

This Provisional PDF corresponds to the article as it appeared upon acceptance. Fully formatted PDF and full text (HTML) versions will be made available soon.

Fluxomers: A new approach for ^{13}C metabolic flux analysis

BMC Systems Biology 2011, **5**:129 doi:10.1186/1752-0509-5-129

Orr Srour (srouro@tx.technion.ac.il)
Jamey D Young (j.d.young@vanderbilt.edu)
Yonina C Eldar (yonina@ee.technion.ac.il)

ISSN 1752-0509

Article type Methodology article

Submission date 10 August 2010

Acceptance date 16 August 2011

Publication date 16 August 2011

Article URL <http://www.biomedcentral.com/1752-0509/5/129>

Like all articles in BMC journals, this peer-reviewed article was published immediately upon acceptance. It can be downloaded, printed and distributed freely for any purposes (see copyright notice below).

Articles in BMC journals are listed in PubMed and archived at PubMed Central.

For information about publishing your research in BMC journals or any BioMed Central journal, go to

<http://www.biomedcentral.com/info/authors/>

Fluxomers: A new approach for ^{13}C metabolic flux analysis

Orr Srour¹, Jamey D. Young^{2,*} and Yonina C. Eldar^{1,*}

¹Dept. of Electrical Engineering, Technion–Israel Institute of Technology, 32000 Haifa, Israel.

²Department of Chemical and Biomolecular Engineering, Vanderbilt University, TN, USA.

Email: Jamey D. Young - j.d.young@vanderbilt.edu; Yonina C. Eldar - yonina@ee.technion.ac.il;

*Corresponding author

Abstract

Background: The ability to perform quantitative studies using isotope tracers and metabolic flux analysis (MFA) is critical for detecting pathway bottlenecks and elucidating network regulation in biological systems, especially those that have been engineered to alter their native metabolic capacities. Mathematically, MFA models are traditionally formulated using separate state variables for reaction fluxes and isotopomer abundances. Analysis of isotope labeling experiments using this set of variables results in a non-convex optimization problem that suffers from both implementation complexity and convergence problems.

Results: This article addresses the mathematical and computational formulation of ^{13}C MFA models using a new set of variables referred to as *fluxomers*. These composite variables combine both fluxes and isotopomer abundances, which results in a simply-posed formulation and an improved error model that is insensitive to isotopomer measurement normalization. A powerful fluxomer iterative algorithm (FIA) is developed and applied to solve the MFA optimization problem. For moderate-sized networks, the algorithm is shown to outperform the commonly used 13CFLUX cumomer-based algorithm and the more recently introduced OpenFLUX software that relies upon an elementary metabolite unit (EMU) network decomposition, both in terms of convergence time and output variability.

Conclusions: Substantial improvements in convergence time and statistical quality of results can be achieved by applying fluxomer variables and the FIA algorithm to compute best-fit solutions to MFA models. We expect that the fluxomer formulation will provide a more suitable basis for future algorithms that analyze very large scale networks and design optimal isotope labeling experiments.

Background

Metabolic Pathway Analysis

Metabolism is the complete set of chemical reactions taking place in living cells. These chemical

processes form the basis of all life, allowing cells to grow, reproduce, maintain their structure and respond to environmental changes. Metabolic reac-

tions are divided into groups called metabolic pathways, which are typically constructed heuristically according to their connectivity and presumed function [1]. Each metabolic pathway is characterized by a set of chemical reactions that transform substrates into end products while generating intermediate byproducts. Due to its importance in medicine and biotechnology, metabolic pathway research has become a highly active field of investigation [2].

Initially, the structure of metabolic pathways was examined by identifying their intermediate compounds. Subsequently, the various biochemical reactions connecting these compounds were mapped. Due to the success of this research, the topological structure of many metabolic pathways is nowadays fully documented [3]. The next step in the progression of metabolic pathway research involves quantification of the rates of these various chemical reactions, also known as “fluxes”. The values of these rates are affected by various environmental conditions and can change rapidly in response to perturbations. Nevertheless, if the environmental parameters are held fixed and stable, the network can attain a steady state in which the concentrations of all network metabolites are assumed constant over time. This, of course, implies that the rates of their input and output reactions must balance. The latter imposes a set of linear constraints on the metabolic fluxes, known as “stoichiometric balance equations” [4]. Unfortunately, since the number of unknown fluxes typically exceeds the number of independent stoichiometric balances, these constraints are insufficient to completely identify the metabolic network. In order to overcome this lack of information, additional constraints must be provided to the stoichiometric mathematical model to estimate the values of the network fluxes [5].

¹³C Isotope Labeling Experiments

Various experimental techniques have been developed to enable measurement of intracellular metabolic fluxes, either directly or indirectly. One of these approaches makes use of isotope labeling experiments. In this method, the metabolic system is administered a known amount of an isotopically labeled substrate (such as glucose labeled with ¹³C at specific atom positions). By measuring the re-

sulting labeling patterns of intracellular metabolites after steady state has been achieved, additional flux information is obtained.

One major drawback of this experimental approach is the high complexity and computational intensity of the metabolic flux analysis (MFA) required to interpret these labeling measurements. In their series of articles, Wiechert et al. [6–9] constructed a systematic approach for performing this analysis. They show that measurements of the relative abundance of various isotope isomers, also known as “isotopomers”, contain enough information to fully identify the metabolic fluxes of the network. Formulating the problem using isotopomer variables (or a transformed set of isotopomer variables referred to as “cumomers”), Wiechert et al. posed the flux estimation problem as a non-convex least-squares minimization, assuming random error is added to their isotopomer measurements. The resulting high-dimensional non-convex problem suffers from various drawbacks, such as slow convergence and notable probability of attaining local minima. Several optimization algorithms have been developed in order to address these drawbacks. Early approaches used iterative parameter-fitting algorithms [8], evolutionary algorithms [10] and simulated annealing [11]. Furthermore, several investigations have been conducted in order to assess the accuracy of these results [9, 12, 13]. Recently, a novel method to decompose the metabolic network into Elementary Metabolite Units (EMUs) was introduced [14] and implemented into the OpenFLUX software package [15]. This decomposition effectively reduces the size of the optimization problem by efficiently simulating only those isotopomers that contribute to the measurement residuals. Nevertheless, all of these algorithms suffer from several major drawbacks due to the standard isotopomer-flux variables used in formulating the optimization problem:

- *Presence of unstable local minima:* due to the non-convex nature of the objective function.
- *Complex mathematical representation and computational implementation.* This results in the need for ad-hoc algorithms and mathematical analysis, and long running times are required for reliable convergence.

The OpenFLUX implementation, for example, may require several dozens of convergence iterations with various initial values in order to achieve acceptable probability of obtaining the optimal set of fluxes in any one of its attempts. In addition, due to the chosen objective function, it is also commonly required to estimate *scaling factors* for each isotopomer measurement, because of the fact that the available experimental techniques are only capable of measuring isotopomer fractions up to a proportional scaling factor (see Mollney et al. [9] for further details).

Our Contribution

This article introduces a new set of variables for simulating ^{13}C isotope labeling experiments. The main idea underlying this reformulation is that, instead of treating fluxes and isotopomer variables separately, we identify a set of “isotopically labeled fluxes” as our state variables of interest. We refer to these variables as *fluxomers*. Fluxomers combine flux variables with isotopomer variables and consequently reduce the complexity and nonlinearity of the original isotopomer balance equations. In this article, we show that by reformulating the flux estimation problem in terms of fluxomer variables, it is possible to construct an algorithm that has the following key benefits:

- Provides efficient computation of *all* isotopomers in a metabolic pathway
- Is robust to measurement noise (i.e., suppresses the effects of measurement errors) and initial conditions
- Eliminates the need for measurement scaling factor estimation
- Poses the problem using simple mathematical expressions, allowing the use of generic optimization algorithms

The rest of the article is constructed as follows. The Results and Discussion section illustrates the advantage of our approach via simulation results comparing fluxomer variables to the commonly used cumomer approach and the more recently introduced EMU approach. The Methods section presents the

detailed formulation of the fluxomers optimization problem and the fluxomers iterative algorithm (FIA) that provides a reliable and efficient method for solving it. All source code and executables for our algorithms are freely available at the author’s website [16].

Results and Discussion

We compared our FIA algorithm to the widely used MFA software 13CFLUX [17], which relies on the cumomer approach, and to the more recent OpenFLUX [15] software, which is based on the EMU [14] approach. In order to compare the methods, we conducted flux estimations for various well-studied metabolic pathways. Our first example is based upon the tutorial which Wiechert et al. provide with their 13CFLUX software: the *Embden-Meyerhof* and *Pentose Phosphate* metabolic pathways of *Escherichia coli* [17]. This example compares the running time and robustness of both algorithms in response to input noise. Our second example compares the results and performance of FIA to both an ad-hoc method and the OpenFLUX algorithm for the analysis of lysine production by *C. glutamicum*, as described by Becker et al. [18] and Quek et al. [15].

