BIOINFORMATICS ORIGINAL PAPER

Vol. 28 no. 5 2012, pages 687–693
doi:10.1093/bioinformatics/btr716

Systems biology

Advance Access publication December 30, 2011

influx_s: increasing numerical stability and precision for metabolic flux analysis in isotope labelling experiments

Serguei Sokol[1,2,3], Pierre Millard[1,2,3] and Jean-Charles Portais[1,2,3,*]

[1]INSA, UPS, INP, LISBP, Université de Toulouse, 135 Avenue de Rangueil, F-31077 Toulouse, [2]INRA, UMR792, Ingénierie des Systèmes Biologiques et des Procédés, F-31400 Toulouse and [3]CNRS, UMR5504, F-31400 Toulouse, France

Associate Editor: Trey Ideker

ABSTRACT

Motivation: The problem of stationary metabolic flux analysis based on isotope labelling experiments first appeared in the early 1950s and was basically solved in early 2000s. Several algorithms and software packages are available for this problem. However, the generic stochastic algorithms (simulated annealing or evolution algorithms) currently used in these software require a lot of time to achieve acceptable precision. For deterministic algorithms, a common drawback is the lack of convergence stability for ill-conditioned systems or when started from a random point.

Results: In this article, we present a new deterministic algorithm with significantly increased numerical stability and accuracy of flux estimation compared with commonly used algorithms. It requires relatively short CPU time (from several seconds to several minutes with a standard PC architecture) to estimate fluxes in the central carbon metabolism network of *Escherichia coli*.

Availability: The software package influx_s implementing this algorithm is distributed under an OpenSource licence at http://metasys.insa-toulouse.fr/software/influx/

Contact: jean-charles.portais@insa-toulouse.fr

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on July 26, 2011; revised on December 1, 2011; accepted on December 25, 2011

1 INTRODUCTION

Metabolic flux analysis (MFA) aims at quantifying the actual rates of biochemical reactions occurring in living cells. In recent decades, MFA has been increasingly used to identify novel metabolic pathways (Fischer and Sauer, 2003, Peyraud *et al.*, 2009), for in-depth understanding of metabolism (Nicolas *et al.*, 2007, Perrenoud and Sauer, 2005, Sauer *et al.*, 2004). It is extensively used in biotechnology to improve the metabolic properties of industrially relevant organisms (Becker *et al.*, 2007, van Gulik *et al.*, 2000). More recently, MFA has been successfully integrated with other omics tools (transcriptomics, proteomics, metabolomics, etc.) to obtain novel biological insights through systems biology (Ishii *et al.*, 2007, Lemuth *et al.*, 2008, Shimizu, 2004).

The growing interest in MFA underlines the importance of developing reliable tools. The present contribution particularly addresses the need for accurate and stable algorithms for solving the least-squares problem that underlies the calculation of fluxes in MFA.

In a stationary metabolic system, the biochemical reactions which occur in a cell can be described by the following stoichiometric linear equation:

$$Sv = 0$$

where S is $m \times n$ stoichiometric matrix, m rows and n columns correspond to the number of metabolites and reactions, respectively, v is the vector of all net fluxes. Each component of the vector v expresses a net flux, i.e. the net quantity of material converted by a particular reaction per time unit. The whole equation system expresses the mass conservation law in the metabolic system. At metabolic (quasi-)steady-state, the intracellular concentrations of metabolites are kept constant.

For most metabolic systems, the stoichiometry matrix S is under-determined, i.e. the number of equations m is lower than the number of fluxes n. Some fluxes can be measured experimentally. This is generally true of input and output fluxes, but is usually not enough to allow the calculations of all fluxes in the system. The remaining degrees of freedom, so-called free fluxes, need additional equations to be calculated. This can be achieved using different approaches. For example, flux balance analysis (FBA) requires maximization of some linear cost function like biomass yield (Edwards *et al.*, 2001). In the approaches using isotope labelling experiments (ILE) discussed in this article, additional relationships between fluxes come from the measurement of the labelling patterns (or isotopomer distributions) of selected metabolites. Currently, these measurements can be made by mass spectrometry (mass isotopomers) or by Nuclear magnetic resonance (NMR) (positional isotopomers).

The MFA-ILE approach was developed in the 1950s when ^{14}C radioactive isotopes were used to elucidate fragments of carbon metabolism in rat liver (Strisower *et al.*, 1951, Weinman *et al.*, 1950). Since the 1980s–1990s, a stable isotope ^{13}C has preferably been used instead of the radioactive ^{14}C. For many years, the equations describing the label distribution in a given metabolic network and their solution were derived by hand (Heath, 1968). In the early 1990s, general mathematical descriptions of the labelling problem were introduced (Schuster *et al.*, 1992, Wiechert, 1994, Zupke and Stephanopoulos, 1994). This generalization led to a need to solve algebraic systems of high dimensions (often ill-conditioned) to find the labelling state of a given metabolic network. This paved the way for the intensive use of applied mathematics in the MFA field.

