

Fast Flux Module Detection Using Matroid Theory

Arne C. Müller^{1,2,3}, Frank J. Bruggeman⁸, Brett G. Olivier^{4,5,7},
and Leen Stougie^{4,6}

¹ Department of Mathematics and Computer Science, Freie Universität Berlin,
Arnimallee 6, 14195 Berlin, Germany

arne.mueller@fu-berlin.de

² International Max Planck Research School for Computational Biology and
Scientific Computing (IMPRS-CBSC), Max Planck Institute for Molecular Genetics,
Inhnestrasse 63-73, 14195 Berlin, Germany

³ Berlin Mathematical School (BMS), Berlin, Germany

⁴ Centre for Mathematics and Computer Science (CWI),
Science Park 123, 1098 XG Amsterdam, The Netherlands

⁵ Molecular Cell Physiology, VU University,
De Boelelaan 1087, 1081 HV, Amsterdam, The Netherlands

⁶ Operations Research, VU University,

De Boelelaan 1085, 1081 HV, Amsterdam, The Netherlands

⁷ Netherlands Institute for Systems Biology, Amsterdam, The Netherlands

⁸ Systems Bioinformatics, VU University,

De Boelelaan 1087, 1081 HV, Amsterdam, The Netherlands

Abstract. *Flux balance analysis* (FBA) is one of the most often applied methods on genome-scale metabolic networks. Although FBA uniquely determines the optimal yield, the pathway that achieves this is usually not unique. The analysis of the optimal-yield flux space has been an open challenge. *Flux variability analysis* is only capturing some properties of the flux space, while *elementary mode analysis* is intractable due to the enormous number of elementary modes. However, it has been found by Kelk et al. 2012, that the space of optimal-yield fluxes decomposes into *flux modules*. These decompositions allow a much easier but still comprehensive analysis of the optimal-yield flux space.

Using the mathematical definition of module introduced by Müller and Bockmayr 2013, we discovered that flux modularity is rather a local than a global property which opened connections to matroid theory. Specifically, we show that our modules correspond one-to-one to so-called *separators* of an appropriate matroid. Employing efficient algorithms developed in matroid theory we are now able to compute the decomposition into modules in a few seconds for genome-scale networks. Using that every module can be represented by one reaction that represents its function, in this paper, we also present a method that uses this decomposition to visualize the interplay of modules. We expect the new method to replace flux variability analysis in the pipelines for metabolic networks.

Keywords: metabolic networks, FBA, flux modules, matroid theory.

1 Introduction

The metabolic capabilities and behaviors of biological cells are often modeled using metabolic networks. A metabolic network is constituted of a set of chemical compounds and a set of reactions describing the possible transformations of compounds. In the last years it became possible to reconstruct such networks on the genome-scale. This means that on one hand nearly all the reactions that can happen in a biological cell are included. On the other hand such networks consist of thousands of reactions.

Constraint based methods have proven to be very successful in the analysis of metabolic networks [19,20]. In constraint based methods no detailed information on reaction kinetics is needed. Often, the knowledge of reaction stoichiometries is sufficient. Each reaction is described by a column of the stoichiometric matrix S , which has an entry for each chemical compound; s_{ij} the i, j -th entry of S is the number of molecules of compound i consumed ($s_{ij} < 0$), produced ($s_{ij} > 0$), or not involved ($s_{ij} = 0$) in reaction j . Typically, the network is assumed to be in equilibrium, i.e. every metabolite is produced at the same rate as it is consumed. This gives rise to linear constraints on the set of feasible flux-vectors v (pathways) through the network, formally written as

$$Sv = 0$$

Together with bounds ℓ, u on reaction rates (thermodynamic information in form of irreversibilities, limitation of nutrient uptake rates, etc.) we obtain a polyhedron of feasible flux-vectors:

$$P = \{v : Sv = 0, \ell \leq v \leq u\} \quad (1)$$

Among the most prominent analysis methods is *flux balance analysis* (FBA) [28,17,21]. It is, for example, used to compute the optimal biomass yield that can be achieved by a cell under some growth medium [5]. It also computes an optimal flux distribution.

Such an optimum is easily found by solving a linear programming problem of the following type:

$$\max\{cv : Sv = 0, \ell \leq v \leq u\}$$

However, in general such optimal flows are not unique [13]. If this is ignored, it can lead to wrong predictions of by-product flux rates [11].

Kelk et al. [9] showed that many reactions have fixed flux rate in all optimal solutions. The remaining variability is due to variability of the fluxes on a number of relatively small subnetworks, which we call *flux modules*. Each such a module has in each optimum specific fixed input and output compounds. Müller and Bockmayr [15] used this property to formalize the notion of flux-module in a mathematically rigorous way. This allowed them to show that every optimal yield *elementary flux mode* (EFM) [23] is a concatenation of reactions with fixed flux and an elementary mode of each of the flux modules.

