Institut für Mathematik

AG Mathematics in Life Sciences

Dissertation

zur Erlangung des Grades Dr. rer. nat.

Algorithms for the constraint-based analysis of metabolic networks

Eingereicht von: László Dávid

Betreuer: Prof. Dr. Alexander Bockmayr

September 2014

Erstgutachter:	Prof. Dr. Alexander Bockmayr
Zweitgutachter:	Prof. Dr. Ernst Althaus
Tag der Disputation:	17 Dezember 2014

To my loving wife.

Abstract

Constraint-based analysis of genome-scale metabolic networks has become increasingly important for describing and predicting the cellular behavior of living organisms. The steady-state constraints provide a reliable framework without the need for additional kinetic details of the system. As the number of metabolic network reconstructions and their level of detail continually increases, many computational tools for their analysis become unpractical to use. This invites for more efficient algorithms and tools for the analysis of metabolic networks.

We have a two-fold aim with this work. On one hand side, our goal is to design new algorithms that improve the efficiency of some of the existing methods. Secondly, we aim to create additional tools that fill in gaps in our toolbox for metabolic network analysis. These methods will provide additional insight into the structure of metabolic networks and ultimately broaden our understanding of cellular systems.

In the first part of the thesis we focus on improving flux coupling analysis (FCA). We prove that solving certain linear programs to feasibilty is sufficient to correctly deduce most coupling information. As a result of this and other refinements we design the FFCA algorithm that improves the efficiency of existing algorithms in the literature. FFCA is further developed by proving that all fully coupled reactions can be computed algebraically, without the need to solve linear programs (LP). Additionally, we show how utilizing the transitive nature of the coupling relations and reusing existing LP solutions can dramatically decrease running-time. Using these improvements we create the F2C2 algorithm, which is orders of magnitude faster than FFCA.

Traditionally FCA is performed on the unconstrained steady-state flux space, hence the FCA relations are generic to the underlying network. In order to derive additional coupling information from the metabolic networks, we extend the concepts of FCA to the constrained flux space, where any number of additional linear constraints can be imposed on the reactions. Constrained flux coupling analysis (CFCA) is proven to reveal coupling information that is only visible under special environmental conditions. We study the relationship between the FCA and CFCA relations and present an efficient algorithm to compute the latter.

In our next effort we study whether relations similar to FCA could also be applied for metabolites. We introduce the concept of metabolic activity coupling (MAC), which finds implicative relations between the momentary production and consumption of different metabolites.

Elementary flux modes (EM) are a canonical representation of the steady-state flux space and are important for the structural analysis of metabolic networks. The complete enumeration of EMs is closely related to the enumer-

ation of vertices of polyhedra, thus a hard problem. Our novel method for finding EMs containing several predefined reactions can be used to identify pathways that synthesize a desired target from one or more source metabolites, and also to compute EMs that cross through several predefined intermediary reactions. While the problem solved belongs to the class of NP-hard problems, we show that for current-generation networks the method is still applicable in practice.

EMs are most importantly used as a steady-state chain of reactions between sources and products. Without the manual analysis of individual EMs, little is known about their interior topology. For this purpose, we introduce and explore several concepts of ordering for the set of reactions of an EM. The purpose of these concepts is two-fold as they aid not only in the visual representation, but also in providing additional insight into the structure of the analyzed EM. We present graph-theoretic algorithms with which the presented notions can be computed efficiently.

To deal with the combinatorial complexity of an exhaustive EM enumeration, a trend in metabolic network analysis is the study of isolated subnetworks of the system. We show that such methods may result in undesired artifacts, and as an alternative solution, we present a method that aims at projecting the flux space onto lower dimension, while preserving key features of the network. We show that in certain cases such methods can be applied in situations where an exhaustive EM enumeration would otherwise fail.

Zusammenfassung

Constraint-basierte Analyse von genomweiten metabolischen Netzwerken ist zunehmend immer wichtiger geworden, um das Zellverhalten von Lebenswesen zu beschreiben und vorherzusagen. Die sogenannten 'steady- state' Bedingungen bieten einen verlässlichen Rahmen, ohne zusätzliche kinetische Einzelheiten des Systems zu brauchen. Da die Anzahl des Wiederaufbaus des Stoffwechselnetzwerks und deren Detaillierungsgrad fortwährend steigt, sind viele Rechenmitteln für deren Analyse ungeeignet. Dies gibt Anlass für effizientere Algorithmen und Mitteln für die Analyse von metabolischen Netzwerken.

Wir haben mit dieser Arbeit ein zweifaches Ziel. Auf der einen Seite ist es unser Ziel neue Algorithmen zu gestalten, die die Laufzeit von einigen bestehenden Verfahren verbessern. Auf der anderen Seite streben wir an zusätzliche Werkzeuge zu erstellen, die die Lücken in unserem Werkzeugkasten für Stoffwechselnetzwerkanalyse erfüllen. Diese Verfahren werden zusätzlichen Einblick in die Struktur des metabolischen Netzes geben und schließlich unser Verständnis der zellulären Systeme erweitern.

Im ersten Teil der Arbeit konzentrieren wir uns darauf, die Flusskopplungsanalyse (FCA) zu verbessern. Wir beweisen, dass es ausreichend ist, bestimmte lineare Programme zur Ausführbarkeit zu lösen, um die meisten Verkopplungsinformationen richtig herzuleiten. Als Ergebnis dieser und anderer Verfeinerungen entwickeln wir den FFCA Algorithmus, der die Effizienz der in der Literatur bestehenden Algorithmen verbessert. FFCA wird weiter entwickelt, indem wir beweisen, dass alle vollständig gekoppelten Reaktionen algebraisch berechnet werden können, ohne das lineare Programme (LP) gelöst werden müssen. Zusätzlich zeigen wir, wie durch die Anwendung der Transitivität der Koppelbeziehungen und die Wiederverwendung der bestehenden LP, die Laufzeit sich drastisch vermindern kann. Indem wir diese Verbesserungen verwenden, schaffen wir den F2C2 Algorithmus, der um Größenordnungen schneller als FFCA ist.

Herkömmlich wird FCA auf der uneingeschränkten stillstehenden flux space durchgeführt, demzufolge sind die FCA Beziehungen auf dem zugrunde liegenden Netzwerk typisch. Um zusätzliche Kopplungsinformationen aus den metabolischen Netzwerken ableiten zu können, erweitern wir die Vorstellungen der FCA auf die eingeschränkte flux space, in welchem eine beliebige Anzahl von zusätzlichen linearen Einschränkungen auf die Reaktionen zur Folge haben kann. Eingeschränkte Flusskopplung Analyse (CFCA) ist erwiesen, um die Kopplungsinformationen zu offenzulegen, die nur unter besonderen Umweltbedingungen sichtbar ist. Wir untersuchen die Beziehung zwischen den FCA und CFCA Beziehungen und stellen einen effizienten Algorithmus dar, um letztere zu errechnen.

In unserem nächsten Versuch untersuchen wir, ob Beziehungen, die ähn-

lich der FCA sind, auch für Metaboliten angewendet werden könnten. Wir stellen das Konzept der Stoffwechselaktivität Kopplung (MAC) vor, das implicative Beziehungen zwischen der derzeitigen Herstellung und Verbrauch von verschiedenen Metaboliten findet.

Elementare Flussmoden (EM) sind eine anerkannte Darstellung der 'steady-state' Flussraum und bedeutend für die Strukturanalyse von metabolischen Netzwerken bedeutend. Die vollständige Zählung der EM-en ist eng mit der Zählung von Eckpunkte der Polyeder verbunden und somit ein schwieriges Problem. Unsere neue Methode, um EM-en, die mehrere vordefinierten Reaktionen beinhalten, zu finden, kann auch verwendet werden, um Wege, die ein gewünschtes Ziel aus einem oder mehreren Quellmetaboliten zu synthetisieren, zu identifizieren, und auch EM-en zu berechnen, die sich durch mehrere vordefinierte Vermittlerreaktionen kreuzen. Während das gelöste Problem zur Art der NP- harten Probleme gehört, zeigen wir, dass für die current-generation Netzwerke das Verfahren in der Praxis noch anwendbar ist.

EM-en werden vor allem als Kette von Reaktionen zwischen Quellen und Produkten verwendet. Ohne die manuelle Analyse zu den einzelnen EM-en ist unser Wissen über ihre innere Topologie begrenzt. Zu diesem Zweck stellen wir mehrere Konzepte für den Abruf dieser Reihe von Reaktionen eines EM vor und erforschen diese. Der Zweck dieser Konzepte ist zweifach, da sie nicht nur in der visuellen Darstellung helfen, sondern auch zusätzliche Einsicht in die Struktur des analysierten EM geben. Wir stellen graphentheoretische Algorithmen dar, mit denen die vorgestellten Entwürfe effizient berechnen werden können.

Um mit der kombinatorischen Komplexität einer erschöpfenden EM Aufzählung umzugehen, ist es bei der Stoffwechselnetzwerkanalyse die Tendenz, die isolierten Teilnetze des Systems zu untersuchen. Wir zeigen, dass solche Methoden zu unerwünschten Artefakten führen können und als eine alternative Lösung präsentieren wir einen Weg, der anstrebt, den Flussraum auf die untere Dimension zu projektieren, während die wichtigsten Features des Netzwerks beibehalten werden. Wir zeigen, dass in bestimmten Fällen solche Verfahren in Situationen eingesetzt werden können, wo eine erschöpfende EM Aufzählung sonst scheitern würde.

Acknowledgments

First and foremost I would like to express my deepest heartfelt gratitude to my supervisor, Alexander Bockmayr, for accepting me into his workgroup and giving me the opportunity to pursue my PhD studies in the field of Systems Biology. He always allowed me the freedom to choose the research areas which interested me most, making the research process extremely enjoyable. I am also grateful for the unlimited time he committed to our discussions as well as proofreading my papers and thesis. Without his continuous guidance and encouragement this work would not have been completed.

I would also like to thank Ernst Althaus for accepting to read and evaluate my thesis.

I am grateful to Heike Siebert and Andreas Paffenholz for allowing me to tutor their courses, which helped me experience a different aspect of the Ph.D. life. Heike's work ethic and always positive attitude was very inspiring.

My stay in Berlin was possible in part thanks to the Berlin Mathematical School (BMS). Receiving a BMS-stipend during my first two years in Berlin has allowed me to fully focus on my studies. I would like to thank the BMS One-Stop office for their continuous assistance.

My colleagues whom I shared an office with, Sayed-Amir Marashi, Ling Sun and Marco Rügen contributed greatly to the friendly environment I was lucky to work in. The random coffee breaks, the discussions covering all aspects of life, the occasional chess games and table-tennis matches are all memories which I am very fond of. Moreover, Amir with his deep biological knowledge helped me so much in giving meaning to abstract biological concepts. Our discussions led numerous times to coauthored publications.

I would furthermore like to thank Abdelhalim Larhlimi, Yaron Goldstein and Arne Reimers for the many discussions relating to my research topic. With Halim on two occasions this also led to coauthored publications. The proof of Prop. 5.5 is based on ideas from Yaron.

I also thank the rest of the workgroup members, namely Katja Geiger, Shahrad Jamshidi, Hannes Klarner, Corinna Heldt, Aljoscha Palinkas, Alexandra Grigore, Kirsten Thobe, Firdevs Topcu-Alici and Adam Streck for creating a welcoming environment.

A big thank you goes to my friend, Máthé Jenő, who proofread various chapters of the thesis.

I also want to express my deep appreciation to my parents and two younger sisters for their unconditional love and for always motivating me to study. They represented a safe harbor in stressful times.

Above all, I want to thank Gyöngyi, my beloved wife, for her continuous support, for her patience and her love. She gave me the strength to finish this thesis, therefore this work is dedicated to her.

Contents

Chapter 1

Introduction

Contents

1.1 Motivation

Of all the existing matter, living organisms are arguably the most interesting ones to study. Systems biology allows researching them with mathematic rigour. Advances in molecular biology allowed a more precise reconstruction of biological systems, thereby increasing the popularity of systems biology. An integral part of systems biology is the study of metabolic systems.

There are many rewards and benefits associated with the proper understanding of the metabolism of living cells. It has the potential to increase our quality of life via the design and development of new, more efficient, and less harmful drugs. It can have a positive impact on economy by allowing the production of key strategic resources, such as ethanol, propanol or (bio)butanol. These resources may be produced cheaper and in a higher quantity than with traditional methods. Last but not least it serves the purpose of elucidating the understanding of our surrounding, feeding a drop to the everlasting human thirst for knowledge.

1.2 Mathematical notations and preliminaries

In this section we present the essential mathematical concepts used in the main chapters of the thesis.

1.2.1 Linear algebra and set theory

We denote with $\mathbb{N}$ the set of natural number, with $\mathbb{Z}$ the set of integers and with $\mathbb{R}$ the set of real numbers. $\mathbb{R}_+$ represents the non-negative halfspace, while $\mathbb{R}^n$ represents the n-dimensional vector space over $\mathbb{R}$. Any vector $v \in \mathbb{R}^n$ is a column vector, while v^t represents the transposed row vector. We reference the i^{th} coordinate (where $i \in \{1, ..., n\}$) of vector v as v_i.

Let $C \subseteq \mathbb{R}^n$. The orthogonal space of C is defined as $C^\perp := \{x \in \mathbb{R}^n \mid \forall y \in C, \sum_{i=1}^n y_i x_i = 0\}$.

The support of a vector is the set of indices (or coordinates) in which the vector differs from zero.

Definition 1.1. *Let $v \in \mathbb{R}^n$. The support of v is $supp(v) = \{i \in \{1, \ldots, n\} \mid v_i \neq 0\}$.*

Let S be a set with n elements (i.e., $|S| = n$) and let $I \subseteq \{1, ..., n\}$ with $k := |I| \leq n$. Then S_I will denote the subset of S consisting of elements $\{s_{i_1}, s_{i_2}, ..., s_{i_k}\}$. We will use a similar notation for vectors. Let $v \in \mathbb{R}^n$ and $I \subseteq \{1, ..., n\}$ with $k := |I| \leq n$ and $k > 0$. Then $v_I \in \mathbb{R}^k$ with $v_I := (v_{i_1}, v_{i_2}, ..., v_{i_k})$.

