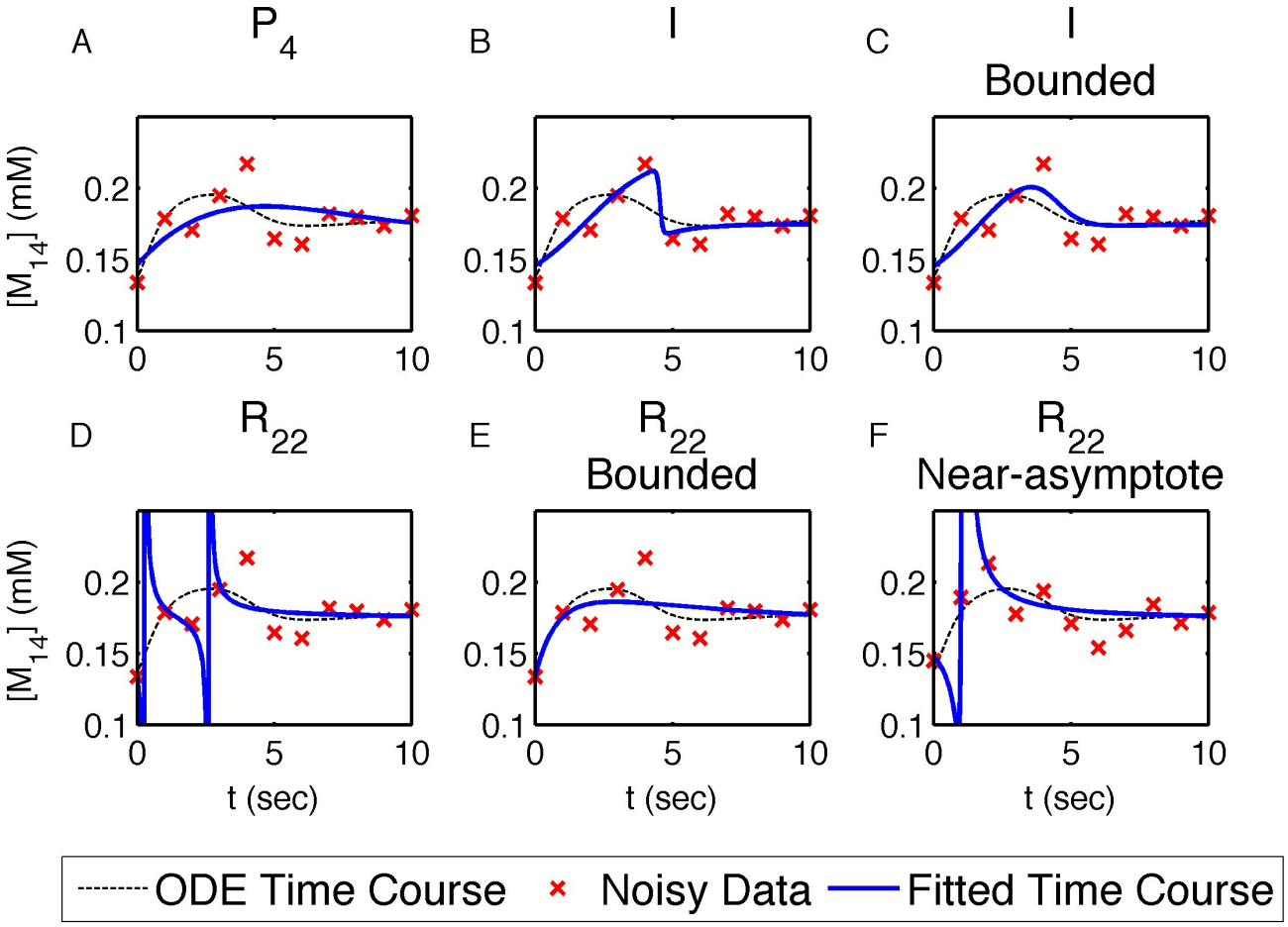


Fig. 3. Performance of different fitting functions for fitting concentration trajectories. Thin, dotted black lines are the original synthetic data. Red crosses are the noisy time course data used to fit the functions. Solid blue lines are the function fitted to the data. A) Polynomial curves were consistent but typically not very accurate. B) The sigmoidal impulse function performed well but sometimes exhibited steep derivatives. C) Constraining the parameter space for the impulse function prevented this behavior. D) The rational function R_{22} can exhibit unphysical asymptotes in the time interval of the data due to a polynomial term in the denominator. E) Constraining the parameter space for R_{22} prevents such asymptotes. F) However, near-asymptote behavior can still occur in the rational functions, despite the parameter restrictions, when the value of the denominator polynomial becomes sufficiently small. Note: A-E all use the same noisy data set.



691 Table 2. Average rank of function accuracy using the Direct Fit method on the *E. coli*
692 model.
693

Average Rank of Metric	P_2	P_3	P_4	P_5	R_{11}	R_{22}	R_{31}	I
Concentration Accuracy	3.68	4.13	2.50	2.94	3.94	2.33	4.83	1.74
Derivative Accuracy	3.18	3.45	2.48	3.08	3.58	2.61	3.77	2.18

694

Fig. 4. Quantitative assessment of function accuracy across metabolites in the *E. coli* model.
The impulse function performs consistently well across most metabolites for both (A) concentration and (B) derivative accuracy. The resampling method improves the performance of a number of functions for both (C) concentration and (D) derivative accuracy. Error metrics are normalized to average metabolite concentrations (see Methods) for easier visualization and are presented in log-transformed format.