While the method by Kelk et al. [9] required the enumeration of exponentially many vertices of a flux polyhedron (which are related to the optimal yield EFMs),

Müller and Bockmayr [15] showed a way to find the modules without needing to compute all extreme solutions. Their method however relied on many runs of *flux variability analysis* (FVA) [3,13]. In FVA for each reaction r of the network the following two linear optimization problems are solved to obtain the minimal and maximal possible flux rate:

$$\max / \min \{v_r : Sv = 0, \ell \leq v \leq u\}$$

Although faster than EFM enumeration, the method is very sensitive to numerical instabilities and analyses of genome-scale networks could still take several hours.

The most important result in this paper is an extremely simple method allowing to compute the flux-modules in a few seconds for genome-scale metabolic networks. The method, described in Section 2, is based on the observation that the modules correspond to the separators of the linear matroid defined by the columns of the stoichiometric matrix that belong to reactions with variable optimal flux. We will explain all these technical concepts in Section 1.1. The efficiency of our method is demonstrated in Section 3 by application to several genome-scale metabolic networks.

Flux modularity highly depends on the growth conditions. In particular, interesting flux modules can usually only be found in the optimal flux space. Hence, it is of high importance to understand how the decomposition of modules changes under different growth conditions and objective functions. Since with our new method, module computation has become so fast, we can simply compute and compare modules under many different growth conditions and compare the results. Essential for this is a visualization method that shows the interplay of modules in the context of the whole network. In Sec. 2.2 we present a method that automatically generates such a visualization using a clever compression based on flux modules. Results of that method applied to a set of genome-scale metabolic networks can be found in Sec. 3.2.

1.1 Definitions and Preliminaries

We use \mathcal{M} to denote the set of metabolites, \mathcal{R} to denote the set of reactions. We abuse the notation for sets also for their size. $S \in \mathbb{R}^{\mathcal{M} \times \mathcal{R}}$ denotes the stoichiometric matrix. For $b \in \mathbb{R}^{\mathcal{M}}$, we analyze flux spaces $P \subseteq \{v \in \mathbb{R}^{\mathcal{R}} : Sv = b\}$. We observe that $b = 0$ leads to the standard steady-state assumption. Here, we also allow $b \neq 0$ to simplify notation in the context of modules. Furthermore, the space of optimal-yield fluxes is again a polyhedron and can be written in this form, too [15]. In this paper we will show that we can reduce the analysis of P to the analysis of linear vector spaces, i.e. flux spaces defined by the kernel of S : $\ker(S) := \{v \in \mathbb{R}^{\mathcal{R}} : Sv = 0\}$. We use subscripts to index flux through reactions, i.e., v_r denotes flux through reaction r . The support of flux-vector v is denoted by $\text{supp}(v) := \{r \in \mathcal{R} : v_r \neq 0\}$. We will also be interested in the flux through a subset of reactions $A \subseteq \mathcal{R}$. Hence, we write v_A to denote the components of v corresponding to the reactions in A and we use S_A to denote the stoichiometric

matrix that only contains the columns corresponding to the reactions in A . We define the projection $\text{pr}_A(P) := \{v_A : v \in P\}$.

Definition 1. [Flux Module, [15]] $A \subseteq \mathcal{R}$ is a P -module if there exists a $d \in \mathbb{R}^{\mathcal{M}}$ s.t. $S_A v_A = d$ for all $v \in P$. We call d the interface flux of the module.

In contrast to the definition in [15], we also allow $A = \emptyset$ to be a module, which together with \mathcal{R} we call the trivial modules. We present here some useful properties of modules proven in [15]. They may also help the reader to get some intuition on the concept of module.

Proposition 1. *Properties of Modules.*

- (i) If disjoint sets A and B are P -modules then $A \cup B$ is a P -module;
- (ii) If A and B are P -modules and $B \subset A$ then $A \setminus B$ is a P -module.

The rest of this section is devoted to an introduction to the relevant concepts from Matroid Theory [18], which is a generalization of graph theory and linear algebra. A matroid is defined by a universe of elements and subsets of them that have some independence structure.

Definition 2. Given a universe \mathcal{U} and a family \mathcal{A} of independent subsets of \mathcal{U} . Then $\{\mathcal{U}, \mathcal{A}\}$ is a matroid if it satisfies the following conditions.

- $\emptyset \in \mathcal{A}$;
- If $A \in \mathcal{A}$ and $A' \subset A$, then $A' \in \mathcal{A}$;
- If $A, A' \in \mathcal{A}$ and A' contains more elements than A , then there exists an element $e \in A' \setminus A$, such that $A \cup \{e\} \in \mathcal{A}$.