[*]To whom correspondence should be addressed.

Wiechert *et al.* (1999) introduced a variable change based on the cumomer (*cum*ulated isotopo*mer*) concept. This approach allowed a large non-linear system to be decomposed into a cascade of smaller linear problems, one per cumomer weight and thus greatly simplified the computational needs for labelling state calculations. A general software exploiting the cumomer approach `13CFlux` was then published, which provides flux estimation based on all types of labelling data (mass and positional isotopomers) (Wiechert *et al.*, 2001).

Antoniewicz *et al.* (2007) proposed another variable change based on elementary metabolite unit (EMU) concept. The EMU framework is based on an efficient decomposition of an atom transition network, which identifies the minimum amount of information needed to simulate the experimental data without any loss of information. This led to a significant reduction in the dimension of the labelling problem. As a consequence, the computation of labelling variables now requires much less time and memory. A software based on EMUs, called OpenFlux, is now available (Quek *et al.*, 2009).

A comparable gain in the computation of cumomers was announced by Weitzel *et al.* (2007). These authors showed that often a highly dimensional linear system describing cumomers of a given weight can be decomposed into series of smaller linear subsystems corresponding to so-called strongly connected components (SCCs). To the best of our knowledge, there is currently no publicly available software exploiting this approach. For a more complete review of recent progress in MFA-ILE, including both experimental and mathematical improvements, see Tang *et al.* (2009).

The EMU and SCC concepts led to computational gains by improving the mathematical formulation of the labelling data problem involved in residue calculation. Another potential source of improvement is the optimization process, i.e. the iterative process by which the simulated labelling data are fitted to the experimental data.

The two steps—i.e. residue calculation and optimization—of the flux calculation process are independent, and can be improved separately. It is likely that future software will combine the most efficient solutions for each calculation step.

Here, we present an original deterministic algorithm called non-linear least squares with inequality constraints (NLSIC) with its improved optimization process. It solves MFA-ILE problems with good convergence robustness without sacrificing convergence speed. NLSIC also enhances the numerical accuracy of a final solution. Moreover, it did not suffer from the local minima problem in the cases we had at hand. We discuss this aspect in the Section 5.

In this article, the performance of the NLSIC algorithm is illustrated by solving a MFA-ILE problem. But this algorithm can be advantageously applied to a larger framework of non-linear least-squares problems with inequality constraints. The NLSIC algorithm is particularly suited to solve constrained least-squares problems with ill-conditioned or even rank-deficient Jacobian.

We start with the formulation of the mathematical problem in MFA-ILE. We then describe the NLSIC algorithm and its practical implementation for flux estimation. The `influx_s` software is further validated in the numerical section by comparing its performance to that of several other algorithms available in the widely used 13CFlux software package. For this purpose, the different algorithms were applied to publicly available data from Zamboni *et al.* (2009).

2 PROBLEM FORMULATION

In this section, we use the same conventions and notations as in Möllney *et al.* (1999), Wiechert *et al.* (1999). The free fluxes and free scaling parameters in a given metabolic system can be estimated using a least-squares problem that can be written as follows:

$$\underset{\Theta,\omega}{\arg\min}\, T(\Theta,\omega)=||F_w(\Theta)-w||^2_{\Sigma_w}+||F_y(\Theta,\omega)-y||^2_{\Sigma_y} \qquad (1)$$

Here T is a cost function representing the sum of squared weighted errors. Its arguments, Θ a free flux vector and ω a free scale vector, are the free parameters that are adjusted during the minimization process. Vectors w and y are the vectors of measured fluxes and labelling data, respectively, whereas vector functions F_w and F_y represent the data simulations matching measured values w and y. Matrices Σ_w and Σ_y are covariance matrices characterizing the experimental noise in flux and labelling data, respectively. They are often assumed to be diagonal as the noise is expected to be uncorrelated.

The solution of (1) must satisfy linear inequality constraints

$$U\begin{pmatrix}\Theta\\ \omega\end{pmatrix}\geq c \qquad (2)$$

where U is an inequality matrix which is multiplied by a compound vector of free parameters Θ and ω, c is a right-hand side vector. Inequalities express some biological conditions which might not be naturally respected by solving the unconstrained problem (1). These conditions can include the unidirectionality of some reactions, or the existence of lower or upper bounds for the flux values, and so on.

