Hierarchical decomposition of metabolic networks using $k$-modules

Arne C. Reimers*1

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

Abstract
The optimal solutions obtained by flux balance analysis (FBA) are typically not unique. Flux modules have recently been shown to be a very useful tool to simplify and decompose the space of FBA-optimal solutions. Since yield-maximization is sometimes not the primary objective encountered in vivo, we are also interested in understanding the space of sub-optimal solutions. Unfortunately, the flux modules are too restrictive and not suited for this task. We present a generalization, called $k$-module, which compensates the limited applicability of flux modules to the space of sub-optimal solutions. Intuitively, a $k$-module is a sub-network with low connectivity to the rest of the network. Recursive application of $k$-modules yields a hierarchical decomposition of the metabolic network, which is also known as branch decomposition in matroid theory. In particular, decompositions computed by existing methods, like the null-space-based approach, introduced by Poolman et al. [2007] J. Theor. Biol. 249, 691–705 can be interpreted as branch decompositions. With $k$-modules we can now compare alternative decompositions of metabolic networks to the classical sub-systems of glycolysis, tricarboxylic acid (TCA) cycle, etc. They can be used to speed up algorithmic problems [theoretically shown for elementary flux modes (EFM) enumeration] and have the potential to present computational solutions in a more intuitive way independently from the classical sub-systems.

Introduction
Constraint based methods have proven to be very successful in the analysis of metabolic networks [1,2], which are used to model metabolic capabilities and predict behaviours of organisms. In contrast with kinetic models, constraint-based metabolic network models do not aim to predict a single phenotype, but a space of biologically possible phenotypes. This is achieved by excluding unrealistic phenotypes using constraints. This reduces the data requirements enormously such that also large models with thousands of reactions can be built.

Because of the size of the networks, however, even the interplay of very simple constraints can yield very complex and high-dimensional solution spaces that are very hard to comprehensively analyse. This is already the case for networks solely based on the steady-state assumption and irreversibility constraints, which are the basic assumptions for methods like flux balance analysis (FBA) [3,4] and related methods. The steady-state assumption states that every metabolite must be produced at the same rate as it is consumed. Formally, a vector of reaction rates (flux vector) $\mathbf{v} \in \mathbb{R}^R$ is in steady-state if it satisfies

$$S\mathbf{v} = 0,$$

where $S$ is the stoichiometric matrix. We use $\mathcal{R}$ to denote the set of all reactions and $\mathcal{M}$ to denote the set of all metabolites.

With a set $\text{Irrev} \subseteq \mathcal{R}$ of reactions that are only allowed to operate in forward direction, the full steady-state flux space is obtained, as given below:

$$\{ \mathbf{v} \in \mathbb{R}^R : S\mathbf{v} = 0, \mathbf{v}_{\text{Irrev}} \geq 0 \}$$

Although extreme pathways [5] or elementary flux modes (EFM) [6,7] can comprehensively characterize the solution space based on easily understandable pathways, the number of pathways explodes with the size of the network. This makes these approaches only applicable to small networks.

Therefore, many methods try to determine only special properties of the network. For example, FBA asks for the maximal biomass yield for a given uptake of nutrients [3,4]. Although the space of optimal yield fluxes (optimal yield flux space) also contains many solutions [8,9]; Kelk et al. [10] discovered a method that allows a comprehensive pathway-based description for the optimal yield flux space of many genome-scale networks. They observed that the optimal yield flux space can be decomposed into flux modules. For a flux space $P \subseteq \mathbb{R}^R$, a $P$-module is a set of reactions $A \subseteq \mathcal{R}$ for which there exists a vector $d \in \mathbb{R}^M$ with

$$S_A\mathbf{v}_A = d \text{ for all } \mathbf{v}_A \in P,$$

where $S_A$ denotes the sub-matrix of $S$ with only columns corresponding to the reactions in $A$. Similarly, $\mathbf{v}_A$ is the sub-vector of $\mathbf{v}$ with only entries corresponding to reactions in $A$. With this definition, originally introduced by Müller and Bockmayr [11], the flux modules can be efficiently computed [12]. By computing the pathways through each module, a comprehensive pathway-based description can be obtained.
The reactions in $A = \{r_1, \ldots, r_{10}\}$ have three boundary metabolites and form a 2-module.

