MARKOV CHAIN MONTE CARLO METHODS TO ANALYZE THE STEADY-STATE FLUX SOLUTION SPACE OF METABOLIC NETWORK MODELS

Thomas Latzko, Wolfgang Wiechert and Katharina Nöh  
IBG-1: Biotechnology & JARA-HPC  
Forschungszentrum Jülich GmbH  
D-52425 Jülich  
Germany  
E-Mail: k.noe@fz-juelich.de

Uwe Jaekel  
Department of Mathematics and Technology  
University of Applied Sciences Koblenz  
D-53424 Remagen  
Germany

KEYWORDS  
Monte Carlo, Markov Chain Monte Carlo, Volume Estimation, Steady-State Flux Solution Space, Polytope, HPC

ABSTRACT

The steady-state flux solution space of a metabolic network model represents a convex polytope. Determining the size and shape of the flux solution space can give valuable information for a better understanding of the fluxes operating metabolic processes within living cells. For realistic problems, the polytope is usually high dimensional. Computing its volume analytically is proven to be \#P-hard (Dyer and Frieze 1998). Therefore, Monte Carlo methods are used to approximate the volume numerically. To cope with high-dimensional problems (> 20 dimensions), naive Monte Carlo approaches, however, generally fail. We propose a new Markov Chain Monte Carlo method in combination with a Hit and Run algorithm that is able to cope with the volume estimation problem in high dimensions. Based on quantitative measures we analyze the performance of our algorithm. By application to an example metabolic network model of realistic size we could demonstrate reasonable running times even without taking parallelization into account. The results show that Markov Chain methods are capable to analyze the size and shape of flux solution spaces for large-scale network models. Thus, flux space volume estimation has the potential to become a new member of the computational toolset for constraint based modeling.

INTRODUCTION

In Metabolic Engineering and Systems Biology, the characterization of non-measurable metabolic fluxes is vitally important. Constraint-based modeling aims to describe metabolic processes under steady-state conditions by a set of mass balances equations (Palsson 2006). These mass balances define the flux solution space which is bounded by additionally inequality constraints. For realistic models the number of equations is usually not sufficient to determine a unique flux solution. In the last decades, numerous methods and computational tools have been developed that seek for single flux solutions under additional assumptions, see (Llaneras and Pico 2008) for a review. Complementary to these approaches, we aim to explore the complete steady-state flux solution space by means of its shape and size. In this context, the volume of the flux solution space can be interpreted as an indicator for the fidelity of a particular flux solution. Moreover, the volume characterizes the flexibility of the cellular behavior.

Mathematically speaking, the steady-state flux solution space of metabolic networks is a linear polytope $P$ (Schuster et al. 1999). For large-scale metabolic network, corresponding polytopes are typically high-dimensional having $\sim 20$ dimensions upwards. Exploration of polytopes by a naive Monte Carlo sampling approach is, however, only applicable to small scaled problems up to 15 dimensions (Wiback et al. 2004). In high dimensions, advanced methods for sampling are required in order to deduce sound results. Because Monte Carlo methods are well-known to be computational intensive, special focus has to be laid on an efficient and parallelizable implementation of the sampling approaches as this is the key to success for high-dimensional problems.

In this contribution, we propose three sampling-based Monte Carlo methods and analyze their sampling performance for high-dimensional polytopes:

1. a Hit-and-Run algorithm for sampling the inner domain of a polytope;
2. the Fok-Crevier (FC) volume estimator;
3. a new, specially tailored Markov Chain Monte Carlo (MCMC) volume estimator.

These algorithms are compared to a naive Monte Carlo method with respect to convergence, sampling efficiency
and running time.

Based on these methods, we investigate the volume estimation problem which is known to be of $\#P$-hard complexity (Dyer and Frieze 1998). In order to cope with the “curse of dimensionality”, we apply an advanced method suggested in (Jaekel 2011). The scalability of the volume estimation algorithm is demonstrated with an example metabolic network model at hand. Exemplary, advanced insights into the distribution of flux vectors and flux correlations are derived. Finally, we summarize the capability of sampling-based Monte Carlo methods to produce fast and accurate results in high dimensions enabling the use in constraint-based modeling of large-scale metabolic network models.