Compared with the original cumomer formulation (Wiechert *et al.*, 1999), the main difference consists in using a reduced cumomer set in simulations $F_y(\Theta,\omega)$. Usually, simulations are performed with the entire set of isotopomers allowed by the network. But in most cases, the full set of cumomers is not needed to simulate the experimental data. To reduce the number of cumomers to be calculated, the network is read starting from cumomers involved in simulated data. Then, upstream the fluxes, all the precursor cumomers, up to the input ones, are added to the minimum set. This technique can result in a significant reduction in the number of cumomers to be simulated and, hence, in the calculation time. A similar idea was proposed in the EMU framework (Antoniewicz *et al.*, 2007). For example, for a network of *Escherichia coli* described in Section 5, the total number of cumomers in the complete model was 3183, whereas it was only 2271 in the reduced model. The number of cumomers of weight 1 remains unchanged but a significant reduction is obtained for all other weights (Supplementary Material 1). The complexity of solving a dense linear system is $O(n^3)$ where n is the problem size. The theoretical speedup in the cumomer calculations for the above-mentioned example is $\Sigma n^3_{\text{full}}/\Sigma n^3_{\text{reduced}}\approx 2.8$ where the sum involves the problem sizes of all cumomer weights present in the system. The savings are highly dependent on the network and on the measurements concerned. The greatest reduction in the total number of cumomers is expected for the simulation of positional labelling data where only cumomers of weight 1 are used. So all cumomers of higher weights can be excluded from the simulations without loss of information.

For the sake of simplicity, and without loss of generality, we omit writing vector ω in the free parameter vector. We also omit the matrices Σ_w and Σ_y. So, from now on, Θ denotes the whole vector of free parameters. The vector functions F_w and F_y are combined in one vector function F, and vectors w and y in one vector u. Using these new terms, the problem (1, 2) takes the form of a classical non-linear least-squares problem with inequality constraints:

$$\begin{cases} \operatorname{argmin}_\Theta T(\Theta) = ||F(\Theta) - u||^2 \\ U\Theta \geq c \end{cases}$$

A Jacobian matrix at a point Θ defined as $J(\Theta) = \partial F(\Theta)/\partial\Theta$ and a residual vector $r(\Theta) = F(\Theta) - u$, which also depends on Θ, are introduced. Taking into account the first-order Taylor development

$$||r(\Theta + p)|| = ||r(\Theta) + J(\Theta)p|| + O(||p||^2)$$

and searching for a vector p that minimizes the norm $||r(\Theta + p)||$, a linearized incremental form of the problem in the point Θ is obtained:

$$\begin{cases} Jp = -r \\ Up \geq c - U\Theta \end{cases} \tag{3}$$

where the new variable vector p is an unknown correction to the current free parameter vector Θ and the equality sign must be interpreted in the least-squares sense. This linear problem is hereafter referred to as least-squares with inequalities (LSI).

3 NLSIC ALGORITHM

Generally, non-linear, unconstrained optimization problems like (1) can be solved using a wide range of methods which are well described in text books. These methods can be divided into two classes: stochastic and deterministic algorithms. Among the most frequently cited deterministic algorithms are gradient descent, conjugate gradient, Broyden-Fletcher-Goldfarb-Shanno optimization algorithm (BFGS). To deal with inequality constraints, the deterministic algorithms have to be used in conjunction with appropriate techniques, e.g. interior point methods like barrier function or active set. Non-linearity is most often treated using a classical line search approach, sequential quadratic programming (SQP) or a combination of the two approaches. Finally, techniques which are widely used to make these algorithms globally converge, i.e. independent of the distance between the starting point and the convergence point, are trust region and backtracking. All these algorithm aspects—non-linearity, inequalities, globalization and stopping criterion—are detailed hereafter.

3.1 Non linearity

The non-linearity approach in the NLSIC algorithm is close to the classical SQP method. The latter solves a normal (square) linear system at each iteration:

$$Hp = -J^T r$$

where H is a matrix proportional to Hessian, i.e. the matrix of second partial derivatives of the cost function T

$$H_{ij} = (J^T J)_{ij} + \Sigma_k \frac{\partial^2 r_k}{\partial\Theta_i \partial\Theta_j} r_k$$

or more frequently H is taken to be just $J^T J$; while in the NLSIC algorithm, a least-squares problem with rectangular matrix J is solved

$$Jp = -r. \tag{4}$$

This point is crucial for numerical stability and precision even if its numerical cost is higher than solving the normal system. In real-world problems, like in MFA-ILE, the Jacobian J is often ill-conditioned, i.e. $\kappa(J) \gg 1$ (here κ is a condition number) so that the condition of $J^T J$ will be even worse. For example, in l^2-norm $\kappa(J^T J) = \kappa(J)^2$. If $\kappa(J)$ is say 10^7, the problem (4) can still be solved in double precision arithmetics with satisfactory precision; whereas the corresponding normal system has $\kappa(J^T J)$ as high as 10^{14} and will be considered as numerically singular. It is also well known that the matrix condition is a determining factor for convergence stability and for the precision of many numerical methods. By solving a system with matrix J instead of H, condition deterioration is avoided and hence numerical stability and precision are preserved.

3.2 Inequalities

The presence of inequalities makes the optimization problem more difficult from a numerical point of view. Like for SQP (Liu, 2005), for NLSIC, an active set method was chosen to deal with inequalities.

