DISCOPOLIS 2.0: a new recursive version of the algorithm for uniform sampling of metabolic flux distributions with linear programming

Philippe Bogaerts*, Marianne Rooman**

**3BIO-BioControl, **3BIO-BioInfo, Université Libre de Bruxelles*
Av. F.-D. Roosevelt 50 C.P. 165/61, B-1050 Brussels, Belgium
(e-mail: philippe.bogaerts@ulb.ac.be ; mrooman@ulb.ac.be)

Abstract: Metabolic flux values are subject to equality (*e.g.*, mass balances, measured fluxes) and inequality (*e.g.*, upper and lower flux bounds) constraints. The system is generally underdetermined, *i.e.* with more unknown fluxes than equations, and all the admissible solutions belong to a convex polytope. Sampling that polytope allows subsequently computing marginal distributions for each metabolic flux. We propose a new version of the DISCOPOLIS algorithm (DIscrete Sampling of COnvex POlytopes via Linear program Iterative Sequences) that provides the same weight to all the samples and that approximates a uniform distribution thanks to a recursive loop that computes variable numbers (called grid points) of samples depending on the fluxes that have already been fixed in former iterations. The method is illustrated on three different case studies (with 3, 95 and 1054 fluxes) and shows interesting results in terms of flux distribution convergence and large ranges of the marginal flux distributions. Three consistent criteria are proposed to choose the optimal maximum number of grid points.

Keywords: Metabolic network, Metabolic Flux Analysis, Flux Balance Analysis, Flux Variability Analysis, Most Accurate Fluxes, sampling methods, uniform sampling, hit-and-run methods, underdetermined systems, constrained-based modeling

1. INTRODUCTION

The determination of steady-state values of fluxes in metabolic networks boils down to solving a system of linear algebraic equations subject to linear inequality constraints. That system can be obtained from the metabolite mass balances under the assumption that they do not accumulate inside cells. Additional equations can be provided through measurements of exchange fluxes between cells and culture medium, and inequality constraints often correspond to lower and upper flux bounds. The system of equations is most of the time underdetermined, meaning that there are more unknowns (metabolic fluxes) than equations (mass balances and measured fluxes). Flux Variability Analysis (FVA) (Mahadevan and Schilling, 2003) or Flux Spectrum Approach (FSA) (Llaneras and Picó, 2007) can provide, for each flux, its minimum and maximum values among all the admissible solutions by solving linear programs (LPs). The same information can also be deduced from Elementary Flux Modes or Extreme Pathways (Klamt and Stelling, 2003). To reduce system underdetermination, several methods have been proposed. Flux Balance Analysis (FBA) (Orth et al., 2010a) is based on an objective cost function, made of a linear combination of fluxes, whose optimization represents some optimal metabolic behavior of the cells, *e.g.*, their growth maximization. Other methods aim at extending the set of inequality constraints, either in a systematic way (Nikdel et al., 2018) or based on some biological assumptions, *e.g.*, regarding overflow metabolism (Richelle et al., 2016; Bogaerts et al., 2017). Some methods force the determination of a unique solution, *e.g.,* by using Most Accurate Fluxes (MAF) (Mhallem Gziri and Bogaerts, 2019).

Another solution to cope with system underdetermination is to sample the convex polytope of the admissible solutions. The latter is defined by the intersection of half planes (corresponding to the abovementioned inequality constraints) in a space of reduced dimension that is obtained after eliminating the set of equality constraints (mass balances and measured fluxes). Marginal distributions of each flux can be obtained from that sampling. Hit-and-run algorithms (Smith, 1984) are Markov Chain Monte Carlo (MCMC) methods that sample the solution space via a random walk but have the disadvantage to often get stuck in some regions of the polytope if it has an irregular shape with highly elongated directions. The artificial centering hit-and-run method (ACHR) (Kaufman and Smith, 1998) tries to circumvent that problem and has been later improved, in terms of computational efficiency and more extensive exploration of the polytope, by the optimized general parallel sampler (OPTGP) (Megchelenbrink, 2014). Haraldsdóttir et al. (2017) proposed the coordinated hit-and-run with rounding (CHRR) method that first determines the ellipsoid with largest volume which can be inscribed in the polytope and, secondly, computes the rounding transformation that transforms this ellipsoid into a unit ball. The same rounding transformation is then applied to the convex polytope of flux solutions, hence leading to a much more efficient sampling. CHRR is shown to outperform ACHR and OPTGP in terms of computation efficiency and convergence in two recent studies by Herrmann et al. (2019) and Fallahi et al. (2020).

Bogaerts and Rooman (2019) proposed the DISCOPOLIS algorithm that samples the convex polytope via iterative sequences of LPs to constrain the solutions inside that polytope, taking into account all the previously estimated fluxes. Weights were associated to the samples to ensure the uniformity of their distribution. The drawback of this method is that the weights rapidly tend to zero in irregular shaped, highly constrained, polytopes and that, consequently, only a small subset of samples significantly contributes to the flux distribution.