**MATHEMATICAL FORMULATION**

**Steady-State Flux Solution Space**

Given a metabolic network model consisting of $m$ metabolites and $n$ reactions, the corresponding steady-state flux solution space is the set of all flux distributions $v$ that satisfy the so-called steady-state assumption

$$ S \cdot v = 0. \quad (1) $$

Here $S \in \mathbb{R}^{m \times n}$ is the stoichiometric matrix whose elements, the stoichiometric coefficients $s_{ij}$, relate the proportions of the $i$th metabolite processed by the $j$th reaction. In typical network models the number of metabolites is far less than the number of reactions, i.e. $m << n$, leading to underdetermined systems (1). Without loss of generality we assume that $S$ has full row rank.

The set of all valid flux distributions fulfilling (1) defines the $n \times d$-dimensional null space $N$ of the matrix $S$ consisting of (non-unique) orthonormal basis vectors $N_i$ (i.e., columns of $N$). Here, $d$ can be interpreted as the metabolic network’s degree of freedom that defines the dimension of the resulting flux solution polytope. The null space $N$ of $S$ contains important information about the network models’ capabilities. For instance, the correlation coefficient $\phi_{ij}$ between the fluxes $v_i$ and $v_j$ of two reactions can be readily deduced (Poolman et al. 2007)

$$ \phi_{ij} = \frac{N_i \cdot N_j^T}{\sqrt{N_i \cdot N_i} \sqrt{N_j \cdot N_j}}. \quad (2) $$

One numerical option to compute the null space matrix $N$ is the singular value decomposition (SVD):

$$ S = U \cdot \Sigma \cdot V^T, \quad (3) $$

where $U \in \mathbb{R}^{n \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices and $\Sigma$ is a $m \times n$ rectangular diagonal matrix with the non-negative eigenvalues (Golub and Loan 1996). The null space matrix $N$ is spanned by the last $d$ columns of $V$.

In addition to the stoichiometry, usually the flux solution space is bounded by known or at least plausible constraints

$$ v_{\text{min}} \leq v \leq v_{\text{max}}. \quad (4) $$

With these constraints, the solution space of (1) can be described as follows

$$ \mathcal{P} = \left\{ x \in \mathbb{R}^d \mid \begin{bmatrix} N & -N \end{bmatrix} \cdot x \leq \begin{bmatrix} v_{\text{max}} \\ -v_{\text{min}} \end{bmatrix} \right\}. \quad (5) $$

**METHODS**

**Naive Monte Carlo Random Sampling**

A straightforward approach to determine the volume $\text{vol}(\mathcal{P})$ of the flux polytope $\mathcal{P}$ in (5) is to relate it to a known reference volume. The reference volume $\mathcal{R}$, usually a hyper-rectangle, is chosen as an outer bounding box of $\mathcal{P}$. Within $\mathcal{R}$, uniformly distributed random sampling points can be easily generated. A naive MC method determines the volume of $\mathcal{P}$ by simple rejection sampling: Whenever a sample drawn is found in the polytope it is marked as “accepted”, otherwise it is “rejected”. Clearly, the sequence of accepted points is uniformly distributed in $\mathcal{P}$. For the number of total samples $K$ and $M$ the number of “accepted” ones that hit the polytope, an estimator for the volume of $\mathcal{P}$ is derived according to (Binder 1979)

$$ \text{vol}(\mathcal{P}) \approx \text{vol}(\mathcal{R}) \cdot \frac{M}{K}. \quad (6) $$

Despite of its simplicity, the main limitation of the method is that usually the relative volume between the polytope $\mathcal{P}$ and the reference volume $\mathcal{R}$ decreases rapidly with increasing dimensionality $d$ of $\mathcal{P}$. This results in progressively low probabilities to generate “rejected” points in the inner domain of $\mathcal{P}$.

**Hit-and-Run Sampling**

A more efficient approach that bypasses “rejected” samples while allowing to draw asymptotically uniformly distributed samples in $\mathcal{P}$, is the well-known Hit-and-Run (HR) algorithm (Smith 1984). Starting with a feasible initial point $x_0 \in \mathcal{P}$, a direction vector $e$ is drawn from a uniform distribution over a $d$-dimensional hyper-sphere. Moving along the vector $e$ in both directions till constraints are violated gives two distances $a$ in negative and $b$ in positive direction. In turn, a random number, $\lambda$, that is uniformly distributed between $a$ and $b$ is used to construct a new iterate $x_{k+1}$

$$ x_{k+1} = x_k + \lambda e, \quad k = 0, 1, \ldots K. \quad (7) $$
The construction rule (7) results in a Markov Chain that converges towards the uniform distribution with increasing sample size $K$. The HR algorithm converges in polynomial time to a uniform distribution (Lovsz 1999), i.e., HR mixes fast.