As an overdetermined linear system has to be solved at each non-linear iteration, one particular method called non-negative least squares (NNLS) suits this purpose well (Lawson and Hanson, 1974). To reduce the LSI problem to an NNLS problem, an intermediate problem called LDP has to be formulated as detailed hereafter.

Let us start by formulating an NNLS problem. Given a rectangular $m \times n$ matrix A and right-hand side vector b of size m, find a vector x of size n such that $x_i \geq 0$, $\forall i$. In short:

$$\begin{cases} Ax = b \\ x_i \geq 0, \quad \forall i \end{cases} \tag{5}$$

here again, the equality sign must be interpreted in the least-squares sense.

The NNLS algorithm solves this problem by combining QR decomposition (Björck, 1996) of the matrix A with an active set method. One attractive feature of this algorithm is that QR decomposition is not recalculated each time a new active set is tried. Only those elements of matrices Q and R that are affected by changes in the active set, are updated. Another feature worth mentioning is that an appropriate active set is rapidly found in only a few iterations, at least in the problems we had to deal with. An interface to this algorithm in R language is available under OpenSource licence (http://cran.r-project.org/web/packages/nnls/).

It can be shown that an NNLS problem can be set as a reformulation of another so-called least distance programming (LDP) problem (Lawson and Hanson, 1974). An LDP problem is formulated in the following way: find a vector z of smallest norm l^2 satisfying a set of inequalities

$$Ez \geq f \tag{6}$$

where E and f are any real rectangular matrix and vector of appropriate sizes, respectively. Given an LDP problem, an equivalent NNLS problem can be obtained by setting

$$A = \begin{pmatrix} E^T \\ f^T \end{pmatrix}$$

$$b = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

Once NNLS is solved for x, a residual vector $v = Ax - b$ is calculated and the solution for LDP is given by $z_i = -\frac{v_i}{v_{n+1}}$, for $i = 1, \ldots, n$. The case $v_{n+1} = 0$ corresponds to infeasible inequality constraints (6).

In turn, the LDP solution can be used to address the LSI problem (3). To reduce the problem (3) to (6), the following variable change is performed

$$z = Rp + Q^T r$$

where Q and R are components of QR decomposition of the Jacobian $J = QR$. We then put

$$E = UR^{-1}$$

$$f = c - U(\Theta - R^{-1}Q^T r).$$

Once LDP with E and f is solved for z, a vector p can be calculated as $p = R^{-1}(z - Q^T r)$.

It can happen that during iterations the Jacobian becomes rank deficient. In this case, we provide an option to continue iterations with an approximate least norm solution. A strict least norm solution always exists and is unique if the inequalities are feasible. A description of the method chosen to obtain an approximate least norm solution for p satisfying inequality constrains is beyond the scope of this article. Interested readers can refer to the source code of the function `lsi_ln()` in the file `nlsic.R` distributed with the `influx_s` software.

To summarize, starting from the LSI problem, an equivalent LDP problem is formulated which in turn is reformulated as an NNLS problem. Having solved NNLS, we go backwards in the same way: first the solution for LDP and then the solution for the original LSI problem are calculated.

3.3 Globalization

The last but not least component of the NLSIC algorithm is the use of a backtracking algorithm to ensure the global convergence of the non-linear iterations. The backtracking was derived and successfully used to solve systems of non-linear equations and later for non-linear unconstrained optimization (Dennis and Schnabel, 1996). The key idea behind this method is to iteratively shorten a candidate vector p if at some point Θ the norm $||r(\Theta+p)||$ is much higher than predicted by the linear model norm $||r(\Theta)+Jp||$. If r is differentiable in the neighbourhood of Θ, a sufficiently small vector p can always be found such that two norms are close enough. Backtracking ensures that the increment vector p remains in the validity domain of linear approximation (3), and hence ensures the norm $||r(\Theta)||$ is reduced during non-linear iterations. Near the solution, where the first value of p found is sufficiently small to satisfy backtracking conditions, a very rapid convergence of Newton-like methods is often observed.

Algorithm 1 NLSIC

```
1. Set initial values $i := 0$, $\Theta_0$
2. Solve LSI problem $J_i p_i = -r_i$ subject to $Up_i \geq c - U\Theta_i$
3. If $||p_i||$ is sufficiently small or the limit of iteration number
   is reached, stop iterations with $\Theta := \Theta_i + p_i$
4. While backtracking condition (7) is not satisfied or
   maximal number of backtrack iterations is reached, iteratively
   set $p_i := \alpha p_i$ with $0 < \alpha < 1$
5. Set $\Theta_{i+1} = \Theta_i + p_i$
6. Set $i := i + 1$ and restart with p. 2.
```

More formally, the backtracking method consists in iteratively setting $p := \alpha p$ with some positive constant $\alpha < 1$ until the following backtrack condition is fulfilled

$$||r(\Theta+p)|| \leq ||r(\Theta)|| + \beta r^T Jp \qquad (7)$$

with some constant $0 < \beta < 1$. Parameters α and β are chosen *a priori* by the user.