In this contribution, we propose a new recursive version of the DISCOPOLIS algorithm that, on the one hand, gives the same weight to all the samples and, on the other hand, introduces a new mechanism for approximating a uniform distribution via a recursive determination of the fluxes that takes into account the narrowing of the intervals of admissible solutions due to the fluxes already fixed in former iterations. We compare the new DISCOPOLIS version with the CHRR method and show through case studies that the former is able to uncover larger ranges of admissible flux values.

The paper is organized as follows. Section 2 defines the convex polytope of flux solutions for a metabolic network. Section 3 presents the DISCOPOLIS 2.0 algorithm and highlights the main differences with DISCOPOLIS 1.0. Section 4 illustrates its use, first, on a 3D toy example, secondly, on the core metabolic network of *Escherichia coli* (Orth et al., 2010b) and, finally, on the genome-scale metabolic network of *Pseudomonas putida* (Nogales et al., 2015). Conclusions and perspectives are proposed in Section 5. In Annex, we present an erratum to our previous paper (Bogaerts and Rooman, 2019).

2. THE CONVEX POLYTOPE OF METABOLIC FLUXES

Concatenating the mass balances of the intracellular metabolites that do not accumulate with the measurements of exchange fluxes, the metabolic flux distribution v (made of n fluxes v_i) is constrained by a set of n_e linear equations

$$A_e v = b_e \tag{1}$$

with $v \in \Re^n$, $A_e \in \Re^{n_e \times n}$, $b_e \in \Re^{n_e}$.

The fluxes are also subject to n_i inequality constraints (typically, lower and upper flux bounds) which can be concatenated into

$$Av \leq b \tag{2}$$

with $A \in \Re^{n_i \times n}$ and $b \in \Re^{n_i}$.

Taking into account the equality constraints (1), it is shown in (Bogaerts and Rooman, 2019) how to reduce the problem to the definition of the polytope (2) in a space of reduced dimension with $v \in \Re^{n-n_e}$ and $A \in \Re^{n_i \times (n-n_e)}$.

3. THE DISCOPOLIS 2.0 ALGORITHM

Version 2.0 of the DISCOPOLIS (DIscrete Sampling of COnvex POlytopes via Linear program Iterative Sequences) algorithm is presented in Fig. 1 (main routine) and Fig. 2 (recursive routine). The user chooses the total number of samples (N) and the maximum number of grid points (S^{MAX}). The samples are generated in subsets, each corresponding to a different instance of the while loop in the main routine (Fig. 1, line 6), and this until the total number of samples N is reached. For each subset of samples, S^{MAX} represents the maximum number of values, for a given flux v_i, that are randomly selected. We describe hereunder how the samples in a given subset are obtained.

A first flux index i is randomly selected in $[1,n]$ (Fig. 1, line 8). The recursive routine DISCOPOLIS_recursive_loop is then called at line 12 for randomly selecting S^{MAX} flux values v_i uniformly over $[v_i^{MIN}, v_i^{MAX}]$, these lower and upper bounds having been computed through FVA. For each flux value v_i, a new flux index i^{new} is randomly chosen ($i^{new} \neq i$), and the same routine is recursively called for computing $S^{new} \leq S^{MAX}$ flux values $v_{i,new}$, and so forth until the last flux index is reached, which stops the recursive call to the loop.

The main inputs of the recursive routine (Fig. 2) are the index i of the selected flux, the set I of indexes of the fluxes that have not yet been selected, the number $S \leq S^{MAX}$ of flux values v_i to be randomly drawn (S^{MAX} when called by the main routine), the lower and upper bounds $v_i^{LOW} \geq v_i^{MIN}$ and $v_i^{UP} \leq v_i^{MAX}$ for the flux v_i. These bounds are determined through LPs taking into account the values of all the fluxes previously fixed in the recursive loop via the matrix A_{eq} and the vector b_{eq} (which are empty when called by the main routine).

For each of the S values of the flux indexed by i to be determined in the recursive loop, a value v_i is randomly selected on $[v_i^{LOW}, v_i^{UP}]$ (line 6), except if v_i^{UP} - v_i^{LOW} is smaller than a threshold value ε, in which case it is set as the arithmetic mean of v_i^{UP} and v_i^{LOW} (line 8). If that flux index i is the last to be selected ($I = \varnothing$) (line 10), then a new sample is obtained with specified values for the n fluxes (line 11); we then go over to the next randomly chosen value of v_i in the loop, without a new call to the recursive loop. If the total number of samples N is reached, then the loop is broken (line 13). If that flux i is not the last to be determined in the flux distribution ($I \neq \varnothing$), then its fixed value appears as a new equality constraint in the matrix A_{eq}^{new} and the vector b_{eq}^{new} (line 18), except if the difference v_i^{UP} - v_i^{LOW} is below the threshold ε, in which case the flux v_i is naturally constrained without the need for an additional equality constraint (line 17). Then a new flux index i^{new} is randomly selected in the set I of flux indexes that remain to be set (line 20) and is subsequently withdrawn from that set (line 21). The new lower and upper bounds v_i^{LOWnew} and v_i^{UPnew} of that flux are determined through LPs taking into account all the flux values that have been fixed in the previous iterations (lines 22 and 23). The number S^{new} of fluxes $v_{i,new}$ is computed in line 24. It corresponds to the maximum number of grid points S^{MAX} multiplied by $(v_i^{UPnew}-v_i^{LOWnew})/(v_i^{MAX}-v_i^{MIN})$, which corresponds to the relative decrease of the new flux range due to all the other fluxes that have been previously fixed. S^{new} is of course lower bounded by 1. These S^{new} fluxes $v_{i,new}$ are then randomly generated with a new call to the recursive loop (line 25).