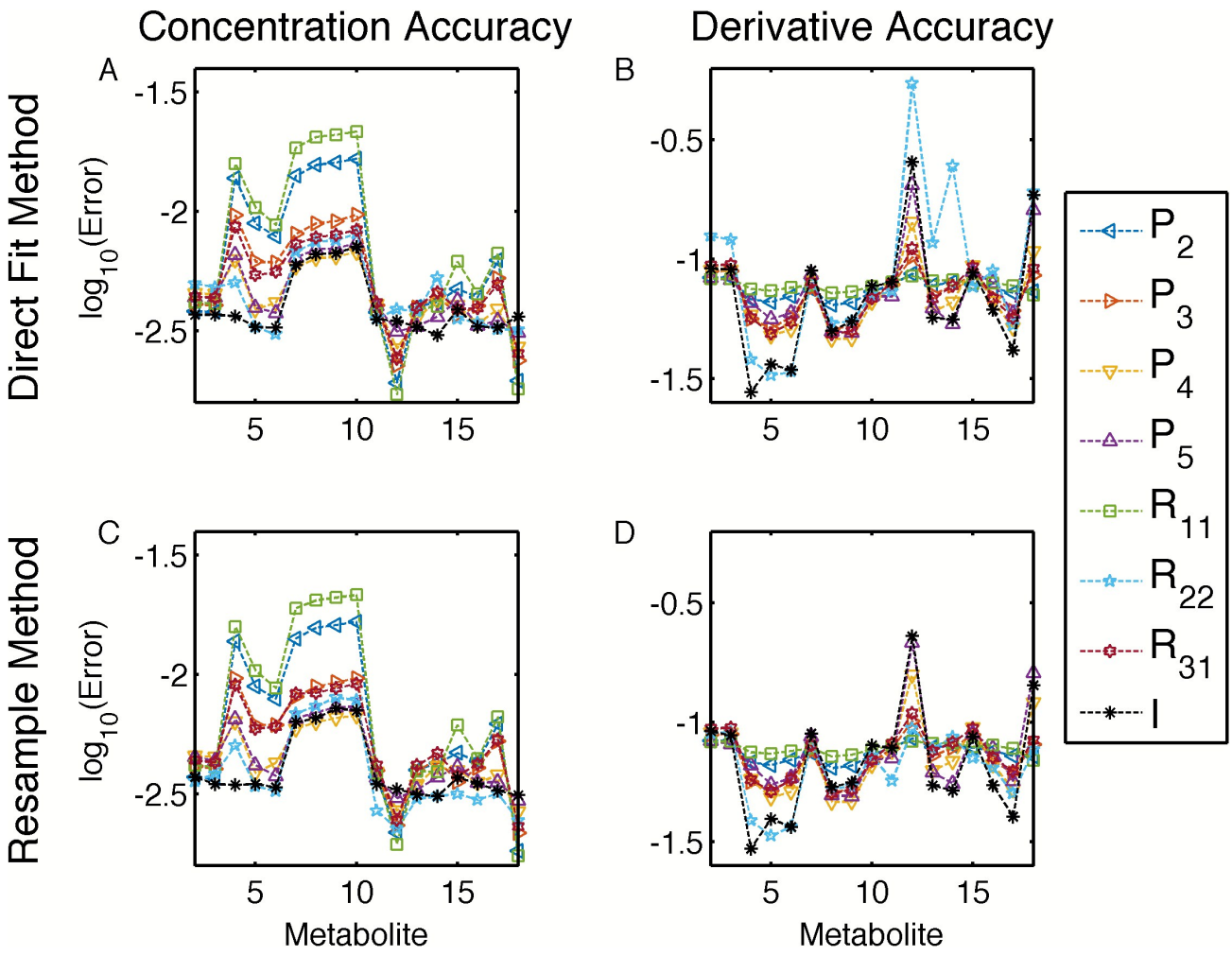
