Electronic Supplementary Material (ESI) for Molecular BioSystems.

# Supplementary File 1

Supplementary Figures and Methods: *Improved metabolite profile smoothing for flux estimation*

## Table of Contents

## Table of Figures

Table S1. Table of Model Initial Conditions. Initial conditions listed here were used to generate synthetic data as a gold standard from each model as described in the Methods section of the main text.

| Chassagnole[1] (*E. coli*) | | |
|---|---|---|
| # | Metabolite | Conc (mM) |
| 1 | Extracellular Glucose | 1.670E+00 |
| 2 | Glucose-6-Phosphate | 3.480E+00 |
| 3 | Fructose-6-Phosphate | 6.000E-01 |
| 4 | Fructose-1,6-bisphosphate | 2.720E-01 |
| 5 | Dihydroxyacetonephosphate | 1.670E-01 |
| 6 | Glyceraldehyde-3-Phosphate | 2.180E-01 |
| 7 | 1,3-diphosphosphoglycerate | 8.000E-03 |
| 8 | 3-Phosphoglycerate | 2.130E+00 |
| 9 | 2-Phosphoglycerate | 3.990E-01 |
| 10 | Phosphoenol pyruvate | 2.670E+00 |
| 11 | Pyruvate | 2.670E+00 |
| 12 | 6-Phosphogluconate | 8.080E-01 |
| 13 | Ribulose-5-phosphate | 1.110E-01 |
| 14 | Xylulose-5-phosphate | 1.380E-01 |
| 15 | sedoheptulose-7-phosphate | 2.760E-01 |
| 16 | Ribose-5-phosphate | 3.980E-01 |
| 17 | Erythrose-4-phosphate | 9.800E-02 |
| 18 | Glucose-1-Phosphate | 6.530E-01 |

| Hynne[2] (*S. cerevisiae*) | | |
|---|---|---|
| # | Metabolite | Conc (mM) |
| 1 | Extracellular glucose | 3.330E-02 |
| 2 | Cytosolic glucose | 3.700E-03 |
| 3 | Glucose-6-Phosphate | 5.708E-01 |
| 4 | Fructose-6-Phosphate | 7.190E-02 |
| 5 | Fructose 1,6-bisphosphate | 5.090E-02 |
| 6 | Dihydroxyacetone phosphate | 2.851E-01 |
| 7 | Glyceraldehyde 3-phosphate | 1.240E-02 |
| 8 | 1,3-Bisphosphoglycerate | 0.000E+00 |
| 9 | Phosphoenolpyruvate | 6.300E-03 |
| 10 | Pyruvate | 6.540E-02 |
| 11 | Acetaldehyde | 1.268E-01 |
| 12 | Extracellular acetaldehyde | 1.100E-01 |
| 13 | EtOH | 3.754E+00 |
| 14 | Extracellular ethanol | 3.210E+00 |
| 15 | Glycerol | 3.642E-01 |
| 16 | Extracellular glycerol | 1.462E-01 |
| 17 | Extracellular cyanide | 5.564E+00 |
| 18 | AMP | 6.055E-01 |
| 19 | ADP | 1.757E+00 |
| 20 | ATP | 1.571E+00 |
| 21 | NAD | 7.787E-01 |
| 22 | NADH | 2.013E-01 |

Fig. S1. Validation of the Hynne model. cf. Hynne 2002, Fig. 6[2]. Model conditions described in Hynne et al. were used with the SBML model from the BioModels database[3] and Copasi 6[4] to reproduce steady oscillations, consistent with those shown in Fig. 6 of Hynne et al. The time interval depicted in this graph was selected to eliminate the presence of transients, and only depict the steady-state oscillations, while otherwise being consistent with Hynne et al.