**Fok-Crevier Volume Estimator**

While HR and the naive Monte Carlo method sample the inner domain of the convex polytope $\mathcal{P}$ and compare them with a reference volume, an alternative approach for volume estimation relies on surface-sampling of $\mathcal{P}$. Here, instead of integrating over the whole domain $\Omega(\mathcal{P})$, integration over the corresponding surface $\partial\Omega$ reduces the dimensionality of the problem by one from $d$ to $d-1$. Next, we outline the main ideas to derive calculation rules for volume estimation starting from the Gaussian divergence theorem:

$$V = \int_\Omega 1 \cdot d^d x = \int_\Omega \nabla \cdot F(x) d^d x$$

$$= \int_{\partial\Omega} F(x) \cdot n \cdot d^{d-1} x,$$  

where $F$ is a divergence free vector field and $n$ is the normal vector of the surface $\partial\Omega$ in outer direction. Assuming that the origin of the coordinate system is located inside the polytope $\mathcal{P}$, the volume and the vector field is set to $F(x) = x/d$. Each point $x$ lying on the surface of $\mathcal{P}$ can be written as $x = c \cdot \lambda(x)$, where $\lambda(x)$ is the distance from the origin in the direction of $c$. Using these properties, the integrand for the volume computation can be simplified

$$V = \int_{\partial\Omega} F \cdot n \cdot e \cdot n \cdot \lambda^{d-1}(x) \ dx = \int_{\partial\Omega} \lambda^d(x) \ d\omega.$$

Solving the integral (10) by means of Monte Carlo integration (Jaekel 2011) leads to the estimator of Fok and Crevier (FC)

$$\hat{V}_K = \frac{A_{\mathcal{B}_d}}{d} \sum_{k} \lambda_k^d,$$  

where $A_{\mathcal{B}_d}$ is the surface of the $d$-dimensional unit sphere and $K$ is the number of samples (Fok and Crevier 1989).

One iteration of the sampler is performed by drawing a direction $e$ uniformly distributed on the boundary of the unit sphere, computing the distance $\lambda$ to the border of the polytope and summing up the integrand $\lambda^d$ in (11). The term $\lambda^{d-1}(x)/n \cdot e$ in (10) represents the projection of the unit sphere to the boundary of $\mathcal{P}$. Notably, for high dimensions $d$, integrands tend to be sharply peaked functions. Hence, only few high values for $\lambda$ will effectively contribute to the volume estimator while the contribution of the predominant fraction of sampled values is actually negligible. This gives reason for limited convergence properties of the FC volume estimator.

**Markov Chain Monte Carlo Volume Estimation**

Markov Chain Monte Carlo (MCMC) methods, like the prominent Metropolis-Hastings algorithm (Hastings 1970), are versatile approaches to estimate the volume $V$ given by (10). The characteristics of MCMC can be used to increase the convergence rate of the FC estimator (11) by preferably sampling in peaked regions, i.e., close to the vertices of $\mathcal{P}$. Markov Chain methods require probability densities to perform a non-uniformly distributed importance sampling. As the distance $\lambda(x)$ introduced before is positive by construction, the integrand in (10) can be interpreted as a probability density normalized by the volume $V$

$$f(x) = \frac{\lambda^d(x)}{V \cdot d}.$$  

In order to derive a MCMC-based volume estimator, a family of probability density functions $p_\beta$ is introduced

$$p_\beta = \frac{1}{Z(\beta)} f^\beta(x) = \frac{1}{Z(\beta)} \exp(\beta \ln f(x))$$