The indexing of matrices is defined as follows. Let $A \in \mathbb{R}^{m \times n}$ be a matrix and $i, j \in \mathbb{N}$ with $0 < i \leq m$ and $0 < j \leq n$. Then A_{ij} represents the entry of the matrix corresponding to the i^{th} row and j^{th} column. For $I \subseteq \{1, 2, ..., m\}$ and $I \subseteq \{1, 2, ..., n\}$ with $0 < k = |I| \leq m$ and $0 < l = |J| \leq n$, the matrix A_{IJ} corresponds to the submatrix of A where rows in I and columns in J have been selected (i.e., $(A_{IJ})_{uv} = A_{i_u j_v}$).

Definition 1.2 (Linear, affine, conic and convex combination)**.** *Let $\{v^1, v^2, ..., v^k\} \subset \mathbb{R}^n$. Then*

- *$x := \sum_{i=1}^k \lambda_i v^i$ for some $\lambda_1, \lambda_2, ..., \lambda_k \in \mathbb{R}$ is a linear combination of the vectors.*
- *$x := \sum_{i=1}^k \lambda_i v^i$ for some $\lambda_1, \lambda_2, ..., \lambda_k \geq 0$ is a conic combination of the vectors.*
- *$x := \sum_{i=1}^k \lambda_i v^i$ for some $\lambda_1, \lambda_2, ..., \lambda_k \in \mathbb{R}$ with $\sum_{i=1}^k \lambda_i = 1$ is an affine combination of the vectors.*
- *$x := \sum_{i=1}^k \lambda_i v^i$ for some $\lambda_1, \lambda_2, ..., \lambda_k \geq 0$ with $\sum_{i=1}^k \lambda_i = 1$ is a convex combination of the vectors.*

Definition 1.3 (Linear independence). *A set of vectors $\{v^1, v^2, ..., v^k\} \subset \mathbb{R}^n$ are linearily independent if $0 \in \mathbb{R}^n$ cannot be written up as their linear combination, unless $\lambda_1 = \lambda_2 = ... = 0$.*

Definition 1.4 (Linear subspace). *$S \subseteq \mathbb{R}^n$ is a linear subspace if the following two conditions hold:*

- *if $u, w \in S$ then also $u + w \in S$.*
- *if $u \in S$ and $c \in \mathbb{R}$ then also $c \cdot u \in S$.*

Definition 1.5 (Kernel). *Let $A \in \mathbb{R}^{m \times n}$ be a matrix. The kernel (or null space) of A is defined as $kern(A) := \{x \in \mathbb{R}^n \mid Ax = 0\}$. The kernel of A is a linear subspace of $\mathbb{R}^n$.*

Definition 1.6 (Basis). *Let $S \subseteq \mathbb{R}^n$. A set of vectors $B := \{b^1, b^2, ..., b^k\} \subset \mathbb{R}^n$ is a basis of S if and only if the following conditions hold:*

- *For any $v \in S$, v can be written as a linear combination of the vectors in B.*
- *The vectors in B are linearily independent.*

Definition 1.7 (Dimension). *Let $S \subseteq \mathbb{R}^n$ and B a basis of S. Then the dimension of S is $dim(S) := |B|$.*

In the following we recall a few notions from set theory.

Definition 1.8 (Relation). *Let X, Y be two sets, and $G \subseteq X \times Y$. The triple $R := (X, Y, G)$ is a relation. If $(x, y) \in G$ then we say that 'x is related to y' and the shortened notation xRy can be used. If $X = Y$ we will simply say that R is a relation over X.*

Definition 1.9 (Reflexivity). *Let R be a relation over X. R is reflexive if for all $x \in X$, xRx holds.*

Definition 1.10 (Irreflexivity). *Let R be a relation over X. R is irreflexive if for all $x \in X$, xRx is false.*

Definition 1.11 (Transitivity). *Let R be a relation over X. R is transitive if for all $x, y, z \in X$ with xRy and yRz, then xRz.*

Definition 1.12 (Antisymmetry). *Let R be a relation over X. R is antisymmetric if for all $x, y \in X$ if xRy and yRx, then $x = y$.*

Definition 1.13 (Symmetry). *Let R be a relation over X. R is symmetric if for all $x, y \in X$, xRy implies yRx.*

Definition 1.14 (Totality). *Let R be a relation over X. R is total or complete if for all $x, y \in X$, either xRy or yRx (or both) holds.*

Definition 1.15 (Partial order). *A relation R over X is said to be a partial ordering relation if it is reflexive, transitive and antisymmetric. X is said to be partially ordered under R.*

Definition 1.16 (Total order). *A relation R over X is a total ordering relation if it is a partial ordering relation and in addition R is total.*

1.2.2 Polyhedral theory

Definition 1.17 (Convex cone). *Let $C \subseteq \mathbb{R}^n$ with $C \neq \emptyset$. C is a convex cone if for all $x, y \in C$, any conic combination of x and y is an element of C.*

Definition 1.18 (Equivalent rays). *Let C be a convex cone. A non-zero element $x \in C$ is called a ray. Two rays $x, y \in C$ are equivalent, written $x \cong y$, if $x = \lambda y$, for some $\lambda > 0$.*

Definition 1.19 (Extreme ray). *A ray x in C is extreme if there do not exist rays $y, z \in C, y \ncong z$ such that $x = y + z$.*

Definition 1.20 (Polyhedron). *A set $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ with $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ is a (convex) polyhedron.*

Definition 1.21 (Polyhedral cone). *A set $C = \{x \in \mathbb{R}^n \mid Ax \leq 0\}$ with $A \in \mathbb{R}^{m \times n}$ is a polyhedral cone.*

Therefore, a polyhedral cone is a special type of polyhedron, where the right-hand side of the inequality constraints is $0 \in \mathbb{R}^m$.

Definition 1.22 (Lineality space). *Let $C = \{x \in \mathbb{R}^n \mid Ax \leq 0\}$ be a polyhedral cone. The lineality space of C is $lin.space(C) = \{x \in \mathbb{R}^n \mid Ax = 0\}$.*

Definition 1.23 (Pointed cone). *A polyhedral cone C is pointed if $lin.space(C) = \{0\}$.*

Definition 1.24 (Finitely generated cone). *A convex cone C is finitely generated if there exist vectors $g^1, g^2, ..., g^k$, called generators, such that $C = \{\lambda_1 g^1 + ... + \lambda_k g^k \mid \lambda_1, ..., \lambda_k \geq 0\}$.*

Theorem 1.25 (Farkas-Minkowski-Weyl). *A convex cone is polyhedral if and only if it is finetely generated.*

Definition 1.26 (Face). *Let P be a polyhedron and $F \subseteq P$. F is called a face of P if $F = P$ or $F = P \cap \{x \in \mathbb{R}^n \mid a^t x = b\}$ and $P \subseteq \{x \in \mathbb{R}^n \mid a^t x \geq b\}$*

Definition 1.27 (Minimal face). *Let P be a polyhedron and $F \neq \emptyset$ a face. F is a minimal face if there is no face $F' \neq \emptyset$ of P with $F' \subsetneq F$.*

Definition 1.28 (Dimension). *Let C be a polyhedral cone and F a face. The dimension of F is equal to the dimension of its linear span, i.e., $dim(F) = dim(\{x \in \mathbb{R}^n \mid x$ can be written up as the linear combination of elements in $F\})$.*

Definition 1.29 (Minimal proper face). *Let C be a polyhedral cone and $F \neq \emptyset$ a face of C. F is a minimal proper face if $dim(F) = dim(lin.space(C)) + 1$.*

Proposition 1.30 (Farkas' Lemma). *Let $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Then exactly one of the following is true:*

- *There exists $x \in \mathbb{R}^n$ such that $Ax = b$ and $x \geq 0$.*
- *There exists $y \in \mathbb{R}^m$ such that $y^t A \geq 0$ and $y^t b < 0$.*

For a proof of the Farkas' Lemma we refer the reader to [100].

1.2.3 Linear and mixed integer programming

Definition 1.31 (Linear program). *Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$. The problem of computing $argmin/argmax\{c^t x \mid Ax \leq b\}$ is called a linear program (LP). We will use the shortened notation of $min/max\{c^t x : Ax \leq b\}$*

Definition 1.32 (Feasible solution). *Let $max\{c^t x : Ax \leq b\}$ be a linear program. Then $x' \in \mathbb{R}^n$ is a feasible solution if $Ax' \leq b$ holds.*

Definition 1.33 (Optimal solution). *Let $max\{c^t x : Ax \leq b\}$ be a linear program. Then a feasible solution $x^* \in \mathbb{R}^n$ is an optimal solution if for all feasible solutions $x' \in \mathbb{R}$, $c^t x^* \geq c^t x'$ holds.*

A linear programming problem can be solved to optimality using the Ellipsoid-method in polynomial time [100], although this is very slow when implemented in practice. The fastest LP solving algorithms employ Simplex-based or Interior point methods [100].

Definition 1.34 (Mixed integer program). *A linear programming problem with the additional constraint that some or all variables need to belong to the set of integers ($\mathbb{Z}$) is called a mixed integer (linear) program (MIP).*

The computational complexity of solving MIPs is NP-hard [36]. However, there are a number of tools that use branch-and-cut algorithms together with different heuristics that are able to solve MIP problems 'efficiently' in practice (i.e., Gurobi [41]). For further reference on linear and mixed integer programming we recommend [100].

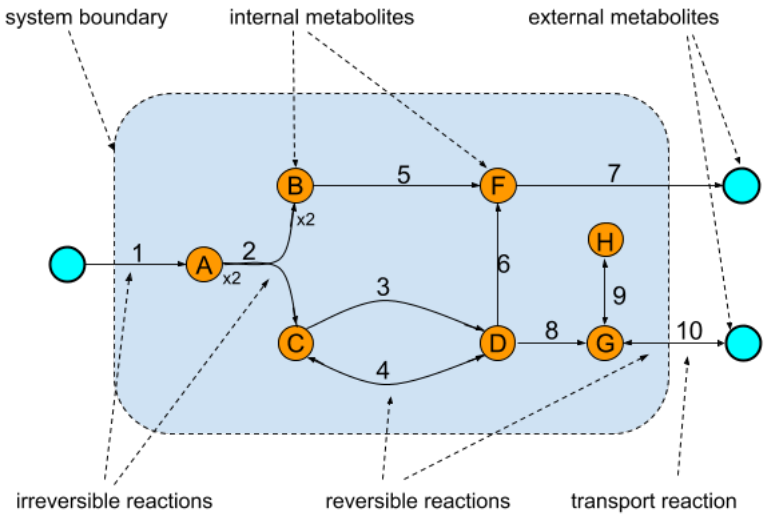


Figure 1.1: Ingredients of a metabolic network model.

1.3 An overview of metabolic networks

Metabolic networks describe a collection of biochemical reactions that occur on a cellular level, where most of the reactions are enzymatic in nature (i.e., catalyzed by enzymes). Each metabolic reaction acts on metabolites in a way that converts one or more reactant metabolites into one ore more product metabolites. Reactions that involve common metabolites are adjacent, therefore forming an interconnected network. Figure 1.1 illustrates an example metabolic network with its typical components:

- The *system boundary* is represented by the dashed rectangle. The objects within the system boundary are considered to be *internal*, while the ones outside it are consider to be *external* to the modeled cell.

- *Metabolites* are denoted by the circles in the figure. We classify metabolites as either *external* (blue) or *internal* (orange) to the system. Typically, we are interested in the metabolites internal to the system, thus the external metabolites are only drawn symbolically for completeness of the figure. Formally, for a system with m metabolites we denote with $\mathcal{M} \subset \mathbb{N}$, where $|\mathcal{M}| = m$, the total set of internal metabolites. Typically $\mathcal{M} = \{1, 2, ..., m\}$.

- *Reactions* are the directed hyper-edges in the drawing, with the arrows pointing in the direction of the metabolites produced by the reaction. We distinguish *internal* reactions (i.e., those which are only adjacent to internal metabolites) from *transport* reactions (i.e, those which are

adjacent to both internal and external metabolites). A transport reaction is either producing internal metabolites or consuming them, but not both.

Definition 1.35 (Metabolic reaction)**.** *Let $\mathcal{M}$ be the set of internal metabolites and let $X, Y \subseteq \mathcal{M}$. Then $r := (X, Y)$ is a metabolic reaction in the system if and only if $X \cup Y \neq \emptyset$ and $X \cap Y = \emptyset$. X is called the set of reactants and Y is the set of products.*

Assuming the metabolic network contains n metabolic reactions, we denote the total set of reactions with $\mathcal{R} = \{r_1, r_2, ..., r_n\}$. For simplicity and convenience, we will refer to 'reaction r_i' as 'reaction i' in cases when this does not introduce ambiguity. In fact, the set of reactions will be often denoted as $\mathcal{R} = \{1, 2, ..., n\}$.

Throughout the thesis n and m represent natural numbers, therefore the sets $\mathcal{M}$ and $\mathcal{R}$ are finite.

The relationship between a reaction and its incident metabolites is furthermore characterized by the *stoichiometric coefficient*. This is a scalar number, representing the relative amount produced or consumed from the given metabolite by the reaction in relation to the other metabolites adjacent to the same reaction. For example, looking at reaction '2' in the previous figure, it consumes two 'A' metabolites while it produces two 'B' metabolites and a 'C' metabolite. Therefore the stoichiometric coefficient of reaction '2' in metabolite A is -2, in metabolite B it is 2, while in metabolite 'C' it is 1. The general consenus is that unitary stoichiometric values are not explicitly represented in figures and we will use this notation throughout the figures of the thesis. If a reaction does not consume nor produce a metabolite then its stoichiometric coefficient is zero.

In order to show the importance of the stoichiometric coefficients consider the following example. Assume that metabolites 'A', 'B' and 'C' correspond to the chemical species' H_2O, H_2 and O_2. Then reaction 2 simply denotes the well known reaction $2H_2O \rightarrow 2H_2 + O_2$.

We note that by Def. 1.35 no reaction represented in a metabolic model will both consume and produce the same metabolite. If there was such a reaction, it could be represented with a simpler reaction by considering the net difference of its sources and products. For example, let us assume we had the following reaction $2A + B \rightarrow A + C$. Then the simplified reaction $A + B \rightarrow C$ would result in the same effect on the concentration levels of metabolites 'A', 'B' and 'C' as the previous one.

We can write all the stoichiometric coefficients of a metabolic network in a concise form by considering the *stoichiometric matrix S*.