$k$-modules

Let us consider the network shown in Figure 1 and the set $A = \{r_1, \ldots, r_{10}\}$ of reactions. The metabolites $B = \{m_1, m_2, m_3\}$ are each involved in a reaction of $A$ and also in one of the other reactions. These metabolites form the boundary of $A$ and therefore connect $A$ to the rest of the network. Therefore, we call them the boundary metabolites $B(A)$ of $A$:

$$B(A) := \{ m \in M : S_{mu} \neq 0 \neq S_{vu} \exists r \in A, v \notin A \}$$

We observe that for any set of reactions $A$, we can compute how well $A$ is connected to the rest of the network using the formula below:

$$\mu(A) := |B(A)|.$$  

We argue that a set of reactions $A$ with low connectivity $\mu(A)$ should be easy to analyse by itself, because the interaction with the rest of the network that could influence the interpretation of $A$ is low.

However, we also observe for the set $A$ from Figure 1 that $m_1$ is always produced at the rate by which $m_2$ is consumed (in real networks this can happen for example with currency metabolites like ATP and ADP). Hence, the interaction of $A$ through $m_2$ is already given by the interaction through $m_1$ with the rest of the network. To deal with such redundancies, we use the concept of $k$-modules.

A set of reactions $A \subseteq R$ is a $P$-$k$-module, if there exists a $d \in R^k$ and a matrix $D \in R^{M \times k}$ (interface) such that for every $v \in P$ there exists an $\alpha \in R^k$ with

$$S_{A^v} = d + D\alpha$$

We simply write $k$-module if $P$ is the full steady-state flux space and the network contains no blocked reactions.

The set $A$ from the example of Figure 1 is a 2-module, because we can choose

$$D = m_1 \begin{pmatrix} 1 & 0 \\ m_2 & -1 \\ m_3 & 0 \end{pmatrix}$$

The notion of $k$-module allows us now to define an alternative connectivity function that does not only consider network topology but also stoichiometries:

$$\lambda(A) := \min \{ k : A \text{ is a } k \text{-module} \}$$
To improve the reader’s understanding of $k$-modules and their relation to the previously defined $P$-modules, we here list a few properties that hold in general:

- $\mu(A) = \mu(R \setminus A)$
- $\lambda(A) = \lambda(R \setminus A)$
- $\lambda(A) \leq \mu(A)$
- $\lambda(\bar{A}) \leq |A|$ (note that $\mu(A) \leq |A|$ does not hold in general)
- $\lambda(\bar{A}) \leq |R \setminus A|$
- $A$ is a $P$-module if and only if $A$ is a $P$-$\emptyset$-module (i.e., $A$ is a $\emptyset$-module if $P$ is the full steady-state flux space without blocked reactions).

Furthermore, we want to remark that the simplifications [12] that make an efficient computation of flux modules possible, also apply to $k$-modules. This means that as soon as all reactions with fixed reaction rate have been identified, the connectivity function $\lambda$ can be computed based on linear algebra alone [14], i.e. we have:

$$\lambda(A) = \lambda(A \cap V),$$

where $V$ is the set of reactions with variable flux. Furthermore, if all reactions can carry variable flux, then $\lambda$ depends on the stoichiometric matrix alone. For $V = R$, this leads to the surprising result that [14]:

$$\lambda(A) = \text{rank}(S_A) + \text{rank}(S_{R \setminus A}) - \text{rank}(S_R).$$

Since the rank of a matrix can be efficiently computed, we can also compute $\lambda$ efficiently. For example, in the case of the Escherichia coli iAF1260 network and its sub-system annotations, we computed that glycolysis/gluconeogenesis has a connectivity of 15 and the citric acid cycle has a connectivity of 12. A complete list can be found in the Supplementary Material.