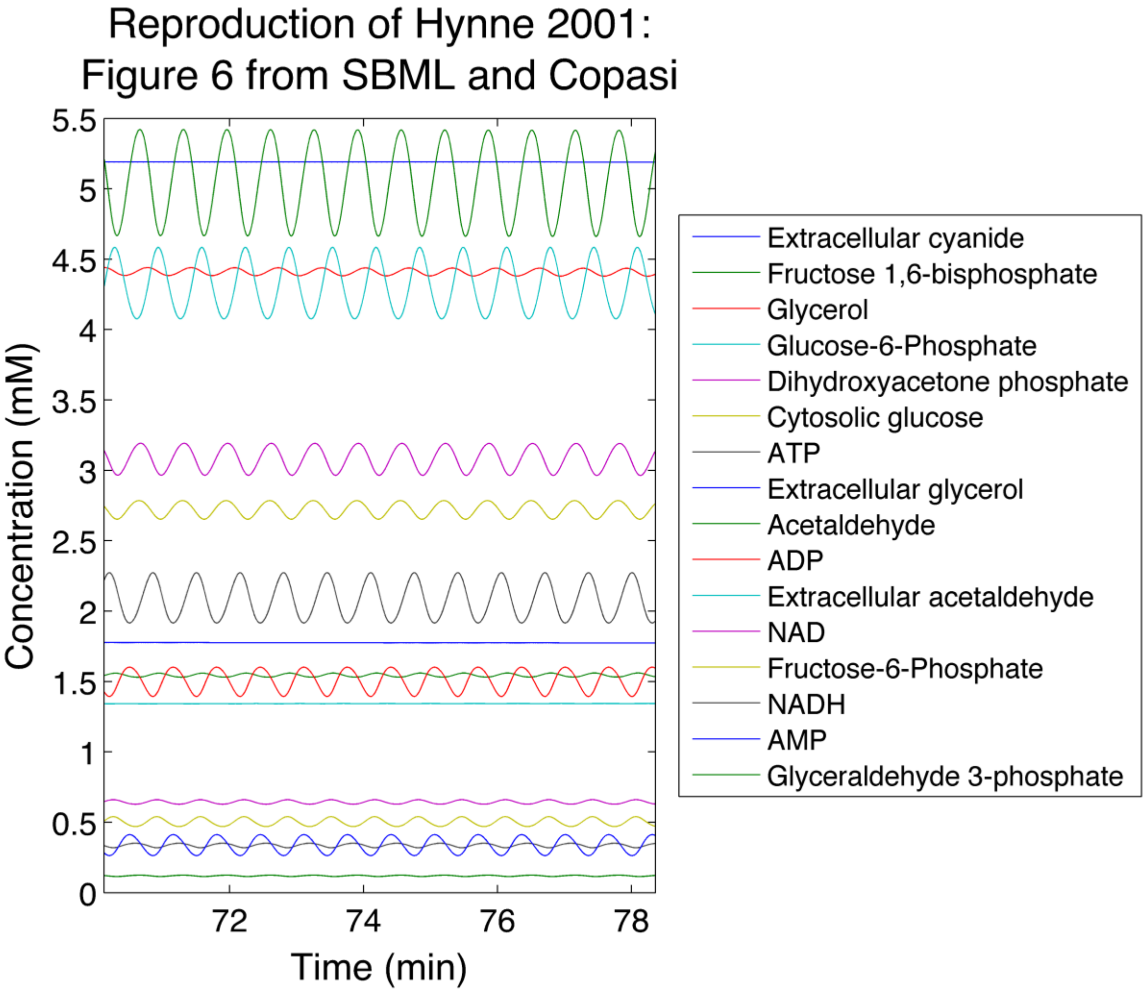


Fig. S2. The data used in the *S. cerevisiae* model introduces an additional challenge for the impulse model: some metabolites do not reach steady-state during the time window used. The impulse function qualitatively captures their behavior, but does not perform as strongly for these metabolites. Concentrations were fit using the Resampling method, for 50 noisy time courses and 250 resampled time courses each. Solid black lines indicate the synthetic data. Dashed black lines indicate the 15% CoV window above and below the concentration of the synthetic data, used to generate noisy time courses. Blue lines indicate functional fits to individual noisy time courses. A) The metabolite concentration does not reach steady-state; for some noisy time courses, the time delay identified in the resulting fit may be driven by noise, rather than an underlying feature in the noiseless data. B) As with the E. *coli* model, dynamics with short timescales are fundamentally more difficult to capture.

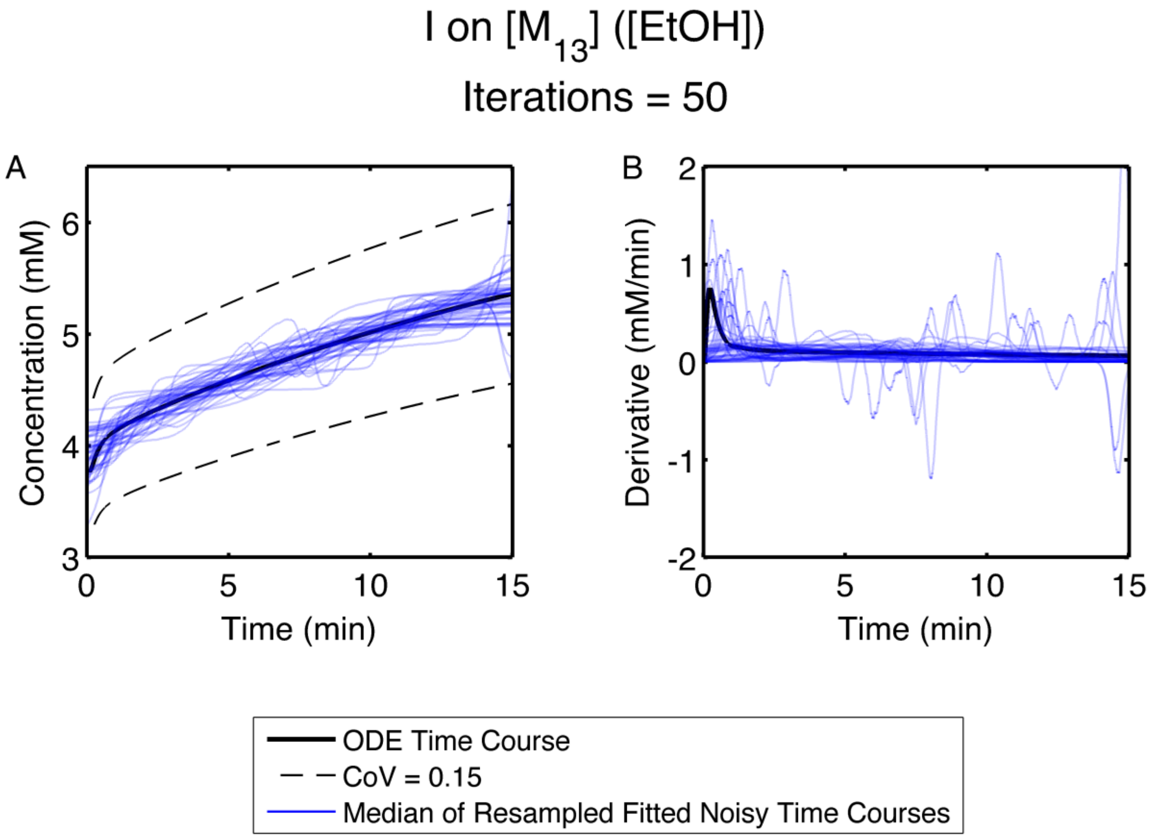


Table S2. Unabridged comparison of functions and methods with the *E. coli* model. An alternative method for ranking accuracy is calculated such that the median function accuracy across noisy time courses is calculated for each metabolite and function, and the functions are ranked once per metabolite using a single median value instead of independently ranking functions for each noisy time course. This metric is referred to as "Median Accuracy" in the table.