Definition 1.36 (Stoichiometric matrix)**.** *Let $S \in \mathbb{R}^{m \times n}$. Then S is the stoichiometric matrix of the system if for all $i \in \{1, ..., m\}$ and for all $j \in$*

$\{1, ..., n\}$, *s_{ij}, representing the entry in the i^{th} row and j^{th} column is equal to the stoichiometric coefficient of metabolite 'i' in reaction 'j'.*

The stoichiometric matrix is very similar to the incidence matrix of graphs. For the metabolic network in Figure 1.1, its stoichiometric matrix is summarized in 1.1.

$$S = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array} \begin{array}{c} \begin{array}{cccccccccc} r_1 & r_2 & r_3 & r_4 & r_5 & r_6 & r_7 & r_8 & r_9 & r_{10} \end{array} \\ \begin{pmatrix} 1 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} \end{array} \tag{1.1}$$

Notice that in S the columns corresponding to the transport reactions are always non-negative or non-positive, while columns corresponding to internal reactions contain both positive and negative entries. This property holds in general and it is a convenient way of finding the transport reactions from the stoichiometric matrix.

A second property by which reactions can be classified is their *irreversibilty*. Due to thermodynamic properties, some reactions are termed *irreversible* and they can only function in their specified direction. Opposed to this are the *reversible* reactions which can function in either directions. In Figure 1.1 there are three reversible reactions, reactions 4, 9 and 10. They can be recognized from the figure as the ones that have an arrowhead pointing towards each incident metabolite. In the stoichiometric matrix, columns corresponding to reversible reactions can be multiplied by -1 without actually changing the topology of the network. In order to avoid confusion, for reversible reactions we will use the term *forward direction* to refer to the direction that is specified in the stoichiometric matrix, while the *reverse direction* is the opposite sense.

Definition 1.37 (The set of irreversible reactions)**.** *The set of irreversible reactions is denoted with $Irr \subseteq \mathcal{R}$.*

Naturally, the set of reversible reactions can be deduced by considering the set difference $Rev := \mathcal{R} - Irr$.

1.3.1 Metabolic network modeling

There are several different ways in which metabolic networks can be modeled mathematically. In *dynamic modeling*, each metabolite is modeled by an ordi-

nary differential equation (ODE). If we denote with $C(t) : \mathbb{R} \to \mathbb{R}^{\mathcal{M}}$ the concentration function associated with the metabolites and with $v(t) : \mathbb{R} \to \mathbb{R}^{\mathcal{R}}$ the flux rates associated with the reactions, then we can describe the changes in concentration for every metabolite as summarized in equation 1.2.

$$\frac{d}{dt}C(t) = S \cdot v(t) \tag{1.2}$$

Assuming knowledge of the enzyme-kinetic rate laws (called also kinetic parameters), as well as the initial concentration levels of every metabolite, it is possible model the evolution and to predict the state of the system at any given time. Unfortunately, in practice both sets of information are very hard to measure and it is available only for a limited number of reactions [6]. Approaches have been developed that work with partial kinetic data [47, 112].

Stochastic modeling can be used to cope with non-deterministic properties of biological networks [37]. In [69] the authors use Stochastic Logic Programming to estimate the flux rates of reactions in metabolic pathways.

Within the framework of this thesis we will use *constraint-based modeling* (CBM), which gives up the idea of modeling the exact dynamic behavior of the underlying system [20, 88]. Rather than finding the exact trajectory of concentration changes and flux distributions, constraint-based methods limit the set of possible behaviors according to our current level of knowledge about the system. These limitations on the set of behaviors are imposed in the form of constraints. The more we know about the system, i.e., the more constraints we add, the narrower the total set of potential behaviours will be. In spite of partial knowledge about metabolic systems, constraint-based methods have been used with success to a great extent [9].

In CBM there are two very frequent sets of constraints which are typically imposed on the system. These sets of constraints originate from the observation that in general metabolic reactions occur on a much faster timescale than the changes in the environment or within the system itself [88, 116]. This leads to the (quasi) *steady-state assumption* which assumes that the system will reach a steady-state, where concentrations and flux rates are constant (over a certain amount of time). The steady-state assumption has a direct consequence for Eq. 1.2, since its left term becomes zero. The resulting *stoichiometric constraints* are summarized in Eq. 1.3. Note that now $v \in \mathbb{R}^n$, i.e., we have an algebraic equation instead of an ODE.

$$0 = S \cdot v \tag{1.3}$$

A second set of constraints relates to the irreversible reactions. By definition, these reactions can only take place in their forward direction, hence any flux value associated with these must be non-negative [87]. The *thermodynamic constraints* in Eq. 1.4 formulate this observation mathematically.

$$v_i \geq 0, \forall i \in Irr \tag{1.4}$$

The stoichiometric constraints and thermodynamic constraints are commonly referred together to as *steady-state constraints*. Beside the steady-state constraints, sometimes additional knowledge is known about the system in form of reaction bounds, also known as *capacity constraints* [89]. When lower and upper bounds are known for a reaction r_i, then these can be simply imposed as in Eq. 1.5. Here $l_i \in \mathbb{R}$ is the lower bound and $u_i \in \mathbb{R}$ is the upper bound of the reaction.

$$u_i \geq v_i \geq l_i \tag{1.5}$$

As mentioned above the two most common constraints are the stoichiometric and thermodynamic ones. In order to impose these, one only needs to know the stoichiometric matrix and the set of the irreversible reactions associated with the system to be modeled. Hence we give the following definition for a metabolic network model.

Definition 1.38 (Metabolic network model)**.** *Let $\mathcal{M}$ be a set of m metabolies and $\mathcal{R}$ be a set of n reactions. Furthermore let $S \in \mathbb{R}^{m \times n}$ be a stoichiometric matrix and $Irr \subseteq \mathcal{R}$ the set of irreversible reactions. Then $\mathcal{N} := (S, Irr, \mathcal{M}, \mathcal{R})$ is a metabolic network model.*

Def. 1.38 gives a concise definition of a metabolic network model in the constraint-based modeling framework. In some cases when we define metabolic networks, we will omit the sets $\mathcal{M}$ and $\mathcal{R}$ from the definition. In specific, when the set $\mathcal{M}$ corresponds to $\{1, 2, ..., m\}$ and the set $\mathcal{R}$ corresponds to $\{1, 2, ..., n\}$ then we will simply define metabolic networks as $\mathcal{N} := (S, Irr)$.

Every subset of the reactions leads to a *metabolic subnetwork*.

Definition 1.39 (Metabolic subnetwork)**.** *Let $\mathcal{N} := (S, Irr, \mathcal{M}, \mathcal{R})$ be a metabolic network model and let $\mathcal{R}' \subseteq \mathcal{R}$ with $n' := |\mathcal{R}'|$. $N' = (S', Irr', \mathcal{M}, \mathcal{R}')$ with $S' := S_{\mathcal{M}\mathcal{R}'} \in \mathbb{R}^{m \times n'}$ and $Irr' = Irr \cap \mathcal{R}'$ is a metabolic subnetwork of $\mathcal{N}$.*

In practice, the stoichiometric matrix and the set of irreversible reactions belonging to a metabolic network is the result of a process called *metabolic network reconstruction* [28]. Advances in molecular biology as well as the existence of a wide range of bibliomic data allow for the reconstruction of genome-scale metabolic networks at a fast pace [28]. Several online databases have been created where full genome-scale reconstructions of different microorganisms are accessible. Such databases are BiGG [94] and BioCyc [14] just to name a few. Having access to these collections of metabolic networks is critical and invaluable when developing tools for the analysis of metabolic networks since they allow the testing of new tools on relevant biological data.

In the following subsections we introduce the main concepts commonly found in the constraint-based analysis of metabolic networks. Furthermore, we present a number of methods from the literature that were employed with great success.

1.3.2 The steady-state flux cone

In the following we consider a metabolic network $\mathcal{N} = (S, Irr)$ with m internal metabolites and n reactions. If steady-state conditions hold, i.e., there is no net production or consumption of internal metabolites, the set of all feasible flux distributions defines a polyhedral cone [61, 117].

Definition 1.40 (Steady-state flux cone)**.** *For a metabolic network $\mathcal{N} = (S, Irr)$, let $C := \{v \in \mathbb{R}^n \mid S \cdot v = 0, v_i \geq 0$ for all $i \in Irr\}$. C is called the (steady-state) flux cone.*

Definition 1.41 (Active reactions)**.** *Let $v \in C$. Then the reactions in $supp(v)$ are called the active reactions of v.*

It is important to note that the structure of a metabolic network may further constrain the reactions in the sense that some might never become active.

Definition 1.42 (Blocked reaction)**.** *Given the steady-state flux cone C, let $i \in \{1, \ldots, n\}$ be a reaction. If $v_i = 0$, for all $v \in C$, reaction i is called blocked, otherwise i is unblocked.*

Looking again at the example from Fig. 1.1, reaction 9 is actually a blocked reaction. Indeed, should the flux corresponding to reaction 9 be non-zero, it would result in violating the steady-state conditions for metabolite H.

From an algorithmic viewpoint computing all the blocked reactions in a metabolic network is considered an easy task. Indeed, assuming we are interested whether reaction r_i is blocked or not, consider the two LPs in Eq. 1.6.

$$\begin{aligned} o_i &= \min \ \{v_i \colon Sv = 0,\ v_{Irr} \geq 0\}, \\ O_i &= \max \ \{v_i \colon Sv = 0,\ v_{Irr} \geq 0\}. \end{aligned} \tag{1.6}$$

Then r_i is blocked if either one of the following two points holds:

- r_i is irreversible and $O_i = 0$.
- r_i is reversible and $O_i = o_i = 0$.

If a reaction r_i is given as reversible but only one of O_i, o_i is zero then r_i can essentially be treated as an irreverible reaction. In case o_i is the non-zero term it is necessary to multiply the column in S corresponding to the reaction by -1 before changing the reaction's reversibility type to irreversible.

Larhlimi and Bockmayr [64] show that the reversibility type of reactions is a key concept in the analysis of metabolic networks and further refine it by introducing the concepts of *fully reversible* and *pseudo-irreversible* reactions.

Definition 1.43 (Reversibility types)**.** *A reversible reaction $i \in Rev$ is called fully reversible if there exists a flux vector $v \in C$ such that $v_i \neq 0$ and $v_j = 0$ for all $j \in Irr$. Otherwise, reaction i is called pseudo-irreversible.*

Using the reversibility type of reactions, we can define the following reaction sets:

- $Frev = \{i \mid i \text{ is fully reversible}\}$,
- $Prev = \{i \mid i \text{ is pseudo-irreversible and there exist } v^+, v^- \in C \text{ such that } v_i^+ > 0, v_i^- < 0\}$,
- $Irev = \{i \mid i \notin Frev \cup Prev \text{ and } v_i \neq 0 \text{ for some } v \in C\}$,
- $Blk = \{i \mid i \text{ is blocked}\}$.

Note that the above reaction sets are disjoint and their union is equal to the set of all reactions, i.e., $Blk \cup Irev \cup Prev \cup Frev = \mathcal{R}$. Given a metabolic network one can compute the above defined four reaction sets in polynomial time (Additional file 1 in [24]).

The uncertainty of direction associated with reversible reactions is sometimes undesirable. In such cases a common way to deal with the problem is to replace the reversible reactions by two opposing irreversible reactions. With this transformation one can essentially get rid of all the reversible reactions. We will refer to this modified network where all the reversible reactions have been replaced as the *reconfigured network* and to the new flux cone as the *reconfigured flux cone*. This operation, however, also has some bad consequences. For one, the flux space increases in dimension, possibly doubling the dimension of the flux space. Moreover, it can hide certain properties of the network. In Chapter 2 we will discuss this aspect in more detail.

1.3.3 Elementary flux modes

The concept of elementary flux modes [105, 101, 104, 106, 84] was introduced as a tool to structurally characterize steady-state flux cones.

Definition 1.44 (Elementary flux mode)**.** *A flux vector $e \in C$ is called an elementary flux mode (EM) if there is no vector $v \in C \setminus \{0\}$ such that $supp(v) \subsetneq supp(e)$.*

According to Def. 1.45, each EM represents a minimal set of reactions that can work together in steady-state. An alternative but equivalent definition can be formulated as follows (proof in [106]).

Definition 1.45 (Elementary flux mode)**.** *A flux vector $e \in C$ is called an elementary flux mode if there are no vectors $e', e'' \in C \setminus \{0\}$ such that $e = e' + e''$, $supp(e') \subsetneq supp(e)$ and $supp(e'') \subsetneq supp(e)$.*

The concept of elementary modes is closely related to that of extreme rays [115].

Proposition 1.46. *Let $e^1, e^2 \in C$ be two elementary modes. If $supp(e^1) = supp(e^2)$, then $e^1 \cong e^2$ holds (i.e., the two rays associated with the elementary modes are equivalent).*

Proposition 1.47. *Let $v \in C \setminus \{0\}$ be a steady-state flux vector. Let $\mathcal{N}' = (S', Irr')$ be the subnetwork induced by the reactions in $supp(v)$. Then the following are equivalent:*

- *v is an elementary mode.*
- *$dim(kern(S')) = 1$;*

A proof for Prop. 1.46 and Prop. 1.47 can be found in [39]. The importance of Prop. 1.47 is in that it gives a method for checking the elementarity of any feasible flux vector v. One only needs to consider a subnetwork consisting of the active reactions of v and compute a basis for the kernel of S', the stoichiometric matrix of the subnetwork. Then v was an elementary mode if and only if the kernel of S' has a single generator.

In the following we denote with E the set of all pairwise non-equivalent EMs of the system $E = \{e^1, e^2, \ldots, e^s\}$.

Corollary 1.48. *The number of elementary modes in E in a metabolic network is finite.*

Proof. Since there are at most 2^n different supports and according to Prop. 1.46 for each support there corresponds at most one non-equivalent elementary mode, there are at most 2^n elementary modes. □

In [39], the author computes a tighter upper bound on the number of elementary modes using lattice theory.

In order to provide an example, we revisit the metabolic network from Fig. 1.1. In this network there is a total of 5 elementary modes, described below (each row in the matrix corresponds to one elmenetary mode).

$$E = \begin{array}{c} \\ e^1 \\ e^2 \\ e^3 \\ e^4 \\ e^5 \end{array} \begin{array}{c} \begin{array}{cccccccccc} r_1 & r_2 & r_3 & r_4 & r_5 & r_6 & r_7 & r_8 & r_9 & r_{10} \end{array} \\ \begin{pmatrix} 2 & 1 & 1 & 0 & 2 & 1 & 3 & 0 & 0 & 0 \\ 2 & 1 & 0 & 1 & 2 & 1 & 3 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 2 & 0 & 2 & 1 & 0 & 1 \\ 2 & 1 & 0 & 1 & 2 & 0 & 2 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{1.7}$$

Proposition 1.49. *Let $\{e^1, e^2, \ldots, e^s\}$ be the set of non-equivalent elementary modes of a metabolic network. Then for all $v \in C \setminus \{0\}$ there exists $\lambda_1, \lambda_2, ..., \lambda_s \geq 0$ such that $v = \sum_{i=1}^{s} \lambda_i e^i$.*

Prop. 1.49 is proven in [106] and it is an exceptionally important result, since it states that any flux distribution can be characterized as a non-negative linear combination of elementary modes. As consequence, the development of methods for the computation of EMs has become an active research area over the past years [34, 130, 118, 115, 25, 91, 92].