As the original backtracking method was designed to be used in an unconstrained context, it is necessary to check that it can be used in the context of inequality constraints. This is easy to do by observing that if two vectors Θ and p are such that Θ and $\Theta + p$ satisfy the inequalities, i.e. $U\Theta \geq c$ and $U(\Theta+p) \geq c$ then by multiplying both inequalities by positive numbers $(1-\alpha)$ and α respectively and summing them, we obtain $U(\Theta + \alpha p) \geq c$. This means that for any α, the intermediate vector of backtracking iterations $\Theta + \alpha p$ cannot leave the feasibility domain delimited by the inequalities.

As mentioned above, due to backtracking, the sequence of norms $||r(\Theta_i)||$ monotonously decreases. This sequence has a low bound 0, and is hence necessarily convergent. This guarantees that the whole NLSIC algorithm is globally convergent under the assumption that there is no significant problem due to finite precision arithmetics. Strictly speaking, the convergence point may differ from the global minimum but achieving convergence independently of the starting point is already appreciable.

3.4 Stopping criterion

The stopping criterion in NLSIC is based on the norm of the correction vector p and not on the variation of the cost function T as it is usually the case. This point is important for ill-conditioned systems, as relatively big correction vectors sometimes produce very small variations in cost function. In this situation, the error control based on T variations could stop iterations relatively far from the solution. In practice, this could be mistakenly interpreted as falling in a local minimum. In a nutshell, the NLSIC algorithm can be classified as a sequential LSI algorithm (based on NNLS) with backtracking globalization.

4 IMPLEMENTATION

Input data for the MFA-ILE problem, i.e. metabolic network, carbon transitions in reactions and measurements are all supplied in a plain text file. The input file must be in the FTBL format developed for the wide-spread `13CFlux` software (http://www.13cflux.net)

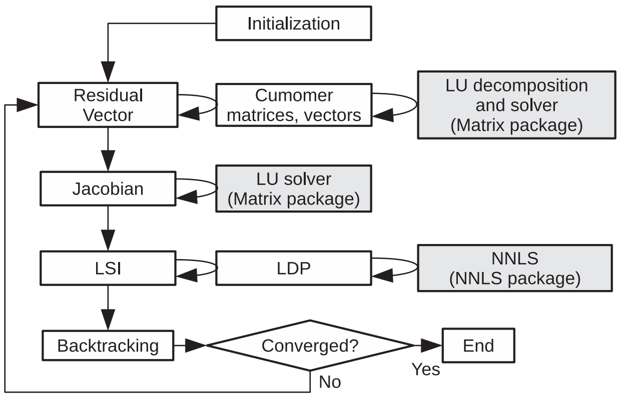
Initialization
Residual Vector
Cumomer matrices, vectors
LU decomposition and solver (Matrix package)
Jacobian
LU solver (Matrix package)
LSI
LDP
NNLS (NNLS package)
Backtracking
Converged?
End
Yes
No

Fig. 1. Flow diagram of the NLSIC algorithm. The third part software packages are in the grey boxes. Curved arrows mean 'need a call to …', straight arrows mean 'go to the next block'.

(Wiechert *et al.*, 2001). A module of influx_s package parsing input file and generating an executable code is written in Python (http://python.org). The executable program calculating flux estimations (specific to each network) is automatically generated in R language (http://www.r-project.org).

The NLSIC algorithm is also programmed in R. As it is an interpreted language, for the sake of calculation speed, the most time consuming linear algebra operations: *QR* and *LU* decompositions, NNLS algorithm and so on, are programmed in FORTRAN and C in several third part software distributed independently of influx_s. A small part of the influx_s package concerning sparse matrix updates is written in FORTRAN. This part of the executable code is compiled only once and is not regenerated for every new network input. The use of sparse matrices (http://cran.r-project.org/web/packages/Matrix/) greatly reduces memory and CPU requirements, especially in the case of large networks which are transformed into huge matrices. Third part software used by NLSIC is indicated in Figure 1 in grey boxes.

For the sake of rigorous simulations, an option to include the growth flux μM into the stoichiometric balance is provided. Even if its inclusion does not have a major impact on the final results in most common situations (data not shown), users can themselves check that this is true in their own case. It should be noted that this option requires absolute concentrations for intracellular metabolites, which may be not very easy to measure.

Confidence intervals of the free and dependent fluxes can be assessed using two methods in influx_s: by linearized statistics as proposed in Möllney *et al.* (1999) or by Monte Carlo simulations. The latter is programmed for parallel execution on multi-core architecture where a third part R package multicore is available.

The influx_s software was developed and tested as a command line tool on a Linux platform but potentially it can be run on any platform where Python and R are installed. At the time of writing, the multicore package is not stable enough on Windows platforms and consequently, until a stable version is available, users will have to run such simulations on a single core on their Windows platform. The package influx_s is available at http://metasys.insa-toulouse.fr/software/influx/ under OpenSource licence. All third part libraries used in our software are also freely available at their respective sources.