As a very relevant example, a set of vectors in $\mathbb{R}^{\mathcal{R}}$, together with their linearly independent subsets form a matroid; a so-called *linear matroid*. Another example, less relevant here, is a graph with subsets of its edges that form forests as independent sets; a so-called *graphic matroid*. Matroid theory has already been used in the past to describe metabolic networks [16,2]. Indeed, many concepts from metabolic networks also exist in matroid theory. For example, flux modes in metabolic networks correspond to cycles in matroid theory; i.e., dependent sets of a matroid. Elementary flux modes correspond to circuits; i.e., minimal dependent sets. The only difference is, that in matroid theory we only talk about the support. I.e., $A \subseteq \mathcal{R}$ is a cycle if and only if there exists a flux mode v with $A = \text{supp}(v)$. Similarly, a circuit $C \subseteq \mathcal{R}$ is a cycle with minimal support.

However, matroid theory is a wide field with additional concepts that have not yet been studied for metabolic networks. We show some of these concepts, that originate from linear algebra, directly in terms of the linear matroid represented by the columns of the stoichiometric matrix S , which is the matroid that we use to describe metabolic networks. In our notation \mathcal{R} is the set of columns of S .

- $A \subseteq \mathcal{R}$ is called a basis if it is a maximal independent set.
- $r(A)$ is the rank of S_A , i.e. the size of the largest independent subset of A .
- The dual matroid is the matroid represented by a kernel matrix of S .

Also graph theory introduces some further useful concepts into matroid theory. Typical examples are the circuits and cycles. Another property is connectivity. A graph is 2-connected if for any two edges, there exists a circuit that contains both. If a graph is not 2-connected, then the union of connected components is called a separator; i.e., any of the two sides of a partition of a graph into two parts A, B that are not 2-connected to each other. We can understand this as that there exists no flow circulation between A and B . It follows that the interface flux of A and B has to be zero. In matroid theory separators are defined using the matroid rank function:

Definition 3 (Separator). *A set A is defined to be a separator if and only if*

$$r(A) + r(\mathcal{R} \setminus A) = r(\mathcal{R}).$$

In Sec. 2.1 we show how the flux modules of a metabolic network correspond one-to-one to the separators of the corresponding matroid. We then use matroid theory to derive a very fast and simple algorithm for finding modules. It is based on a result by Krogdahl [12]. The runtime results on a set of genome-scale metabolic networks are presented in Sec. 3.1.

2 Methods

2.1 Finding Modules Efficiently

We first show that it is sufficient to analyze modularity as a local property of one point in the inside of the flux space, implying that we can ignore reaction reversibilities and simply analyze a subvector-space (Thm. 1). This allows to describe modularity in terms of matroid separators (Thm. 2), which we then exploit in designing an efficient algorithm to compute modules.

To make the first step, consider a point x *inside* the flux space and a neighbourhood of it (Fig. 1). This neighbourhood captures all the characteristics needed to analyse modularity of the whole flux space. We only have to deal with the term “inside”. Since $P \subseteq \{v \in \mathbb{R}^{\mathcal{R}} : Sv = b\}$, it follows that P is of lower dimension in $\mathbb{R}^{\mathcal{R}}$. Hence, we will only consider the interior relative to $\ker(S)$. However, if we have reactions with fixed flux rate, P will also have lower dimension than

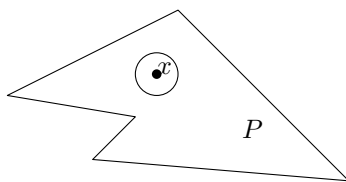


Fig. 1. Viewed from a point x inside the flux space, the flux space looks like a linear vector space and the bounds are not important

$\ker(S)$. Therefore, we will restrict to reactions with variable flux rate, which we define by:

$$\begin{aligned} V &:= \{r \in \mathcal{R} : v_r^{\max} \neq v_r^{\min}\}, \text{ where} \\ v_r^{\max} &:= \sup\{v_r : v \in P\} \\ v_r^{\min} &:= \inf\{v_r : v \in P\} \end{aligned} \tag{2}$$

This restriction does not destroy the module property:

Observation 1. *It holds for all $A \subseteq V$ that A is P -module $\Leftrightarrow A$ is $\text{pr}_V(P)$ -module.*

To guarantee that we can find a x inside the flux space after we restricted to reactions with variable flux rate, we require that P is convex. Theoretically, there are weaker conditions that are also sufficient.

Theorem 1. *If $P \subseteq \{v \in \mathbb{R}^{\mathcal{R}} : Sv = b\}$, is convex, it holds for all $A \subseteq \mathcal{R}$*

$$A \text{ is } P\text{-module} \Leftrightarrow A \cap V \text{ is } \ker(S_V)\text{-module.}$$

The proof can be found on the supplementary website.