The computational complexity of enumerating all EMs is not known [1]. However, there exist several algorithms and software packages, for an exhaustive enumeration in a given metabolic network [34, 118, 130, 115]. These methods are mainly based on the double description method [33], and while they work very well for small networks, due to the possibly exponential number of EMs, they may fail for medium or large genome-scale networks. One workaround to this problem used frequently in the scientific literature is the deletion of several reactions from the network [80, 102, 97, 95, 108, 111, 15, 109, 128, 129, 55, 120, 54], therefore simplifying the analyzed system. As shown in [73] (see also Chapter 7) this can lead to the introduction of artificial EMs into the system.

Alternative concepts to EMs exist in the literature, i.e., extremal currents [17] and extreme pathways [96]. For a comparison of the differences and similarities between these three concepts, we refer the reader to [63].

1.3.4 Minimal metabolic behaviors

A drawback of elementary modes is that while they are a generating set, they are not minimal. Indeed, if we look again at the elementary modes in 1.7 we quickly realize that $e^1 = e^2 + e^5$ holds. In fact having only e^2, e^4 and e^5 would be sufficient to generate any feasible flux distribution of the network. To tackle this redundancy, Larhlimi and Bockmayr [65] introduced the concept of minimal metabolic behaviors (MMB) and the reversible metabolic space (RMS).

Definition 1.50 (Metabolic behavior)**.** *A metabolic behavior is a set of irreversible reactions $D \subseteq Irr$ such that $D \neq \emptyset$ and there exists a flux vector $v \in C$ with $D = supp(v) \cap Irr$.*

Definition 1.51 (Minimal metabolic behavior)**.** *A metabolic behavior D is minimal if there doesn't exist a metabolic behavior D' with $D' \subsetneq D$.*

Proposition 1.52 ([65])**.** *The minimal metabolic behaviors of a network are in a 1-1 correspondence with the minimal proper faces of C. Moreover, for a minimal proper face G of C, represented by $g \in G \setminus lin.space(C)$ its corresponding minimal metabolic behavior D can be obtained as $D := supp(g) \cap Irr$.*

Prop. 1.52 is proven in [65] and it shows a method how MMBs can be computed in practice. In essence one can use any available tool for the computation of the generators of minimal proper faces, i.e., cddTool [33], and then simply deduce the MMB from their support vector.

Indirect applications of MMBs include but are not limited to flux coupling analysis [63] (see also Chapter 2) and control-effective flux analysis [63].

1.3.5 Flux coupling analysis

Flux coupling analysis (FCA) is concerned with describing dependencies between reactions [12]. The stoichiometric and thermodynamic constraints not only determine all possible steady-state flux distributions over a network, they also induce coupling relations between the reactions. If a non-zero flux through a reaction in steady-state implies a non-zero flux through another reaction, then the two reactions are said to be coupled.

In the following, we assume that the flux cone is not trivial, i.e., not all reactions are blocked. We define the (un)coupling relationships between reactions as follows.

Definition 1.53 (Flux coupling relations)**.** *Let i, j be two unblocked reactions. The (un)coupling relationships $\longrightarrow$, $\longleftrightarrow$, $\Longleftrightarrow$ and $\not\longrightarrow$ are defined in the following way:*

- $i \longrightarrow j$ *if for all* $v \in C$, $v_i \neq 0$ *implies* $v_j \neq 0$.
- $i \longleftrightarrow j$ *if for all* $v \in C$, $v_i \neq 0$ *is equivalent to* $v_j \neq 0$.
- $i \Longleftrightarrow j$ *if there exists* $\lambda \neq 0$ *such that for all* $v \in C, v_j = \lambda v_i$.
- $i \not\longrightarrow j$ *if there exists* $v \in C$ *such that* $v_i \neq 0$ *and* $v_j = 0$.

Reactions i and j are fully (resp. partially, directionally) coupled if the relation $i \Longleftrightarrow j$ (resp. $i \longleftrightarrow j$, $i \longrightarrow j$) holds. Otherwise, i and j are uncoupled.

Note that $i \Longleftrightarrow j$ (resp. $i \longleftrightarrow j$) is equivalent to $j \Longleftrightarrow i$ (resp. $j \longleftrightarrow i$). In addition, $i \Longleftrightarrow j$ implies $i \longleftrightarrow j$, which in turn is equivalent to $i \longrightarrow j$ and $j \longrightarrow i$.

The flux coupling relations can be computed from EMs as noted in [72]. For a broader overview on the main existing approaches to FCA, we refer the reader to Chapter 2 (Subsection 2.2).

Marashi and Bockmayr [72] further refine uncoupled relations and introduce the concepts of *mutually exclusive* and *sometimes coupled* reaction pairs.

Definition 1.54 (Mutually exclusive and sometimes coupled)**.** *Two uncoupled reactions $r_i \in \mathcal{R}$ and $r_j \in \mathcal{R}$ are mutually exclusive if and only if for all $e \in E$, $e_i * e_j = 0$. Otherwise r_i and r_j are sometimes coupled.*

By the previous definition, mutually exclusive reactions are never simultaneously active in any elementary mode.

From a computational perspective, deciding between the four basic coupling relations can be done in polynomial time [24], while distinguishing between mutually exclusive and sometimes coupled pairs is an NP-complete problem [2].

FCA has been used for exploring various biological questions such as network evolution [78, 82, 136], gene essentiality [78], gene regulation [79, 75] or analysis of experimentally measured fluxes [113, 11].

1.3.6 Elementary flux patterns

Kaleta et al. [50] introduced the concept of elementary flux patterns (EFP) as an alternative to elementary mode analysis. EFPs allow the study of metabolic subnetworks in the context of the original network, without the need to delete the reactions not belonging to it. We assume to know a metabolic network $\mathcal{N} = (S, Irr, \mathcal{R}, \mathcal{M})$ with an associated flux cone C, and a subnetwork of $\mathcal{N}$, $\mathcal{N}' = (S', Irr', \mathcal{R}', \mathcal{M}')$.

Definition 1.55 (Flux pattern)**.** *A set of reactions $f \subseteq \mathcal{R}'$ is a flux pattern if and only if there exists $v \in C$, such that $f \subseteq supp(v)$ and $(\mathcal{R}' - f) \cap supp(v) = \emptyset$.*

Definition 1.56 (Elementary flux pattern)**.** *A flux pattern f is elementary if and only if there do not exist flux patterns $g_1, g_2, ..., g_k$ (for $k \geq 2$) with $g_i \neq f$ for all $i \in \{1, ..., k\}$, such that $\cup_{i=1}^{k} g_i = f$.*

Therefore EFPs are non-decomposable subsets of reactions. The authors in [50] show a method how to iteratively enumerate EFPs based on mixed integer programming. Assuming the network consists only of irreversible reactions and some EFPs have been already computed and stored in the set P, one can compute a new EFP by solving MIP_EFP [50].

$$
\begin{array}{lll}
\min & \sum_{i \in R'} b_i & \\
\text{s.t.} & Sv & = 0 \\
& b_i & \leq v_i \leq M b_i \quad \forall i \in \mathcal{R}' \\
& b_i - h_i & \geq 0 \quad \forall i \in \mathcal{R}' \\
& \sum_{i \in f} (b_i + h_i) & \leq |f| \quad \forall f \in P \\
& \sum_{i \in R'} h_i & \geq 1 \\
& v & \geq 0 \\
& b_i, h_i & \in \{0, 1\} \quad \forall i \in \mathcal{R}'
\end{array}
\qquad \text{(MIP\_EFP)}
$$

In the previous MIP formulation, M is a 'suitably large' scalar constant. If the MIP is infeasible then all EFPs have been computed already. Otherwise,

for an optimal solution (v^*, b^*, h^*), $e := supp(v^*) \cap \mathcal{R}'$ is a new EFP. For a proof of MIP_EFP we refer the reader to the Supplemental material in [50].

EFPs are in a 1-to-many relation with EMs. Therefore, from any EFP a unique EM can be computed (the converse is not true) [50].

EFPs have been used to anwer various biological questions related to the subnetworks of various microorganisms [50]. Indirect applications of EFPs are, for example, the computation of flux coupling relations (see Section 2.2.2).

1.3.7 Other methods

The spectrum of methods employed for the constraint-based analysis of metabolic networks is certainly much larger than the few we chose to present in this chapter [68]. There are other very successful methods that found great interest within the scientific community. These include optimization-based methods like flux balance analysis (FBA) [127, 32] or flux variability analysis (FVA) [71], but also methods involving minimal cut sets (MCS) [58, 42] and random sampling of the flux cone [89]. If the reader is interested in learning more about the wide range of methods we recommend the following review articles: [10, 56, 122, 139].

1.4 Organization of the thesis

The thesis consists of an introductory chapter (Chapter 1) and six main chapters containing original and collaborative research (Chapters 2-7). The content of some chapters has already been published in peer-reviewed journals. When this is the case, it is indicated at the beginning of the corresponding chapter. When using material from already published papers, only those materials were used where I appear either as first author, or joint first author.

Chapter 2 is about flux coupling analysis (FCA). First, we give an overview of all the methods existing in the literature. Then we devise two novel methods for the same purpose. Our first method, FFCA is based on checking the feasibility of linear programs. Although not by a large margin, FFCA is faster than other existing approaches. While solving LP problems is fast, solving a large number of them can hinder the performarnce of the application. We use this observation to improve on FFCA and show that we can skip solving LP problems for many pairs of reactions, using algebraic methods instead. The resulting algorithm, called F2C2 is orders of magnitude faster than FFCA, reducing the task of flux coupling from the range of days to the range of minutes for genome-scale networks. Both FFCA and F2C2 are publicly available Matlab tools [24, 66].

In Chapter 3 we extend the concept of FCA to constrained flux spaces and define constrained flux coupling analysis (CFCA). In addition to imposing

restrictions on individual fluxes, we also allow the introduction of any number of linear constraints on the reactions. Thus CFCA can be looked upon as a generalization of FCA. We present an LP based algorithm for computing CFCA relations and using a similar analysis as in Chapter 2, we reduce the number of LPs necessary to be solved. We study the relationship between FCA and CFCA relations and present mathematical rules for cases in which FCA relations determine the CFCA ones.

In Chapter 4 we explore the idea of introducing a similar concept to FCA for metabolites instead of reactions. We define metabolite activity coupling (MAC) as the FCA counterpart for metabolites. Oddly enough, when looking at the problem from a computational standpoint, it turns out that in some cases, MAC is a much harder problem than FCA or CFCA. In specific, when there are reversible reactions present in the network, we give a conjecture which states that deciding between fully coupled and partially coupled metabolites becomes a co-NP hard problem. Nevertheless, we present a hybrid LP and MIP based method for computing MAC relations.

Elementary flux modes are a canonical representation of the steady-state flux space. The complete enumeration of EMs is considered a hard problem both in terms of runtime complexity and storage space required. Thus, a targeted search of EMs becomes the more viable route when considering practical applications. Targeting conditions could be, for example, the inclusion of certain reactions. Existing methods in the literature allow the enumeration of EMs that contain a single target reaction. In Chapter 5 we generalize such methods using the Farkas-Lemma, and provide a MIP-based solution that can constrain any number of reactions to be included in the EMs found. Using various mathematical results we present a formulation of the algorithm that only needs to solve one MIP to feasibility and one LP to optimality per EM found. The resulting Matlab tool is publicly available [23].

In Chapter 6 we continue to work with EMs and try to find a solution to one of their practical weaknesses. In specific, the reactions of an elementary mode do not implement a natural order, and little is known about their interior structure. We devote this chapter to exploring various notions of ordering, as well as applying them on EMs. Ordering based on reachability properties leads to several options on how to represent EMs visually. We discuss the positive and negative aspect of each.

In Chapter 7 we present and analyze various existing methods for the analysis of metabolic subnetworks. We show that isolated subnetworks are almost never a good idea to be used since they introduce undesired artifacts in form of artificial elementary modes, as well as break up FCA relationships. We propose a method, ProCEM, that relies on the projection of the flux space onto the reactions of interest, and prove why this method should be preferred compared to the existing ones. While projection of polyhedra is a well-studied

hard problem, we can still find examples where selecting a suitable subnetwork makes performing it possible.

Chapter 2

Improving the Efficiency of Flux Coupling Analysis

Contents

Remark: This chapter is a synthesis of our results that were originally published in [24] and [66].

2.1 Introduction

Flux coupling analysis (FCA) is concerned with describing dependencies between reactions [12]. The stoichiometric and thermodynamic constraints not only determine all possible steady-state flux distributions over a network, they also induce coupling relations between the reactions. For instance, some reactions may be blocked, i.e., unable to carry flux under steady-state conditions. If a non-zero flux through a reaction in steady-state implies a non-zero flux through another reaction, then the two reactions are said to be coupled (see Def. 1.53 for a formal definition). FCA has been used for exploring various biological questions such as network evolution [78, 82, 136], gene essentiality

[78], gene regulation [79, 75] or analysis of experimentally measured fluxes [113, 11]. Having a time efficient implementation of FCA is important in such studies.

After introducing the main existing algorithms for flux coupling analysis, we propose in this chapter two new algorithms, which significantly speed up the calculation of flux coupling. In our first algorithm (FFCA) we build on existing improvements, and extend the method using the observation that typically solving an LP to feasibility is faster than solving a similarly complex one to optimality.

For the second algorithm (F2C2), we first reduce the stoichiometric model as much as possible, eliminating redundancies and finding trivial coupling relations. In a second step, we use inference rules to dramatically reduce the number of linear programming problems that have to be solved. We prove the efficiency of our algorithms by comparing them with the most recent existing approaches. We show that FCA can now be quickly performed even for very large genome-scale metabolic networks.

2.2 Existing approaches to flux coupling analysis

In this subsection, we briefly recall the main existing approaches to flux coupling analysis. For additional information and technical details on the implementation of these algorithms, we refer the reader to Additional file 1 in [24].