| Metric Ranked | Direct Fit Method | | | | | | | | Resampling Method | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $R_{11}$ | $R_{22}$ | $R_{31}$ | I | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $R_{11}$ | $R_{22}$ | $R_{31}$ | I |
| Concentration Accuracy | 3.68 | 4.13 | 2.50 | 2.94 | 3.94 | 2.33 | 4.83 | 1.74 | 4.02 | 4.16 | 2.44 | 3.11 | 4.22 | 1.83 | 5.32 | 1.90 |
| Derivative Accuracy | 3.18 | 3.45 | 2.48 | 3.08 | 3.58 | 2.61 | 3.77 | 2.18 | 3.38 | 3.40 | 2.50 | 3.07 | 3.68 | 2.16 | 4.66 | 2.20 |
| Concentration Robustness | 2.00 | 3.00 | 5.05 | 6.49 | 1.00 | 7.18 | 4.00 | 6.98 | 2.00 | 3.04 | 5.26 | 6.83 | 1.00 | 5.96 | 4.05 | 7.25 |
| Derivative Robustness | 2.00 | 3.00 | 5.00 | 6.05 | 1.00 | 7.68 | 4.00 | 7.19 | 2.00 | 3.04 | 5.05 | 6.20 | 1.00 | 6.85 | 4.05 | 7.30 |
| Median Concentration Accuracy | 4.19 | 4.43 | 2.41 | 3.36 | 3.68 | 2.38 | 5.11 | 1.53 | 4.26 | 4.68 | 2.46 | 3.79 | 3.95 | 1.50 | 5.47 | 2.02 |
| Median Derivative Accuracy | 3.34 | 3.60 | 2.39 | 2.72 | 3.03 | 2.86 | 3.96 | 2.37 | 3.49 | 3.67 | 2.62 | 2.97 | 3.03 | 2.39 | 4.75 | 2.07 |

| Metric Ranked | $P_2$ | | $P_3$ | | $P_4$ | | $P_5$ | | $R_{11}$ | | $R_{22}$ | | $R_{31}$ | | I | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 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 |
| Concentration Robustness | 3.24 | 3.64 | 5.20 | 5.84 | 9.59 | 10.11 | 12.44 | 13.24 | 1.21 | 1.48 | 13.68 | 11.10 | 7.66 | 7.38 | 14.06 | 13.39 |
| Derivative Robustness | 3.19 | 3.71 | 5.15 | 5.96 | 9.33 | 9.73 | 11.53 | 11.95 | 1.42 | 1.26 | 15.08 | 12.67 | 7.58 | 7.38 | 14.62 | 13.48 |
| Median Concentration Accuracy | 6.43 | 7.56 | 8.15 | 9.05 | 4.65 | 3.07 | 6.48 | 6.95 | 6.04 | 6.19 | 3.30 | 1.81 | 9.50 | 10.49 | 3.56 | 2.82 |
| Median Derivative Accuracy | 5.72 | 6.05 | 6.65 | 6.86 | 4.23 | 3.76 | 5.57 | 4.63 | 4.99 | 4.35 | 4.85 | 3.39 | 7.01 | 9.17 | 3.34 | 2.75 |

Table S3. Unabridged comparison of functions and methods with the *S. cerevisiae* model.

| Metric Ranked | Direct Fit Method | | | | | | | | Resampling Method | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $R_{11}$ | $R_{22}$ | $R_{31}$ | 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 |
| Concentration Robustness | 2.00 | 3.04 | 5.38 | 6.62 | 1.00 | 5.51 | 4.24 | 7.43 | 2.10 | 3.41 | 5.78 | 7.11 | 1.00 | 3.54 | 4.47 | 7.79 |
| Derivative Robustness | 1.83 | 3.00 | 4.88 | 5.89 | 1.05 | 7.27 | 4.20 | 7.44 | 2.00 | 3.00 | 5.12 | 6.30 | 1.00 | 6.54 | 4.00 | 7.84 |
| Median Concentration Accuracy | 3.49 | 4.03 | 4.21 | 3.45 | 4.77 | 1.26 | 4.67 | 2.30 | 4.37 | 4.50 | 4.17 | 3.27 | 4.32 | 1.14 | 5.01 | 2.41 |
| Median Derivative Accuracy | 3.12 | 3.87 | 4.27 | 2.90 | 4.49 | 2.25 | 4.59 | 1.48 | 3.37 | 4.20 | 4.39 | 3.09 | 4.73 | 1.69 | 4.95 | 1.60 |

| Metric Ranked | $P_2$ | | $P_3$ | | $P_4$ | | $P_5$ | | $R_{11}$ | | $R_{22}$ | | $R_{31}$ | | I | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 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 |
| Concentration Robustness | 3.19 | 3.99 | 5.58 | 6.57 | 10.26 | 11.00 | 12.88 | 13.79 | 1.31 | 1.35 | 10.81 | 5.84 | 8.31 | 7.93 | 15.00 | 14.32 |
| Derivative Robustness | 3.19 | 3.55 | 5.33 | 5.66 | 9.20 | 9.83 | 11.49 | 11.92 | 1.53 | 1.20 | 14.72 | 11.90 | 8.09 | 7.11 | 15.35 | 14.20 |
| Median Concentration Accuracy | 6.98 | 8.27 | 7.63 | 8.24 | 7.62 | 7.72 | 5.91 | 6.02 | 6.47 | 6.91 | 1.73 | 1.44 | 7.92 | 9.46 | 4.45 | 4.12 |
| Median Derivative Accuracy | 5.30 | 4.91 | 7.21 | 7.72 | 7.74 | 8.11 | 4.98 | 4.84 | 7.34 | 8.73 | 4.66 | 2.09 | 8.84 | 9.43 | 2.77 | 1.87 |