By Thm. 1 we can restrict our attention to the analysis of linear vector spaces. Hence, in the following we will only analyse polyhedra of the form $P = \ker(S)$. We will relate modules of $\ker(S)$ to separators of the matroid defined by the columns of S . Remember the explanation of a separator in a graph in terms of the non-existence of a flow circulation in Section 1.1 and observe, that every module in $\ker(S)$ also has interface flux 0 since $0 \in \ker(S)$.

Formally, we obtain the following theorem, the proof of which is deferred to the supplementary website <https://sourceforge.net/projects/fluxmodules/>.

Theorem 2. *$A \subseteq \mathcal{R}$ is a $\ker(S)$ -module if and only if A is a separator in the matroid represented by S .*

The characterization of modules as separators of matroids allows to compute the flux-modules of a metabolic network efficiently. Since separators and modules are closed under disjoint union, it suffices to describe the set of *minimal nontrivial separators* (modules).

Definition 4 (Minimal Module). *A P -module $\emptyset \neq A \subseteq \mathcal{R}$ is called minimal if there exists no P -module $B \neq \emptyset$ with $B \subset A$.*

To understand the algorithm for finding the modules, we observe that the minimal non-trivial separators are the connected components of the matroid. In the context of graph-theory these are called 2-connected components (Note the inconsistency of the terminology between matroid and graph theory. The connected components in graph-theory are something different.) Formulated in matroid-terminology we recall the following graph-theoretic characterization of 2-connected component: For any 2 elements (columns of S in the linear matroid, edges in the graph) in the same connected component there exists a minimal dependent set (circuit) that contains them both. For pairs of elements of different

connected components this is not true. It turns out that this also holds for matroids in general (Proposition 4.3.4 in [18]).

Theoretically, we could now build a graph $G = (V, E)$, where V is the set of reactions defined in (2) and there is an edge between two reactions (columns of S_V) if and only if there exists a circuit that contains both. The connected components (in the graph-theoretic sense) of G will be the minimal separators. Since the number of circuits explodes exponentially, it is not efficient to enumerate all circuits in order to compute the connected components of the graph G . Indeed, this is also not necessary and it suffices to look at a special set of circuits, so called *fundamental circuits* [27].

A set of fundamental circuits is obtained as follows: We start by finding a basis X of the matroid; i.e., a maximal independent set, which we compute by Alg. 1. Notice that, starting from the empty set, the algorithm grows X by adding elements only if this keeps X independent. Since we try to add all elements to X , it follows that at the end of the algorithm, X will be a basis of the linear matroid represented by S_V .

Let $Y := V \setminus X$. Clearly, for every $r \in Y$, adding r to X will create a cycle $C^r \subseteq X \cup \{r\}$. It is easy to see that C^r is actually a circuit, which is called fundamental circuit. In Alg. 1 the fundamental circuits are constructed simultaneously with constructing X . This gives us a so-called *partial representation*.

We now build, by Alg. 2, the graph $G' = (V, E')$, where two reactions are connected by an edge if there exists a fundamental circuit that contains both. Krogdahl and Cunningham showed that the connected components of G' , found by Alg. 2, are precisely the minimal separators of the matroid [4,12].

To each circuit C there exists a flux vector v that is unique up to scaling with $C = \text{supp}(v)$, $Sv = 0$. If we enter for every circuit in B the corresponding flux values from v , we obtain a null-space matrix of S . Hence, this approach can be understood as computing a block-diagonalization of the null-space matrix. Approaches like this in the context of stoichiometric matrices have already been studied in [24]. However, [24] does not use matroid theory and it is unclear whether their method will always compute the finest block-diagonalization.

Here we recapitulate all the steps for finding the modules of the optimal flux space of a metabolic network.

1. Determine the optimal value by LP;
2. Set the objective function equal to the optimum value and add it as a constraint;
3. For each reaction r maximize and minimize the flux through r in the optimal flux space;
4. Determine the set V of reactions for which the maximum and the minimum are not equal;
5. Select the set of columns S_V corresponding to V of the stoichiometric matrix S and neglect the non-negativity constraints; i.e., irreversibilities, directions of the reactions;

Algorithm 1. Computes a basis X and its set of fundamental circuits of a matroid represented by S

```

function ComputePartialRepresentation( $S$ )
 $\mathcal{C} = \emptyset$ 
 $X = \emptyset$ 
for  $r \in V$  do
  check feasibility of  $S_X v = -S_r$ 
  if feasible then
     $C := \text{supp}(v) \cup \{r\}$ 
     $\mathcal{C} := \mathcal{C} \cup \{C\}$ 
  else
     $X := X \cup \{r\}$ 
  end if
end for
return  $\mathcal{C}$ 

```

Algorithm 2. Computes the modules of $\{v : S_V v = 0\}$

```

function ComputeModules()
 $\mathcal{C} = \text{ComputePartialRepresentation}(S_V)$ 
Build Graph  $G = \{V, E\}$  with  $(x, y) \in E$  iff there exists  $C \in \mathcal{C}$  with  $x, y \in C$ .
 $\mathcal{A} = \text{find connected components of } G$  (e.g. using depth-first search).
return  $\mathcal{A}$ 

```

6. Apply Alg. 2 to compute the minimal modules \mathcal{A} of $\{v \in \mathbb{R}^V : S_V v = 0\}$.
7. \mathcal{A} is the set of minimal modules that contain reactions in V . The reactions with fixed flux are all minimal modules by themselves.