2.2.1 Flux Coupling Finder algorithm (FCF)

The most widely used method for FCA is the Flux Coupling Finder (FCF) algorithm [12]. It determines blocked reactions (see Def. 1.42) as well as coupled reactions by solving a sequence of linear programming (LP) problems. The FCF algorithm requires that each reversible reaction is split into two irreversible reactions, one forward and one backward. This could hinder the application of FCF to determine flux coupling in large genome-scale metabolic networks. Indeed, splitting reversible reactions results in an increase in the number of variables (resp. constraints) by $|Rev|$ (resp. $2|Rev|$).

Assuming there are no reversible reactions, for a pair of non-blocked reactions i and j, the FCF algorithm solves two LP problems (2.1) to identify the minimum and maximum allowed flux rate through reaction i, while the flux through j is set to a unit value.

$$\begin{aligned} L_{ij} &= \min \ \{v_i\colon Sv = 0,\ v_j = 1,\ v \geq 0\}, \\ U_{ij} &= \max \ \{v_i\colon Sv = 0,\ v_j = 1,\ v \geq 0\}. \end{aligned} \tag{2.1}$$

Based on L_{ij} and U_{ij} the flux coupling relations are deduced as follows:

- $j \longrightarrow i$ holds if and only if $L_{ij} \neq 0$
- $i \longrightarrow j$ holds if and only if $U_{ij} \neq +\infty$
- $i \longleftrightarrow j$ holds if and only if $L_{ij} \neq 0$ and $U_{ij} \neq +\infty$
- $i \Longleftrightarrow j$ if and only if $L_{ij} = U_{ij} > 0$.

In the following we prove the second one of the four cases above. That is, $i \longrightarrow j$ holds if and only if $U_{ij} \neq +\infty$. The proofs for the other three are trivial.

Proof. "⇒" Assume by contradiction that $i \longrightarrow j$ and $U_{ij} = +\infty$. Let us assume that we use the Simplex algorithm to solve the maximization LP (2.1). Since the LP is unbounded, the algorithm will terminate after finding an extreme ray d. That is, for all $x \in C$ and for all $\theta > 0$ we have $x + \theta d \in C$. Clearly, $d_i > 0$ and $d_j = 0$, since otherwise for a large enough θ, the feasibility in the j-th coordinate would be violated. For $x = 0$ and $\theta = 1$ we have $d \in C$, which is a contradiction to $i \longrightarrow j$.

"⇐" Assume by contradiction that $U_{ij} \neq +\infty$ and $i \not\longrightarrow j$ hold. Since U_{ij} is finite, let y be an optimal solution of the maximization problem (2.1). Thus $y_i = U_{ij}$ and $y_j = 1$. Since $i \not\longrightarrow j$, then there exists $x \in C$ with $x_i > 0$ and $x_j = 0$. Clearly, $x + y$ is a feasible point, with $U_{ij} < x_i + y_i$ which contradicts the optimality of y. □

A limitation of FCF is that for reactions that originate from splitting reversible reactions, the computed flux coupling relationships can be very different from the one existing in the original network. For example, in Fig. 2.1(left), reactions r_1 and r_2 are fully coupled, while in the corresponding reconfigured network, Fig 2.1(right), all reaction pairs are uncoupled. In order to deduce the real couplings between the reactions, a non-trivial post-processing step is needed, which is not detailed in [12]. We describe this post-processing step in Additional file 1 of [24].

On the computational side, the total number of LP problems that are being solved in FCF is increasing quadratically with the number of reactions. Since all reaction pairs are explored exhaustively, this can lead to a relatively large number of LP problems. This strategy may not scale well for genome-scale models of complex microorganisms, which involve several thousands of reactions.

The FCF algorithm has been successfully used for finding coupling relations in a number of metabolic networks [12, 82, 11, 79, 78, 113, 136, 75].

Figure 2.1: The result of splitting reversible reactions. **Left:** a metabolic network with two reversible reactions; **Right:** the same network when reactions are split into a forward and a backward arc)

2.2.2 FCA based on elementary flux patterns (EFP-FCA)

Recently, Kaleta et al. [50] introduced the concept of elementary flux patterns (EFPs) (see Def. 1.56) for the analysis of minimal active reactions in a "subnetwork", which account for possible steady-state flux distributions in a (much) larger metabolic network. They also presented a method based on mixed-integer linear programming (MILP) to compute EFPs. Kaleta et al. suggested that EFPs can be used for characterizing flux coupling relations (see Supplemental Material in [50]).

Consider a subnetwork with exactly two unblocked reactions, i and j. Based on the EFPs of this subnetwork, we can deduce the coupling relations as follows:

- $i \nleftrightarrow j$ holds if and only if the EFPs of the subnetwork are $\{i\}$ and $\{j\}$
- $i \longrightarrow j$ holds if and only if the EFPs of the subnetwork are $\{i, j\}$ and $\{j\}$
- $j \longrightarrow i$ holds if and only if the EFPs of the subnetwork are $\{i, j\}$ and $\{i\}$
- $i \longleftrightarrow j$ holds if and only if the subnetwork has one EFP, $\{i, j\}$

Thus, from computing the set of EFPs for a subnetwork corresponding to every pair of reactions, we can compute all flux coupling relations.

With this method, it is not possible to distinguish between partial and full coupling, since flux patterns only contain the information about the activity or inactivity of the fluxes, but not the actual flux rates.

2.2.3 Reversibility-based flux coupling analysis

In FCF, every reversible reaction is split into a forward and a backward reaction. This splitting procedure results in an increase in the number of LPs and also in the size of each LP to be solved. Moreover, a non-trivial post-processing step is required to infer the flux coupling relations of the

original reversible reactions. For all these reasons, it might be better to perform flux coupling analysis without splitting the reversible reactions. An alternative implementation of FCF without splitting, referred to as W-FCF, is presented in Additional file 1 of [24].

Larhlimi and Bockmayr [64] show that depending on the reversibility type of the reactions (see Def. 1.43), only certain flux coupling relations can occur. These cases are summarized in Obs. 2.1. In every other case $i \nleftrightarrow j$ holds.

Observation 2.1 ("Reversibility-Type pruning" (RT-pruning))**.** *Let i and j be two different unblocked reactions. The two reactions can be coupled only if one of the following 4 cases holds:*

1. *$i, j \in Irev$: In this case, i and j can be directionally, partially, fully or uncoupled.*
2. *$i \in Irev$ and $j \in Prev$: The only possibility is $j \longrightarrow i$ or $i \nleftrightarrow j$.*
3. *$i, j \in Prev$: In this case, we can only have $i \Longleftrightarrow j$ or $i \nleftrightarrow j$.*
4. *$i, j \in Frev$: In this case, we can only have $i \Longleftrightarrow j$ or $i \nleftrightarrow j$.*

We note that the statement in Obs. 2.1 is not precise. The mentioned coupling possibilities are the "strongest" possible ones, where the "strength" of a coupling type is defined by the intuitive ordering between them. That is $\nleftrightarrow \; < \; \longrightarrow \; < \; \longleftrightarrow \; < \; \Longleftrightarrow$.

Consider the following two LPs, with optimal solutions v^1 resp. v^2.

$$\begin{aligned} v^1 &:= \min \ \{v_j \colon Sv = 0,\ v_i = 0,\ v_k \geq 0,\ k \in Irr\}, \\ v^2 &:= \max \ \{v_j \colon Sv = 0,\ v_i = 0,\ v_k \geq 0,\ k \in Irr\}. \end{aligned} \tag{2.2}$$

Observation 2.2 (Flux coupling between (pseudo-)irreversible reactions)**.** *Let $i \in Irev$ and $j \in Prev$. Then the coupling relation can be decided based on v^1 and v^2 (cf. 2.2) as follows:*

- *if $v^1_j = v^2_j = 0$ then $j \longrightarrow i$.*
- *otherwise $i \nleftrightarrow j$.*

The number of LPs solved for a pair $(i, j) \in Irev \times Prev$ in FCF or W-FCF is four. Obs. 2.2 reduces this number to two.

The previous two observations reduce the number of reaction pairs for which the coupling have to be explicitly computed, as well as the number of LPs required by each pair. Also, in many of the cases, it is enough to determine the reversibility types of reactions i and j, and then to check if the corresponding coupling relation is possible. We will refer to an implementation

of W-FCF that incorporates the RT-pruning and also the improvement from Obs. 2.2, as WR-FCF (see Additional file 1 of [24]).

In [63] a further improvement is suggested.

Observation 2.3 (*Prev*/*Frev*-based improvement (PF-improvement))**.** *If $i, j \in Prev$ or $i, j \in Frev$, then the following are equivalent:*

- $i \Longleftrightarrow j$
- *For all v such that $Sv = 0$, $v_i = 0$ implies $v_j = 0$.*

The importance of Obs. 2.3, is that the stoichiometric constraints uniquely determine whether $i \Longleftrightarrow j$ holds, independently of the thermodynamic constraints. Thus, considering the following LP problem and its optimal solution v^*

$$v^* := max\{v_j : Sv = 0,\ v_i = 0\}, \tag{2.3}$$

we can state that $i \Longleftrightarrow j$ holds if and only if $v^*_j = 0$. Alternatively, a kernel basis (see Def. 1.5) of S can be used to check whether the condition holds.

An enhanced version of FCF, which takes into account all of the above-mentioned improvements, has been suggested in [63], but no implementation is available. We have implemented this method and will refer to it under the name, WRP-FCF. The different versions of FCF presented in this subsection have been compared with the other FCA approaches (see Sect. 2.3.2).

2.2.4 FCA based on minimal metabolic behaviors (MMB-FCA)

Larhlimi and Bockmayr [64, 65] have proposed a different strategy for flux coupling analysis. In this approach, rather than solving LPs, a minimal set of generating vectors of the flux cone is computed.

Let us denote with $G := \{g_1, g_2, ..., g_s\}$ the set of generators for the minimal proper faces (see Def 1.29) of $C \backslash lin.space(C)$ (one generator per minimal proper face). Let $B := \{b_1, b_2, ..., b_t\}$ be a vector basis of $lin.space(C)$. In combination with the RT-pruning, the coupling relation for any pair of reaction is inferred based on the co-appearance of non-zero fluxes in the generating vectors, as follows [63]:

1. $i, j \in Irev$:
   - if $\forall k \in \{1, 2, ..., s\}, g_k(i) \neq 0 \implies g_k(j) \neq 0$, then $i \longrightarrow j$ holds.
   - if $\forall k \in \{1, 2, ..., s\}, g_k(j) \neq 0 \implies g_k(i) \neq 0$, then $j \longrightarrow i$ holds.
   - if $\forall k \in \{1, 2, ..., s\}, g_k(i) \neq 0 \Leftrightarrow g_k(j) \neq 0$, then $i \longleftrightarrow j$ holds.

- if $\exists\lambda \in \mathbb{R}$, such that, $\forall k \in \{1, 2, ..., s\}, g_k(i) = \lambda g_k(j)$, then $i \Longleftrightarrow j$ holds.
- otherwise $i \nleftrightarrow j$ holds.

2. $i \in Irev$ and $j \in Prev$:

- if $\forall k \in \{1, 2, ..., s\}, g_k(j) \neq 0 \implies g_k(i) \neq 0$ then $j \longrightarrow i$ holds.
- otherwise $i \nleftrightarrow j$ holds.

3. $i, j \in Prev$:

- if $\exists\lambda \in \mathbb{R}$, such that, $\forall k \in \{1, 2, ..., s\}, g_k(i) = \lambda g_k(j)$ then $i \Longleftrightarrow j$ holds.
- otherwise $i \nleftrightarrow j$ holds.

4. $i, j \in Frev$:

- if $\exists\lambda \in \mathbb{R}$, such that, $\forall k \in \{1, 2, ..., s\}, g_k(i) = \lambda g_k(j)$ and $\forall k \in \{1, 2, ..., t\}, b_k(i) = \lambda b_k(j)$, then $i \Longleftrightarrow j$ holds.
- otherwise $i \nleftrightarrow j$ holds.

The algorithm relies on computing the complete set of MMBs (see Def. 1.51) of a network, the size of which can grow exponentially with the number of reactions. In worst case, however, when every reaction is irreversible, the MMBs are in a one-to-one correspondence with EMs (see Def. 1.45) [63]. Thus MMB-FCA is not expected to work for large, genome-scale networks.

2.3 FFCA: Feasibility-based flux coupling analysis

2.3.1 Description of the algorithm

A linear programming problem is typically solved by iterating through a sequence of feasible solutions (e.g. Simplex method variants or interior point methods) [99]. Each of the intermediate solutions has a no worse objective value than its predecessors, and the sequence 'converges' to the optimal solution. Moreover, by employing heuristic preprocessing steps, solvers can be optimized to find a first feasible solution fast [41]. Finding a first feasible solution is often referred to as "Phase I" of LP solving.

The above described reasons motivate a new approach in which we deduce coupling relations based on testing the feasibility of LP problems, rather than solving LP problems to optimality. The feasibility-based flux coupling analysis tool (FFCA) will take into account the previous improvements (Sect. 2.2.3) as well, in particular the RT-pruning and the PF-improvement.

Finding reversibilities and blocked reactions

In the first step of FFCA, we analyze the flux capability of every reaction. For a reaction i, consider the following two sets:

$$\begin{aligned} P_1(i) &:= \{v\colon Sv = 0,\ v_i = 1,\ v_k \geq 0,\ k \in Irr\}, \\ P_2(i) &:= \{v\colon Sv = 0,\ v_i = -1,\ v_k \geq 0,\ k \in Irr\}. \end{aligned} \tag{2.4}$$

Then $i \in Blk$ if one of the following two cases holds:

- $i \in Irr$ and $P_1(i) = \emptyset$.
- $i \notin Irr$ and $P_1(i) = P_2(i) = \emptyset$.

The nonblocked reactions have to be further classified into $Irev$, $Prev$ and $Frev$. This classification method is different to the one used in [63]. A reaction i belongs to the set $Irev$ if and only if one of the following holds:

- $i \in Irr$ and $P_1(i) \neq \emptyset$.
- $i \notin Irr$ and $P_1(i) = \emptyset$ and $P_2(i) \neq \emptyset$.
- $i \notin Irr$ and $P_1(i) \neq \emptyset$ and $P_2(i) = \emptyset$.

Let $T := \{i\colon P_1(i) \neq \emptyset, P_2(i) \neq \emptyset\}$ and consider the subnetwork $(S', \emptyset)$ of the original network, where reactions not in T are removed. Let $B := \{b_1, b_2, ..., b_t\}$ be a basis of $ker(S')$. Then a reaction $i \in T$ can be classified as follows:

- $i \in Frev$ if and only if $\exists k \in \{1, 2, ..., t\}$ with $b_k(i) \neq 0$.
- $i \in Prev$ otherwise.