FIA vs. 13CFLUX Comparison: Embden-Meyerhof and Pentose Phosphate Pathways

In this section we examine a network representing the *Embden-Meyerhof* and *Pentose Phosphate* pathways of *E. coli*, which is based upon the tutorial supplied by Wiechert et al. as part of their 13CFLUX software package. Since our FIA implementation natively supports 13CFLUX input files (i.e. “FTBL” files), the same input files can be used for both algorithms. (Note, however, that FIA does not require definition of free fluxes nor initial values, and thus these are simply ignored when imported). Figure 1 shows the simple network used along with the nomenclature used in previous publications. In addition to the network structure, the models are provided with flux and isotopic measurements as shown in Table 1.

First, we examined the output of the two algorithms for the traditional “noiseless” input file. In order to run the analysis, 13CFLUX requires the user to define a set of “free fluxes” along with their associated initial values [7]. Note that a bad choice of free fluxes or their associated values can result in poor algorithmic performance (both in computation time and accuracy). In fact, under various initial guesses the algorithm did not converge at all. As for FIA, none of the above is required. Since the network along with the given measurements are well defined, in the noiseless case the two algorithms returned similar values for unidirectional fluxes, as can be seen in Table 2. Some slight disagreements were observed for the bi-directional fluxes, which are more poorly identified.

We next compared the algorithms’ sensitivities to noise. In a series of 10 experiments, white Gaussian noise was added to all of the measured isotopomer values, and the outputs and computation times for both algorithms were recorded. As can be seen in Figure 2, unidirectional fluxes remain quite constant and hardly suffer from the added experimental error (for both algorithms). However, the bi-directional fluxes are affected by the added noise. 13CFLUX suffers from a higher variance spread of the estimated values than FIA (thus is more sensitive to the added measurement noise). Note that the difference arises not only due to the mathematical model used, but also due to the stability properties of the optimization method chosen.

We next examined the computational performance of the two methods. Table 3 shows the algorithm running time for convergence (in seconds). The average running time for 13CFLUX was 133 seconds, while for FIA this time was 7 seconds. The running time ratio (13CFLUX/FIA) for individual experiments varied between $\times 9$ to $\times 75$.

FIA vs. OpenFLUX Comparison: Lysine Production by *C. glutamicum*

In this section we examine the analysis of the central metabolism of two lysine-overproducing strains of *Corynebacterium glutamicum*: ATCC 13032 ($lysC^{fbr}$) and its $P_{EFTU}fbr$ mutant. Both express feedback-resistant isoforms of the aspartokinase en-

zyme *lysC*, while the latter is additionally engineered to overexpress the glycolytic enzyme fructose-1,6-bisphosphatase. The example is based upon the measurements provided by Becker et al. [18], who implemented an ad-hoc program to estimate the values of various metabolic fluxes. In their more recent article introducing the OpenFLUX software package [15], Quek et al. chose to compare their results to those of Becker et al. Therefore, we will expand upon their comparison using our FIA implementation. The input file for FIA was constructed using the measurements and pathway structure given in [18] and [15]. As described in [15], the published mass isotopomer fractions were modified for mass interference from non-carbon backbone isotopes using the molecular formula of the amino acid fragments. FIA supports automatic generation of the naturally occurring isotopes correction matrix when the measured molecular formulas are supplied. This adjusts the measured fluxomers vector appearing in the objective function during the process of optimization. If necessary, it is possible not to use this feature but instead to directly supply the algorithm with the corrected measurement values.

When comparing the running times of FIA with OpenFLUX, the different algorithmic approaches of the two must be kept in mind. While OpenFLUX requires the user to supply it with sets of free fluxes, FIA requires no free fluxes nor initial values. OpenFLUX rapidly evaluates dozens of different optimization cycles with random initial values and seeks the best fitting result among them, while FIA uses only one single (longer) run. As such, the convergence probability of OpenFLUX depends on the number of attempts and random values generated during its operation, while the FIA results do not depend on any random value. Furthermore, in its analysis, EMU based algorithms evaluate only the fluxes necessary for measurement comparison, and thus their running time depends both on the metabolic network structure and the amount and location of the given measurements. FIA, on the other hand, can supply the entire set of metabolic fluxes at any given time, with no additional computation requirement (which depends mainly on the network structure).

Measured fluxes as constants

First, we ran the exact same simulation as Quek et al. performed in their article. They supply very accurate (mean error in the order of 0.15 mol%) values for the label measurements, and used the given measured fluxes as if they were noiseless measurements (thus as constants). We start by comparing the simulation time for this simple case. According to [15] and as validated by us using our computer, OpenFLUX required 50 iterations of about 16 seconds each in order to find a decent minimal point, hence about 800 seconds in total. While so, the FIA analysis took 60 seconds for initial analysis and matrices creation, and 300 further seconds for convergence, thus 360 seconds as a whole. Regarding the simulation results, as one can see in Table 4 and Table 5, the fluxes are very close to those calculated before, and the estimated fluxes FIA returned had the lowest residual value compared to the other methods.

Measured fluxes as measurements

We can also run the same optimization, but weight the given flux measurements by their variances. When running this optimization using OpenFLUX (again using 50 iterations), the amount of time was greatly increased, and ended in around 48 minutes. For FIA, on the other hand, the running time was the same as before, thus about 6 minutes. Comparing the results of the algorithms, OpenFLUX suffered from severe convergence problems. Most of its iterations ended without converging at all, while those that did converge yielded useless results, far from the measurements. FIA, on the other hand, succeeded in converging for all scenarios. For the wildtype lysine producing pathway, the results were very close to the ones before (since the fluxes and measurements were quite accurate). For the mutant example, which was less accurate, a reduction of the residual value was achieved by small changes to the measured fluxes. Fluxes and residual values can be examined in Table 4 and Table 5.

Using non-normalized MS measurements

We now show that FIA can easily use incomplete or non-normalized measurements by examining its performance in the example above. The supplied MS measurements were normalized to the $n+1$ backbone carbon atoms of the measured metabolites. Instead of using the supplied normalized data, we multiply each set of metabolite measurements by a random constant number. By doing so, we simulate the case in which only the first 3 (2 for GLY) MS peaks were measured, and had not been normalized. The original and supplied non-normalized measurement values can be found in Table 4. Note that the values were corrected by the molecular formulas of the measured fragments (again, can be automatically performed by FIA). In the absence of normalized data, FIA gave estimated fluxes very close to the previous cases, with very low residual values, as can be seen in Table 5. The running time of the algorithm was not affected by the change.

Conclusions

The main contribution of this article is the introduction of *fluxomers*—a new set of state variables used to simulate ^{13}C metabolic tracer experiments. The fluxomers approach allows the central optimization problem of MFA to be reformulated as a sequence of quadratic programs, which form the basis of the fluxomers iterative algorithm (FIA). Both fluxomers and FIA result in several important benefits compared to flux-isotopomer variables. Among these advantages are (i) a reduction in algorithm running time required for simulation of isotopomer distributions and metabolic flux estimation, (ii) reduced sensitivity to measurement noise and initial flux values and (iii) availability of complete isotopomer information for a given network (as opposed to the EMU approach, which only supplies partial information) without the need for user specification of free fluxes or initial flux values. Additionally, the error model used by the FIA algorithm has the advantage that it depends solely upon isotopomer *ratios* rather than complete isotopomer *fractions*, and therefore it eliminates the need to estimate a normalization factor for each measured isotopomer distribution. Our current results show significant improvements even with re-

gards to simplistic tracer experiments (the running times have been improved by an order of $\times 3$ to $\times 20$ compared to the 13CFLUX algorithm, and about $\times 2$ to $\times 8$ compared to the OpenFLUX implementation). It is important to note that the total time required to obtain an MFA solution is controlled both by (i) the time of each iteration and (ii) the number of optimization iterations that are required to achieve a reliable solution. While a single OpenFLUX iteration is certainly faster than a single iteration of FIA, the FIA algorithm was expressly constructed to provide high reliability in achieving the optimal solution. Therefore, FIA was able to consistently find a better optimal solution in less total time in comparison to the other algorithms examined. Furthermore, extending the fluxomers formulation to other global optimization techniques is straightforward. We expect that reformulating more sophisticated MFA problems—for example, involving optimal experimental design or large-scale metabolic networks—in terms of fluxomer variables will lead to dramatic enhancements of algorithmic efficiency and robustness.