The reduction of the number of fluxes (from S^{MAX} to S^{new}) is proportional to the relative decrease of the flux range and allows approximating a uniform distribution of the samples. By decreasing the number of fluxes that are computed in the narrowed interval v_i^{UPnew} - v_i^{LOWnew}, we compensate for the corresponding increase of probability associated to the fluxes in that new interval.

```
Inputs : solution polytope defined by $A$ and $b$; number of samples $N$;
    maximum number of grid points $S^{MAX}$; minimum and maximum values
    of the fluxes $v_i^{MIN}$ and $v_i^{MAX}$ ($i \in [1,n]$) obtained with Flux Variability
    Analysis
Outputs : $N$ samples $v(k) \in \mathbb{R}^n$  ($k \in [1,N]$)

$I^{tot} = [1,n]$;  /* define set of all indexes $i$ of all the fluxes $v_i \in v$
$L_i = v_i^{MAX} - v_i^{MIN}$;  /* compute for each flux $v_i$ the distance
    between minimum and maximum values
$\varepsilon = 10^{-5}$;  /* set the minimum threshold for $v_i^{UP} - v_i^{LOW}$;
$m = 0$;  /* initialize number of computed samples
while $m < N$ do
    $A_{eq}^{new} = \emptyset$; $b_{eq}^{new} = \emptyset$;  /* initialize empty matrices for equality
    constraints
    Randomly select an index $i$ in $I^{tot}$;
    $I^{new} = I^{tot} \setminus i$;  /* remove index $i$ from set $I$;
    $v_i^{LOWnew} = v_i^{MIN}$; $v_i^{UPnew} = v_i^{MAX}$; $i^{new} = i$; $S^{new} = S^{max}$;
    $flag\_stop = 0$;  /* $flag\_stop = 1$ when $m = N$ and 0 when $m < N$
    DISCOPOLIS_recursive_loop ($S^{new}$, $i^{new}$, $I^{new}$, $v_i^{LOWnew}$, $v_i^{UPnew}$,
        $A_{eq}^{new}$, $b_{eq}^{new}$, $L_i$, $\varepsilon$, $m$)  /* compute $S^{new}$ admissible values of the
        $i^{new}$-th flux $v_{i,new}$
end
```

Fig. 1. DISCOPOLIS 2.0 main routine.

```
Inputs : $S$, $i$, $I$, $v_i^{LOW}$, $v_i^{UP}$, $A_{eq}$, $b_{eq}$, $L_i$, $\varepsilon$, $m$, $N$, $S^{MAX}$

Output : $S$ admissible values of the $i$-th flux $v_i$, $m$, $flag\_stop$

for $s = 1$ to $S$ do
    if $flag\_stop = 1$ then
        break  /* go out of the loop if $m \geq N$
    end
    if $v_i^{UP} - v_i^{LOW} > \varepsilon$ then
        $v_i = (v_i^{LOW} + \varepsilon) + (v_i^{UP} - v_i^{LOW} - 2\varepsilon) * $ rand;
          /* uniform sampling of $v_i$ (rand is a real number uniformly
             distributed in [0,1])
    else
        $v_i = (v_i^{LOW} + v_i^{UP}) / 2$;  /* use of the center of the new
          solution interval if it is smaller than $\varepsilon$
    end
    if $I = \emptyset$ then
        $m = m + 1$;
        if $m \geq N$ then
            $flag\_stop = 1$;
        end
    else
        $A_{eq}^{new} = A_{eq}$ ; $b_{eq}^{new} = b_{eq}$ ;
        if $v_i^{UP} - v_i^{LOW} > \varepsilon$ then
            Extend $A_{eq}^{new}$ and $b_{eq}^{new}$ to account for last fixed $v_i$;
        end
        Randomly select an index $i^{new}$ in $I$;
        $I^{new} = I \setminus i^{new}$ ;  /* remove index $i^{new}$ from set $I$;
        $v_i^{LOWnew} = \min_v v_i$ computed with LP subject to $A*v \leq b$
          and $A_{eq}^{new} * v = b_{eq}^{new}$;
        $v_i^{UPnew} = \max_v v_i$ computed with LP subject to $A*v \leq b$
          and $A_{eq}^{new} * v = b_{eq}^{new}$;
        $S^{new} = \max(1, \mathrm{round}(S^{MAX} * (v_i^{UPnew} - v_i^{LOWnew}) / L_i))$;  /* number
          of grid points remaining in the new constrained solution interval
        DISCOPOLIS_recursive_loop ($S^{new}$, $i^{new}$, $I^{new}$, $v_i^{LOWnew}$, $v_i^{UPnew}$,
          $A_{eq}^{new}$, $b_{eq}^{new}$, $L_i$, $\varepsilon$, $m$)  /* compute $S^{new}$ admissible values of the
          $i^{new}$-th flux $v_{i,new}$
    end
end
```

Fig. 2. DISCOPOLIS 2.0 recursive routine.