Deducing flux coupling relations

1. $i, j \in Irev$: In this case, we check the feasibility of two systems of linear inequalities:

$$\begin{aligned} P_3(i,j) &:= \{v\colon Sv = 0,\ v_i = 1,\ v_j = 0,\ v_k \geq 0,\ k \in Irr\}, \\ P_4(i,j) &:= \{v\colon Sv = 0,\ v_i = 0,\ v_j = 1,\ v_k \geq 0,\ k \in Irr\}. \end{aligned} \tag{2.5}$$

   Based on the feasibility of the sets $P_3(i,j)$ and $P_4(i,j)$ we deduce coupling relations as follows:

   - $i \nleftrightarrow j$ holds if and only if $P_3(i,j) \neq \emptyset$ and $P_4(i,j) \neq \emptyset$

- $i \longrightarrow j$ holds if and only if $P_3(i,j) = \emptyset$
- $j \longrightarrow i$ holds if and only if $P_4(i,j) = \emptyset$
- $i \longleftrightarrow j$ holds if and only if $P_3(i,j) = \emptyset$ and $P_4(i,j) = \emptyset$

We note that, to distinguish between fully and partially coupled pairs in this case, one has to use other methods, e.g. computing enzyme subsets [87] or solving the LPs in 2.1, as in the FCF algorithm.

2. $i \in Irev$ and $j \in Prev$: The only possible coupling relation is $j \longrightarrow i$ (in the reverse direction $i \not\longrightarrow j$ always holds). Hence, $P_3(i,j)$ will always be feasible and does not need to be tested. However, we need to check the feasibility of $P_4(i,j)$. Additionally, since j can take negative values, one more system should be checked for feasibility:

$$P_5(i,j) := \{v\text{: } Sv = 0,\ v_i = 0,\ v_j = -1,\ v_k \geq 0,\ k \in Irr\}. \tag{2.6}$$

The coupling relations are deduced as follows:

- $j \longrightarrow i$ holds if and only if $P_4(i,j) = \emptyset$ and $P_5(i,j) = \emptyset$
- $i \not\longleftrightarrow j$ holds if and only if $P_4(i,j) \neq \emptyset$ or $P_5(i,j) \neq \emptyset$

3. $i,j \in Prev$ or $i,j \in Frev$: In this case, studying the set $P_6(i,j)$ suffices to compute the coupling relations.

$$P_6(i,j) := \{v\text{: } Sv = 0,\ v_i = 0,\ v_j = 1\}. \tag{2.7}$$

Based on $P_6(i,j)$ we conclude that:

- $j \Longleftrightarrow i$ holds if and only if $P_6(i,j) = \emptyset$.
- $i \not\longleftrightarrow j$ holds if and only if $P_6(i,j) \neq \emptyset$.

To perform FFCA, a method is needed to check the feasibility of a system of linear (in)equalities. In practice, this can be done by using any available LP solver. The set of linear (in)equalities is used as the constraints of the LP problem, together with a constant objective function. Clearly, any feasible solution will be also an optimal solution, and therefore, the LP solver will finish after finding the first feasible solution (i.e. after Phase I). For example, checking the feasibility of $P_3(i,j)$ is equivalent to solving the following LP:

$$\min\ \{0\text{: } Sv = 0,\ v_i = 1,\ v_j = 0,\ v_k \geq 0,\ k \in Irr\}, \tag{2.8}$$

In Table 2.1, we compare the characteristics of the FFCA approach to the other FCA methods described in Subsection 2.2.3.

2.3.2 Computational results

To compare the different approaches, namely FCF, MMB-FCA, EFP-FCA and FFCA, we implemented all of them in Matlab [74]. A benchmark set of six metabolic network models was used for the evaluation. The number of unblocked reactions in these models ranges from 18 to 765. Table 2.2 summarizes the running times, while Table 2.3 reports on the resulting coupling relations. One can see that in all cases FFCA is 2 to 3 times faster than FCF and orders of magnitude faster than EFP-FCA. The table also shows that FFCA is more appropriate for FCA in genome-scale networks. MMB-FCA is the fastest method for the three smallest networks. However, for the middle-sized *H. pylori* network and especially for the large networks of *S. cerevisiae* and *E. coli*, FFCA proves to be faster than MMB-FCA. The computation time required for MMB-FCA rapidly grows when the number of reactions increases. This is due to the possibly exponential size of the set of generating vectors which has to be computed before finding the coupled reactions (see Section 4.4 in [63]). EFP-FCA, which is based on solving mixed-integer linear programs, turns out to be much slower than other methods. Although the concept of elementary flux patterns is very useful in the analysis of subnetworks, its applicability in full FCA seems to be limited.

Both the FCF algorithm and the current implementation of FFCA solve LPs for flux coupling analysis. Here we summarize the main differences between the two methods:

- When an LP is solved in FFCA, finding the first feasible solution is sufficient, while the LPs should be solved to optimality in case of the FCF algorithm.

- In the FCF method, in contrast to FFCA, every reversible reaction is split into two (forward and backward) irreversible reactions. This step slows down the procedure and increases the size of each LP to be solved.

- For computing the flux coupling relations between any pair of reactions, we always need at most two LPs in FFCA, while in FCF sometimes more LPs have to be solved. For example, for computing the coupling relation between an irreversible and a reversible reaction (after splitting), four LPs are solved.

- The RT-pruning to reduce the number of possible coupled reaction pairs is not considered in FCF, but is in FFCA.
- The PF-improvements are not considered in FCF, but they are in FFCA.

While the above differences improve the run-time speed of flux coupling, the complexity of the two algorithms is similar. That is, $O(n^2 \times L(n,m))$, where $L(n,m)$ is the complexity of solving a linear problem with n variables and m constraints. Moreover, assuming there are only irreversible reactions in the network, the two algorithms solve the same number, n^2, of LPs.

In Table 2.2 the computational running times of these methods are also shown. As expected, the three versions of the improved FCF algorithm are faster than the classical FCF algorithm, while they are still slower that FFCA.

Metabolic network models

Nine metabolic networks were used in this study:

- ILLUSNET network from [64]
- RBC : metabolic network of red blood cell [134]
- EC core: central metabolic network of *E. coli* [83]
- *H. pylori* genome-scale metabolic network [121]
- *S. cerevisiae* (yeast) genome-scale metabolic network [29]
- The 2003 genome-scale metabolic network of *E. coli* (iJR904) [90]
- The 2007 genome-scale metabolic network of *E. coli* (iAF1260) [30]
- The metabolic network of *H. sapiens* (Recon 1) [28]
- The metabolic network of human hepatocyte (HepatoNet1) [38]

For FCA, all uptake reactions were assumed to be able to carry non-zero fluxes.

2.3.3 Computational properties of FFCA

Flux coupling analysis of genome-scale networks can be very time consuming. Only recently (see e.g. [136]) FCA has been used for some of the new genome-scale metabolic networks that contain more than 2000 reactions. To further illustrate our approach, we have applied FFCA to three of these very large networks. The

Table 2.1: General comparison of different approaches to flux coupling analysis

Method name	Preprocessing	Main procedure: Type of linear program and solution	Main procedure: Further distinguishing of $\longleftrightarrow$ and $\Longleftrightarrow$ required?	Postprocessing for rev. reactions?
MMB-FCA	computing MMBs + reaction classification	n/a	No	No
EFP-FCA	splitting reversible reactions	MILP, optimal	Yes	Yes
FCF	splitting reversible reactions	LP, optimal	No	Yes
W-FCF	n/a	LP, optimal	No	No
WR-FCF	reaction classification	LP, optimal	No	No
WRP-FCF	reaction classification	LP, optimal	No	No
FFCA	reaction classification	LP, feasible	Yes	No

Table 2.2: CPU running time (in seconds) required for flux coupling analysis of the benchmark networks when CLP [18] is used as the LP solver

	no. of unblocked reactions	MMB-FCA	EFP-FCA	FCF	W-FCF	WR-FCF	WRP-FCF	FFCA
ILLUSNET	18	0.01	26.3	0.25	0.14	0.09	0.06	0.06
RBC	38	0.05	152	1.39	0.80	0.68	0.64	0.62
EC core	63	0.22	585	6.58	3.03	3.13	2.19	2.15
H. pylori	217	69.8	$>$ 1 day	196	83.6	67.0	63.3	60.7
S. cerevisiae	639	$>$ 1 day	$>$ 1 day	8.5×10^3	4.0×10^3	3.4×10^3	3.3×10^3	3.1×10^3
E. coli(iJR904)	765	$>$ 1 day	$>$ 1 day	1.2×10^4	7.4×10^3	6.3×10^3	5.9×10^3	5.6×10^3

Table 2.3: Number of flux (un-)coupling relations for each of the benchmark networks (Table 2.2)

	no. of unblocked reactions	total no. of reactions	fully coupled	partially coupled	directionally coupled	uncoupled
ILLUSNET	18	19	6	0	9	138
RBC	42	42	21	0	15	825
EC core	63	63	19	0	9	1925
H. pylori	217	480	164	3	309	22960
S. cerevisiae	639	1144	692	90	2214	200845
E. coli(iJR904)	765	930	2567	68	6208	283387

Table 2.4: Flux coupling analysis of new generation genome-scale metabolic networks

	no. of unblocked reactions	total no. of reactions	fully coupled	partially coupled	directionally coupled	uncoupled	running time (sec)
H. sapiens (Recon. 1)	2018	3284	1555	4	3503	2030091	1.59×10^5
E. coli (iAF1260)	1878	2075	2563	143	12791	1747006	1.66×10^5
Human hepatocyte (HepatoNet1)	2309	2309	1463	201	1593	2661329	1.35×10^5

results are reported in Table 2.4. We can see that FFCA needs 37-46 hours for each of the networks to perform a complete FCA. The numbers of the resulting flux coupling relations are also given in the table. Since FFCA includes the RT-pruning and PF-improvement, it might be interesting to see how the number of coupling relations (and also the running time of FFCA) depends on the number of reversible reactions in a network. This problem is studied in Additional file 2 of [24] and it is empirically shown that the numbers of uncoupled pairs increase (and the running times generally decrease) with the increase in the number of reversible reactions.

2.4 F2C2: A fast tool for the computation of flux coupling in genome-scale metabolic networks

In this section we build on the performance advantage of the FFCA algorithm and present additional preprocessing steps as well as algorithmic improvements that reduce the total running time of the algorithm. The preprocessing steps occur before using linear programming to calculate flux coupling between reactions, and try to reduce the number of variables and constraints of the subsequent LP problems. The network reduction is mainly based on the removal of trivially blocked reactions and the merging of the stoichiometric columns corresponding to trivially coupled reactions [87, 62, 35, 126]. For computing a subset of the fully coupled reactions, one can also use the kernel of the stoichiometric matrix [62]. We will prove that under some easily satisfiable conditions, all fully coupled reactions can be found. Alternatively, one can apply some reduction rules which require only a simple parsing of the stoichiometric matrix and are not time consuming. This alternative strategy allows avoiding potential numerical instabilities related to the computation of a basis of the kernel.

Once the improvements have been detailed and an outline of the F2C2 algorithm has been presented, we benchmark it against the FFCA algorithm.

2.4.1 Preprocessing

Certain metabolites, called *dead-end metabolites* [62], are produced (resp. consumed) by some reactions without being consumed (resp. produced) by other reactions. This concept is illustrated in Fig. 2.2 where the dead-end metabolite H is produced by reaction 13 without being consumed by any of the remaining reactions.

As stated below, reactions which are consuming or producing dead-end metabolites are blocked.

Observation 2.4 (Dead-end metabolite)**.** *Let $k \in \{1, \ldots, m\}$ be a metabolite. Then, the following hold:*

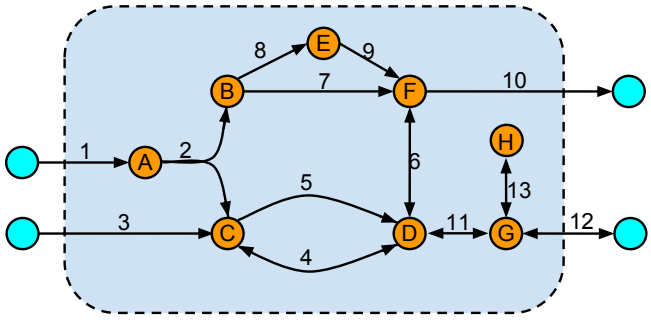


Figure 2.2: Exemplary metabolic network MetNet. It consists of eight metabolites $(A, \ldots, H)$, and thirteen reactions $(1, \ldots, 13)$, whereof six reactions are irreversible. Metabolites are depicted as nodes while reactions are depicted as arrows. Reversible reactions are indicated by double arrowheads.

- *If there exists a reaction i such that $S_{ki} \neq 0$ and $S_{kj} = 0$ for all $j \neq i$, then reaction i is blocked.*
- *If there exists a set of reactions $I \subseteq Irr$ such that $S_{ki} > 0$ (resp. $S_{ki} < 0$) for all $i \in I$ and $S_{kj} = 0$ for all $j \notin I$, then all reactions $i \in I$ are blocked.*

In each of these cases, k is called a dead-end metabolite.

Certain metabolites are involved in exactly two reactions. For instance, in the network MetNet depicted in Fig. 2.2, metabolite E is produced/consumed only by reactions 8 and 9. The following observation states that the fluxes through reactions involving such metabolites are proportional to each other.

Observation 2.5 (Trivial Full Coupling (TFC))**.** *Let i and j be two reactions such that, for some metabolite $k \in \{1, \ldots, m\}$, $S_{ki} \neq 0$, $S_{kj} \neq 0$ and $S_{kl} = 0$ for all $l \neq i, j$. Then, reactions i and j are either blocked or fully coupled.*

The identification of dead-end metabolites and their corresponding blocked reactions allows us to reduce the number of metabolites and reactions that matter for identifying coupled reactions. As stated in the following observation, the removal of the rows (resp. columns) in the stoichiometric matrix corresponding to dead-end metabolites (resp. blocked reactions) does not influence the flux coupling between reactions.