Methods

In the following, we show how to construct and solve MFA problems using fluxomer variables. First we define and explain the basic properties of fluxomers. Then we show how to express MFA balance equations and measurements in terms of fluxomers. Finally, we formulate the MFA optimization problem and present the FIA algorithm for solving it. Throughout this section we use boldface uppercase letters \mathbf{A} to denote matrices, lowercase boldface letters \mathbf{x} to denote vectors, and lowercase letters u for scalars. We use the $\langle \circ \rangle$ product $\mathbf{z} = \mathbf{x} \circ \mathbf{y}$ to represent the element-wise product vector, i.e. $z_i = x_i y_i$. The model formulation will be illustrated using the simple metabolic network shown in Figure 3.

Fluxomers overview

Traditional MFA approaches construct distinct variables for each flux and for each possible labeling state (isotopomer) associated with all metabolites in the

network. Fluxomers, on the other hand, are a composite of these two and therefore allow the network state to be described using only one variable type.

Definition 1 (Fluxomer) *A fluxomer is the rate that a metabolic reaction transfers labeling from one or more specific substrate isotopomers into product isotopomers.*

Taking each fluxomer to be a transformation from one set of labeled atoms into others, we can write its labeling state as an array of binary elements representing the state of each atom it consumes (0 representing an unlabeled atom and 1 representing a labeled atom). Thus, $f_i(1001)$ is a fluxomer of reaction i consuming 4 atoms, with its first and last atoms labeled and two middle atoms unlabeled. When using x as an index for one (or more) of the atoms, we denote a sum of fluxomers where the indicated atom can be either labeled or unlabeled (e.g., $f_i(1x01)$ is the sum of $f_i(1001)$ and $f_i(1101)$). See Figure 3b for a detailed example.

Traditional metabolic fluxes and isotopomer variables can be easily expressed using fluxomers. We start with metabolic fluxes, which are just a sum of their associated fluxomers. For the simple network in Figure 3b we have:

$$\begin{aligned}
 f_{upt} = f_1 &= \sum f_1(ijkl) = f_1(xxxx) \\
 f_2 &= \sum f_2(ij) = f_2(xx) \\
 f_3 &= \sum f_3(ij) = f_3(xx) \\
 f_{out} = f_4 &= \sum f_4(ijkl) = f_4(xxxx) \\
 f_5 &= \sum f_5(ijkl) = f_5(xxxx) \\
 f_{5r} &= \sum f_{5r}(ijkl) = f_{5r}(xxxx).
 \end{aligned} \tag{1}$$

We can also express isotopomer abundances in terms of fluxomer variables for the same example. Because of the assumption that enzymes do not differentiate between the various isotopomers of a given metabolite, the isotopomers within each metabolite pool are distributed uniformly across the outgoing fluxes emanating from that pool. Therefore, the fractional abundance of a given isotopomer within a metabolite pool will determine the fractional contribution of its corresponding fluxomers to the fluxes leaving

that pool:

$$\begin{aligned}
A_{ijkl} &= f_1(ijkl)/f_1 \\
B_{ij} &= f_2(ij)/f_2 = f_5(ij)/f_5 \\
C_{ij} &= f_4(xxij)/f_4 = f_{5r}(ij)/f_{5r} \\
D_{ij} &= f_4(ijxx)/f_4 = f_3(ij)/f_3 \\
E_{ijkl} &= f_{out}(ijkl)/f_{out}.
\end{aligned} \tag{2}$$

Fluxomer balance equations

We now examine the fluxomer balance equations that describe how fluxomers are propagated through the metabolic network. These balance equations represent the main mathematical device for calculating steady-state fluxomer values for a given network. For ease of notation, let us define the vector of metabolic fluxes in our system by $\mathbf{u} \in \mathcal{R}^n$ and the vector of fluxomers as $\mathbf{x} \in \mathcal{R}^m$. As shown above, the metabolic fluxes are calculated from a linear transformation of the fluxomers. Denoting this linear transformation matrix as \mathbf{U} , we can write $\mathbf{u} = \mathbf{U}\mathbf{x}$. We now assume that we are given a certain \mathbf{u} vector and wish to calculate the fluxomers in our system. We start by considering balances on “simple fluxomers”, i.e. those that originate exclusively from a single metabolite pool. (An example of a simple fluxomer is $f_5(01)$ in Figure 3, which derives solely from pool B.) Under conditions of metabolic and isotopic steady state, the rate of 01-labeled molecules entering pool B must balance the rate that 01-labeled molecules leave that pool. Therefore, we can construct a balance on fluxomers around pool B as

$$f_5(01) + f_2(01) = f_1(01xx) + f_3(01) + f_{5r}(01). \tag{3}$$

However, according to eq. 2 the left-hand side of this equation can be re-expressed as $B_{01}(f_5 + f_2) = \frac{f_5(01)}{f_5}(f_5 + f_2)$. Substituting this latter result into the flux balance equation and solving for the fluxomer $f_5(01)$ yields

$$\begin{aligned}
f_5(01) &= \frac{f_5}{f_5 + f_2} (f_1(01xx) + f_3(01) + f_{5r}(01)) \\
&= g(\mathbf{u})(\mathbf{h}^T \mathbf{x}),
\end{aligned} \tag{4}$$

where $g(\mathbf{u})$ is a function of \mathbf{u} alone, and \mathbf{h} is a constant vector. Thus, for this simple case we can solve for the outgoing fluxomer $f_5(01)$ directly in terms of

the fluxomers entering pool B and the total fluxes f_2 and f_5 leaving pool B.

We now turn to the more complex situation in which the output fluxomer originates from more than one metabolic pool. For example, consider fluxomer $f_4(0001)$ coming from pools C and D . Here, the fraction of 0001-labeling carried by flux f_4 is proportional to the abundance of 01-labeling in C and 00-labeling in D :

$$\begin{aligned}
f_4(0001) &= f_4 C_{01} D_{00} \\
&= f_4 \left(\frac{f_1(xx01) + f_5(01)}{f_4} \right) \left(\frac{f_2(00)}{f_3 + f_4} \right) \\
&= g(\mathbf{u})(\mathbf{h}_1^T \mathbf{x})(\mathbf{h}_2^T \mathbf{x}).
\end{aligned} \tag{5}$$

As before, the outgoing fluxomer $f_4(0001)$ can be expressed solely in terms of g —a pure function of \mathbf{u} (always a rational function of outgoing fluxes)—and a product of linear projections of \mathbf{x} .

Without loss of generality, we restrict ourselves to fluxes coming from at most 2 metabolic pools (referred to subsequently as the “left” and “right” pools). When the system reaches steady state, we have

$$\mathbf{x} = \mathbf{g}(\mathbf{u}) \circ (\mathbf{H}_1 \mathbf{x}) \circ (\mathbf{H}_2 \mathbf{x}), \tag{6}$$

where \mathbf{g} is a function $\mathcal{R}^n \rightarrow \mathcal{R}^m$, and $(\mathbf{H}_1, \mathbf{H}_2)$ are two $m \times m$ matrices. This equation allows for the output fluxomers emanating from a specific metabolite pool to be expressed in terms of the total flux vector \mathbf{u} and the fluxomers entering the pool. This enables each outgoing fluxomer to be solved “locally” for the incoming fluxomers. Note that this local calculation does not involve any matrix inversions or other expensive computational procedures. If there are no recycle loops in the network so that all possible paths through the network are non-self-intersecting, this equation can be used to solve sequentially for all “downstream” fluxomers in terms of previously calculated “upstream” fluxomers. In the presence of recycle loops an iterative approach can be constructed to solve for the fluxomers while still avoiding repeated matrix inversions.

Constructing the system matrices

The matrices $\mathbf{H}_1, \mathbf{H}_2 \in \mathcal{R}^{m \times m}$ are defined by

$$\begin{aligned} (\mathbf{H}_1)_{ij} &= 1, \text{ if } x_j \text{ enters the left (for } \mathbf{H}_2, \text{ right)} \\ &\quad \text{source metabolic pool in a reaction} \\ &\quad \text{for which } x_i \text{ is a product} \\ (\mathbf{H}_1)_{ij} &= 0, \text{ otherwise.} \end{aligned} \quad (7)$$

The function $\mathbf{g} : \mathcal{R}^m \rightarrow \mathcal{R}^n$ is defined as

$$g_i(\mathbf{u}) = \frac{\mathbf{g}_{1i}^T \mathbf{u}}{(\mathbf{g}_{2i}^T \mathbf{u}) + (\mathbf{g}_{3i}^T \mathbf{u})}, \quad (8)$$

with $\mathbf{g}_{1i}, \mathbf{g}_{2i}, \mathbf{g}_{3i} \in \mathcal{R}^n$ given by

$$\begin{aligned} (\mathbf{g}_{1i})_j &= 1, \text{ if the fluxomer } x_i \text{ is part of the flux } f_j, \\ (\mathbf{g}_{1i})_j &= 0, \text{ otherwise} \\ (\mathbf{g}_{2i})_j &= 1, \text{ if flux } f_j \text{ exits the left source pool} \\ &\quad \text{in a reaction for which } x_i \text{ is a product,} \\ (\mathbf{g}_{2i})_j &= 0, \text{ otherwise} \\ (\mathbf{g}_{3i})_j &= 1, \text{ if flux } f_j \text{ exits the right source pool} \\ &\quad \text{in a reaction for which } x_i \text{ is a product,} \\ (\mathbf{g}_{3i})_j &= 0, \text{ otherwise.} \end{aligned} \quad (9)$$