## Supplementary Methods

### *Calculating equation robustness*

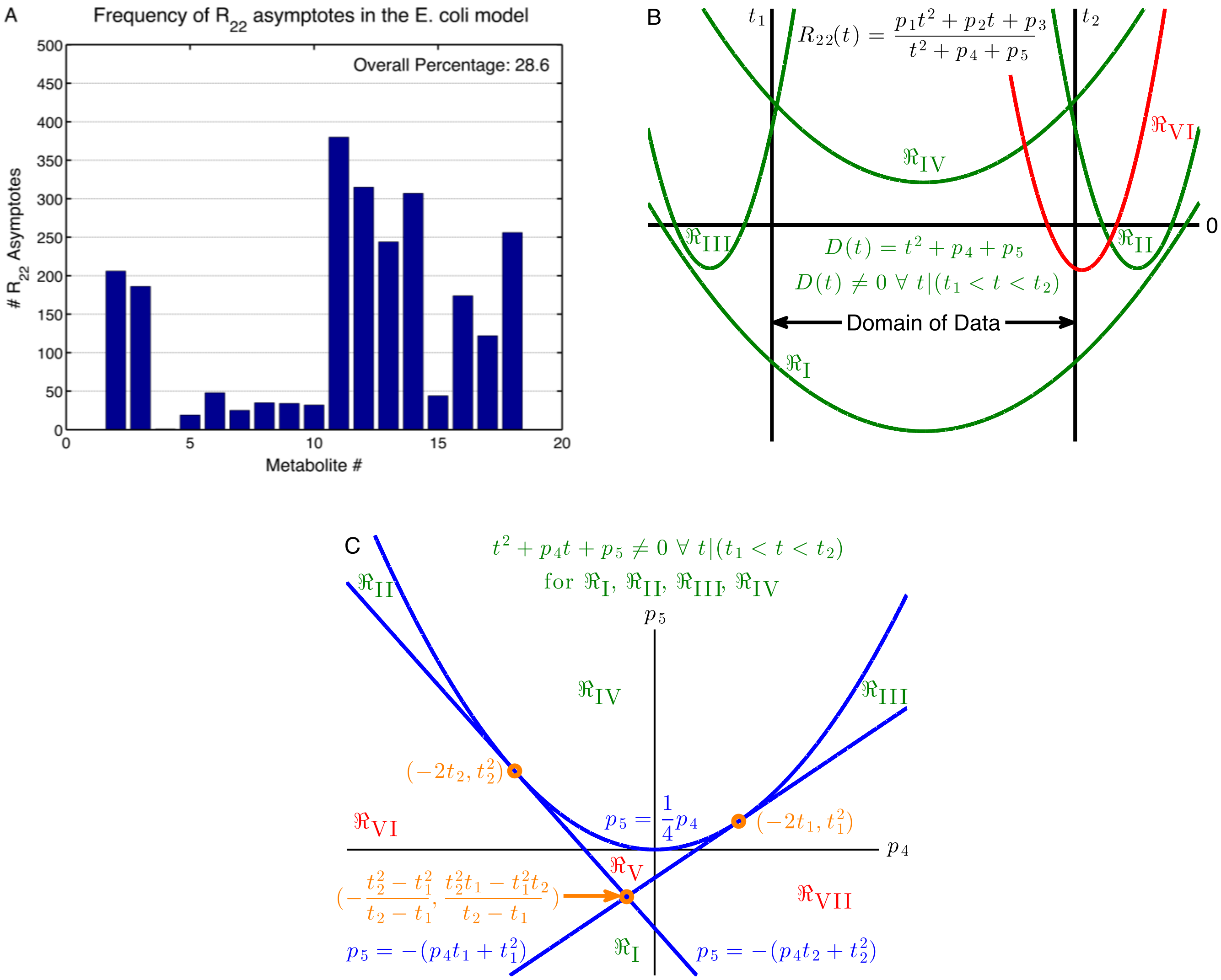
The robustness metric captures the variance of a particular function for a given metabolite, as determined by its consistency across multiple noisy time courses. A function that frequently returns very similar values at each time point, even if that value is inaccurate, would be considered more robust. Robustness is calculated as

$$Robustness_{i,j} = \left(\frac{1}{\mu}\right)\left(\frac{1}{n_m}\right)\sum_{m}\sqrt{\sum_{k}\frac{\left(f_{i,j,m}(t_k) - \widetilde{f_{i,j}(t_k)}\right)^2}{n_I \cdot S}}$$

where $n_m$ is the number of noisy time courses and $\widetilde{f_{i,j}(t_k)}$ is the median value of $f_{i,j,m}(t_k)$ across noisy time courses. For calculating derivative accuracy, the derivative values $f'_{i,j,m}(t_k)$ and $\widetilde{f'_{i,j}(t_k)}$ are substituted in place of $f_{i,j,m}(t_k)$ and $\widetilde{f_{i,j}(t_k)}$.

Because the Robustness is calculated by finding the variation across fits to different noisy time courses, it is only meaningful on the level of assessing performance for a given metabolite. We report Robustness in the form of averaged rank performance along with the function accuracy for easier comparison in Supplementary Tables 2 and 3.

Fig. S3. Parameter domain and bounding for the rational function denominators. A) In 28.6% of cases, the best-fit $R_{22}$ concentration trajectory in the *E.coli* model contains an asymptote in the domain of the data, leading to the qualitatively invalid fitted concentration behavior shown in Fig 3D. B) Some $R_{22}$ denominator polynomials (colored red) contain roots in the time interval of the data, leading to asymptotes in the resulting concentration trajectory. Others (green) produce qualitatively valid trajectories. Polynomials are labeled by the region in Fig S3C from which parameters $p_4$ and $p_5$ were taken. C) Multiple regimes exist in the parameter space spanned by the $R_{22}$ denominator polynomial, different root positions for the denominator polynomial. Blue lines designate the boundaries between regions. Green text ($\mathcal{R}_{\mathrm{I}}$, $\mathcal{R}_{\mathrm{II}}$, $\mathcal{R}_{\mathrm{III}}$, $\mathcal{R}_{\mathrm{IV}}$) indicates regions that preclude problematic roots, while red text ($\mathcal{R}_{\mathrm{V}}$, $\mathcal{R}_{\mathrm{VI}}$, $\mathcal{R}_{\mathrm{VII}}$) indicates regions that produce qualitatively invalid behaviors. In practice, the valid $R_{22}$ denominator parameter regions are modified to simplify solver implementation by grouping sub-regions of $R_{\mathrm{IV}}$ instead with $R_{\mathrm{II}}$ and $R_{\mathrm{III}}$.