Indeed, the uniform sampling on $[v_i^{MIN}, v_i^{MAX}]$ has a probability $(v_i^{MAX} - v_i^{MIN})^{-1}$ while the uniform sampling on the narrower interval $[v_i^{LOWnew}, v_i^{UPnew}]$ has a greater probability $(v_i^{UPnew} - v_i^{LOWnew})^{-1}$. This compensation was obtained by associating weights to the samples in the former version of DISCOPOLIS (Bogaerts and Rooman, 2019).

Note that this algorithm does not correspond to a Markov chain Monte Carlo method. Each iteration of the while loop in the main routine generates a subset of samples issued from the recursive calls to DISCOPOLIS_recursive_loop. The samples in a given subset have necessarily common fluxes. On the contrary, the subsets are completely independent. For a given number of samples N, the number of independent subsets obtained in the while loop decreases when the maximum number of grid points S^{MAX} increases. The limit case $S^{MAX} = 1$ generates N independent samples. However, in that case, the abovementioned mechanism for approximating a uniform distribution is not active anymore.

Here are some significant advantages of DISCOPOLIS version 2.0 with respect to version 1.0:

- All the samples have the same weight and contribute equally to the flux marginal distributions, whereas the fraction of samples with very low weights was rapidly increasing with the number of grid points in version 1.0.
- The choices of the tuning parameters (total number of samples N and maximum number of grid point S^{MAX}) can be *a posteriori* tested as will be illustrated in the case studies.
- The computational efficiency of this recursive version is higher and is increasing with S^{MAX} given that it reduces the number of LPs to be solved. The run time decrease depends of course on the specific case study and on the tuning parameters but a factor of 4 can easily be reached.

4. CASE STUDIES

4.1 Toy Example: 3 fluxes

We consider the 3D polytope defined by fluxes $v^T = [\mathrm{v}_1\ \mathrm{v}_2\ \mathrm{v}_3]$ belonging to the intersection of half-lines (2) with

$$A^T = \begin{bmatrix} -1 & 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0.4 \\ 0 & -1 & 0 & 1.5 & -1.5 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 1.2 \end{bmatrix} \qquad (3)$$

$$b^T = \begin{bmatrix} 0 & 0 & 0 & 2 & 0.5 & 1 & 0.5 & 1 \end{bmatrix}$$

The shape of this convex polytope is shown in Fig. 3. A set of 10^4 samples uniformly distributed in the polytope were obtained with the rejection algorithm (Rubinstein, 1982). This algorithm samples uniformly each flux v_i on the interval $[v_i^{MIN}, v_i^{MAX}]$ corresponding to its lower and upper bounds. If the obtained flux distribution v satisfies the inequality constraints $Av \leq b$, it is kept as an additional sample, otherwise it is rejected. The procedure is repeated until the requested number of samples is obtained. This rejection algorithm has the double advantage to be very simple and to lead to a genuine uniform distribution of the samples. It is however not usable with high dimensional spaces and irregular shaped polytopes as the fraction of rejected samples dramatically increases.

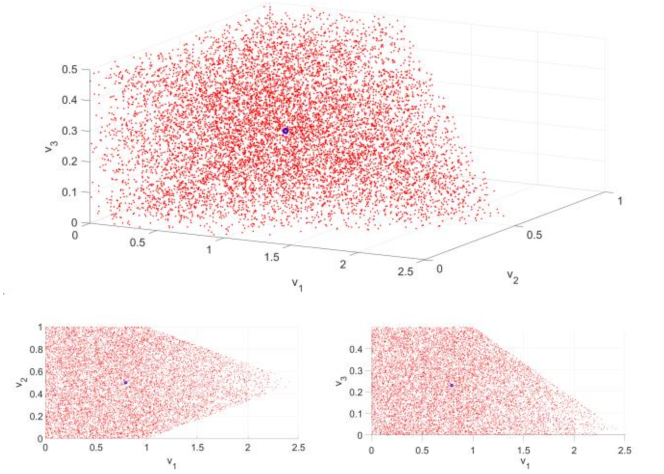

Fig. 3. Toy example: 3 different views of 10^4 samples (red dots) uniformly distributed in the polytope $Av \le b$, defined in (3), obtained with the rejection algorithm. The blue circle corresponds to the mean of the flux distribution $\overline{v}^T = \begin{bmatrix} 0.79 & 0.50 & 0.23 \end{bmatrix}$.