In matrix form,

$$\mathbf{g}(\mathbf{u}) = \frac{\mathbf{G}_1 \mathbf{u}}{(\mathbf{G}_2 \mathbf{u}) \circ (\mathbf{G}_3 \mathbf{u})}. \quad (10)$$

Isotopomer measurement formulation

In the following, we develop a systematic method for expressing measured isotopomer variables using fluxomer notation. The final result of the analysis shows that isotopomer measurements can be written simply as the norm of a linear transformation of fluxomers, thus $Err \sim \|\mathbf{A}\mathbf{x}\|^2$. First, we briefly summarize the available isotopomer measurements provided by Nuclear Magnetic Resonance (NMR) and Mass Spectrometry (MS) methods. We then discuss the mathematical modeling of these measurements using fluxomer variables.

Available isotopomer measurements

MFA experiments are typically carried out by (i) introducing a labeled substrate into a cell culture at metabolic steady state, (ii) allowing the system to reach an isotopic steady state, and (iii) measuring isotopomer abundances of metabolic intermediates and byproducts using either MS or NMR analysis. These two measurement techniques provide qualitatively different information about isotopic labeling.

- ^1H NMR: Measures the fractional ^{13}C enrichment of each proton-bound carbon atom, irrespective of the labeling of its neighboring carbon atoms. Both ^{12}C and ^{13}C atoms are distinguishable in the same spectrum, and therefore the peak areas corresponding to different carbon isotopes can be normalized directly.
- ^{13}C NMR: Quantifies isotopomers based on the presence of multiplet peaks (e.g., doublets, triplets, doublet doublets, etc.) in the spectrum caused by two or more neighboring ^{13}C atoms. Because ^{12}C atoms are undetectable by ^{13}C NMR, it is impossible to quantify the overall fraction of each isotopomer unless ^1H NMR spectra are simultaneously obtained. Instead, only the relative ratio of different isotopomers can be assessed by ^{13}C NMR.
- MS: This technique is usually preceded by some form of chromatographic separation (GC or LC) to resolve mixtures into their individual components. These components are then ionized and fragmented in the MS ion source. The ionized particles are separated according to their masses by an electromagnetic filter, and a detector measures the relative abundance of each mass isotopomer. These abundances can be normalized to a fractional scale if all MS peaks corresponding to a particular ion are simultaneously measured.

Previous studies based on flux-isotopomer variables have modeled the measurement error as Gaussian noise added to the fractional isotopomer enrichments. Therefore, if $\hat{\mathbf{y}}$ is the vector of measured isotopomer fractions, this model states that $\hat{\mathbf{y}} = \mathbf{y} + \mathbf{e}$, where \mathbf{e} is the Gaussian error term. However, a more accurate error model would add the measure-

ment noise directly to the physically measured values. The motivation for the traditionally chosen error model is its relative simplicity when expressed using flux-isotopomer variables. Furthermore, since some isotopomers of a specific metabolite may be unmeasurable, the isotopomer fractions cannot be experimentally determined in many cases. This implies the need for an alternative error model that avoids these shortcomings.

Measurement Error Model

We denote the measured isotopomer abundances by a vector $\hat{\mathbf{m}}$. For NMR analysis, the elements of $\hat{\mathbf{m}}$ are proportional to the areas under the different spectral peaks. For MS, they are proportional to the integrated ion counts associated with each mass isotopomer. Since $\hat{\mathbf{m}}$ is the measured quantity, the correct error model is an addition of Gaussian noise so that $\hat{\mathbf{m}} = \mathbf{m} + \mathbf{e}$, where \mathbf{m} is the “true” measurement value. The measured isotopomer fractions $\hat{\mathbf{y}}$ are then expressed as

$$\hat{y}_j = \frac{\hat{m}_j}{\sum_i \hat{m}_i} = \frac{m_j + e_j}{\sum_i (m_i + e_i)}. \quad (11)$$

Let ϵ_j represent the residual between the model-predicted and experimentally measured abundance of a single isotopomer. After multiplying eq. 11 by $\sum_i (m_i + e_i)$ and rearranging, the residual expression becomes

$$\epsilon_j = m_j - \hat{y}_j \left(\sum_i m_i \right) = e_j - \hat{y}_j \sum_i e_i, \quad (12)$$

where ϵ_j is a sum of Gaussian variables. Noting that each measurement m_j is simply proportional to a linear combination of fluxomers, the residual expression eq. 12 takes the form

$$\boldsymbol{\epsilon} = [\mathbf{diag}(\hat{\mathbf{y}})\mathbf{T} - \mathbf{V}]\mathbf{x}, \quad (13)$$

where \mathbf{T} and \mathbf{V} are transformation matrices needed to convert fluxomers to isotopomer measurements and the **diag** operator converts its vector argument into a diagonal matrix. The resulting expression is both a simple sum of Gaussian vectors and affine in \mathbf{x} .

The advantage of this objective function is that it only depends upon the relative isotopomer intensities in the vector $\hat{\mathbf{y}}$ but does not depend upon how

these intensities have been normalized (as long as the transformation matrix \mathbf{T} is constructed accordingly). This eliminates the need to estimate optimal normalization factors that are required by previous algorithms in order to convert experimental measurements into isotopomer fractions. This is true for both MS and NMR measurements, either when conducted alone or used together in the same experiment.

The MFA optimization problem using fluxomers

Now that we have defined both the isotopomer measurements and the feasible solution set, we can formulate the least-squares MFA optimization problem in terms of fluxomer variables. Our objective is to find the flux vector \mathbf{u} that minimizes the measurement error. In addition to the fluxomer balances, usually upper bounds \mathbf{u}_{ub} are provided for all fluxes. As has been proven by Wiechert et al. [6–9], once the inputs to the system and \mathbf{u} are set, the solution (\mathbf{x}, \mathbf{u}) is unique. In other words, the steady-state fluxomer balance equation, eq. 6, is actually an implicit definition of $\mathbf{x}(\mathbf{u})$. With this in mind, the MFA optimization problem can be simply defined as

$$\min_{\mathbf{u} \in \mathcal{Q}} \|\mathbf{A}\mathbf{x}(\mathbf{u}) - \mathbf{b}\|^2 \quad (14)$$

with

$$\mathcal{Q} = \left\{ \mathbf{u} : \begin{array}{l} \mathbf{S}\mathbf{u} = 0 \\ \mathbf{0} \leq \mathbf{u} \leq \mathbf{u}_{\text{ub}} \end{array} \right\},$$

where \mathbf{A} selects the measured elements of the fluxomers vector $\mathbf{x}(\mathbf{u})$, \mathbf{b} contains their associated values, and \mathbf{S} is the stoichiometric matrix of the reaction network. Note that \mathbf{b} may contain non-zero elements only when associated with measurements of absolute flux values. For isotopomer measurements, the associated elements of \mathbf{b} are zero.

Eq. 14 can be solved using various non-convex global optimizing techniques. These optimizers typically require the user to provide subroutines for computing the value of the objective function and its first derivatives at various points along their convergence path. Furthermore, evaluation of the function $\mathbf{x}(\mathbf{u})$ and its derivatives are the main (practically only) time-consuming procedures when solving the

optimization problem in eq. 14. The mathematical formulation of eq. 14 is similar to the optimization problem resulting when using the labels and fluxes variables, with one exception - the implicit formula for $\mathbf{x}(\mathbf{u})$. As shown above, using fluxomers we are able to formulate the propagation equation (and thus solving $\mathbf{x}(\mathbf{u})$) as a multiplication of homogeneous functions of fluxes, and second order functions of fluxomers. Using labels and fluxes, formulating the same equation results in a sum of functions of the same structure, and the homogeneous separation property vanishes. The following sections exploit this unique property of the fluxomers propagation equation in order to achieve great reduction in the system computational complexity, leading to the FIA algorithm.

Fluxomers Iterative Algorithm (FIA)

This section deals with the evaluation of $\mathbf{x}(\mathbf{u})$ along with its gradient using the fluxomer formulation. First, we show that $\mathbf{x}(\mathbf{u})$ can be calculated iteratively while avoiding repeated matrix inversions. Then, we demonstrate how the number of iterations can be reduced using a Newton-type gradient-based algorithm. Finally, we explain how it is possible to greatly increase the sparsity of the system using a simple linear transformation of variables, which further reduces the number of iterations needed for convergence.