### *On asymptotes in the fitted rational functions*

The rational functions are composed of polynomial terms in both the numerator and denominator, ranging from first to third order in the numerator, and first to second order in the denominator. However, we observed that fitting the noisy time courses with rational functions on some occasions produced asymptotes in the time interval of the data (Fig. 2D). For example, and as shown in Fig. S3A, we observed that in nearly 29% of the noisy time courses we generated for the *E. coli* model, the optimal $R_{22}$ parameters led to asymptotes in the resulting fitted concentration time course. This asymptotic behavior occurs when the best fit parameters for the fitted noisy time courses correspond to denominator polynomials with roots in the time interval of the data; in Fig. S3B, we depict this scenario in the case of the $R_{22}$ denominator polynomial.

Since this behavior is unphysical and can undermine the relevance or usefulness of the fitted concentration or derivative values, we modified our parameter fitting routines to exclude these pathological cases by introducing bounds on the denominator parameters. Specifically, as suggested by the conditions depicted in Fig. S3B, rejecting parameter values which produce roots in the denominator polynomial in the time interval of the data (spanning the interval between $t_1$ and $t_2$ in Fig. S3B) will prevent this problematic behavior from occurring. As implemented, these parameter restrictions include the option for an additional buffer on each end of the time interval to prevent asymptotes from occurring at the edges of the data. By default, this buffer is calculated from the time data used to fit the function to have a magnitude of ΔT, where ΔT is the time interval between sequential data points in the time course to be fit. (As a concrete example, the E. coli model is simulated from 0 seconds to 20 seconds, and the fitted data is sampled at 1 second intervals; the corresponding buffers lead to $t_1 = -1$ and $t_2 = 21$m which prevent the roots of the denominator from falling in the interval of -1 to 21.)

### *Bounding the first order denominator polynomials of the $R_{11}$ and $R_{31}$ functions*

In the case of $R_{11}$ and $R_{31}$, the denominator is a first order polynomial. Because the permissible parameter value regions are discontinuous, the most efficient way to determine the optimal parameters is to identify the optimal parameters in each permitted region, and pick the set with the lower RMSD. Solver Region I enforces that $p_i < -t_2$, and Solver Region II enforces $p_i > -t_1$. where i = 3 in the $R_{11}$ model, and i = 5 in the $R_{31}$ model.

As described below, we fit the data using multiple sets of random initial conditions. For each set of initial conditions, we perform two parameter optimizations: one for each permissible parameter region. The resulting parameter set for a given set of initial conditions is the set that produced the lower RMSD. Parameter optimizations were performed using `fmincon()` in MATLAB.

### *Bounding the second order denominator polynomial of the $R_{22}$ function*

In the case of $R_{22}$, the denominator is a second order polynomial. Examples of these polynomials are shown in Fig. S3B; the position of the roots relative to the interval ($t_1$, $t_2$) is determined by the parameters $p_4$ and $p_5$. By bounding the values these parameters may take, we can prevent the resulting trajectory from producing asymptotes in the interval of the data.

We construct these bounds by starting with the denominator polynomial equation,

$$D(t) = t^2 + p_4 \cdot t + p_5$$

And the equation for the roots of a quadratic polynomial,

$$r = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$

In this instance, a = 1, b = $p_4$, and c = $p_5$, and hence:

$$r = \frac{-p_4 \pm \sqrt{p_4^2 - 4 \cdot p_5}}{2}$$

We isolate the square root term to get

$$2 \cdot r + p_4 = \pm\sqrt{p_4^2 - 4 \cdot p_5}$$

and square both sides to get

$$4 \cdot r^2 + 4 \cdot r \cdot p_4 + p_4^2 = p_4^2 - 4 \cdot p_5$$

We then solve for $p_5$, and get

$$p_5 = -(r \cdot p_4 + r^2)$$

We set r = $t_1$ and r = $t_2$ to produce the main divisions in the space spanned by $p_4$, and $p_5$; these equations are plotted in Fig. S3C. The value of $p_5$ relative to these divisions determines the placement of the roots of the corresponding denominator polynomial.

In some cases, a quadratic equation may have no real roots. This situation will never lead to asymptotes in the resulting $R_{22}$ function, and so is also acceptable. We determine the boundary of this condition by setting the discriminant $p_4^2 - 4 \cdot p_5 < 0$. Solving for $p_5$, we get

$$p_5 > \frac{1}{4} p_4^2$$

Under these conditions, the resulting roots are imaginary and do not produce asymptotes, regardless of the values of $t_1$ and $t_2$. This boundary is plotted in Fig. S3C.

Table S4. The parameters $p_4$ and $p_5$ determine the behavior of the denominator polynomial in $R_{22}$ by determining the location of its roots. The space spanned by these parameters can be divided into 7 regions, based on the position of each of these roots relative to the interval $(t_1, t_2)$.