We notice that steps 3 (and therefore 4) of the algorithm can be parallelized in a trivial way, reducing the computation times even further.

2.2 Visualization

We develop a visualization tool to help us understand how the decomposition of modules changes under different growth conditions and objective functions. By the definition of module, the reactions inside a module have together a fixed function (the interface flux). Hence, we can represent the module by a single reaction with a fixed flux in the genome-scale network. The stoichiometry of the representing reaction is precisely the interface flux of the module.

This way we can create a compressed network that contains all the reactions with fixed flux rates and artificial reactions that represent the modules. This compressed network has the following advantages:

- The number of reactions carrying flux is compressed (a module with many reactions, is represented by a single reaction).
- All the reactions in the compressed network have a fixed flux rate.

Unfortunately, the number of fixed reactions is still very large. This prevents automatic visualization of the network and the role of the modules containing variable reactions is obfuscated. However, reactions that have a fixed flux rate can also be grouped together into modules by Prop. 1.

Theoretically, we could group all reactions with a fixed flux rate into 1 module. This would result in a compressed metabolic network consisting of $k+1$ reactions, where k is the number of minimal modules containing reactions with variable flux rates. In particular, the module containing all fixed reactions will likely also contain the biomass- and nutrient-uptake reactions. If we want to understand the role of the modules for biomass production or nutrient uptake, this is not very useful. Moreover, modules of variable reactions may disconnect reactions with fixed flux rates from each other. Such disconnected reactions are important for the mediation between modules and should also be displayed separately. Hence, we decided to build a compressed network as follows:

1. Given: A collection Mod of interesting modules (selected by the user). Mod has to cover all reactions with variable flux rates. Typically Mod contains all minimal modules of variable reactions, a module containing the biomass reaction and modules containing the nutrient uptake reactions.
2. We compute the set $\mathcal{R}_{\text{Mod}} := \{r \in \mathcal{R} : r \in M \exists M \in \text{Mod}\}$ of reactions in interesting modules.
3. We compute the set $\mathcal{R}_B := \{r \in \mathcal{R} \setminus \mathcal{R}_{\text{Mod}} : v_r = 0 \forall v \in P\}$ of blocked reactions.
4. We compute the set $\mathcal{M}_{\text{Mod}} := \{m \in \mathcal{M} : \exists r \in \mathcal{R}_{\text{Mod}} \text{ such that } m \in \text{supp}(S_r)\}$ of metabolites involved in the interesting modules.
5. We consider the metabolic network, where \mathcal{R}_{Mod} , \mathcal{R}_B and \mathcal{M}_{Mod} are removed. It is represented by the stoichiometric matrix $S' := S_{\mathcal{M} \setminus \mathcal{M}_{\text{Mod}}, \mathcal{R} \setminus (\mathcal{R}_{\text{Mod}} \cup \mathcal{R}_B)}$.
6. We compute the connected components Mod_F of S' . We do so by defining the incidence matrix of a bipartite graph, the nodes of which on one side of the bipartition correspond to the rows of S' , and the ones on the other side to the columns of S' , and there is an edge between row-node i and column-node j if and only if $S'_{ij} \neq 0$. The column-nodes represent the reactions in $\mathcal{R} \setminus (\mathcal{R}_{\text{Mod}} \cup B)$, and the corresponding reactions of the connected components of this bipartite graph, whence Mod_F , forms a partition of $\mathcal{R} \setminus (\mathcal{R}_{\text{Mod}} \cup B)$. Clearly, every $A \in \text{Mod}_F$ is a module, since Mod_F only contains fixed reactions.
7. We represent each module in Mod, Mod_F by a single reaction with the corresponding interface flux. Let \mathcal{M}_0 be the set of metabolites that have a net interface flux of 0 in all these modules. We suppress \mathcal{M}_0 , since they would just show up as isolated metabolites. We obtain a metabolic network with metabolites $\mathcal{M}' := \mathcal{M} \setminus \mathcal{M}_0$ and reactions $\mathcal{R}' := \text{Mod} \cup \text{Mod}_F$.
8. We remove reactions disconnected from the network that contain the target reaction, e.g. because of modules that form thermodynamically infeasible cycles or otherwise have no role in the metabolism.

In practice, this results in medium-scale networks that can automatically be visualized with graph-drawing software like `GraphViz` [8].