**Branch decomposition**

Although we can compute $\lambda(A)$ efficiently for a given $A$, we are still left with the problem of finding ‘interesting’ sets of reactions $A$ with low $\lambda(A)$. In particular, it is not really clear what an ‘interesting’ set of reactions is. We observe by the properties mentioned above that there are many sets of reactions, where $\lambda(A)$ is low (for example if $A$ contains only one reaction or if $A$ contains all but one reaction), but which are clearly not interesting.

We conclude that we want to find sets of reactions $A$ where $A$ is large, the complement $R \setminus A$ is large and $\lambda(A)$ is low. However, if $A$ contains many reactions, we will be interested to understand $A$ more deeply. Hence, we want to be able to split $A$ recursively into smaller $k$-modules. This leads us to the concept of branch decompositions and branch width [17,18]. Branch width is related to the more popular concept of tree width, which is used to measure how tree-like a graph is. In contrast with tree width, branch-width has a natural extension to matroids and, thus, also to metabolic networks.

A branch-decomposition is a sub-cubic tree, i.e. all nodes have either degree 3 or they are leaves. Every leaf is uniquely associated to a reaction of the network. An example is shown in Figure 2. We observe that if we remove an edge $e$ of the tree, we get two connected components. Let $A$ be the set of reactions associated to the leaves of one of the connected components. We observe that $R \setminus A$ is the set of reactions associated to the leaves of the other connected component. Hence, we can annotate the edge $e$ with the value of the connectivity function $\lambda(A) = \lambda(R \setminus A)$. Note that, alternatively, we can do the same with the connectivity function $\mu$. Furthermore, we observe that by deleting the edge $e$, we get two rooted binary trees, each rooted at a vertex that was incident to $e$. These rooted binary trees give a straightforward rule on how to recursively split the $k$-module $A$ (respectively the $k$-module $R \setminus A$) into two smaller $k$-modules. In the example of Figure 2 the reaction set $A = \{r_{11}, r_{12}, r_{13}, r_{14}\}$ would be split into the reaction set $\{r_1, r_2, r_5, r_6, r_7, r_8\}$ with connectivity 2 and the fully coupled reactions $\{r_9, r_4\}$ with connectivity 1.

The largest value with which an edge of a branch-decomposition is annotated is called the branch width of the branch decomposition. For the example of Figure 1, we see in Figure 2 a branch-decomposition with branch width 4, because $A' = \{r_{15}, r_{12}, r_{13}, r_{14}\}$ is only a 4-module, i.e. $\lambda(A') = 4$. We can now turn the question of finding an interesting $k$-module into the question of finding a branch-decomposition with low branch width. In particular, we can consider a metabolic network modular if we can find a branch-decomposition with low branch width.

**Computing branch decompositions for metabolic networks**

Unfortunately, it is NP-hard to find a branch-decomposition with minimal branch width. There exist theoretical results on how to solve this problem in polynomial time for low branch-width instances [19,20]. However, these algorithms are not
Figure 3 | Excerpt of branch decomposition computed using a variant of Poolman’s method [22] for the *E. coli* core network

All non-dashed leaves belong to the pentose phosphate pathway.

---

likely to be practical, which is why we need to use heuristics. Heuristics have been developed by Ma et al. [21]. Core idea is to compute a similarity matrix by computing the similarity for each pair of reactions. Interestingly, Ma et al. [21] use a similarity measure, which is closely related to the one introduced by Poolman et al. [22]. Indeed, the decomposition computed by Poolman et al. [22] is a branch decomposition. In Figure 3 an excerpt of the branch decomposition computed for an *E. coli* core network [23] is shown. The full version can be found in the Supplementary Material.