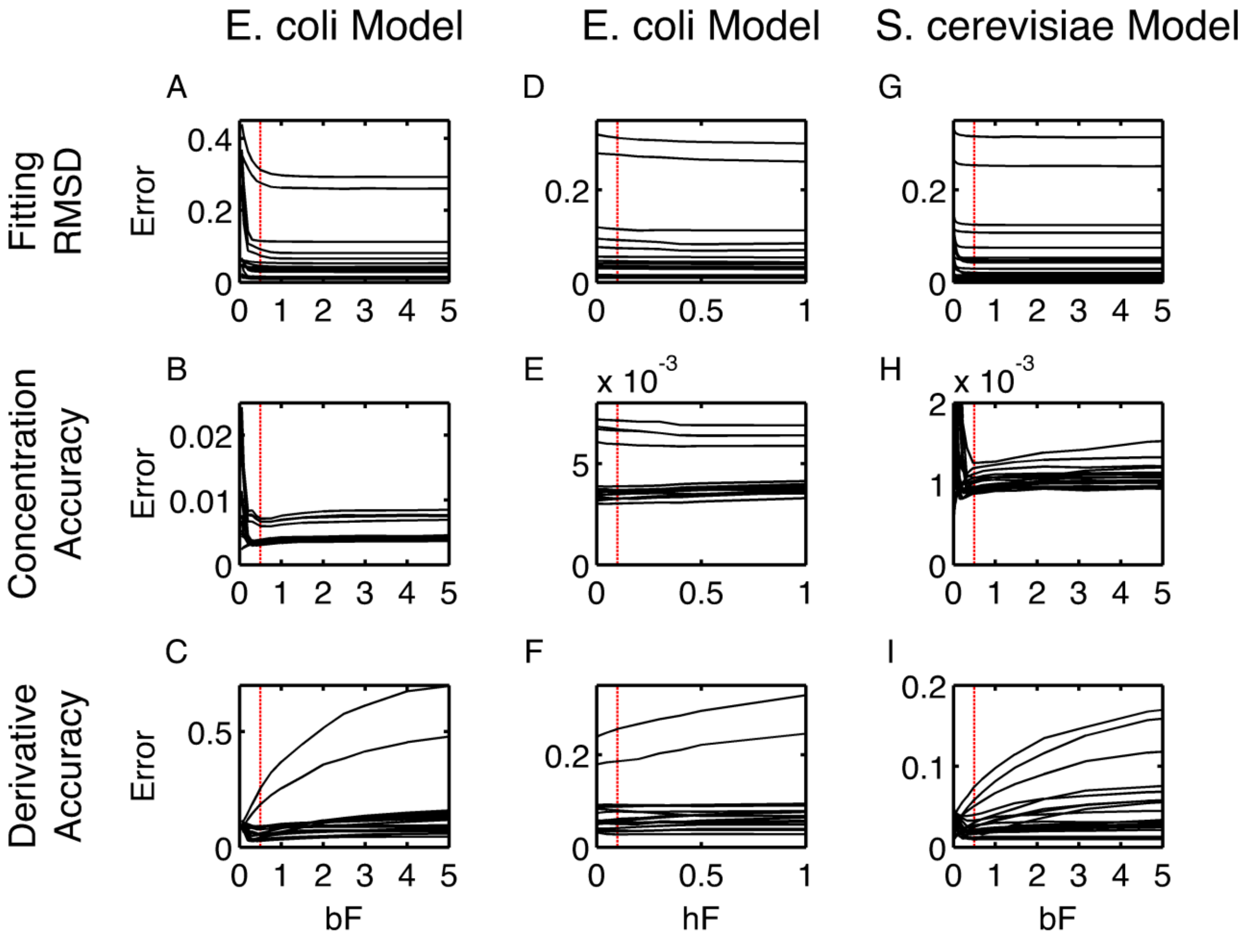
| Region | Constraints | Description |
|---|---|---|
| $\mathcal{R}_{\mathrm{I}}$ | $p_5 < -(t_1 \cdot p_4 + t_1^2)$<br>$p_5 < -(t_2 \cdot p_4 + t_2^2)$ | $r_1 < t_1$<br>$r_2 > t_2$ |
| $\mathcal{R}_{\mathrm{II}}$ | $p_5 > -(t_2 \cdot p_4 + t_2^2)$<br>$p_5 < \frac{1}{4} p_4^2$ | $r_1 > t_2$<br>$r_2 > t_2$ |
| $\mathcal{R}_{\mathrm{III}}$ | $p_5 > -(t_1 \cdot p_4 + t_1^2)$<br>$p_5 < \frac{1}{4} p_4^2$ | $r_1 < t_1$<br>$r_2 < t_1$ |
| $\mathcal{R}_{\mathrm{IV}}$ | $p_5 > \frac{1}{4} p_4^2$ | $r_1 \epsilon \mathbb{C}$<br>$r_2 \epsilon \mathbb{C}$ |
| $\mathcal{R}_{\mathrm{V}}$ | $p_5 > -(t_1 \cdot p_4 + t_1^2)$<br>$p_5 > -(t_2 \cdot p_4 + t_2^2)$<br>$p_5 < \frac{1}{4} p_4^2$ | $t_1 < r_1 < t_2$<br>$t_1 < r_2 < t_2$ |
| $\mathcal{R}_{\mathrm{VI}}$ | $p_5 > -(t_1 \cdot p_4 + t_1^2)$<br>$p_5 < -(t_2 \cdot p_4 + t_2^2)$ | $t_1 < r_1 < t_2$<br>$r_2 > t_2$ |
| $\mathcal{R}_{\mathrm{VII}}$ | $p_5 < -(t_1 \cdot p_4 + t_1^2)$<br>$p_5 > -(t_2 \cdot p_4 + t_2^2)$ | $r_1 < t_1$<br>$t_1 < r_2 < t_2$ |

The values of $p_4$ and $p_5$ relative to these three boundaries determine the position of the denominator polynomial roots. These curves delineate seven regions in the space spanned by $p_4$ and $p_5$; these regions are depicted graphically in Fig. S3C, and described mathematically in Table S4.