Solving the fluxomer balance equations

A simple approach for computing \mathbf{x} given \mathbf{u} is to imitate nature. Once a metabolic network reaches steady state (namely, when \mathbf{u} is constant), changing its input labeling does not affect its flux values \mathbf{u} , but only influences the labeling of its intermediate metabolite pools. The metabolite labeling patterns become gradually mixed and propagated throughout the network until isotopic equilibrium is reached. Accordingly, a simple approach for solving eq. 6 is by using its iterative representation (which is similar to the process taking place in nature):

$$\mathbf{x}_{t+1} = \mathbf{g}(\mathbf{u}) \circ (\mathbf{H}_1 \mathbf{x}_t) \circ (\mathbf{H}_2 \mathbf{x}_t),$$

where \mathbf{x}_t is the fluxomer vector at iteration t and \mathbf{x}_{t+1} is the fluxomer vector at iteration $t + 1$. In order to simulate the steady-state labeling, we initialize the system with the vector \mathbf{x}_0 in which only the input fluxomers are labeled and all others are unlabeled. By recursively substituting \mathbf{x} back into the equation, steady state is eventually reached and the final value of \mathbf{x} is obtained. (This equation represents a non-linear time-invariant Markov chain.) For the *Embden-Meyerhof* and *Pentose Phosphate Pathway* example in the Results and Discussion section, it takes a few hundred iterations to achieve complete stability of the solution (maximal fluxomer value change on the order of $1e-17$). Algorithm convergence for a given input vector is retrieved exactly as in the real biological system, and thus a unique solution always exists (for realistic metabolic networks).

We now show it is possible to reach pathway convergence in much fewer iterations. First, we write eq. 6 as

$$\mathbf{F}(\mathbf{x}, \mathbf{u}) = \mathbf{g}(\mathbf{u}) \circ (\mathbf{H}_1 \mathbf{x}) \circ (\mathbf{H}_2 \mathbf{x}) - \mathbf{x}. \quad (15)$$

Now, in order to find the values of (\mathbf{x}, \mathbf{u}) one needs to solve $\mathbf{F}(\mathbf{x}, \mathbf{u}) = 0$ while holding \mathbf{u} constant. We choose to use one of the classic and powerful algorithms for finding roots of an equation, the well known Newton-Raphson [19] [20] [21] method. Roughly speaking, the change of the \mathbf{x} vector at each iteration is calculated by

$$\mathbf{x}_{t+1} = \mathbf{x}_t - (\mathbf{F}'_{\mathbf{x}}(\mathbf{x}, \mathbf{u}))^{-1} \mathbf{F}(\mathbf{x}, \mathbf{u}),$$

with $\mathbf{F}'_{\mathbf{x}}(\mathbf{x}, \mathbf{u}) = \frac{\partial \mathbf{F}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}}$. The main concern now is the evaluation of the expression $(\mathbf{F}'_{\mathbf{x}}(\mathbf{x}, \mathbf{u}))^{-1} \mathbf{F}(\mathbf{x}, \mathbf{u})$. Here, it turns out that due to the decomposable nature of $\mathbf{F}(\mathbf{x}, \mathbf{u})$, the derivative $\mathbf{F}'_{\mathbf{x}}$ at a point (\mathbf{x}, \mathbf{u}) is the simple matrix

$$\mathbf{F}'_{\mathbf{x}}(\mathbf{x}, \mathbf{u}) = (\mathbf{g}(\mathbf{u}) \circ (\mathbf{H}_1 \mathbf{x})) \mathbf{H}_2 + (\mathbf{g}(\mathbf{u}) \circ (\mathbf{H}_2 \mathbf{x})) \mathbf{H}_1 - \mathbf{I}. \quad (16)$$

Therefore, finding $\mathbf{r} = (\mathbf{F}'_{\mathbf{x}}(\mathbf{x}, \mathbf{u}))^{-1} \mathbf{F}(\mathbf{x}, \mathbf{u})$ is equivalent to solving the linear system of equations

$$(\mathbf{F}'_{\mathbf{x}}(\mathbf{x}, \mathbf{u})) \mathbf{r} = \mathbf{F}(\mathbf{x}, \mathbf{u}). \quad (17)$$

In order to determine the root of the propagation equation, FIA starts with an iteration or two using Newton's correction and then continues with the simple "natural" approach. Applying this method to the *Embden-Meyerhof* and *Pentose Phosphate Pathway* example in the Results and Discussion section,

only a few dozen iterations are now needed. In the next section we show how to reduce both the number of variables and the number of iterations required for convergence by another order of magnitude, without affecting system convergence stability.

Reducing system complexity

The following section introduces a mathematical approach for reducing the number of nonzero elements in our system. Variable reduction techniques such as the recently developed Elementary Metabolite Unit (EMU) network decomposition [14] were developed for application to systems that are modeled using flux-isotopomer variables. Fluxomers and the FIA algorithm, as opposed to prior approaches, allow us to effectively reduce the number of system variables using a simple linear transformation on \mathbf{x} . Our main goal here is to find a transformation for the fluxomer vector \mathbf{x} , $\mathbf{y} = \mathbf{K}\mathbf{x}$ that:

- Reduces the number of its nonzero elements.
- Reduces the computational complexity of solving eq. 16.
- Eases the evaluation of eq. 15.

From eq. 16 we see that the greatest expense is due to inversion of a sum of two linear transformations (\mathbf{H}_1 and \mathbf{H}_2) of \mathbf{x} . From the iterative propagation equation, eq. 15, we see that \mathbf{x} is iteratively calculated by computing the product of the same two matrices. Had it been possible to find a sparse, close-to-diagonal representation for both \mathbf{H}_1 and \mathbf{H}_2 by simply multiplying them by the matrix from the right, both problems would be solved.

In order to accomplish the above, we examine the properties of the concatenation of these two matrices which we denote by \mathbf{H} . Next we find the LU factorization of \mathbf{H} ,

$$\begin{pmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{pmatrix} = \mathbf{H} = \mathbf{L}_H \mathbf{U}_H = \begin{pmatrix} \mathbf{L}_{H1} \\ \mathbf{L}_{H2} \end{pmatrix} \mathbf{U}_H, \quad (18)$$

with \mathbf{L}_H lower triangular and \mathbf{U}_H upper triangular matrices. The matrix \mathbf{L}_{H1} contains the first m rows of \mathbf{L}_H and \mathbf{L}_{H2} contains the last m rows of \mathbf{L}_H . Our

new set of variables now becomes $\mathbf{y} = \mathbf{U}_H \mathbf{x}$, and the new propagation equation is

$$\mathbf{U}_H [\mathbf{g}(\mathbf{u}) \circ (\mathbf{L}_{H1}\mathbf{y}) \circ (\mathbf{L}_{H2}\mathbf{y})] - \mathbf{y} = 0. \quad (19)$$

When expressed in terms of the variable \mathbf{y} , our system becomes much more sparse. This is illustrated in Figure 4 which shows \mathbf{H}_1 , \mathbf{H}_2 , \mathbf{L}_{H1} , \mathbf{L}_{H2} and \mathbf{U}_H for the *Embden-Meyerhof* and *Pentose Phosphate Pathway* example. The transformation has two essential benefits:

1. Reduced computational complexity — note that the derivative $\mathbf{F}'_{\mathbf{x}}$ depends upon the matrices \mathbf{H}_1 and \mathbf{H}_2 which have already been factored, and thus solving Newton's step is straightforward.
2. Fewer iterations needed for convergence.

As a matter of fact, this transformation reduced the number of iterations needed for convergence of the simple *E. coli* example to a total of 5.