Observation 2.6 (Reduction Rule 1)**.** *Let D be a set of dead-end metabolites and let B be a set of blocked reactions. For convenience, suppose $B = \{n - |B| + 1, \ldots, n\}$. Let S' be the submatrix of S formed by the rows S_{k*} with $k \notin D$ and the columns S_{*l} with $l \notin B$. Let $Irr' = Irr \setminus B$. Then, for all pairs of reactions $i \notin B$ and $j \notin B$,*

- $j \longrightarrow i$ *if and only if* $v'_i = 0$ *implies* $v'_j = 0$, *for all* $v' \in \mathbb{R}^{n-|B|}$ *such that* $S'v' = 0$ *and* $v'_p \geq 0$ *for all* $p \in Irr'$.
- $i \Longleftrightarrow j$ *if and only if there exists* $\lambda' \neq 0$ *such that* $v'_j = \lambda' v'_i$, *for all* $v' \in \mathbb{R}^s$ *with* $S'v' = 0$ *and* $v'_p \geq 0$ *for all* $p \in Irr'$.

The next observation shows that two fully coupled reactions can be represented by only one column in the stoichiometric matrix, without altering the flux coupling between reactions.

Observation 2.7 (Reduction Rule 2)**.** *Let* k, l *be two reactions such that for all* $v \in C, v_l = \lambda v_k$ *for some* $\lambda \neq 0$. *For convenience, suppose* $l = n$ *and* $\lambda > 0$. *Let* S' *be the* $m \times (n-1)$ *matrix defined by* $S'_{*p} = S_{*p}$ *for all* $p \neq k, l$ *and* $S'_{*k} = S_{*k} + \lambda S_{*l}$. *Let* $Irr' = (Irr \cup \{k\}) \setminus \{l\}$ *if* $l \in Irr$, *and* $Irr' = Irr$ *otherwise. Then, for all pairs of reactions* $i \neq l$ *and* $j \neq l$,

- $j \longrightarrow i$ *if and only if* $v'_i = 0$ *implies* $v'_j = 0$, *for all* $v' \in \mathbb{R}^{n-1}$ *such that* $S'v' = 0$ *and* $v'_p \geq 0$ *for all* $p \in Irr'$.
- $i \Longleftrightarrow j$ *if and only if there exists* $\lambda' \neq 0$ *such that* $v'_j = \lambda' v'_i$, *for all* $v' \in \mathbb{R}^{n-1}$ *with* $S'v' = 0$ *and* $v'_p \geq 0$ *for all* $p \in Irr'$.

Note that when applying the reduction rules of Observations 2.6 and 2.7, further metabolites and reactions may fulfill the conditions of Observations 2.4 and 2.5. Accordingly, we apply these reduction rules in an iterative way. As an illustration, the reduction of the network MetNet depicted in Fig. 2.2 involves two iterations. In the first one, metabolite H and reaction 13 are removed, the pairs of reactions $(1, 2)$ and $(8, 9)$ are merged and metabolites A and E are removed. In the second iteration, the equivalent reactions $(11, 12)$ are merged and metabolite G is removed. The preprocessed network depicted in Fig. 2.3 contains only four metabolites and nine reactions.

Certain fully coupled reactions could not be identified using Observation 2.5. The following lemma proves that all fully coupled reaction pairs can be deduced from the kernel $kern(S) = \{v \in \mathbb{R}^n \mid Sv = 0\}$ of the stoichiometric matrix after the removal of all blocked reactions.

Lemma 2.8. *Let* (S, Irr) *be a metabolic network with all* n *reactions unblocked. For a pair of reactions* (i, j) *the following are equivalent:*

- $i \Longleftrightarrow j$.
- *there exists* $\lambda \in \mathbb{R} \setminus \{0\}$ *such that* $v_i = \lambda v_j$, *for all* $v \in kern(S)$.

Proof. "$\Leftarrow$" Immediate.
"$\Rightarrow$" Since $i \Longleftrightarrow j$, there exists $\lambda \neq 0$ such that $v_i = \lambda v_j$ for all $v \in C$.

Assume by contradiction that there is $v \in kern(S)$ such that $v_i \neq \lambda v_j$.

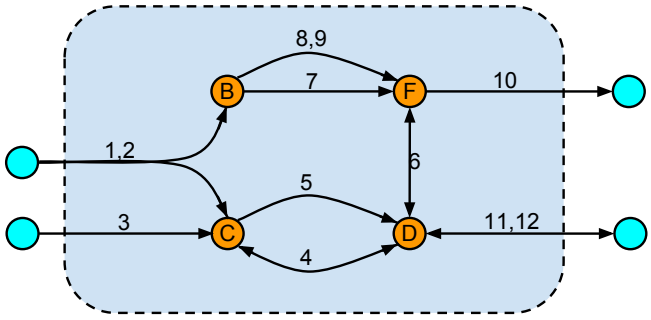


Figure 2.3: Exemplary metabolic network MetNet after preprocessing. Dead-end metabolites and blocked reactions were removed and fully coupled reactions were merged iteratively. This resulted in the removal of the blocked reaction 13 and the merging of the pairs of equivalent reactions $(1,2)$, $(8,9)$ and$(11,12)$

Let $L = \{l \in Irr \mid v_l < 0\}$. Since every reaction is unblocked, for every $l \in L$ there exists $g^{(l)} \in C$ with $g_l^{(l)} = 1$.

Now let $w := v - \sum_{l \in L} v_l g^{(l)}$. Clearly, $w \in kern(S)$ and $w_l > 0$ for all $l \in Irr$, thus $w \in C$. However, $w_i \neq \lambda w_j$, contradicting $i \Longleftrightarrow j$. The required statement follows. □

In analogy with the WRP-FCF and FFCA approaches, we identify the reversibility type of reactions in order to apply linear programming only in those cases where coupling relationships can occur [64]. To compute the sets $Irev, Prev, Frev$ and Blk we use the procedure for reaction classification described in Section 2.3.1. Note that applying the above reduction rules beforehand reduces the number of variables and constraints in the LP problems used for the classification of reactions.

Based on the results above, we propose to apply the preprocessing procedure given in Table 2.5 before identifying coupled reactions using linear programming. We show later that the preprocessing step turns out to be crucial for obtaining an efficient flux coupling algorithm.

2.4.2 Algorithmic improvements

In certain metabolic networks, the conversion of a set of substrates into a set of products can be made by different reactions having the same stoichiometry. A simple example of such reactions are isoenzymes which make the same conversion of substrates into products. This concept is illustrated in Fig. 2.2 where both reactions 4 and 5 convert C into D in the same way. This holds also for reaction 7 and the merged equivalent reactions $(8,9)$ in Fig. 2.3, showing that the network preprocessing may simplify the identification of

Table 2.5: Main steps of the preprocessing procedure

Step	Rule
1.	Identify dead-end metabolites and the corresponding blocked reactions.
2.	Apply Reduction Rule 1 to remove the rows (resp. columns) corresponding to dead-end metabolites (resp. blocked reactions) from the stoichiometric matrix.
3.	Apply the TFC rule to determine reactions which are proportional to each other and update their reversibility type.
4.	Apply Reduction Rule 2 to keep only one column for each set of reactions which are proportional to each other.
5.	Repeat Steps 1-4 until neither a dead-end metabolite nor a pair of fully coupled reactions can be identified.
6.	Update the reversibility type of each reaction and remove the columns corresponding to blocked reactions from the stoichiometric matrix.
7.	Use a basis of the kernel of the stoichiometric matrix to identify fully coupled reactions. This step is optional as the kernel computation may lead to numerical problems.
8.	Classify reactions according to their reversibility type.

such alternative conversions. The flux coupling of such reactions is trivial using the following lemma.

Lemma 2.9 (Trivial Uncoupling (TUC)). *Let $i, j \in Irev$ and $k, l \in Prev \cup Frev$ be four reactions. Then, the following holds:*

- *If $S_{*i} = \alpha S_{*j}$ for some $\alpha > 0$, then $p \not\rightarrow i$ and $p \not\rightarrow j$ for all $p \notin Blk$.*
- *If $S_{*i} = \alpha S_{*j}$ for some $\alpha < 0$, then $i \not\rightarrow p$ (resp. $j \not\rightarrow p$) for all $p \notin Blk \cup \{j\}$ (resp. $p \notin Blk \cup \{i\}$).*
- *If $S_{*i} = \alpha S_{*k}$ for some $\alpha \neq 0$, then $p \not\rightarrow i$ and $i \not\rightarrow p$ for all $p \notin Blk$ and $k \not\rightarrow q$ for all $q \notin Blk \cup \{i\}$.*
- *If $S_{*k} = \alpha S_{*l}$ for some $\alpha \neq 0$, then $p \not\rightarrow k$ and $k \not\rightarrow p$ for all $p \notin Blk \cup \{l\}$ and $l \mapsto q$ and $q \mapsto l$ for all $q \notin Blk \cup \{k\}$.*

Proof. The proofs of the four statements are similar. We only consider the first one.

Suppose $S_{*i} = \alpha S_{*j}$ for some $\alpha > 0$ and let us prove $p \not\rightarrow i$.

Let $p \notin Blk$. Then there exists $v \in C$ such that $v_p \neq 0$. Let $w \in \mathbb{R}^n$ such that $w_i = 0$, $w_j = \alpha v_i + v_j$ and $w_q = v_q$ for all $q \neq i, j$. We have $w \in C$ with $w_i = 0$ and $w_p \neq 0$, so the claim follows. □

The next observation states that metabolites which are involved only in irreversible reactions and consumed or produced by exactly one reaction define trivial directional couplings between these reactions.

Observation 2.10 (Trivial Directional Coupling (TDC)). *Let k be some metabolite such that $S_{kl} = 0$ for all $l \in Frev \cup Prev$. Let $P = \{i \mid S_{ki} > 0\}$ and $N = \{j \mid S_{kj} < 0\}$. If $card(P) = 1$ (resp. $card(N) = 1$), then $j \longrightarrow i$ (resp. $i \longrightarrow j$) for all $(i, j) \in P \times N$.*

Since directional flux coupling is a transitive relation, the flux (un)coupling between many reactions can simply be deduced from dependencies between reactions whose flux (un)coupling has been determined beforehand. This allows us to significantly reduce the total number of LP problems to be solved. Examples of such inferred flux (un)couplings are given in Fig. 2.3. According to the TDC rule, we have $(8, 9) \longrightarrow (1, 2)$. By solving the LP problems (2.1), we obtain $(8, 9) \not\longrightarrow 10$. We can easily conclude that $(1, 2) \not\longrightarrow 10$.

Table 2.6 shows the inferred flux (un)coupling relations we can deduce from known relationships between reactions.

Table 2.6: Transitivity inferred flux (un)coupling

Known flux coupling	$i \Longleftrightarrow j$	$i \longleftrightarrow j$	$j \longrightarrow i$	$i \longrightarrow j$
$k \Longleftrightarrow i$	$k \Longleftrightarrow j$	$i \longleftrightarrow j$	$j \longrightarrow k$	$k \longrightarrow j$
$k \longleftrightarrow i$	$k \longleftrightarrow j$	$k \Longleftrightarrow j$ $k \longleftrightarrow j$	$j \longrightarrow k$	$k \longrightarrow j$
$i \longrightarrow k$	$j \longrightarrow k$	$j \longrightarrow k$	$j \longrightarrow k$	
$i \not\longrightarrow k$	$j \not\longrightarrow k$	$j \not\longrightarrow k$		$j \not\longrightarrow k$
$k \not\longrightarrow i$	$k \not\longrightarrow j$	$k \not\longrightarrow j$	$k \not\longrightarrow j$	

For some pairs of reactions, we need to solve at least one LP problem. The optimal solution not only determines the flux coupling between the considered pair of reactions, but also allows one to determine many other uncoupled reactions.

Observation 2.11 (Feasibility Rule). *Let $v \in C$ be a steady-state flux vector and let $I = \{i \mid v_i = 0\}$ and $J = \{j \mid v_j \neq 0\}$. Then $j \not\longrightarrow i$ for all $(i, j) \in I \times J$.*

In general, most irreversible reactions are uncoupled to each other. Accordingly, the LP problems (2.1) used to determine coupled irreversible reactions are often unbounded. This limits the use of the feasibility rule, which requires the calculation of a feasible flux vector. In order to optimally use the feasibility rule, instead of solving the LP problems (2.1) to decide whether two irreversible reactions $i, j \in Irev$ are coupled, we propose to solve the bounded LP problems

$$\begin{aligned} L_{ij} &= \min \{v_i : Sv = 0,\ v_j = 1,\ v_k \geq 0,\ k \in Irr\}, \\ L_{ji} &= \min \{v_j : Sv = 0,\ v_i = 1,\ v_k \geq 0,\ k \in Irr\}, \end{aligned} \tag{2.9}$$

and to use the following scheme:

- $j \longrightarrow i$ (resp. $i \longrightarrow j$) if and only if $L_{ij} \neq 0$ (resp. $L_{ji} \neq 0$),
- $j \Longleftrightarrow i$ if and only if $L_{ij} \neq 0$, $L_{ji} \neq 0$ and $L_{ij} = 1/L_{ji}$.

The following observation states that removing a fully reversible reaction does not alter the flux coupling between (pseudo-)irreversible reactions.

Observation 2.12. *Let $k \in Frev$ be a fully reversible reaction. For convenience, suppose $k = n$. Let S' be the $m \times (n-1)$ matrix defined by $S'_{*p} = S_{*p}$ for all $p \neq k$ and let $Irr' = Irr$. Then, for all pairs of reactions $i \notin Frev$ and $j \notin Frev$,*

- *$j \longrightarrow i$ if and only if $v'_i = 0$ implies $v'_j = 0$, for all $v' \in \mathbb{R}^{n-1}$ with $S'v' = 0$ and $v'_p \geq 0$ for all $p \in Irr'$.*
- *$i \Longleftrightarrow j$ if and only if there exists $\lambda' \neq 0$ such that $v'_j = \lambda' v'_i$, for all $v' \in \mathbb{R}^{n-1}$ with $S'v' = 0$ and $v'_p \geq 0$ for all $p \in Irr'$.*

Let S_{*Rev} be the submatrix defined by the columns in S corresponding to the reversible reactions and let t be the dimension of $kern(S_{*Rev})$. Based on Observation 2.12, we can remove up to t independent fully reversible reactions without altering the flux coupling between (pseudo-)irreversible reactions. Since certain fully reversible reactions may change their reversibility type after the deletion of a fully reversible reaction, we first remove a randomly chosen reaction $i \in Frev$ together with the coupled reactions with i. We calculate the impact of this deletion on the dimension of $kern(S_{*Rev})$. If this dimension decreases by 1, the deletion is maintained; otherwise the removed reactions are restored to the network. This is repeated until t independent fully reversible reactions and their coupled reactions are removed. We assume that the flux coupling between fully reversible reactions is determined beforehand.