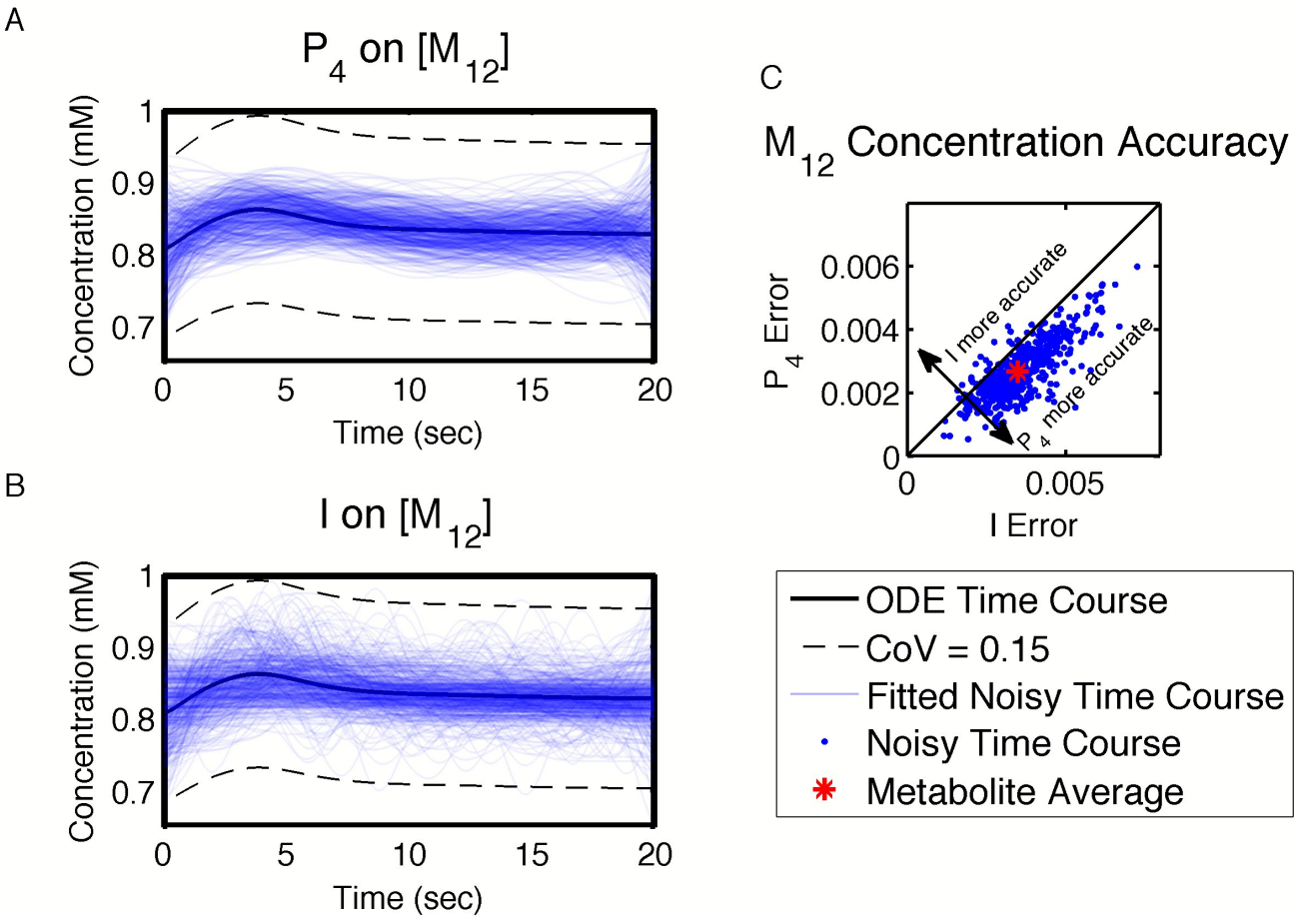