5 NUMERICAL RESULTS

In this section, a publicly available FTBL file Ecoli.ftbl from Supplementary Material 3 of Zamboni *et al.* (2009) was used to compare the NLSIC algorithm implemented in the influx_s software with the three optimization algorithms used in the 13CFlux software (v20050329): Evolution, BFGS and donlp2 (Spellucci, 1993). The 13Cflux software is widely distributed throughout the biological community concerned with MFA and, in the last decade, has become *de facto* reference method for flux calculations.

The file Ecoli.ftbl describes a central metabolic network of the bacterium *E. coli*. The network includes central carbon metabolism (glycolysis, gluconeogenic reactions, pentose phosphate pathways, the tricarboxylic acid cycle, glyoxylate shunt and anaplerotic reactions) and reactions for amino acid biosynthesis. It contains 35 internal metabolites, 68 reactions (of which 16 are reversible and 52 are not reversible). The labelling data includes 193 isotopomer measurements obtained only by gas chromatography–mass spectrometry (GC–MS). Free parameters were composed of 27 free fluxes and 35 scale parameters.

The GC-MS data provided in the FTBL file are not sufficient to determine all free fluxes. So the network is structurally undefined. This leads to rank-deficient Jacobian. So the least norm solution (available in influx_s with an option --ln) that we discussed in Section 3.2, was revealed to be indispensable in this situation. Another problem encountered with this FTBL file was that cumomer balance matrix could become singular when some fluxes vanished to 0. To prevent this happening, the net fluxes over non-reversible reactions were constrained to be over 10^{-4} with an option --clownr=1e-4.

To compare various algorithms, a suite of 10 FTBL files was generated. These files differed only in their random starting points while the network and measurement sections remained the same. Initial flux values were uniformly drawn from [0, 1] interval and the resulting vector was projected on the feasibility domain by solving an LDP problem.

As the Evolution algorithm implemented in 13CFlux does not include a stopping criterion, it was run for a fixed time of 3 h on each input file and the results achieved at this time (and corresponding to the best fit) are shown. The stopping criterion for BFGS and donlp2 algorithms were those set by default in 13CFlux software. The running time for these two algorithms were limited to 15 min. The BFGS algorithms was often stalled at this time, and donp2 stopped before reaching the time limit. The NLSIC algorithm was stopped when the norm $||p||$ was lower than 10^{-5} or the upper limit on the non-linear iteration number (50) was reached. All numerical experiments were run on a laptop with 1 GHz bi-core processor (actually only one core was used) and 2 GB of RAM.

Figure 2 shows box-plots representing the spread of the final cost values for the four methods applied to 10 different starting points and the three deterministic methods: BFGS, donlp2 and NLSIC when they followed the results of the Evolution method. The double algorithm such as a stochastic algorithm followed by a deterministic algorithm is sometimes recommended (Zamboni *et al.*, 2009) to avoid local minima while avoiding prohibitive calculation time. The final cost values for the Evolution algorithm ranged between 156 and 268. The BFGS methods achieved the minima between 138 and 4409 when run alone and between 156 and 268 when it followed the Evolution results. In fact, BFGS rarely improved

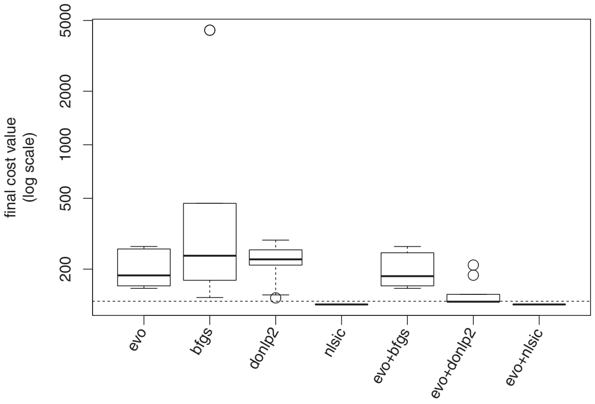

Fig. 2. The spread of final cost values from 10 random starting points for four algorithms: evolution algorithm, BFGS, donlp2 and NLSIC. NLSIC was shown to be the most stable, it converged to the same point in all cases. It was also the most accurate as its final cost value was the lowest among the four algorithms tested. The dashed line corresponds to a level of 132 found in Zamboni *et al.* (2009).

the Evolution results and most often stagnated at the same level of the cost value. This behaviour was also observed in Zamboni *et al.* (2009). The algorithm donlp2 produced better results. Its final cost was spread between 138 and 291 when run alone and between 131 and 211 when preceded by the Evolution algorithm. Finally, the NLSIC algorithm converged to the same value (127) from all starting points and achieved the lowest cost of all the results obtained whether preceded or not by the Evolution algorithm. While multiple convergence points for the BFGS and donlp2 methods could be interpreted as local minima in which the convergence was trapped, in fact this was not the case. Since using the final results of, for example, the Evolution or BFGS algorithms as starting point for NLSIC, it continued convergence until it reached the previously found solution, 127.