Finding $\frac{\partial \mathbf{x}}{\partial \mathbf{u}}$

As discussed above, our optimization problem seeks the minimum of $\|\mathbf{A}\mathbf{x}(\mathbf{u}) - \mathbf{b}\|^2$. In order to converge rapidly, the gradient of the objective function must be computed at each iteration of the algorithm. The key step for computing it is the evaluation of $\frac{\partial \mathbf{x}}{\partial \mathbf{u}}$ (the derivative of the fluxomers as a function of the metabolic fluxes). Since we have an implicit function $\mathbf{F}(\mathbf{x}, \mathbf{u})$ along with a valid solution for $\mathbf{F}(\mathbf{x}, \mathbf{u}) = 0$, the implicit function theorem [22,23] can be used to compute $\frac{\partial \mathbf{x}}{\partial \mathbf{u}}$. Because $\mathbf{F}(\mathbf{x}, \mathbf{u})$ is continuously differentiable around its root, we can write

$$\frac{\partial \mathbf{x}}{\partial \mathbf{u}} = - \left(\frac{\partial \mathbf{F}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{F}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}. \quad (20)$$

In the previous section we showed that $\frac{\partial \mathbf{F}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}}$ can be directly expressed in terms of the system matrices and the vectors \mathbf{x} and \mathbf{u} . The same procedure can

be carried out in order to determine $\frac{\partial \mathbf{F}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}$

$$\begin{aligned} \frac{\partial \mathbf{F}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} = & \text{diag}((\mathbf{H}_1 \mathbf{x}) \circ (\mathbf{H}_2 \mathbf{x})) \left[\frac{\mathbf{G}_1}{((\mathbf{G}_2 \mathbf{u}) \circ (\mathbf{G}_3 \mathbf{u}))} \right. \\ & \left. + \frac{((\mathbf{G}_1 \mathbf{u}) \circ (\mathbf{G}_2 \mathbf{u})) \mathbf{G}_3 + ((\mathbf{G}_1 \mathbf{u}) \circ (\mathbf{G}_3 \mathbf{u})) \mathbf{G}_2}{((\mathbf{G}_2 \mathbf{u}) \circ (\mathbf{G}_3 \mathbf{u}))^2} \right]. \end{aligned} \quad (21)$$

Keeping in mind that $\mathbf{F}_{\mathbf{x}}(\mathbf{x}, \mathbf{u})$ is in its reduced form due to our variable transformation, solving the equation $\left(\frac{\partial \mathbf{F}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \right) \left(\frac{\partial \mathbf{x}}{\partial \mathbf{u}} \right) = -\frac{\partial \mathbf{F}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}$ can be accomplished efficiently.

The initial point

The generation of the initial point for the FIA algorithm is very similar to the standard method used by many interior point algorithms for finding a valid initial point over a convex linear set. We added a fluxes-measurement regularization factor λ in order to generate an initial point closer to the final estimation (and thus speed up the convergence process). The initial point is generated by solving the following simple convex optimization problem:

$$\min_{\mathbf{u}, s \in \mathcal{E}} (-s + \lambda \|\mathbf{A}\mathbf{u} - \hat{\mathbf{u}}\|^2),$$

with

$$\mathcal{E} = \left\{ \mathbf{u} : \begin{array}{l} \mathbf{S}\mathbf{u} = 0 \\ [\mathbf{I}, -\mathbf{I}]\mathbf{u} + [\mathbf{I}, \mathbf{I}][\mathbf{0}, \mathbf{u}_{\mathbf{b}}]^T \geq s \end{array} \right\}$$

with \mathbf{A} a matrix that selects the measurable elements of \mathbf{u} , $\hat{\mathbf{u}}$ the measured elements of \mathbf{u} (if they exist), $\mathbf{0}$ a vector of zeros, and $\mathbf{u}_{\mathbf{b}}$ a vector of the flux upper bounds. The regularization factor λ starts with some large value, and if necessary is reduced until the optimal value of s is greater than 0. Note that when $\lambda \rightarrow 0$ the problem reduces to finding a feasible solution of \mathbf{u} , and thus always has a solution (for well-structured networks).

The algorithm

Summarizing the above discussion leads to the following algorithm for efficient solution of the MFA optimization problem using fluxomers:

i. Matrix preparation (run once per network):

0. Calculate $\begin{pmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{L}_{\mathbf{H}1} \\ \mathbf{L}_{\mathbf{H}2} \end{pmatrix} \mathbf{U}_{\mathbf{H}}$ using LU (PQ) factorization.

ii. Call the optimizer in order to solve

$$\min_{\mathbf{u} \in \mathcal{Q}} \|\mathbf{A}\mathbf{x}(\mathbf{u}) - \mathbf{b}\|^2, \quad \mathcal{Q} = \left\{ \mathbf{u} : \begin{array}{l} \mathbf{S}\mathbf{u} = 0 \\ \mathbf{0} \leq \mathbf{u} \leq \mathbf{u}_{\mathbf{ub}} \end{array} \right\}$$

When requested by the optimizer, return $\mathbf{x}(\mathbf{u})$ and its first derivative:

1. Set $\mathbf{y}_0 = \mathbf{y}_{\text{init}}$.
2. Set $i = 1$.
3. Calculate $\mathbf{y}_i = \mathbf{U}_{\mathbf{H}} [\mathbf{g}(\mathbf{u}) \circ (\mathbf{L}_{\mathbf{H}1} \mathbf{y}_{i-1}) \circ (\mathbf{L}_{\mathbf{H}2} \mathbf{y}_{i-1})]$.
4. If $\|\mathbf{y}_i - \mathbf{y}_{i-1}\|^2 > \epsilon_N$
 - (a) Solve $(\mathbf{F}'_{\mathbf{x}}(\mathbf{x}, \mathbf{u}))\mathbf{r} = \mathbf{F}(\mathbf{x}, \mathbf{u})$.
 - (b) Set $\mathbf{y}_i = \mathbf{y}_i - \mathbf{r}$ according to Newton's method.
5. If $\|\mathbf{y}_i - \mathbf{y}_{i-1}\|^2 > \epsilon_e$ go to 3.
6. Calculate $\mathbf{x} = [\mathbf{g}(\mathbf{u}) \circ (\mathbf{L}_{\mathbf{H}1} \mathbf{y}_i) \circ (\mathbf{L}_{\mathbf{H}2} \mathbf{y}_i)]$.
7. Solve $\left(\frac{\partial \mathbf{F}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \right) \left(\frac{\partial \mathbf{x}}{\partial \mathbf{u}} \right) = -\frac{\partial \mathbf{F}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}$.

The supplied software uses a variant of the “sequential least-squares” algorithm [24, 25] for solving the non-convex optimization problem in eq. 14. This essentially breaks the problem into a sequence of convex optimization problems for which the solution can be readily computed. Note that other algorithms can be easily used with the same procedures described above.

Authors contributions

OS developed and implemented the theory, algorithm, simulations and software. YCE and JDY initiated and supervised the study, and contributed to writing the manuscript. All authors reviewed, edited and approved the final manuscript.

Acknowledgements

JDY is supported by an NSF CAREER Award (CBET-0955251) and by the Vanderbilt Discovery Award program.