Fig. 5. Comparison of the Impulse and P_4 on Metabolite 12 (6-Phosphogluconate) over 500 random noisy time courses.
A) The P_4 polynomial function intrinsically curves upwards or downwards at the ends of the interval, which helps match the early slope in the synthetic data. B) The impulse function exhibits greater variability across different noisy replicates due to the small dynamic concentration range in the synthetic data relative to the noise introduced. Solid black lines indicate the synthetic data. Dashed black lines indicate the 15% coefficient of variation envelope, used to generate the noisy time course data. Blue lines indicate the concentration trajectory of functional fits to individual noisy time courses. C) As a result, the P_4 polynomial consistently fits the synthetic data concentration with lower error than the impulse. Blue dots indicate the error of each function in recapitulating the synthetic data when fit to a particular noisy time course. The red star indicates the average error of the blue dots.



721 Table 3. Average rank of function accuracy using the Resampling Method on the *E. coli*
722 model.
723

Average Rank of Metric	P_2	P_3	P_4	P_5	R_{11}	R_{22}	R_{31}	I
Concentration Accuracy	4.02	4.16	2.44	3.11	4.22	1.83	5.32	1.90
Derivative Accuracy	3.38	3.40	2.50	3.07	3.68	2.16	4.66	2.20

724

Fig. 6. The effect of the Resampling Method on the derivative accuracy of three representative functions.
The error for fitted concentration profiles was determined for both the Direct Fit and Resampling Methods and directly compared. A) For polynomial functions the Resampling Method produces results nearly identical to the Direct Fit method. B) The R_{22} rational function can produce derivative errors several orders of magnitude greater using the Direct Fit method (not shown on these axes) than when using the Resampling Method, making the Resampling Method more accurate on average. C) The impulse function is generally consistent between the Direct Fit and Resampling Methods, but does show some variability. Other metabolites exhibit modest benefits from the Resampling Method relative to the Direct Fit Method.