3 Results

3.1 Runtime of Module Finding

With the new method we can compute all flux modules for the optimal flux space of genome scale networks in about the same time as is needed for conventional flux variability analysis. In Table 1 we see that the new method using matroid theory outperforms the previous methods in orders of magnitude. We used the metaopt toolbox [14] to solve the flux variability subproblems. Unfortunately, we did not have access to all the runtime data of [9] which is why some of the data is missing and the reported runtimes may be only from some of steps in the pipeline. The computations for the matroid approach were obtained by computations on a 4-core desktop computer.

Table 1. Comparison of runtimes for computing modules in the optimal flux space of genome scale networks

Network	using [9]	using [15]	using matroids
<i>E. coli</i> iAF1260	133495sec	755sec	6.4sec
<i>E. coli</i> iJR904	1906sec	162sec	1.9sec
<i>E. coli</i> iJO1366			8.4sec
<i>H. pylori</i> iT341		55.5sec	0.8sec
<i>H. sapiens</i> recon. 1			153.3sec
<i>H. sapiens</i> recon. 2			1131sec
<i>M. barkeri</i> iAF692	1088sec	941sec	1.4sec
<i>M. tuberculosis</i> iNJ661	9317sec	1623sec	4.3sec
<i>S. aureus</i> iSB619		127.8sec	1.2sec
<i>S. cerevisiae</i> iND750			3.0sec

In particular notice that large networks like *Human recon 2* can now also be analyzed. In addition, the new method is numerically much more stable. In the method introduced by [15] it often happens that error tolerances are chosen too small or too large, which causes that linear programs that should be feasible are detected as infeasible etc. This then usually caused the algorithm to abort and the tolerance sometimes needed to be adjusted according to the problem instance.

We experienced that the new matroid based method is much more robust in this respect. Our initial tolerances of 10^{-20} for the optimization step, 10^{-8} for the flux variability and 10^{-9} for the final module computation worked in all cases.

Note, that the other two methods are solving slightly different problems. In [15] we were actually looking for modules in the thermodynamically constrained flux space and in [9], rays and linealities are eliminated prior to module computation.

A comparison between the results of [15] and the new method on *E. coli* iAF1260 revealed that 7 of the modules coincide, 2 modules from the new method contain additional reactions (which have fixed flux under thermodynamic constraints). The remaining modules are computed by the new method, but not by [15] since they again only contain reactions that have fixed flux by thermodynamic constraints (usually those modules are formed by a splitted pair of forward and backward reactions). The differences seem to be small, but a detailed analysis will be subject to future work. Also, we want to point out that, for computing modules, the method by Kelk et al. [9] has to enumerate all the extreme points of the flux polyhedron of optimal fluxes (after some preprocessing), a much harder task. As a result more information than modules is obtained. Hence, the previous works still remain useful.

3.2 Visualization

We used the visualization method presented in Section 2.2 to create visualizations of the above mentioned genome scale networks. The results can be found on the supplementary website. In Tab. 2, we compare the original size of the networks with the size of the compressed networks that are used to visualize the interplay of the flux modules with variable flux rates. Each reaction of the compressed network is a flux module. Every minimal flux module containing reactions with variable flux rates is represented by exactly one reaction. Reactions with fixed flux rate are grouped together. It is interesting to see that although the networks have quite different sizes originally, the compressed sizes do not vary very much.

Table 2. Size of the compressed networks

Network	No. Metabolites (original)	No. Reactions (original)	No. Metabolites (compressed)	No. Reactions (compressed)
<i>E. coli</i> iAF1260	1668	2382	46	25
<i>E. coli</i> iJR904	761	1075	42	17
<i>E. coli</i> iJO1366	1805	2583	49	27
<i>H. pylori</i> iT341	485	554	32	20
<i>M. barkeri</i> iAF692	628	690	35	13
<i>M. tuberculosis</i> iNJ661	826	1025	58	26
<i>S. aureus</i> iSB619	655	743	39	22
<i>S. cerevisiae</i> iND750	1061	1266	57	24

Visualizations of some of the example networks and their modules, using the tool `dot` [7] from the `GraphViz` toolbox, can be found on the supplementary website <https://sourceforge.net/projects/fluxmodules/>. The MATLAB scripts for module detection can be found there as well.

4 Discussion

4.1 Enumeration of Optimal-Yield Pathways

We showed that flux modules [9,15] of genome-scale metabolic networks can efficiently be computed using matroids. We confirmed the previous results that the optimal flux space of most genome-scale metabolic networks decomposes into modules. If we want to compute the set of all optimal yield elementary modes, we theoretically can do this by simply computing the optimal yield elementary modes for each module. Then, we can use the decomposition theorem of [15] and obtain all optimal yield elementary modes of the whole network. There is only a small numerical barrier to be climbed to do this in practice: The EFM enumeration problem for each module appears to be numerically very unstable. Hence it is likely that EFMs are missed if not everything is computed using precise rational arithmetic.