References

1. Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P: *Molecular Biology of the Cell, Fourth Edition*. Garland 2008, [http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0815332181].
2. Steuer R: **Review: On the analysis and interpretation of correlations in metabolomic data.** *Brief Bioinform* 2006, **7**(2):151–158, [http://bib.oxfordjournals.org/cgi/content/abstract/7/2/151].
3. Berg JM, Tymoczko JL, Stryer L: *Biochemistry, Fifth Edition : International Version*. W. H. Freeman 2002, [http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0716746840].
4. Varma A, Palsson BO: **Metabolic Flux Balancing: Basic Concepts, Scientific and Practical Use.** *Nat Biotech* 1994, **12**(10):994–998, [http://dx.doi.org/10.1038/nbt1094-994].
5. Marx A, de Graaf AA, Wiechert W, Eggeling L, Sahm H: **Determination of the Fluxes in the Central Metabolism of Corynebacterium Glutamicum by Nuclear Magnetic Resonance Spectroscopy Combined with Metabolite Balancing.** *Biotechnol Bioeng* 1996, **49**:111–129.
6. Wiechert W, Mollney M, Isermann N, Wurzel M, de Graaf AA: **Bidirectional reaction steps in metabolic networks: III. Explicit solution and analysis of isotopomer labeling systems.** *Biotechnol Bioeng* 1999, **66**(2):69–85.
7. Wiechert W, de Graaf AA: **Bidirectional reaction steps in metabolic networks: I. Modeling and simulation of carbon isotope labeling experiments.** *Biotechnol Bioeng* 1997, **55**:101–17.
8. Wiechert W, Siefke C, de Graaf AA, Marx A: **Bidirectional reaction steps in metabolic networks: II. Flux estimation and statistical analysis.** *Biotechnol Bioeng* 1997, **55**:118–35.
9. Mollney M, Wiechert W, Kownatzki D, de Graaf AA: **Bidirectional reaction steps in metabolic networks: IV. Optimal design of isotopomer labeling experiments.** *Biotechnol Bioeng* 1999, **66**(2):86–103.
10. Schmidt K, Nielsen J, Villadsen J: **Quantitative analysis of metabolic fluxes in Escherichia coli, using two-dimensional NMR spectroscopy and complete isotopomer models.** *J Biotechnol* 1999, **71**(1-3):175–89.
11. Schmidt K, Norregaard LC, Pedersen B, Meissner A, Duus JO, Nielsen JO, Villadsen J: **Quantification of intracellular metabolic fluxes from fractional enrichment and 13C-13C coupling constraints on the isotopomer distribution in labeled biomass components.** *Metab Eng* 1999, **1**(2):166–79.
12. Antoniewicz MR, Kelleher JK, Stephanopoulos G: **Determination of confidence intervals of metabolic fluxes estimated from stable isotope measurements.** *Metab Eng* 2006, **8**(4):324–37.
13. Yang TH, Heinzle E, Wittmann C: **Theoretical aspects of 13C metabolic flux analysis with sole quantification of carbon dioxide labeling.** *Comput Biol Chem* 2005, **29**(2):121–33.
14. Antoniewicz MR, Kelleher JK, Stephanopoulos G: **Elementary metabolite units (EMU): a novel framework for modeling isotopic distributions.** *Metabolic engineering* 2007, **9**:68–86, [http://dx.doi.org/10.1016/j.ymben.2006.09.001].
15. Quek LE, Wittmann C, Nielsen L, Kromer J: **OpenFLUX: efficient modelling software for 13C-based metabolic flux analysis.** *Microbial Cell Factories* 2009, **8**:25, [http://www.microbialcellfactories.com/content/8/1/25].
16. Srour O: **FIA Software**[http://webee.technion.ac.il/Sites/People/YoninaEldar/Info/software/FIA/FIA/FIA%20Software.html].
17. Wiechert W, Möllney M, Petersen S, de Graaf AA: **A universal framework for 13C metabolic flux analysis.** *Metabolic engineering* 2001, **3**(3):265–283, [http://dx.doi.org/10.1006/mben.2001.0188].
18. Becker J, Klopprogge C, Zelder O, Heinzle E, Wittmann C: **Amplified expression of fructose 1,6-bisphosphatase in Corynebacterium glutamicum increases in vivo flux through the pentose phosphate pathway and lysine production on different carbon sources.** *Appl Environ Microbiol* 2005, **71**(12):8587–8596.
19. Ypma TJ: **Historical Development of the Newton–Raphson Method.** *SIAM Review* 1995, **37**(4):531–551, [http://link.aip.org/link/?SIR/37/531/1].
20. Deuffhard P: *Newton Methods for Nonlinear Problems*. Springer 2004, [http://www.amazon.com/Newton-Methods-Nonlinear-Problems-Deuffhard/dp/3540210997%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D3540210997].
21. Ortega JM, Rheinboldt WC: *Iterative Solution of Nonlinear Equations in Several Variables (Classics in Applied Mathematics, 30)*. Society for Industrial Mathematics 1987, [http://www.amazon.com/Iterative-Nonlinear-Equations-Variables-Mathematics/dp/0898714613%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0898714613].
22. Jittorntrum K: **An implicit function theorem.** *Journal of Optimization Theory and Applications* 1978, **25**(4):575–577, [http://www.springerlink.com/content/x746u2457466u402/].
23. Krantz SG, Parks HR: *The Implicit Function Theorem: History, Theory, and Applications*.

Birkhäuser Boston 2002, [<http://www.amazon.com/Implicit-Function-Theorem-History-Applications/dp/0817642854%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0817642854>].

24. Kraft D: **Algorithm 733: TOMP–Fortran modules for optimal control calculations.** *ACM Trans. Math. Softw.* 1994, **20**(3):262–281.
25. Gill PE, Murray W, Saunders MA: **SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization** 2002, **12**(4):979–1006.

Figures

Figure 1 - E.Coli EMP and PPP Metabolic Pathways

The *Embden-Meyerhof* and *Pentose Phosphate* metabolic pathways of *Escherichia coli*.

Figure 2 - Measured fluxes values

Bidirectional fluxes calculated using FIA and 13CFLUX for noisy measurement set.

Figure 3 - Simple metabolic network

(a) Standard network representation. Carbon atoms are drawn explicitly with arrows to indicate atom transitions. Unidirectional arrows represent unidirectional fluxes while bidirectional fluxes (such as flux 5) are represented by bidirectional arrows. (b) Fluxomers representation. Each arrow is a group of fluxomers. X's appear on the appropriate atom positions to indicate summation of divergent fluxomers.

Figure 4 - System matrices complexity reduction

$\mathbf{H}_1, \mathbf{H}_2, \mathbf{L}_{\mathbf{H}_1}, \mathbf{L}_{\mathbf{H}_2}$ and $\mathbf{U}_{\mathbf{H}}$ for the simple *E. coli* example. A substantial reduction in nonzero elements between the \mathbf{H} and \mathbf{L} matrices can clearly be seen.

1 Tables

Table 1 - EMP & PPP simulation data

Values are taken from the example input file included in the 13CFLUX demo. Substrate enrichment values are considered as constants.

Label input data

Flux name	Cumomer Index	Value	STD
GLC	#000000	0.445	-
	#100000	0.500	-
	#000001	0.011	-
	#000010	0.011	-
	#000100	0.011	-
	#001000	0.011	-
	#010000	0.011	-
Rul5P	#1xxxx	0.1979	0.002
	#x1xxx	0.0153	0.002
	#xx1xx	0.0284	0.002
	#xxx1x	0.0122	0.002
	#xxxx1	0.0976	0.002
Ery4P	#1xxx	0.0568	0.002
	#x1xx	0.0229	0.002
	#xx1x	0.0118	0.002
	#xxx1	0.0704	0.002
GA3P	#1xx	0.0330	0.002
	#x1x	0.0126	0.002
	#xx1	0.1207	0.002
PEP	#1xx	0.0330	0.002
	#x1x	0.0126	0.002
	#xx1	0.1207	0.002

Table 2 - Comparison of FIA with 13CFLUX for the simple E. coli metabolic network

Comparison of estimated fluxes and mean-square estimation error using “noiseless” data.

Flux name	FIA		13CFLUX	
	Est. Flux	MSE	Est. Flux	MSE
emp1	0.5100	0.0020	0.5099	0.0023
emp2	0.8500	0.0008	0.8500	0.0007
emp3	0.8500	0.0008	0.8500	0.0007
emp4	1.8700	0.0011	1.8700	0.0006
emp5	1.8700	0.0011	1.8700	0.0006
emp6	1.8700	0.0011	1.8700	0.0006
ppp1	0.5100	0.0019	0.5101	0.0023
ppp2	4.4234	0.5483	4.3281	0.9652
ppp2r	4.0834	0.5485	3.9880	0.9657
ppp3	4.4689	1.0365	2.7370	1.1057
ppp3r	4.2989	1.0368	2.5670	1.1057
ppp4r	4.0768	0.3643	4.1740	1.1608
ppp4	4.2468	0.3640	4.3440	1.1604
ppp5r	0.2538	0.1535	0.2680	0.0654
ppp5	0.4238	0.1531	0.4381	0.0655
ppp6r	0.2550	0.0175	0.2560	0.0194
ppp6	0.4250	0.0171	0.4260	0.0188
upt	1.0200	0.0004	1.0200	0.0001
coOut	0.5100	0.0019	0.5101	0.0023

Table 3 - Algorithm running time comparison for FIA vs. 13CFLUX

Running time is shown in seconds.

FIA	6.63	7.56	5.17	6.85	8.83	5.92	9.53	6.47	6.97	6.77
13CFLUX	59.14	56.93	76	121	65.7	451	81.7	173	177	69.65

Table 4 - Relative mass isotopomer fractions comparison for wild-type and mutant C. glutamicum

Experimental and calculated isotopomer MS fractions. The experimental data and ad-hoc simulation results are taken from Becker et al. [18]. The OpenFLUX results are taken from [15]. The simulated “non-normalized” data is generated by multiplying the given values after natural isotope correction by random factors. Several FIA estimations are provided: using the given fluxes as constants (under “const.”), as measurements (under “meas.”), and when using the simulated non-normalized data (under “ratios”). As can be seen, FIA agrees with previous results (even when the data is used without normalization). For the mutant case, better fits are achieved when allowing the supplied fluxes to change as well.