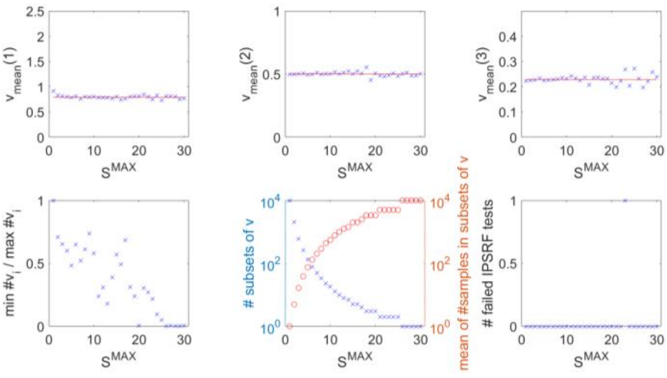

Fig. 4. Toy example: mean values of the 3 fluxes from 10^4 samples obtained with DISCOPOLIS 2.0 using different numbers of grid points S^{MAX} (upper plots, blue crosses, with y scales equal to the flux ranges) compared to the mean of the 10^4 samples obtained with the rejection algorithm (red lines); ratio between the minimum and maximum numbers of random selections among all the fluxes (lower left); number of independent subsets of samples obtained in the while loop (lower central, blue crosses) and their average number of samples (lower central, red circles); number of fluxes for which the IPSRF test fails (lower right).

Fig. 4 shows the results obtained when computing 10^4 samples with DISCOPOLIS 2.0. Except for $S^{MAX} = 1$, *i.e.* the case of 10^4 independent samples without compensation mechanism for keeping uniformity, all the other results for $S^{MAX} \le 10$ have a relative error less than 5%. Higher relative errors (10% and more) appear for $S^{MAX} \ge 15$. In the lower left plot of Fig. 4, we represent the ratio between the minimum and maximum numbers of random selections among all the fluxes (min $\#v_i$ / max $\#v_i$). The random selection of the flux indexes corresponds either to line 8 in the main routine (Fig. 1) or to line 20 in the recursive routine (Fig. 2). Up to $S^{MAX} = 10$, this ratio remains above 0.5, meaning that the flux index that has been the least randomly selected has however not been selected twice less than the most frequently chosen one. Except for 3 cases, all the ratios remain under 0.5 for $S^{MAX} \ge 15$. The lower central plot of Fig. 4 shows that for $S^{MAX} \ge 15$, *i.e.* the threshold above which relative errors of 10% and more appear, there are less than 10 subsets (#subsets of v) that contain an average of 10^3 samples. Given the risk of lack of convergence when the number of independent sets dramatically reduces, the total number (sum over the three fluxes) of failed tests of convergence is shown in the lower right plot of Fig. 4, using the interval-based potential scale reduction factor (IPSRF) test. We use the same test criterion as Fallahi et al. (2020) and Herrmann et al. (2019): the test fails if IPSRF < 0.9 or IPSRF > 1.1. Problems of convergence are only detected for $S^{MAX} = 23$, showing that the test is too optimistic in this low dimensional problem.

We conclude from this toy example that:

- DISCOPOLIS 2.0 recovers the mean of the genuine uniform distribution of samples provided by the rejection algorithm and this with relative errors of less than 5% in case $S^{MAX} \in [2,10]$;

- for values of $S^{MAX} \le 15$ we find that i) #subsets of $v \ge 10$ and ii) min $\#v_i$ / max $\#v_i \ge 0.5$; when these 2 criteria are not satisfied, thus for $S^{MAX} \ge 15$, the mean of the samples exhibits large relative errors (≥ 10%);

- the IPSRF test of convergence appears too optimistic in this low dimensional problem.

4.2 Core Metabolic Network of Escherichia coli*: 95 fluxes (22 in the reduced space)*

This second case study consists of the core metabolic network of *Escherichia coli* (Orth et al., 2010b). The COBRA model (*e_coli_core.mat*) is available in the BiGG Models database (King et al., 2015). It consists of 95 fluxes with upper and lower bounds. As proposed in the supplementary tutorial of Haraldsdóttir et al. (2017), we set the maximum glucose uptake rate to 18.5 mmol/gDW/h and we remove the cellular objective (no FBA). We only consider the aerobic model, with unlimited oxygen uptake. Taking into account the equality constraints from mass balances, the convex polytope is defined by (2) with $A \in \Re^{172\times 23}$. Fig. 5 compares the mean fluxes in the reduced solution space obtained from 10^4 DISCOPOLIS samples with different numbers of grid points S^{MAX} (blue crosses) to the mean fluxes from 10^4 samples obtained with the CHRR algorithm (Haraldsdóttir et al., 2017) and a thinning parameter nSkip set to 5.10^3 (red lines). The DISCOPOLIS mean values obtained vary monotonically with S^{MAX} and tend to a transient plateau for $6 \le S^{MAX} \le 8$. The increase of S^{MAX} before reaching that plateau helps tending to a uniform distribution as explained in Section 3. Fig. 6 shows that the three criteria mentioned in section 4.1 remain satisfied until S^{MAX} reaches values within that plateau: min $\#v_i$ / max $\#v_i \ge 0.5$ for $S^{MAX} \le 8$, #subsets of $v \ge 10$ for $S^{MAX} \le 7$ and none of the IPSRF convergence tests fail for $S^{MAX} \le 6$. For $S^{MAX} \ge 9$, significant variations appear for some mean fluxes and the three criteria are all far from being satisfied. The comparison of the mean fluxes between the two algorithms shows that results are of the same magnitude, although some of them differ significantly, *e.g.*, v_1, v_9, v_{15}, v_{18}.