We noted that the previous methods [9,15] were computing flux modules on slightly different flux spaces (in [9] rays and linealities were removed, in [15] we worked on the thermodynamically feasible flux space). These differences seem to be small but could be of significant biological importance. For example, it could be that due to thermodynamic constraints a reaction is blocked and hence, we can refine the modules. In a follow up work we will (mathematically and empirically) analyse the impact of these differences.

The full flux space is usually not decomposable into modules. In a follow up paper we will generalize the notion of module. This will allow us to find interesting modules also for the full flux space. Furthermore, this will have the potential to derive similar decomposition theorems as in [15] that then will work on the full flux space as well. We think this will be a major step towards EFM enumeration of genome-scale networks.

4.2 Modularity under Different Growth-Conditions

It has been observed that the decomposition into modules depends on the growth condition [9,15]. If we want to understand how the optimal flux space changes if the growth condition is modified, we have to recompute the decomposition into modules. Previously, this was a tedious task. Now it is very simple and fast and it can be done even for very small changes.

We presented a visualization method that shows the interplay of the modules and how they contribute to optimal biomass production. We think that this visualization will be very helpful to detect when a change in a growth condition significantly changes the structure of the optimal flux space.

For the visualization we use the definition of module to lump reactions together. This way we compute a compressed metabolic network that shows the optimal flux distribution with only a small number of reactions. These networks were small enough to be visualized using automated graph drawing tools. Currently, we have only little control on how these networks are drawn, causing the visualization to seem to be very sensitive to changes. In particular it would be

interesting if we could get more robust drawing results for small changes in the network.

4.3 Applications Outside Metabolic Networks

Note that the definition of module of a polyhedron is context free. Every polyhedron can be written in the form $P = \{v \in \mathbb{R}^n : Sv = b, \ell \leq v \leq u\}$, where $S \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $\ell \in \{R \cup \{-\infty\}\}^n$, $u \in \{R \cup \{\infty\}\}^n$ by addition of slack variables. The enumeration of extreme points and extreme rays of such polyhedra is a classical problem in polyhedral combinatorics and computational geometry. In fact, it is a major open problem if this can be done in total polynomial time, i.e., polynomial in input and output. Khachiyan et al. [10] showed that this is impossible if only the extreme points (so without the rays) of an unbounded polyhedron are to be enumerated, unless $P=NP$.

Still, methods have been proposed that do this enumeration. The most prominent ones are variations of the so-called Double Description Method and a method introduced by Fukuda [1,6] (see also the work of Terzer and Stelling [26,25] in the context of metabolic networks). Such methods are much faster on smaller polyhedra, and therefore it makes sense to subject any polyhedron to be enumerated first to our method to see if smaller modules can be found and then apply a favorite enumeration method on the smaller parts.

4.4 Conclusion

In this paper we presented a new method that allows us to compute flux modules very efficiently. This allows us to compute flux modules of many metabolic networks under a large set of different conditions to compare flux modules with existing classical metabolic subsystems like Glycolysis.

Compared to classical metabolic subsystems that, at worst, are arbitrary functional groupings of metabolic reactions/species, flux modules are mathematically well defined. They are structural features only depending on a defined set of conditions (inputs, optimality). This qualifies them as a performance and quality metric for genome-scale metabolic networks. Furthermore, it allows us to investigate the modularity, and simplify genome metabolic networks without the risk of a bias from conventional biological interpretation.

Acknowledgments. We thank Timo Maarleveld for providing us with runtime data of the method by [9].

Authors Contributions

AM and LS found the connection to matroid theory and took care of mathematical correctness. AM provided all detailed proofs. LS supervised and organised the collaboration process. AM, FB and BO developed the visualization method. AM, LS and FB worked on the manuscript. All authors read and approved the manuscript.