Fragment		Wildtype							Mutant				
		Non-normalized	Exp.	Ad-hoc	OpenFLUX	FIA			Exp.	Ad-hoc	OpenFLUX	FIA	
						const.	meas.	ratios				const.	meas.
ALA 260	M0	206.3562	0.5085	0.509	0.509	0.5099	0.5099	0.5097	0.5230	0.525	0.525	0.5247	0.5247
	M1	102.8634	0.3529	0.354	0.354	0.3534	0.3534	0.3537	0.3410	0.342	0.342	0.3425	0.3425
	M2	4.8452	0.1058	0.106	0.106	0.1063	0.1063	0.1062	0.1030	0.104	0.104	0.1037	0.1037
VAL 288	M0	41.4005	0.3455	0.348	0.348	0.3459	0.3458	0.3457	0.3640	0.366	0.366	0.3661	0.3663
	M1	39.6134	0.3983	0.398	0.398	0.3986	0.3986	0.3987	0.3920	0.392	0.392	0.3921	0.3922
	M2	10.7340	0.1845	0.184	0.184	0.1846	0.1846	0.1847	0.1750	0.175	0.175	0.1750	0.1749
THR 404	M0	194.9082	0.3330	0.334	0.334	0.3343	0.3343	0.3340	0.3440	0.344	0.344	0.3439	0.3439
	M1	159.2226	0.3764	0.376	0.376	0.3757	0.3757	0.3759	0.3730	0.371	0.371	0.3715	0.3721
	M2	35.2094	0.1957	0.196	0.196	0.1956	0.1956	0.1957	0.1910	0.192	0.192	0.1920	0.1918
ASP 418	M0	159.9111	0.3343	0.333	0.333	0.3337	0.3337	0.3334	0.3450	0.343	0.343	0.3432	0.3433
	M1	128.3755	0.3732	0.375	0.375	0.3750	0.3750	0.3752	0.3700	0.370	0.371	0.3708	0.3714
	M2	28.7782	0.1955	0.196	0.196	0.1960	0.1959	0.1960	0.1920	0.193	0.192	0.1924	0.1922
GLU 432	M0	3.8009	0.2469	0.25	0.249	0.2474	0.2473	0.2469	0.2570	0.264	0.264	0.2634	0.2624
	M1	4.4232	0.3648	0.366	0.366	0.3661	0.3661	0.3660	0.3650	0.365	0.365	0.3656	0.3658
	M2	1.7429	0.2412	0.239	0.240	0.2406	0.2406	0.2409	0.2360	0.232	0.232	0.2322	0.2327
SER 390	M0	224.9043	0.4497	0.449	0.448	0.4487	0.4488	0.4490	0.4620	0.463	0.463	0.4635	0.4628
	M1	108.4056	0.3576	0.358	0.358	0.3578	0.3578	0.3580	0.3490	0.349	0.349	0.3491	0.3492
	M2	3.5199	0.1428	0.143	0.144	0.1437	0.1437	0.1434	0.1400	0.140	0.140	0.1399	0.1403
PHE 336	M0	250.7079	0.2712	0.274	0.274	0.2764	0.2764	0.2769	0.2870	0.289	0.289	0.2881	0.2874
	M1	303.6304	0.3816	0.381	0.381	0.3817	0.3817	0.3822	0.3800	0.381	0.381	0.3809	0.3806
	M2	129.5861	0.2282	0.228	0.228	0.2263	0.2264	0.2261	0.2200	0.220	0.220	0.2206	0.2210
GLY 246	M0	738.7580	0.7407	0.742	0.742	0.7417	0.7417	0.7421	0.7410	0.743	0.743	0.7426	0.7426
	M1	39.7395	0.1845	0.185	0.185	0.1852	0.1852	0.1849	0.1830	0.184	0.184	0.1844	0.1844
TYR 466	M0	36.7321	0.2344	0.236	0.236	0.2380	0.2380	0.2384	0.2460	0.249	0.249	0.2481	0.2475
	M1	43.7966	0.3530	0.356	0.356	0.3567	0.3567	0.3572	0.3510	0.358	0.357	0.3572	0.3569
	M2	18.6839	0.2423	0.245	0.245	0.2433	0.2433	0.2431	0.2340	0.238	0.238	0.2387	0.2390
TRE 361	M0	34.1048	0.0613	0.062	0.062	0.0612	0.0612	0.0608	0.0880	0.088	0.088	0.0884	0.0884
	M1	327.3441	0.6040	0.607	0.606	0.6051	0.6051	0.6057	0.5730	0.577	0.574	0.5743	0.5742
	M2	27.0318	0.2070	0.207	0.207	0.2084	0.2084	0.2084	0.2130	0.213	0.213	0.2128	0.2126
Sum of weighted residuals				761	684	654	650	718	1735	1461	1451	1308	

Table 5 - Metabolic fluxes comparison for wild-type and mutant *C. glutamicum*

Estimated metabolic fluxes values for the different approaches - the ad-hoc simulation results from Becker et al. [18], the OpenFLUX results [15], and the FIA results for its various simulated scenarios (measured fluxes used as constants, as measurements, and when using ratios of non-normalized data.)

	Wildtype					Mutant			
	Becker	OpenFLUX	FIA			Becker	OpenFLUX	FIA	
			const.	meas.	ratios			const.	meas.
Glucose 6-phosphate isomerase	49.8	51.2	51.9	52.0	51.5	41.6	40.4	42.1	42.5
Glucose 6-phosphate dehydrogenase	46.8	45.0	44.7	44.7	45.1	56.2	57.5	55.7	55.1
Transaldolase	14	13.4	13.3	13.3	13.4	17.5	17.7	17.3	17.0
Transketolase 1	14	13.4	13.3	13.3	13.4	17.5	17.7	17.3	17.0
Transketolase 2	11.9	11.3	11.2	11.2	11.3	15.8	16.4	15.6	15.4
Glyceraldehyde 3-phosphate dehydrogenase	157.5	158.0	158.2	158.6	158.0	160.8	161.0	161.0	160.5
Pyruvate kinase	147.3	148.0	147.8	148.2	147.6	152.6	152.0	152.5	152.0
Pyruvate dehydrogenase	77.5	75.8	74.8	74.9	74.9	87.5	85.2	85.1	79.7
Pyruvate carboxylase - carboxykinase	34.4	35.8	35.9	36.1	35.8	31.5	32.4	32.5	34.9
Citrate synthase	52.5	50.8	49.6	49.7	49.9	67.7	65.4	65.3	58.9
Isocitrate dehydrogenase	52.5	50.8	49.6	49.7	49.9	67.7	65.4	65.3	58.9
Oxoglutarate dehydrogenase	41.2	39.4	38.2	38.3	38.5	59.9	57.6	57.5	50.7
Aspartokinase	11.2	11.2	11.2	11.4	11.2	14.2	14.2	14.2	15.9

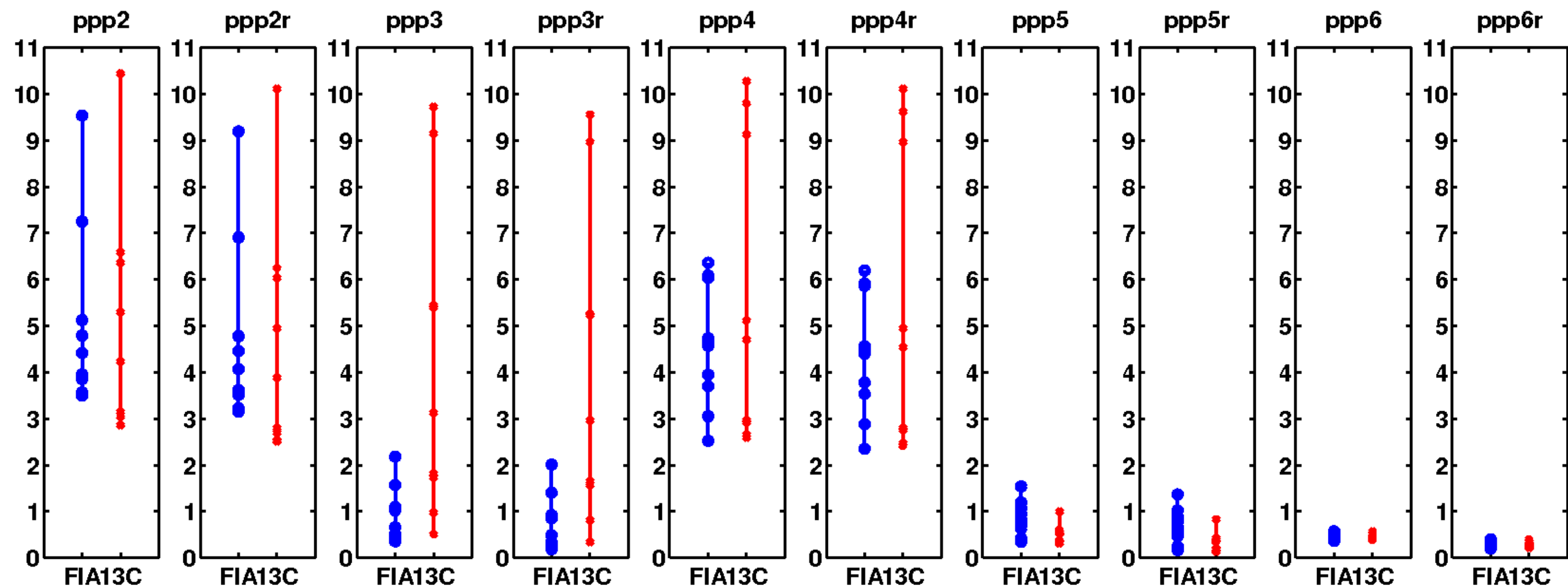


Figure 2