In Table 1 we have listed upper bounds on the branch widths for a set of genome-scale metabolic networks computed using variants of Poolman’s method. The methods for computing the branch decompositions are described in the Supplementary Material and they are implemented in the cbmpy toolbox (cbmpy.sourceforge.net).

**Application: EFM enumeration**

In the study by Reimers and Stougie [14], we have shown that the set of EFM can theoretically be efficiently enumerated if the branch width of the network is low. To be more precise, we show that given a branch decomposition of branch-width $k$, we can enumerate all EFM in time

$$O(|R||EFM|^{2k+2}t)$$

where $t$ is the time needed to solve a linear program (LP) and $|EFM|$ is the number of EFM [14]. Although this is the first result that shows that EFM can be enumerated in total polynomial time, it unfortunately is not very practical due to the $|EFM|^{2k+2}$ term.

---

### Table 1 | Upper bounds on branch-width for some genome-scale metabolic networks

<table>
<thead>
<tr>
<th>Network</th>
<th>Reactions</th>
<th>Branch width</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>E. coli</em> core</td>
<td>95</td>
<td>13</td>
</tr>
<tr>
<td><em>E. coli</em> iJR904</td>
<td>1075</td>
<td>40</td>
</tr>
<tr>
<td><em>E. coli</em> iAF1260</td>
<td>2382</td>
<td>59</td>
</tr>
<tr>
<td>Helicobacter pylori iIT341</td>
<td>554</td>
<td>26</td>
</tr>
<tr>
<td>Homo sapiens recon 1</td>
<td>3742</td>
<td>99</td>
</tr>
<tr>
<td>H. sapiens recon 2</td>
<td>7440</td>
<td>146</td>
</tr>
<tr>
<td>Methanosarcina barkeri iAF692</td>
<td>690</td>
<td>29</td>
</tr>
<tr>
<td>Mycobacterium tuberculosis INH661</td>
<td>1025</td>
<td>35</td>
</tr>
<tr>
<td>Staphylococcus aureus iSB619</td>
<td>743</td>
<td>39</td>
</tr>
<tr>
<td>Saccharomyces cerevisiae ND750</td>
<td>1266</td>
<td>53</td>
</tr>
</tbody>
</table>
Without going into details (we refer to Reimers and Stougie’s work [14] for that), the idea is to enumerate recursively pathways that correspond to EFM through the $k$-modules in the branch decomposition, i.e. the pathways of a $k$-module $C$, which is split into two $k$-modules $A$ and $B$, can be computed from the pathways through $A$ and $B$. The bad runtime bound arises from the fact that it is hard to bind the number of pathways through each $k$-module well enough.

**Conclusion**

In this article we have shown how we can use $k$-modules to measure how well connected sub-systems are to the rest of the network. Whereas this measure is similar to counting the number of metabolites on the boundary, it is stoichiometry-based and hence it smoothly deals with redundancies due to coupled metabolites.

By recursively decomposing a network using branch decompositions, $k$-modules give us a measure of modularity. Unfortunately, it is very hard to compute the best branch decomposition. Therefore, we use heuristics, such as the method by Poolman et al. [22]. Although this method gives us a branch decomposition from which we can recognize familiar sub-networks, its branch width is not very small. In addition, the connectivity of many subsystems is so high compared with their size. Therefore, we conclude that the high branch width computed by our algorithm is probably not due to its lack of a quality guarantee, but because metabolic networks are not very modular (in the $k$-modules sense).

**Acknowledgements**

I thank Timo Maarleveld, Frank Bruggeman, Brett Olivier, Marie-France Sagot and Leen Stougie for insightful discussions.

**Funding**

This work was supported by the European Research Consortium for Informatics and Mathematics through an Alain Bensoussan Fellowship.

**References**


Received 22 June 2015
doi:10.1042/BST20150143