with normalizing constants $Z(\beta), \beta \in [0,1]$. For $\beta = 0$ the resulting directions are uniformly distributed on the unit sphere $\mathcal{B}_d$ meaning that the probability for each direction is the same. In case of $\beta = 1$ samples are preferably located in the neighborhood of the vertices of the polytope $\mathcal{P}$ by giving $\epsilon$ a higher probability to point in direction of a vertex. Although for further reading only probability functions $p_0$ and $p_1$ are required (see also below), for completeness the normalizing constants $Z(\beta)$ are given for the general case

$$Z(\beta) = \int_{\partial\mathcal{B}_d} \exp(\beta \ln(f(\omega))) d\omega,$$

$$Z(0) = A_{\mathcal{B}_d} \frac{2\pi^{d/2}}{\Gamma(d/2)},$$

$$Z(1) = V,$$  

where $Z(0)$ is the surface of the $d$-dimensional unit hyper sphere. In turn, $\beta = 0$ and $\beta = 1$ are used to rewrite the volume $V$ by an expectation value $E_{p_\beta}$ with respect to the density $p_\beta$

$$\frac{\ln Z(1)}{Z(0)} = \int_0^1 \frac{d}{d\beta} \ln Z(\beta) \ d\beta = \int_0^1 \int_{\partial\mathcal{B}_d} \ln f(x(\omega)) p_\beta(x(\omega)) d\omega \ d\beta = \int_0^1 E_{p_\beta} \ln f \ d\beta.$$  

This reformulation is called thermodynamic integration or path sampling (Gelman and Meng 1998).

In this contribution we apply the Metropolis-Hastings (MH) algorithm to draw samples from the probability
densities $p_0$ and $p_1$. As the MH algorithm relies on ratios of these density functions, the normalizing constants $Z(\beta)$ are not required. In order to solve (15), numerical MC integration is applied as before to approximate the volume. Then, utilizing MH to draw samples $x_k$ from the densities $p_β$ gives a MCMC estimator $\hat{E}_K$ for the expectation value

$$\lim_{K \to \infty} \hat{E}_K(\beta) = \lim_{K \to \infty} \frac{1}{K} \sum_k \beta \ln f(x_k) = E_{p_\beta}[\ln f].$$

(16)

depending on $\beta$ and the number of samples $K$ (Jaekel 2011).

Having an estimator for the expectation value $E_{p_\beta}$ of the integrand in (15) under the probability $p_β$ at hand, the corresponding estimator for the volume of the polytope $\mathcal{P}$ can be found using the composite trapezoidal rule

$$\hat{V} = Z(0) \exp \left( \Delta \beta \left( \frac{\hat{E}_K(0) + \hat{E}_K(1)}{2} + \sum_{j=1}^{n-1} \hat{E}_K(j \Delta \beta) \right) \right)$$

(17)

with equally-sized sub-intervals $\Delta \beta = 1/b$ for a given number $b$.

**Tuning the MCMC Estimator**

For sharply peaked probability density functions $p_β$ a Markov chain tends to get stuck in local modes. To circumvent this problem, parallel tempering techniques have been developed (Gregory 2010). Here, the basic idea is to run more than one chain at different, “inverse temperatures” $0 = \beta_1 < \beta_2 < \ldots < \beta_n = 1$ (cf. Fig. 1): a sequence with a low value of $\beta$ explores the whole parameter space while a chain running with a high value of $\beta$ tends to be located close to the peaks.

Finally, to improve the mixing at high temperatures of $\beta$, a swapping rule is introduced which allows an exchange of the states $x_k^{(i)}$ among different temperatures $\beta_k$. The two states $x_k^{(i)}$ and $x_k^{(j)}$ will be swapped with the probability of

$$\min \left\{ 1, \exp \left( (\beta_j - \beta_i) \cdot \left( g \left( x_k^{(i)} \right) - g \left( x_k^{(j)} \right) \right) \right) \right\}.$$

(18)

Basically, this rule enables the Markov Chain to “jump” from one peak to another and therewith permits a faster exploration of all peaks at high $\beta$ values. For the volume estimator (17), $g = \ln f$ is a reasonable choice.

Usually, parallel tempering approaches solely use the density function $p_1$ (Gregory 2010). Instead, we use all temperatures for the numerical integration of the volume estimator (17). In this way, the complete set of computed samples can be used to estimate the volume.