The network was not well defined by the data that were provided, so almost all fluxes are statistically undetermined. Zamboni *et al.* (2009) had to constrain 21 of 27 free fluxes to be able to evaluate the confidence intervals of the remaining fluxes. Of all the free fluxes in the original (not constrained) FTBL file, only the uptake flux was statistically determined. In Figure 3, the box-plots corresponding to this flux are shown for all algorithms tested. Ideally, this value as well as the final cost should not depend on the starting points. When this happens, for example with the Evolution, BFGS and donlp2 algorithms, this numerical instability adds to already present experimental noise and reduces the quality of flux assessment. The NLSIC algorithm was almost free of this drawback as can be seen in Figure 3.

A word about calculation times. Zamboni *et al.* (2009) estimated the time necessary for flux calculation using 13CFlux software to be 1 day. This time included several runs of the Evolution algorithm, optionally followed by some deterministic algorithm(s) and choosing the lowest final cost value of all the runs. In the example given here, NLSIC ran only for several minutes and provided a better minimum, 127, than the 132 found by the authors of the above-cited paper. By considerably shortening the calculation time, our software represents an additional step in the direction of high-throughput flux calculation.

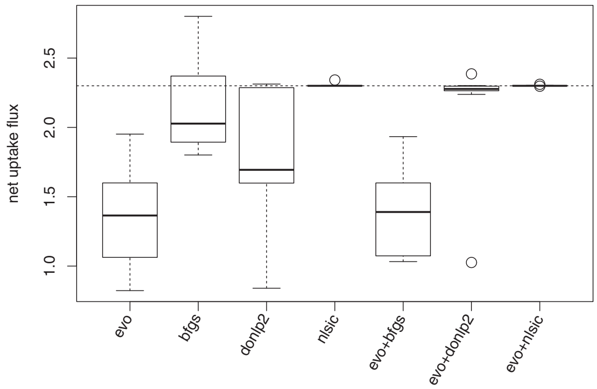

Fig. 3. Spread of the uptake flux estimated by the tested algorithms for 10 random starting points. The NLSIC algorithm produced the most stable results and the closest to the measured value 2.3 represented by the dashed line.

6 CONCLUSION

We developed an algorithm for solving NLSIC, which was used to solve the MFA-ILE problem. In this field, it outperformed widely used algorithms such as BFGS, donlp2 or Evolution algorithms not only in numerical stability but also in the accuracy of the solution. The increased numerical accuracy led us to conclude that a problem of local minima, as often mentioned in the literature dedicated to MFA-ILE, could in some cases be a false problem. Probably, it was lack of precision in convergence which was interpreted as trapping in a local minimum.

The significantly improved computer efficiency, accuracy and reliability of `influx_s` makes the MFA-ILE approach more accessible for a wide biological community interested in fluxomics. We can reasonably expect that the NLSIC algorithm will also provide the same benefits in future MFA-ILE developments dealing with the dynamics of labelling propagation in metabolically stable networks.

The software `influx_s` implementing NLSIC in the MFA-ILE context is distributed under OpenSource licence. It has the same general character and input format as `13CFlux` software as it can take into account all types of labelling data coming both from MS and NMR.

ACKNOWLEDGEMENTS

The authors wish to thank members of the MetaSys team (LISBP, Toulouse) for fruitful discussions.

Funding: French *Agence Nationale de la Recherche* ('ALGOMICS' ANR-08-BIOENERGIES-002); PhD fellowship of P.M. was supported by the *Institut National de la Recherche Agronomique* (INRA) [CJS]; Région Midi-Pyrénées; European Regional Development Fund (ERDF); French Ministry for Higher Education & Research; SICOVAL; INRA.

Conflict of Interest: none declared.

REFERENCES

Antoniewicz,M.R. *et al.* (2007) Elementary metabolite units (EMU): a novel framework for modeling isotopic distributions. *Metab. Eng.*, **9**, 68–86.

Becker,J. *et al.* (2007) Metabolic flux engineering of L-lysine production in *Corynebacterium glutamicum*—over expression and modification of G6P dehydrogenase. *J. Biotechnol.*, **132**, 99–109.

Björck,Å. (1996) *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia.

Dennis,J.E. and Schnabel,R.B. (1996) *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia.

Edwards,J.S. *et al.* (2001) In silico predictions of *Escherichia coli* metabolic capabilities are consistent with experimental data. *Nat. Biotechnol.*, **19**, 125–130.