Selecting $S^{MAX} = 6$, *i.e.* the most conservative choice for which the three criteria are satisfied, we compare in Fig. 7 the marginal probability density functions of each flux obtained

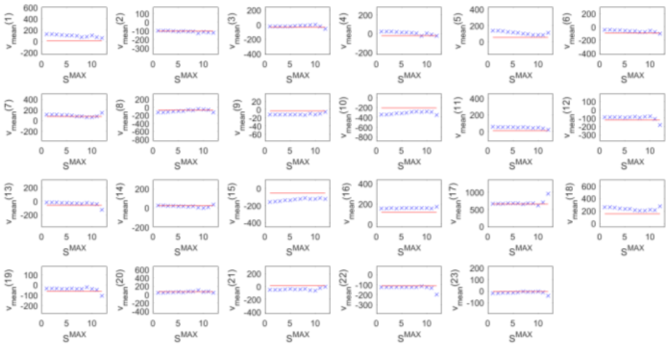

Fig. 5. *E. coli* case study: mean of the 23 fluxes in the reduced solution space from 10^4 DISCOPOLIS samples with different numbers of grid points S^{MAX} (blue crosses, with y scales equal to the flux ranges) compared to the mean fluxes from 10^4 CHRR samples with a thinning parameter nSkip set to 5.10^3 (red lines).

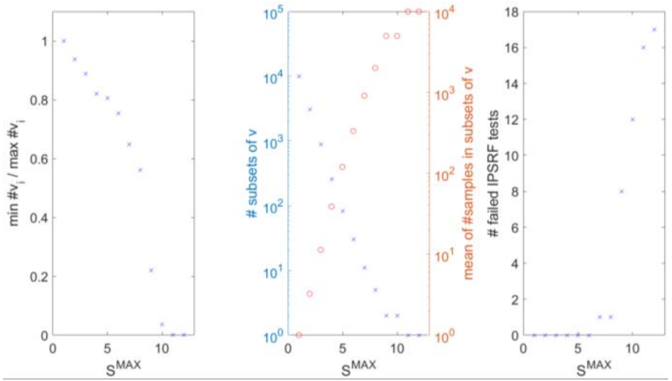

Fig. 6. *E. coli* case study: 10^4 DISCOPOLIS samples: ratio between the minimum and maximum numbers of random selections among all the fluxes (left); number of independent subsets of samples obtained in the while loop (central, blue crosses) and their average number of samples (central, red circles); number of fluxes for which the IPSRF test fails (right).

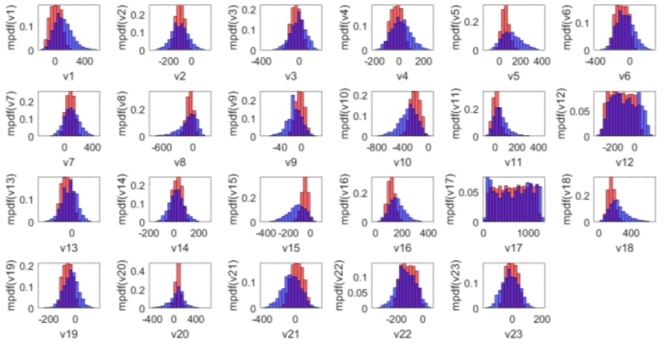

Fig. 7. *E. coli* case study: marginal probability density functions of the fluxes in the reduced solution space from 10^4 DISCOPOLIS samples with S^{MAX} = 6 (blue) and from 10^4 CHRR samples with a thinning parameter nSkip set to 5.10^3 (red).

with DISCOPOLIS (blue histograms) to the ones obtained with CHRR (red histograms). Besides the comparison of the mean values, we can observe here the overall tendency of the DISCOPOLIS algorithm to explore larger ranges of flux values. This can be explained by the total independency of the subsets of *v* obtained in the main routine (Fig. 1), which allows exploring completely different regions of the polytope. This tendency increases when S^{MAX} decreases given that, for the limit case S^{MAX} = 1, all the samples are independent. However, in the latter case, the distribution is not uniform.