**RESULTS**

Having introduced various Monte Carlo algorithms, these methods are to be accounted quantitatively in terms of their mixing rates and computational costs when applied to high-dimensional convex polytopes. Convergence and running time behaviors are analyzed in the context of the volume estimation problem. For the HR sampler the reader is referred to (Lovsz 1999) for an excellent overview. Finally, performance of the FC and the MCMC estimator are compared with a metabolic network of realistic size. First, some comments on the implementation of the introduced Monte Carlo methods are made.

**Monte Carlo-based Volume Estimators – Implementation**

Sampling of high dimensional polytopes is computationally extraordinary expensive. Thus, the effective implementation of sampling-based methods is the key to their successful applicability to real world problems. We implemented the introduced algorithms HR, FC and MCMC in MATLAB (MATLAB R2011a, www.mathworks.com) and C++ (GCC 4.6.3, gcc.gnu.org). Running time by means of speedup for the HR method and the FC estimator are shown in Fig. 2 for both implementations. Notably, for large-scale problems we found the C++ code to be at least one order of magnitude faster than the MATLAB code when using a single thread.

![Figure 1: Probability densities $p_β$ for different inverse temperatures $\beta$ shown as function of the search direction angle. At low values of $\beta$, samples are uniformly distributed on the unit sphere whereas at high temperatures $\beta$ distributions become sharply peaked. This shows that it is unlikely to draw samples in all peaked regions without parallel tempering.](image-url)
Figure 2: Speedup comparison of C++ and MATLAB implementations for FC and MCMC. At low dimensions of (19), the C++ code is up to 40 times faster than the MATLAB code. At high dimensions the speedup ratio drops to a value of about 10. The optimized matrix-vector-multiplication in MATLAB can be given as reason for this behavior.

Next, we analyzed both implementations concerning their data handling efficiency. Again, the C++ implementation clearly outperforms the MATLAB code, in particular in the sampling part. While the MATLAB-based sampler is limited by $O$(million samples), the C++ sampler can easily cope with billions of samples per computational task. As major reason for the sub-optimal performance of the MATLAB code, the Java virtual machine and the memory management of MATLAB itself have been pinpointed.

Of course, MC sampling methods are by construction perfectly suited for a parallel implementation, i.e., by running independent instances of the algorithms in parallel. Here, linear speedup can be expected. This aspect, however, is not further investigated in this contribution.

Volume Estimation Convergence Measurement

To compare the HR, FC and MCMC methods quantitatively, a $d$-dimensional standard simplex given by

$$x_i \geq 0, \ i = 1, \ldots, d \quad \text{and} \quad \sum_{i=1}^{d} x_i \leq 1 \quad (19)$$

is used as test-polytope. The volume of (19) can be derived analytically

$$V = 1/d!.$$  \hspace{1cm} (20)

We define the accuracy of a Monte Carlo method as the number of Monte Carlo samples that have to be generated in order to produce an overall standard deviation of below $P\%$ deviation from the exact volume of the test-polytope. Here, the standard deviation is a more expressive accuracy measure as, e.g., the mean value, because it measures the overall convergence rate of a sampling method. To assess the accuracy of our volume estimators, the standard deviations are used as an indicator for the number of samples that are required to fall below a given precision limit given by $P\%$. For each Monte Carlo method 96 simulations were performed to estimate their standard deviation around the real volume in the test cases. The MCMC algorithm uses 32 parallel chains on different temperatures. The results shown in Figure 3 demonstrate that the MCMC estimator converges in polynomial time while for the FC and the naive Monte Carlo estimator computation time grows exponentially with the dimensions of the problem (19), although FC and the MCMC volume estimators converge asymptotically to the real volume of the polytope. In case of a test-polytope of 70 dimensions, the computation of the 96 simulations takes 5.9 h on two Intel Xeons with 12 cores to reach 1% standard deviation of the real volume. Therefore $K = 26$ million iterations per simulation have to be generated. To obtain 5% accuracy requires $K = 1.02$ million iterations leading to a computational time of only 13 min. In practice we found that good results can be generated in reasonable time even for high-dimensional problems. Hence, running an exploratory short simulation usually provides a good impression on the volume of the test problem.