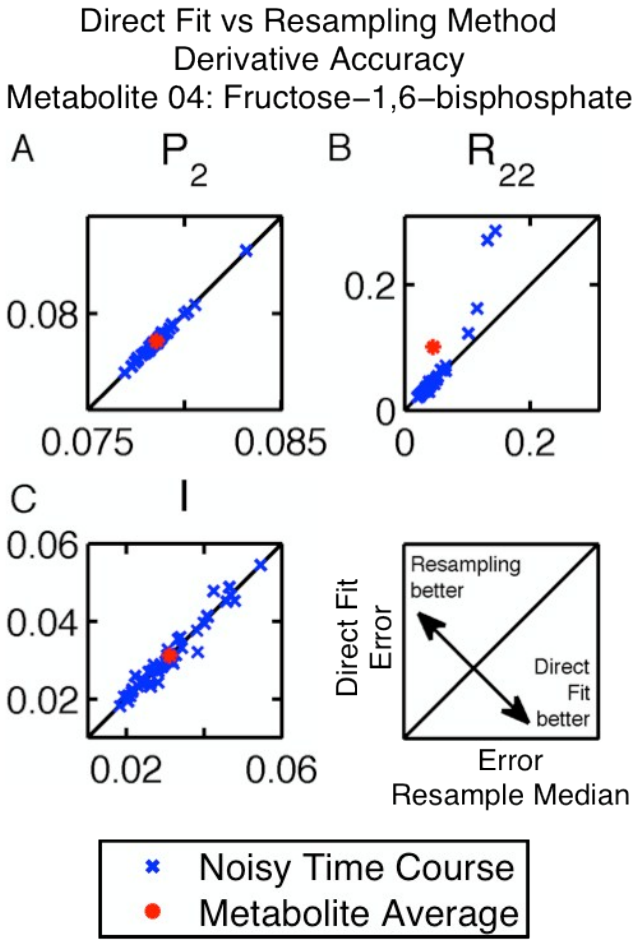


Table 4. Average rank of function and method accuracy using the *E.coli* model. Results from both the Direct Fit (DF) and Resampling (RM) methods are all ranked together to facilitate direct comparison of their performance.

	P_2		P_3		P_4		P_5		R_{11}		R_{22}		R_{31}		I	
Average Rank of Metric	DF	RM	DF	RM	DF	RM	DF	RM	DF	RM	DF	RM	DF	RM	DF	RM
Concentration Accuracy	6.62	6.70	7.36	7.35	3.76	3.94	5.34	5.35	7.17	6.62	3.48	2.55	8.77	10.17	2.60	2.88
Derivative Accuracy	5.40	5.50	6.20	6.21	3.98	4.02	5.12	5.09	6.49	5.85	3.76	3.12	6.33	8.96	3.30	3.17

744 Table 5. Average rank of function accuracy using the *S. cerevisiae* model. Here, the
745 Direct Fit and Resampling Methods are ranked and averaged separately.
746

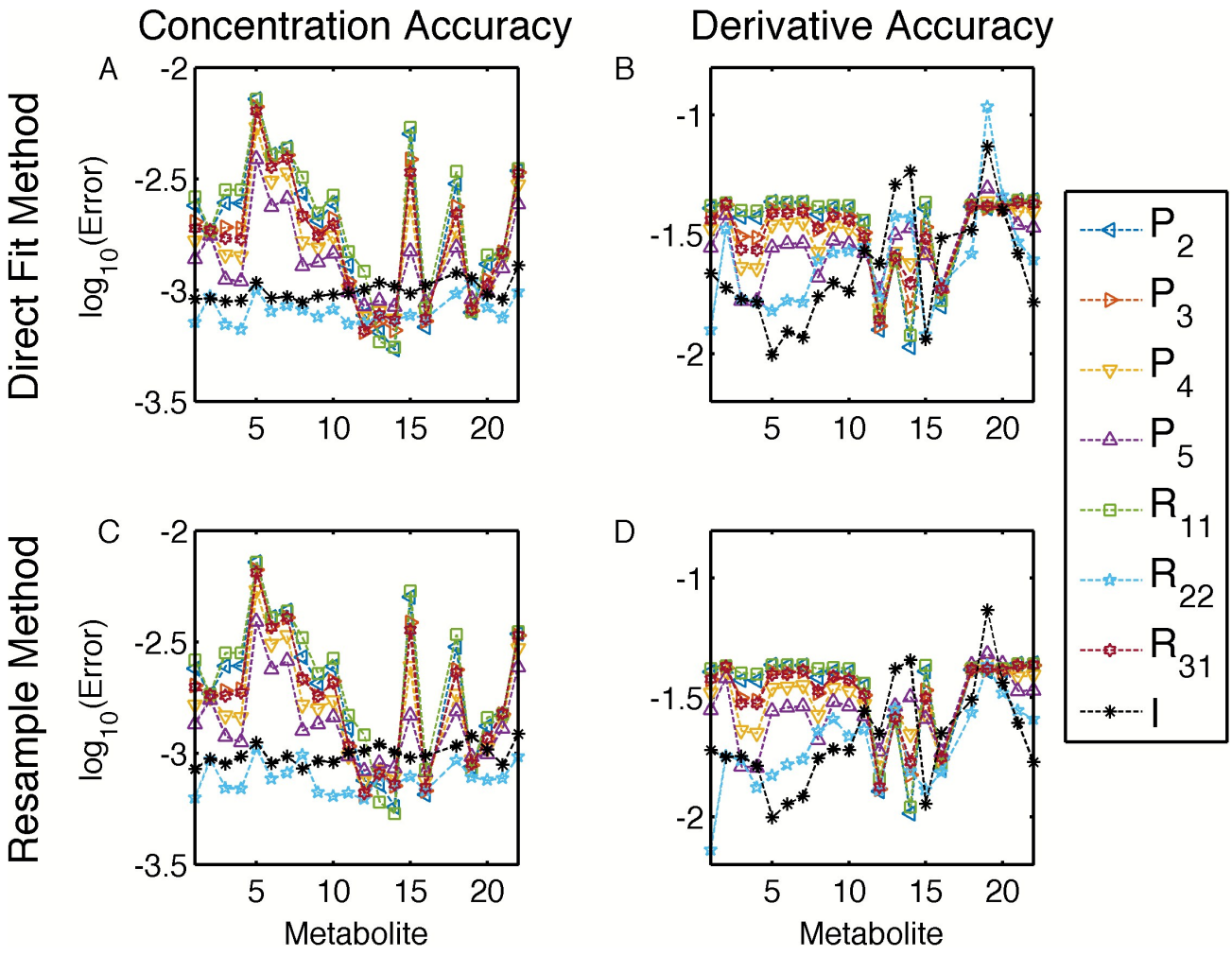
	Direct Fit Method								Resampling Method							
Average Rank of Metric	P_2	P_3	P_4	P_5	R_{11}	R_{22}	R_{31}	I	P_2	P_3	P_4	P_5	R_{11}	R_{22}	R_{31}	I
Concentration Accuracy	4.28	4.00	3.83	3.22	4.81	1.34	4.45	2.07	4.48	4.15	3.90	3.33	4.82	1.24	4.79	2.10
Derivative Accuracy	3.99	3.65	3.55	2.77	4.80	1.95	4.44	1.66	4.39	4.00	3.81	2.92	4.81	1.61	5.06	1.64