Of these seven regions, four produce behavior that is acceptable for our application. Region $\mathcal{R}_{\mathrm{I}}$ represents the case where the roots straddle the time interval of the data. Regions $\mathcal{R}_{\mathrm{II}}$ and $\mathcal{R}_{\mathrm{III}}$ represent the cases where the roots are either both above or both below the boundaries, respectively. Region $\mathcal{R}_{\mathrm{IV}}$ represents the case where there are no real roots. Because Regions $\mathcal{R}_{\mathrm{II}}$ and $\mathcal{R}_{\mathrm{IV}}$ border each other, as do Regions $\mathcal{R}_{\mathrm{III}}$ and $\mathcal{R}_{\mathrm{IV}}$, in practice we re-divide the regions to simplify the parameter bounds assigned to the solver to minimize the number of nonlinear constraints (the orange points in Fig. S3C where the linear boundaries lie tangent to $\mathcal{R}_{\mathrm{IV}}$ mark the values for $p_4$ we chose).

As described below, we fit the data using multiple sets of random initial conditions. For each set of initial conditions, we perform four parameter optimizations, corresponding to the four (adjusted) Parameter Regions we described here and shown in Fig. S3C. From these four regions, we choose the parameter set with the lowest RMSD to represent the parameter set for a given set of initial conditions. Parameter optimizations were performed using `fmincon()` in MATLAB.

Fig. S4. The impact of global parameters on Impulse performance. The impulse function was tested over a range of values for $b_f$ and $h_f$ to determine optimal values for these parameters, and to assess how sensitivity to those values. Solid black lines indicate individual metabolites. Dashed red lines indicate our selected values of $b_f = 0.5$ or $h_f = 0.1$. A-C) $b_f$ was varied in the *E.coli* model, and we found that a value of bf = 0.5 generally worked well based on RMSD during fitting, concentration accuracy, and derivative accuracy. D-F) $h_f$ was varied in the *E.coli* model, and found not to strongly affect the performance of any of our metrics. We chose a value of $h_f = 0.1$ to permit small fluctuations relative to the range of the data due to expected noisiness in the data. G-I) We verified our choice of $b_f = 0.5$ in the S. cerevisiae model, and found that this value indeed worked reasonably well for this model as well.



### *Bounding the Impulse Function*

When we initially implemented the impulse function, we observed that it was prone to producing sharp shifts due to excessive magnitudes for the slope parameters, and prone to getting stuck in insensitive local minima when the time delay parameters are well outside the domain of the data. An example of this behavior is shown in Fig. 2B. Conveniently, the direct correspondence between parameter values and features in the graph of the impulse function made it straightforward to implement effective and beneficial parameter boundaries. These bounds were implemented as constant parameter values, and optimization was performed using `fmincon()` in MATLAB.

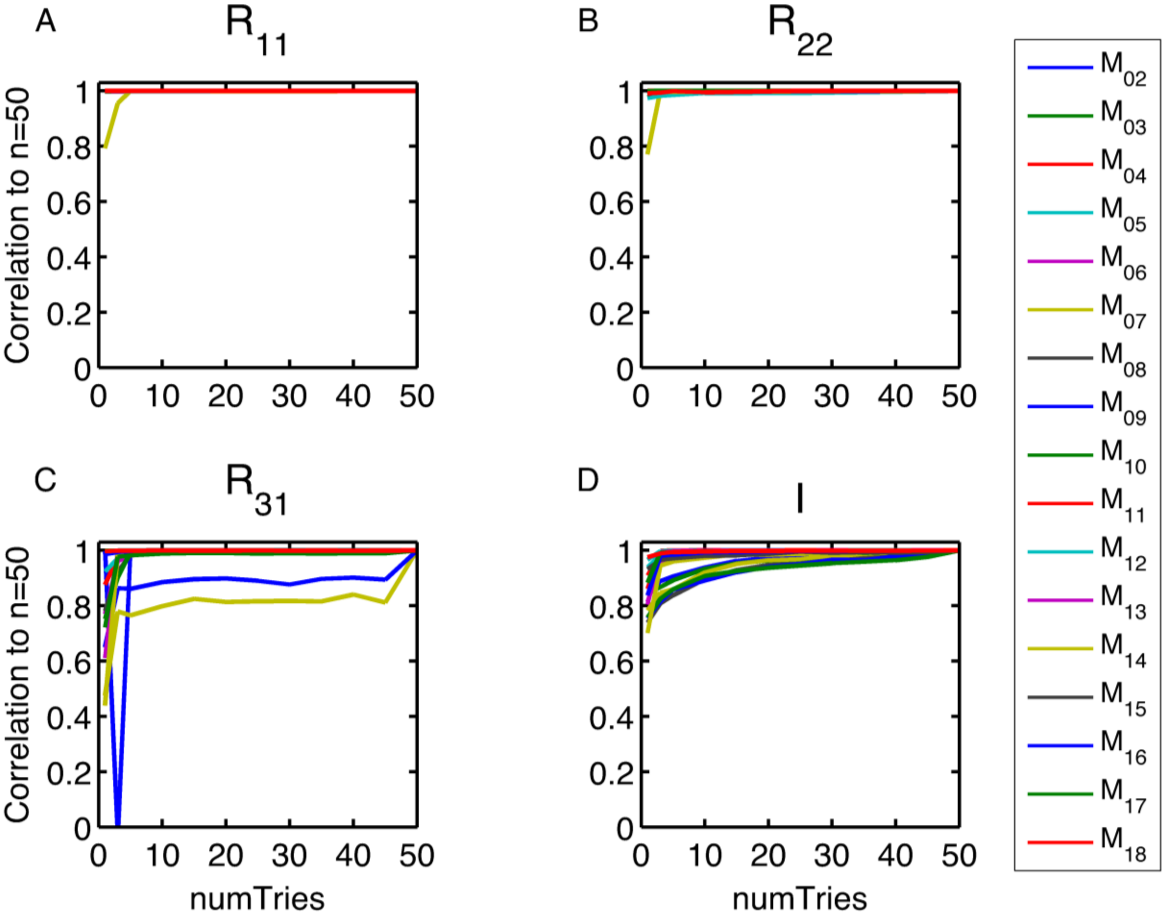
Our first set of bounds is on the time delay parameters. Because there is no basis from the data for modeling a transition outside the interval of the data, we restrict these values to the range of the time values in the data. This has the advantage of removing insensitive local optima from the available parameter space, and helps ensure that the fitting error remains sensitive to parameter values. In effect, we restrict $\tau_1$ and $\tau_2$ to the interval $(t_1,t_2)$. Unlike the rational functions, by default we did not add a buffer to the time interval of the data; for example, in the *E.coli* model, we used $t_1$= 0 seconds and $t_2$ = 20 seconds.