References

1. Avis, D., Fukuda, K.: A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete & Computational Geometry* 8(1), 295–313 (1992)
2. Beard, D.A., Babson, E., Curtis, E., Qian, H.: Thermodynamic constraints for biochemical networks. *Journal of Theoretical Biology* 228, 327–333 (2004)
3. Burgard, A.P., Vaidyaraman, S., Maranas, C.D.: Minimal reaction sets for *escherichia coli* metabolism under different growth requirements and uptake environments. *Biotechnology Progress* 17, 791–797 (2001)
4. Cunningham, W.H.: A combinatorial decomposition theory. PhD thesis, University of Waterloo, Ontario, Canada (1973)
5. Feist, A.M., Palsson, B.Ø.: The biomass objective function. *Current Opinion in Microbiology* 13, 344–349 (2010)
6. Fukuda, K., Prodon, A.: Double description method revisited. In: Deza, M., Manoussakis, I., Euler, R. (eds.) *CCS 1995. LNCS*, vol. 1120, pp. 91–111. Springer, Heidelberg (1996)
7. Gansner, E.R., Koutsofios, E., North, S.C., Vo, K.-P.: A technique for drawing directed graphs. *IEEE Transactions on Software Engineering* 19(3), 214–230 (1993)
8. Gansner, E.R., North, S.C.: An open graph visualization system and its applications to software engineering. *Software - Practice and Experience* 30(11), 1203–1233 (2000)
9. Kelk, S.M., Olivier, B.G., Stougie, L., Bruggeman, F.J.: Optimal flux spaces of genome-scale stoichiometric models are determined by a few subnetworks. *Scientific Reports* 2, 580 (2012)
10. Khachiyan, L., Boros, E., Borys, K., Elbassioni, K., Gurvich, V.: Generating all vertices of a polyhedron is hard. *Discrete Computational Geometry* 39, 174–190 (2008)
11. Khannapho, C., Zhao, H., Bonde, B.L., Kierzek, A.M., Avignone-Rossa, C.A., Bushell, M.E.: Selection of objective function in genome scale flux balance analysis for process feed development in antibiotic production. *Metabolic Engineering* 10(5), 227–233 (2008)
12. Krogdahl, S.: The dependence graph for bases in matroids. *Discrete Mathematics* 19, 47–59 (1977)
13. Mahadevan, R., Schilling, C.H.: The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metabolic Engineering* 5, 264–276 (2003)
14. Müller, A.C., Bockmayr, A.: Fast thermodynamically constrained flux variability analysis. *Bioinformatics* 29(7), 903–909 (2013)
15. Müller, A.C., Bockmayr, A.: Flux modules in metabolic networks. *Journal of Mathematical Biology* (2013) (in press, preprint), <http://nbn-resolving.de/urn:nbn:de:0296-matheon-12084>
16. Oliveira, J.S., Bailey, C.G., Jones-Oliveira, J.B., Dixon, D.A.: An algebraic-combinatorial model for the identification and mapping of biochemical pathways. *Bulletin of Mathematical Biology* 63, 1163–1196 (2001)
17. Orth, J.D., Thiele, I., Palsson, B.Ø.: What is flux balance analysis. *Nature Biotechnology* 28, 245–248 (2010)
18. Oxley, J.: *Matroid Theory*, 2nd edn. Oxford Graduate Texts in Mathematics. Oxford University Press, New York (2011)

19. Jason Papin, A., Stelling, J., Price, N.D., Klamt, S., Schuster, S., Palsson, B.Ø.: Palsson. Comparison of network-based pathway analysis methods. *Trends in Biotechnology* 22(8), 400–405 (2004)
20. Price, N.D., Reed, J.L., Palsson, B.Ø.: Genome-scale models of microbial cells: evaluating the consequences of constraints. *Nature Reviews Microbiology* 2, 886–897 (2004)
21. Santos, F., Boele, J., Teusink, B.: A practical guide to genome-scale metabolic models and their analysis. *Methods in Enzymology* 500, 509 (2011)
22. Schellenberger, J., Que, R., Fleming, R.M.T., Thiele, I., Orth, J.D., Feist, A.M., Zielinski, D.C., Bordbar, A., Lewis, N.E., Rahmanian, S., Kang, J., Hyduke, D.R., Palsson, B.Ø.: Quantitative prediction of cellular metabolism with constraint-based models: the COBRA toolbox v2.0. *Nature Protocols* 6(9), 1290–1307 (2011)
23. Schuster, S., Hilgetag, C.: On elementary flux modes in biochemical systems at steady state. *J. Biol. Systems* 2, 165–182 (1994)
24. Schuster, S., Schuster, R.: Detecting strictly detailed balanced subnetworks in open chemical reaction networks. *Journal of Mathematical Chemistry* 6, 17–40 (1991)
25. Terzer, M.: Large scale methods to enumerate extreme rays and elementary modes. PhD thesis, Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 18538, 2009 (2009)
26. Terzer, M., Stelling, J.: Large-scale computation of elementary flux modes with bit pattern trees. *Bioinformatics* 24(19), 2229–2235 (2008)
27. Truemper, K.: Partial matroid representations. *European Journal of Combinatorics* (1984)
28. Varma, A., Palsson, B.Ø.: Metabolic flux balancing: Basic concepts, scientific and practical use. *Nature Biotechnology* 12, 994–998 (1994)

Supplementary Material

For the supplementary material we have created a sourceforge project <https://sourceforge.net/p/fluxmodules/wiki/Home/>. There you can find

- The proofs for Thm. 1 and Thm. 2.
- The MATLAB code for computing the modules (requires the openCOBRA toolbox [22])
- The MATLAB code for visualising modules (requires the tool dot from GraphViz [7]).
- Computational results and visualizations for the example networks. For each network the results are grouped into one .zip archive.