The equivalent of Fig. 6 in the case of only 10^3 DISCOPOLIS samples leads to a choice of S^{MAX} = 6 (results not shown). The comparison of the marginal probability density functions shows very similar results with N = 10^3 and N = 10^4, hence proving the convergence of the distributions and the fact that they are already reached for N = 10^3 (results not shown).

Finally, the computational time for 10^4 samples (using Matlab R2020b, IBM ILOG CPLEX Studio 12.10 for solving LPs with *cplexlp*, Intel Core i7 at 2.7 GHz with 16 GoRAM) is quite similar: 11.9 min for CHRR and 9.2 min for DISCOPOLIS 2.0, whereas the DISCOPOLIS 1.0 version takes 32.5 min.

We conclude from this case study that:

- the three convergence criteria (based on min #v_i / max #v_i, #subsets of *v* and IPSRF convergence test) are in good agreement and help choosing S^{MAX};

- convergence is guaranteed with both N = 10^3 and N = 10^4;

- DISCOPOLIS explores larger ranges of fluxes than CHRR;

- the computational time is similar with both methods.

4.3 Genome-scale Metabolic Network of Pseudomonas putida*: 1054 fluxes (122 in the reduced space)*

This third case study consists of the genome-scale metabolic network of *Pseudomonas putida* (Nogales et al., 2008). The COBRA model (*iJN746.mat*) is available in the BiGG Models database (King et al., 2015). It consists of 1054 fluxes with upper and lower bounds. As in the previous case study, we remove the cellular objective (no FBA). Taking into account the equality constraints from mass balances, the convex polytope is defined by (2) with $A \in \Re^{1304 \times 122}$. Fig. 8 shows the same plots as in Fig. 6 for this new case study. The three convergence criteria are once again in agreement and lead to the optimal value S^{MAX} = 4, which is the highest value such that min #v_i / max #$v_i \geq 0.5$, #subsets of $v \geq 10$ and no failed IPSRF tests. The equivalent of Fig. 8 in the case of only 10^3 DISCOPOLIS samples leads to a choice of S^{MAX} = 3 (results not shown) and, as in the previous case study, the marginal distributions of the fluxes are similar. Their comparison with the marginal distributions obtained from 10^4 CHRR samples (results not shown) confirms the tendency of the DISCOPOLIS algorithm to explore larger ranges of flux values. The thinning parameter nSkip was set to $1.2.10^5$ to follow the rule of thumb proposed by Haraldsdóttir et al. (2017), *i.e.* nSkip = 8*(dim)^2 where dim is the dimension of the reduced space (dim = 122). Finally, Table 1 compares the percentages of fluxes that fail the IPSRF test and the run times for 10^3 and 10^4 samples with the CHRR and DISCOPOLIS algorithms. The DISCOPOLIS algorithm performs better than CHRR in terms of convergence of the samples distribution but is about 4 times slower than CHRR in this case of a large network.

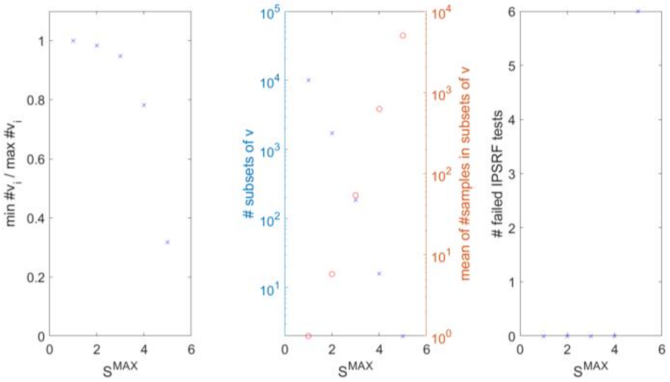

Fig. 8. *Pseudomonas putida* case study: same legend as Fig. 6.

	DISCOPOLIS $N = 10^3$ $S^{MAX} = 3$	DISCOPOLIS $N = 10^4$ $S^{MAX} = 4$	CHRR $N = 10^3$	CHRR $N = 10^4$
% failed IPSRF tests	0 (13.7)	0 (8.9)	85.3 (44.9)	53.3 (30.5)
Run time [h]	3.1	27.3	0.7	7.3

Table 1. *Pseudomonas putida* case study: % of fluxes in the reduced dimension space (dim = 122) that fail the IPSRF test (values in parentheses for the full dimension space, dim = 1054) and run times.

5. CONCLUSIONS AND PERSPECTIVES

Version 2.0 of DISCOPOLIS uses a recursive loop for computing variable numbers (called grid points) of fluxes in order to take into account the fluxes already fixed in former iterations, and gives the same weight to all the samples. This mechanism allows approximating a uniform distribution, as illustrated on a 3D toy example. Two other case studies (95 metabolic fluxes for *E. coli* and 1054 fluxes for *Pseudomonas putida*) show that three different but consistent criteria can be used for choosing the optimal maximum number of grid points S^{MAX}: it should be the highest value which still provides convergence of the sample distribution. In comparison with the CHRR method, run times are similar with the 95 fluxes network but DISCOPOLIS runs 4 times slower than CHRR with the 1054 fluxes network. However, the DISCOPOLIS algorithm explores larger ranges of flux values and is able to satisfy IPSRF convergence tests for all fluxes in the reduced dimension space, even for a large network (1054 fluxes) analyzed with a relatively small number of samples (10^3).