The second set of bounds we introduced restrict the magnitude of the impulse function transition parameters, $\beta_1$ and $\beta_2$; this prevents the transitions from becoming too sharp. The justification for this restriction is that since the data are sampled at a given frequency, there is no justification from the data to model faster dynamics in the resulting function than is present in the data. As such, the sharpness of the impulse transitions is set to be inversely proportional to the size of the time interval used to fit the data. The proportionality factor for this inverse relationship is a global parameter, which we label $b_f$. In effect, we enforce that $|\beta| < \frac{b_f}{\Delta t}$, where $\Delta t$ is the time interval between sequential data points in the time course to be fit. For the *E. coli* dataset, we determined that a value of $b_f$ = 0.5 was generally near optimal for the data, as shown in Fig. S4A-C. The two exceptions to this were Metabolites 12 and 18; for these metabolites, the error in derivative accuracy decreased as the sharpness was more heavily restricted (i.e., as $b_f$ was decreased).

The last set of bounds we introduced restrict the height parameters controlling the initial, intermediate, and steady-state values of the resulting impulse ($h_0$, $h_1$, and $h_2$, respectively). We restrict these parameters to a window defined by the range of the concentration data. The rationale behind this restriction is that there is no basis from the data for the function to model a change in value far outside the range seen in the data. We allowed this window to be extended by an added percentage, represented by the global parameter, $h_f$. For example, a value of $h_f$ = 0.10 corresponds to extending the bounds an additional 10% both above and below the range of the data. As shown in Fig. S4D-F, we found the performance of the impulse in the *E.coli* model to be generally insensitive above relatively small values of $h_f$; a value of $h_f$ = 0.1 was selected to permit some fluctuation due to the expected noisiness of the data.

It is important to note two features of the global parameters $h_f$ and $b_f$. First, they are not parameters in the resulting fitted parameter set; they only determine bounds on those parameters. Second, they were not fit to individual noisy time courses, or even to individual metabolites. Because of this, we hypothesized that we could directly re-use these values from the *E. coli* model with the *S. cerevisiae* model without adjusting them. To validate that this was the case, we tested a range of $b_f$ values on the *S. cerevisiae* model. As shown in Fig. S4G-I, the value of $b_f$ = 0.5 determined from the *E. coli* model was also an effective choice for the *S. cerevisiae* model. This suggests that these global parameters may generalize reasonably well to other models, and as such we did not penalize our metrics to account for them.

Fig. S5. The effect of using multiple solver initial conditions on the consistency of the solver output.
The parameter solver may identify only a local minimum RMSD value. In these cases, seeding the solver with multiple sets of random initial conditions may enable identification of a parameter set that produces a lower RMSD. We tested the effect of multiple optimizations with different initialization values; the parameter numTries indicates the number initial condition sets with which we seeded the solver. Functions were fit to the noisy time courses generated for the Direct Fit method on the *E. coli* model. For each function and metabolite, numTries was varied from 1 to 50. The resulting RMSD values for each value of numTries were then compared to the RMSD values determined when numTries=50 using a Pearson correlation; a higher value indicates that the parameters found for that value of numTries is more consistent with the parameters produced when numTries=50. A) $R_{11}$ requires very few seeds to produce consistent results. B) $R_{22}$ also produces consistent results with few seeds. C) For most metabolites, $R_{31}$ requires few seeds. However, this was not the case for two of the 17 metabolites ($M_{07}$ and $M_{09}$). D) The Impulse required few tries for some metabolites, but some metabolties do improve as numTries is increased. The results in the main text used numTries = 20; we note that this indicates that increasing the number of seeds would likely improve the performance of this function modestly.



*Addressing issues of local minima and solver consistency*

For the rational and impulse functions, our solver routines use initial random parameter values with the built-in MATLAB function `fmincon()` to find the parameter values with the lowest RMSD, subject to the constraints described previously. During our investigation, we found that using a single initial random parameter set would lead to inconsistent RMSD values for the resulting fitted parameters; this indicates that for a single set of initial conditions, there is a real risk of encountering local minima when minimizing the fitting RMSD. To counter this, we seeded our solvers with multiple sets of random initial parameters and selected from the resulting fitted parameters the set that produced the lowest RMSD.

To determine how many sets of random initial parameters was needed for each function, we tested values between 1 and 50 seeds for each function, repeating the procedure for each metabolite in the *E.coli* model using the dataset we produced for the Direct Fit method. The RMSD values of the resulting optimal parameter sets for each noisy time course were then compared against the RMSD values generated using 50 seeds by calculating a Pearson correlation between the RMSD values. As shown in Fig. S5, we found that 20 seeds were sufficient across all functions to ensure consistent results; in many cases, far fewer seeds were necessary to produce RMSD values equivalent to the 50 seeds case. For the results shown in the main text, we used 20 seeds for all four functions.

## References:

1. C. Chassagnole, N. Noisommit-Rizzi, J. W. Schmid, K. Mauch and M. Reuss, *Biotechnol Bioeng*, 2002, **79**, 53-73.
2. F. Hynne, S. Danø and P. G. Sørensen, *Biophys Chem*, 2001, **94**, 121-163.
3. 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 Research*, 2006, **34**, D689-D691.
4. 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.