![Figure 3: Accuracy comparison of the naive Monte Carlo, FC and MCMC method: number of iterations required to obtain a standard deviation of 5%. To demonstrate the scalability of our method, results for a MCMC estimator for a standard deviation of 1% are shown, too.](image_url)
Application Example: *E. coli* Metabolic Network Model

A medium-sized *Escherichia coli* network is chosen to demonstrate the applicability of the methods introduced above. The *E. coli* network consists of 92 reactions and 76 metabolites including the biomass reaction. The corresponding polytope has a dimension of $d = 24$. Fluxes are constrained by $v_{\text{min}} = -10 \leq v \leq 10 = v_{\text{max}}$. Moreover, some specific fluxes are not allowed to be reversible, i.e. $v \geq 0$, due to biological and thermodynamical reasons.

The estimated volume of the polytope is $1.085^{22} \pm 1.0846^{20}$ which is low compared to its corresponding surrounding box ($5.52^{41}$). Due to this imbalance, the naive Monte Carlo method cannot be applied. The FC estimator exhibits slow convergence properties: after 100 million iterations, the estimated value of $9.23^{21} \pm 5.64^{21}$ has a high standard deviation which is about 60% of the real value. In contrast, the MCMC volume estimator converges towards $1.085^{22} \pm 1.0846^{20}$, i.e. the standard deviation is $\leq 1\%$ of the mean value. In this case, the termination criterion is reached after about 638,000 iterations for each simulation.

The HR algorithm is used to estimate the flux distribution by drawing 1 million samples inside the polytope. This can be done in less than 3 seconds on a single computer (Intel core2 Q6600). Examples for sampled flux distributions are shown in Fig. 4. Here, two of the reactions are chosen to have a closer look: (i) the biomass equation (Fig. 4a) which ties several metabolic reaction pathways together and (ii) the pyruvate kinase (Fig. 4b) reaction which is located at a central position of the network. Hence, both reactions are expected to have an impact on the flux solution space.

The biomass equation is constrained by the polytope at a maximum value of 0.42 (computed with linear programming). Close to the maximum of the flux for the biomass reaction, the polytope is sharply peaked. Increasing $v_{\text{min}}$ of the biomass reaction from 0 to 0.84 (20% of the maximal value) leads to a decreased volume of the polytope to only 1.5% of the original volume. Instead, changing the constraints $v_{\text{min}}$ and $v_{\text{max}}$ for the pyruvate kinase reaction does not have a likewise significant impact. Finally, we then compare the correlation of the constrained and the non-constrained system (2). While for the unconstrained system the histidine reaction is highly correlated (-0.5) with the biomass equation, both reactions are not correlated (-0.02) in the constrained system.

Although the given results are exemplary, they demonstrate the potential of sampling based exploration of the flux solution spaces and give impressions on the interpretation capabilities. Such methods are particularly useful for large-scale networks because due to the network’s high degree of connectivity, the impact of constraints and the effect of changes are not intuitive and hardly foreseeable.

**RELATED WORK**

Compared to the Monte Carlo methods used by (Wiback et al. 2004), which are highly limited by the dimension of the problem, both the Hit-and-Run sampler and the MCMC volume estimator can be used for realistic problems in reasonable time. The known deterministic algorithms to determine the volume is effected by the curse of dimensionality and are in general not applicable for high dimensional problem as it is often the case for biological problems of realistic complexity.

**CONCLUSIONS**

In this contribution we assessed sampling-based exploration of high-dimensional convex polytopes from a quantitative perspective. Three methods, the Hit and Run algorithm for sampling the inner domain of the polytope as well as the Fok-Crevier and a MCMC method to estimate the corresponding volume are introduced. We analyzed running time and scalability aspects of these methods with two realizations in MATLAB and C++, respectively. The HR algorithm and the MCMC sampler exhibit significant improvements compared to the naive Monte Carlo method suggested in (Wiback et al. 2004). We demonstrated the capability of the C++ implementation to analyze the flux solution space in reasonable time on a single machine, even in high dimensions $d > 20$.

The characterization of solution spaces in terms of size and shape considerations provides valuable information in the context of Metabolic Engineering research. Ex-
emplary impressions are given that address specific biological questions which are discussed with an E. coli metabolic network of realistic complexity. Our results confirm that our specially tailored MCMC approach can be established as viable tool complementing methods routinely operated in this research field (e.g. MFA, FBA, EFM etc.). Future work will link these insights to routine methods and to further explore the capabilities of the MCMC method to access additional insights of the network model.

REFERENCES