Future work will include further optimizing the algorithm in terms of computational efficiency, testing its use with other complex networks, providing guidelines for the *a priori* choice of the tuning parameters N and S^{MAX}, and further analyzing how the samples extensively explore the convex polytope.

ANNEX: ERRATUM TO Bogaerts and Rooman (2019)

It has been erroneously stated in our previous paper that all the samples computed via the CHRR method belong to the ellipsoid with largest volume that can be inscribed in the polytope, hence implying that "any subsequent hit-and-run sequence of samples provided by the CHRR algorithm necessarily has its mean positioned at the center of this ellipsoid" (and other equivalent claims). This is wrong given that the transformation from the ellipsoid to the unit ball is applied to the convex polytope of solutions, and thus preserves all the admissible solutions.

REFERENCES

Bogaerts, Ph., Mhallem Gziri, K., and Richelle, A. (2017). From MFA to FBA: Defining linear constraints accounting for overflow metabolism in a macroscopic FBA-based dynamical model of cell cultures in bioreactor. *J. Process Control*, 60, 34-47.

Bogaerts, Ph., and Rooman, M. (2019). DISCOPOLIS: An algorithm for uniform sampling of metabolic flux distributions via iterative sequences of linear programs. *IFAC-PapersOnLine*, 52 (26), 269-274.

Fallahi, S., Skaug, H., and Alendal, G. (2020). A comparison of Monte Carlo sampling methods for metabolic network models. *PLoS ONE*, 15 (7), e0235393.

Haraldsdóttir, H., Cousins, B., Thiele, I., Fleming, R., Vempala, S. (2017). CHRR: coordinate hit-and-run with rounding for uniform sampling of constraint-based models. *Bioinformatics*, 33 (11), 1741-1743.

Herrmann, H., Dyson, B., Vass, L., Johnson, G., and Schwartz, J.-M. (2019). Flux sampling is a powerful tool to study metabolism under changing environmental conditions. *NPJ Syst. Biol. Appl.*, 5 (1), 32

Kaufman, D., and Smith, R. (1998). Direction choice for accelerated convergence in hit-and-run sampling. *Oper. Res.*, 46 (1), 84-95.

King, Z., Lu, J, Dräger, A, Miller, P, Federowicz, S, Lerman J., Ebrahim, A, Palsson, B., and Lewis, N. (2015). BiGG Models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Res.*, 44(D1), D515–D522.

Klamt, S., and Stelling, J. (2003). Two approaches for metabolic pathway analysis? *Trends in Biotech.*, 21 (2), 64-69

Llaneras, F., and Picó, J. (2007). An interval approach for dealing with flux distributions and elementary modes activity patterns. *J. of Theor. Biol.*, 246, 290-308.

Mahadevan, R., and Schilling, C. (2003). The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metab. Eng.*, 5, 264-276.

Megchelenbrink, W., Huynen, M., and Marchiori, E. (2014). optGpSampler: an improved tool for uniformly sampling the solution-space of genome-scale metabolic networks. *PLoS ONE*, 9, e86587.

Mhallem Gziri, K., and Bogaerts, Ph. (2019). Determining a unique solution to underdetermined metabolic networks via a systematic path through the Most Accurate Fluxes. *IFAC-PapersOnLine*, 52 (1), 352-357.

Nikdel, A., Braatz, R., and Budman, H. (2018). A systematic approach for finding the objective function and active constraints for dynamic flux balance analysis. *Bioproc. Biosyst. Eng.*, 41, 641-655.

Nogales, J., Palsson, B., and Thiele, I. (2008). A genome-scale metabolic reconstruction of *Pseudomonas putida* KT2440: iJN746 as a cell factory. *BMC Syst. Biol.*, 2: 79.

Orth, J., Thiele, I., and Palsson, B. (2010a). What is flux balance analysis? *Nature Biotechnol.*, 28 (3), 245-248.

Orth, J., Fleming, R., and Palsson, B. (2010b). Reconstruction and use of microbial metabolic networks: the core *Escherichia coli* metabolic model as an educational guide. *EcoSal Plus*, 1(10).

Richelle, A., Mhallem Gziri, K., and Bogaerts, Ph. (2016). A methodology for building a macroscopic FBA-based dynamical simulator of cell cultures through flux variability analysis. *Biochem. Eng. J.*, 114, 50-61.

Rubinstein, R. (1982). Generating random vectors uniformly distributed inside and on the surface of different regions. *Eur. J. Oper. Res.*, 10 (2), 205-209.

Smith, R. (1984). Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions. *Oper. Res.*, 32 (6), 1296-1308.