747

Table 6. Average rank of function and method accuracy using the *S. cerevisiae* model. Results from both the Direct Fit (DF) and Resampling (RM) methods are all ranked together to facilitate direct comparison of their performance.

	P_2		P_3		P_4		P_5		R_{11}		R_{22}		R_{31}		I	
Average Rank of Metric	DF	RM	DF	RM	DF	RM	DF	RM	DF	RM	DF	RM	DF	RM	DF	RM
Concentration Accuracy	7.37	7.82	7.05	7.55	7.14	7.17	5.86	6.02	7.92	7.98	1.85	1.65	7.85	8.98	3.59	3.22
Derivative Accuracy	7.52	7.41	7.16	6.75	6.64	6.74	4.79	4.85	8.23	8.10	2.95	2.14	8.34	9.43	2.72	2.15

Fig. 7. Quantitative assessment of function accuracy across metabolites in the *S. cerevisiae* model.
Results by metric are presented for the Direct Fit Method for (A) concentration accuracy and (B) derivative accuracy, and for the Resampling Method for (C) concentration accuracy and (D) derivative accuracy. Error metrics are normalized to average metabolite concentrations (see Methods) for easier visualization and are presented in log-transformed format.



Notes and References

"Supplementary File 1.pdf" contains Fig. S1-S5, Tables S1-S4, and Supplementary Methods. [PDF, 4.9MB]

"Supplementary File 2.zip" contains an archive of the code used to generate datasets, fit parameter values, calculate metrics, and plot metrics; and descriptions of file contents and directions on use. [ZIP, 8.2MB]