Fischer,E. and Sauer,U. (2003. Metabolic flux profiling of *Escherichia coli* mutants in central carbon metabolism using GC-MS. *Eur. J. Biochem.*, **270**, 880–91.

van Gulik,W.M. *et al.* (2000) Application of metabolic flux analysis for the identification of metabolic bottlenecks in the biosynthesis of penicillin-G. *Biotechnol Bioeng*, **68**, 602–618.

Heath,D.F. (1968) The redistribution of carbon label by the reactions involved in glycolysis, gluconeogenesis and the tricarboxylic acid cycle in rat liver. *Biochem. J.*, **110**, 313–335.

Ishii,N. *et al.* (2007) Multiple high-throughput analyses monitor the response of *E. coli* to perturbations. *Science*, **316**, 593–597.

Lawson,C.L. and Hanson,R.J. (1974) *Solving Least Squares Problem*. Prentice-Hall Series in Automatic Computation. Prentice-Hall, Englewood Cliffs, New Jersey.

Lemuth,K. *et al.* (2008) Global transcription and metabolic flux analysis of *Escherichia coli* in glucose-limited fed-batch cultivations. *Appl. Environ. Microbiol.*, **74**, 7002–7015.

Liu,X.-W. (2005) Global convergence on an active set SQP for inequality constrained optimization. *J. Comput. Appl. Math.*, **180**, 201–211.

Möllney,M. *et al.* (1999) Bidirectional reaction steps in metabolic networks: IV. Optimal design of isotopomer labeling experiments. *Biotechnol. Bioeng.*, **66**, 86–103.

Nicolas,C. *et al.* (2007) Response of the central metabolism of *Escherichia coli* to modified expression of the gene encoding the glucose-6-phosphate dehydrogenase. *FEBS. Lett.*, **581**, 3771–3776.

Perrenoud,A. and Sauer,U. (2005) Impact of global transcriptional regulation by ArcA, ArcB, Cra, Crp, Cya, Fnr, and Mlc on glucose catabolism in *Escherichia. coli. J. Bacteriol.*, **187**, 3171–3179.

Peyraud,R. *et al.* (2009) Demonstration of the ethylmalonyl-CoA pathway by using ^{13}C metabolomics. *Proc. Natl. Acad. Sci. U S A*, **106**, 4846–4851.

Quek,L.-E. *et al.* (2009) OpenFLUX: efficient modelling software for ^{13}C-based metabolic flux analysis. *Microb. Cell Fact.*, **8**, 25.

Sauer,U. *et al.* (2004) The Soluble and Membrane-bound Transhydrogenases UdhA and PntAB Have Divergent Functions in NADPH Metabolism of *Escherichia coli*. *Journal of Biological Chemistry*, **279**, 6613–6619.

Schuster,R. *et al.* (1992) Simplification of complex kinetic models used for the quantitative analysis of nuclear magnetic resonance or radioactive tracer studies. *J. Chem. Soc., Faraday Trans.*, **88**, 2837–2844.

Shimizu,K. (2004) Metabolic flux analysis based on ^{13}C-labeling experiments and integration of the information with gene and protein expression patterns. *Adv. Biochem. Eng. Biotechnol.*, **91**, 1–49.

Spellucci,P. (1993) A SQP method for general nonlinear programs using only equality constrained subproblems. *Math. Program.*, **82**, 413–448.

Strisower,E.H. *et al.* (1951) Conversion of C14-palmitic acid to glucose. I. Normal and diabetic rats. *J. Biol. Chem.*, **192**, 453–463.

Tang,Y.J. *et al.* (2009) Advances in analysis of microbial metabolic fluxes via ^{13}C isotopic labeling. *Mass Spectrom. Rev.*, **28**, 362–375.

Weinman,E.O. *et al.* (1950) Relative rates of conversion of the various carbon atoms of palmitic acid to carbon dioxide by the intact rat. *J. Biol. Chem.*, **184**, 735–744.

Weitzel,M. *et al.* (2007) The topology of metabolic isotope labeling networks. *BMC Bioinformatics*, **8**, 315.

Wiechert,W. (1994) Design of a software framework for flux determination by ^{13}C NMR isotope labelling experiments I. In Gnaiger,E. *et al.* (eds) *What is Controlling Life?* Vol. 3 of *Modern Trends in BioThermoKinetics*. Innsbruck University Press, Innsbruck, pp. 305–310.

Wiechert,W. *et al.* (1999) Bidirectional reaction steps in metabolic networks: III. Explicit solution and analysis of isotopomer labeling systems. *Biotechnol. Bioeng.*, **66**, 69–85.

Wiechert,W. *et al.* (2001) A universal framework for ^{13}C metabolic flux analysis. *Metab. Eng.*, **3**, 265–283.

Zamboni,N. *et al.* (2009) 13c-based metabolic flux analysis. *Nat. Protocols*, **4**, 878–892.

Zupke,C. and Stephanopoulos,G. (1994) Modeling of isotope distributions and intracellular fluxes in metabolic networks using atom mapping matrixes. *Biotechnol. Progr.*, **10**, 489–498.