1. K. Yizhak, E. Gaude, S. Le Dévédec, Y. Y. Waldman, G. Y. Stein, B. van de Water, C. Frezza and E. Ruppin, *eLife*, 2014, **3**, e03641.
2. A. P. Burgard, P. Pharkya and C. D. Maranas, *Biotechnol Bioeng*, 2003, **84**, 647-657.
3. K. Nakahigashi, Y. Toya, N. Ishii, T. Soga, M. Hasegawa, H. Watanabe, Y. Takai, M. Honma, H. Mori and M. Tomita, *Mol Sys Biol*, 2009, **5**, n/a-n/a.
4. M. W. Covert, C. H. Schilling and B. Palsson, *J Theor Biol*, 2001, **213**, 73-88.
5. M. W. Covert, N. Xiao, T. J. Chen and J. R. Karr, *Bioinformatics*, 2008, **24**, 2044-2050.
6. J. Min Lee, E. P. Gianchandani, J. A. Eddy and J. A. Papin, *PLoS Comput Biol*, 2008, **4**, e1000086.
7. C. Cotten and J. Reed, *BMC Bioinformatics*, 2013, **14**, 32.
8. D. McCloskey, B. Ø. Palsson and A. M. Feist, *Mol Sys Biol*, 2013, **9**, 661.
9. C. S. Henry, L. J. Broadbelt and V. Hatzimanikatis, *Biophysical J*, 2007, **92**, 1792-1805.
10. A. Kümmel, S. Panke and M. Heinemann, *Mol Sys Biol*, 2006, **2**, 2006.0034.
11. C. Chassagnole, N. Noisommit-Rizzi, J. W. Schmid, K. Mauch and M. Reuss, *Biotechnol Bioeng*, 2002, **79**, 53-73.
12. R. N. Gutenkunst, J. J. Waterfall, F. P. Casey, K. S. Brown, C. R. Myers and J. P. Sethna, *PLoS Computat Biol*, 2007, **3**, e189.
13. K. van Eunen, J. Bouwman, P. Daran-Lapujade, J. Postmus, A. B. Canelas, F. I. C. Mensonides, R. Orij, I. Tuzun, J. van den Brink, G. J. Smits, W. M. van Gulik, S. Brul, J. J. Heijnen, J. H. de Winde, M. J. Teixeira de Mattos, C. Kettner, J. Nielsen, H. V. Westerhoff and B. M. Bakker, *FEBS J*, 2010, **277**, 749-760.
14. R. Mahadevan, J. S. Edwards and F. J. Doyle, *Biophysical J*, 2002, **83**, 1331-1340.
15. E. O. Voit, G. Goel, I. C. Chou and L. L. Fonseca, in *IET Syst Biol*, Institution of Engineering and Technology, Editon edn., 2009, vol. 3, pp. 513-522.

16. I.-C. Chou and E. Voit, *BMC Syst Biol*, 2012, **6**, 84.
17. G. Goel, I.-C. Chou and E. O. Voit, *Bioinformatics*, 2008, **24**, 2505-2511.
18. N. Ishii, Y. Nakayama and M. Tomita, *Theor Biol Med Model*, 2007, **4**, 19.
19. K. Yugi, Y. Nakayama, A. Kinoshita and M. Tomita, *Theor Biol Med Model.*, 2005, **2**, 42.
20. G. Chechik and D. Koller, *J Comput Biol*, 2009, **16**, 279-290.
21. J. Sivriver, N. Habib and N. Friedman, *Bioinformatics*, 2011, **27**, i392-i400.
22. F. Hynne, S. Danø and P. G. Sørensen, *Biophys Chem*, 2001, **94**, 121-163.
23. N. Le Novère, B. Bornstein, A. Broicher, M. Courtot, M. Donizelli, H. Dharuri, L. Li, H. Sauro, M. Schilstra, B. Shapiro, J. L. Snoep and M. Hucka, *Nucleic Acids Res*, 2006, **34**, D689-D691.
24. S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes and U. Kummer, *Bioinformatics*, 2006, **22**, 3067-3074.
25. R. G. D. Steel and J. H. Torrie, *Principles and procedures of statistics, with special reference to the biological sciences*, McGraw-Hill, New York, 1960.
26. T. Evgeniou, T. Poggio, M. Pontil and A. Verri, *Comput. Stat. Data Anal.*, 2002, **38**, 421-432.