Based on the above results, we propose the Fast Flux Coupling Calculator (F2C2) to determine coupled reactions. The main steps of the F2C2 algorithm are given in Table 2.7.

2.4.3 Computational results and benchmarks

The F2C2 algorithm has been implemented within the Matlab [74] environment, using CLP [18] (via the Mexclp interface [70]) as the underlying linear programming solver. For benchmarking, we analyzed the following genome-scale metabolic networks:

- *Escherichia coli*, $iJR904$ [90]
- *Saccharomyces cerevisiae*, $iND750$ [29]
- *Methanosarcina barkeri*, $iAF692$ [31]

Table 2.7: The main steps of the F2C2 algorithm

Step	Rule
1.	Apply the preprocessing procedure shown in Table 2.5.
2.	Apply the feasibility rule using the feasible solutions obtained when solving the LP problems used in Step 1.
3.	Apply the TDC and TUC rules to determine trivially (un)coupled reactions.
4.	Identify fully coupled reversible reactions by solving the LP problems (2.3). This is not necessary if the kernel of the stoichiometric matrix is used in Step 1.
5.	Determine the dimension t of $kern(S_{*Rev})$ and remove t independent fully reversible reactions and their coupled reactions. This step is optional since t is often small.
6.	Determine the flux coupling between (pseudo-)irreversible reactions by solving the LP problems (2.2) and (2.9).
7.	For each LP problem solved in Step 6, use the inference rules in Table 2.6 in combination with the feasibility rule.

- *Mycobacterium tuberculosis*, $iNJ661$ [45]
- *Escherichia coli*, $iAF1260$ [30]
- *Homo sapiens*, $Recon1$ [28]
- *Escherichia coli*, $iJO1366$ [81]

For the numerically sensitive parts, a tolerance level of `10e-6` was set. All computations were performed using a single Intel Xeon 5160 (3.0 GHz) processor on a 64-bit Debian Linux 6.0 system.

As pointed out in the previous section, part of the performance gain of F2C2 over previous FCA algorithms stems from the fact that the preprocessing steps reduce the network size. This affects the running time on two levels: there are fewer reaction pairs and the LP problems to be solved have reduced size. The dramatic effect of the preprocessing steps on the network size is presented in Table 2.8.

The algorithmic improvements further reduce the number of LP problems to be solved. A direct performance comparison between the FFCA and F2C2 algorithms (including the running times and number of LP problems solved) is summarized in Table 2.9. In all cases, F2C2 outperformed FFCA by several orders of magnitude. In [24] it has been shown that FFCA is more efficient on genome-scale metabolic networks than other flux coupling algorithms.

For an intuitive, visual representation of the individual improvements' impact on the algorithm's performance, a more in-depth analysis has been performed on the recent metabolic network of *E. coli*, $iJO1366$. Four different

sets of improvements were cumulatively switched on, and the linear programs solved were plotted for each case (Fig. 2.4). To better highlight the relevant differences, the following modifications were applied to the results. First, 249 (out of 2582) reactions identified as blocked were removed from the images. This is a common preprocessing step in most FCA algorithms. Secondly, the order of reactions was permuted such that the reactions in *Irev*, *Prev* and *Frev* are clustered together. Additionally, in each of these three clusters, the fully coupled reactions were moved towards the end of the segment.

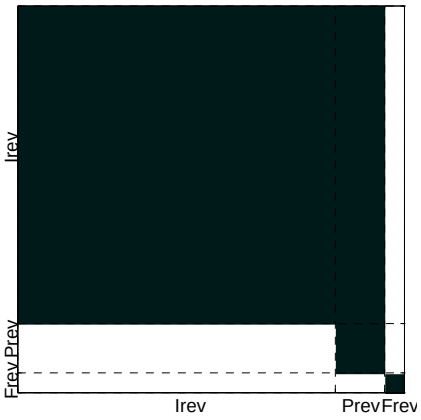
Fig. 2.4(a) plots the LP problems solved in the FFCA algorithm. Applying the simple preprocessing steps without using the kernel (Fig. 2.4(b)), several reactions are found to be fully coupled with others, and as such can be merged together. When Lemma 2.8 is applied (Fig. 2.4(c)), all fully coupled sets are detected. As a consequence, the gray stripes get thicker and the LP problems corresponding to $(Prev, Prev)$ and $(Frev, Frev)$ pairs need not be solved anymore. The use of the algorithmic improvements (Fig. 2.4(d)) filters the pairs in $(Irev, Irev)$ and $(Irev, Prev)$ blocks, greatly reducing the total number of LP problems to be solved.

Table 2.8: Genome-scale metabolic networks with the number of metabolites (♯met.) and reactions (♯reac.) before and after applying the preprocessing steps

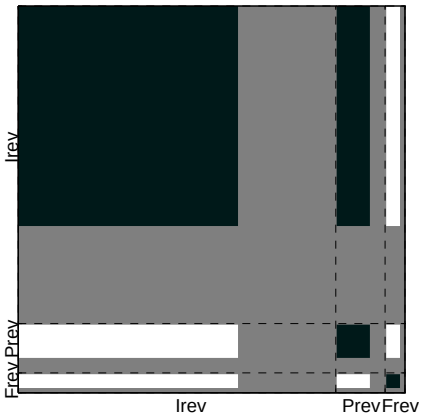
Network name	Original size		Prepr. size	
	♯met.	♯reac.	♯met.	♯reac.
M. barkeri, *iAF*692	628	690	149	221
S. cerevisiae, *iND*750	1061	1266	248	446
M. tuberculosis, *iNJ*661	826	1025	240	418
E. coli, *iJR*904	761	1075	269	565
E. coli, *iAF*1260	1668	2382	604	1272
E. coli, *iJO*1366	1805	2582	651	1376
H. sapiens, *Recon*1	2766	3742	725	1573

Table 2.9: Performance comparison between the FFCA and F2C2 algorithms in terms of the number of LP problems solved (♯LPs) and their total running times (TRT). For the F2C2 algorithm, TRT includes the time (Prepr. RT) required for the preprocessing step. Computation times are given in days (d), hours (h), minutes (m) and seconds (s)

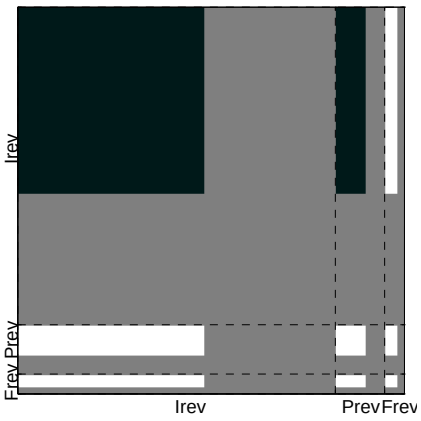
Network	FFCA		F2C2		
	♯LPs	TRT	♯LPs	TRT	Prepr. RT
M. barkeri, *iAF*692	301975	59m 40s	774	7s	5s
S. cerevisiae, *iND*750	472629	1h 50m 17s	1280	21s	15s
M. tuberculosis, *iNJ*661	556504	3h 5m 36s	1506	22s	16s
E. coli, *iJR*904	655437	2h 40m 33s	1580	26s	18s
E. coli, *iAF*1260	4256786	4d 31m 26s	3309	2m 47s	2m
E. coli, *iJO*1366	4877262	4d 5h 30m 46s	3955	3m 55s	2m 45s
H. sapiens, *Recon*1	4566304	4d 18h 3m 37s	3903	5m 20s	4m 9s



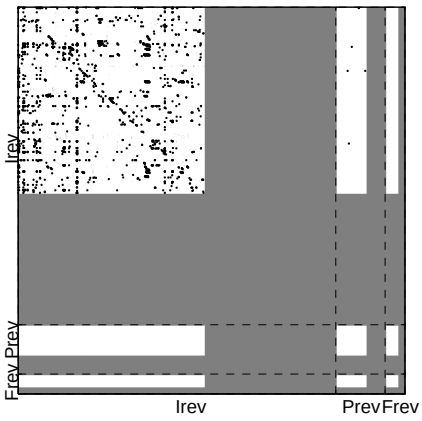
(a) - FFCA algorithm



(b) - FFCA with the preprocessing and without using the kernel



(c) - FFCA with the preprocessing using kernel



(d) - F2C2 algorithm

Figure 2.4: Visualization of the LP problems solved using different algorithms. (**a**) The FFCA algorithm, (**b**) the FFCA algorithm after applying the preprocessing procedure given in Table 2.5 without kernel computation (Step 7), (**c**) the FFCA algorithm after applying the preprocessing procedure and using the kernel of the stoichiometric matrix to identify fully coupled reactions and (**d**) the F2C2 algorithm given in Table 2.7. The dashed lines mark the boundary of the *Irev*, *Prev* and *Frev* regions. Colors: Black - an LP problem is solved for the corresponding (ordered) pair of reactions; Gray - the corresponding LP problem is not solved due to one (or both) reactions being eliminated from the network (a preprocessing improvement); White - corresponding LP problem does not get solved due to an algorithmic improvement.

Summary of the chapter

- All currently existing FCA methods have been reviewed.
- A new algorithm was presented, FFCA, which is based on feasibility-checking of LPs.
- We prove that in a network with no blocked reactions, every full coupling relation can be computed from the kernel of the stoichiometric matrix.
- We described additional mathematical results which reduce the number of LPs necessary for FCA.
- Using the mathematical results we have outlined a new algorithm for FCA, called F2C2.
- We have applied FFCA and F2C2 for large, genome-scale networks. The computational results show that FFCA is slightly faster than the previously described FCA methods. F2C2 is orders of magnitude faster than FFCA.

Chapter 3

Constrained Flux Coupling Analysis

Contents

Remark: This chapter summarizes and extends results originally published in [22].

3.1 Introduction

The previous chapter dealt with classical flux coupling analysis, which identifies dependencies between the activity of reaction fluxes in a metabolic network at steady-state. It can be used for exploring a large range of biological questions such as network evolution, gene essentiality, or gene regulation. While this has found many applications, it turns out that in some cases the flux cone is too general, describes too many metabolic behaviours, and lacks certain desired details. With FCA, we find a coupling relationship between two reactions only if these reactions are coupled to each other in all flux vectors belonging to the cone. Often however, biologists are interested in a particular subspace of the flux cone (e.g. the optimal flux space [53]) or would like to impose additional constraints on the reactions.

In this chapter, we generalize the concept of FCA by allowing additional linear constraints on the reactions. We show that these constraints can be modeled as lower and upper bounds on the reactions of an alternative metabolic network. Based on this, we introduce *constrained flux coupling analysis* (CFCA) and prove that, in many cases, knowing the true bounds of the reactions uniquely determines the constrained coupling relationships. An alternative approach to this problem is presented in [40]. Here, we present a new method to perform CFCA efficiently.

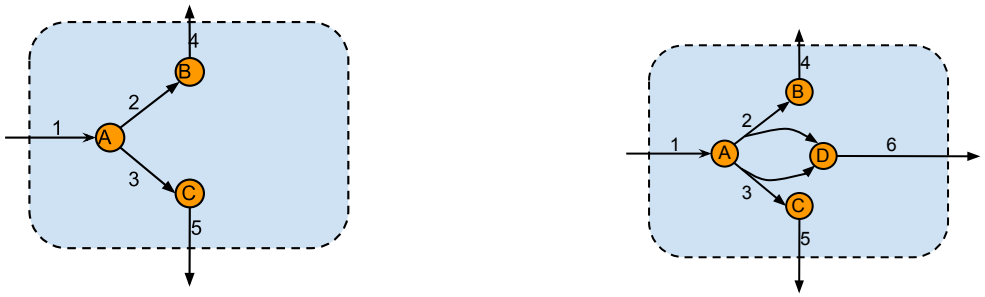


Figure 3.1: The intuitive interpretation of additional constraints

3.2 Methods

In constrained FCA, we would like to impose additional linear constraints on the reactions, which are of the form $\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n \geq \beta$, with $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ not all zero, and $\beta \in \mathbb{R}$. For algorithmic reasons, we will realize such a constraint by considering an alternative metabolic network. It is derived from the original one by introducing an artificial metabolite $m+1$ and an artificial reaction $n+1$. Every reaction $i \in \{1, \ldots, n\}$ for which $\alpha_i > 0$ (resp. $\alpha_i < 0$) will additionally produce (resp. consume) the metabolite $m+1$ with stoichiometric coefficient α_i. The reaction $n+1$ will be an exchange reaction that takes the metabolite $m+1$ out of the system. The mass-balance equation for metabolite $m+1$ is thus given by $\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n - v_{n+1} = 0$. The additional linear constraint is now realized by imposing the lower bound $v_{n+1} \geq \beta$ in the alternative network. The concept is illustrated in Fig. 3.1. Assume we start with the metabolic network on the left and want to impose the additional constraint $v_2 + v_3 \leq 10$. Considering the network on the right and imposing $v_6 \leq 10$ will give the desired result. A similar transformation can also be done for equality constraints, by imposing β both as a lower and upper bound of v_{n+1}. We conclude that additional linear constraints may be realized by imposing lower and upper bounds on suitably defined reactions in an extended network.

3.2.1 Definitions

For a metabolic network with n reactions (possibly after reconfiguration), let L (resp. U) $\subseteq \{1, \ldots, n\}$ be the set of reactions i for which a finite lower bound $l_i \in \mathbb{R}$ (resp. upper bound $u_i \in \mathbb{R}$) has been defined. We will consider all flux vectors in the *(steady-state) flux polyhedron* P, defined by

$$P := \{v \in \mathbb{R}^n \mid Sv = 0,\ v_i \geq l_i, \text{ for all } i \in L,\ v_i \leq u_j, \text{ for all } j \in U\}. \quad (3.1)$$

Note that the standard thermodynamic constraints $v_{Irr} \geq 0$ have not been included here, because irreversible reactions can be defined by having the lower bound 0. Given the flux polyhedron P, we can construct an associated flux cone by defining the irreversible reactions. Let $Irr_P := \{i \in L \mid l_i \geq 0\}$. The *flux cone C_P associated with P* is defined as

$$C_P := \{v \in \mathbb{R}^n \mid Sv = 0,\ v_i \geq 0 \text{ for all } i \in Irr_P\}. \tag{3.2}$$

In the special case $U = \emptyset$ and $l_i = 0$ for all $i \in L$, the flux polyhedron P is equal to the flux cone C (with $Irr = L$). However, in general $P \subseteq C_P$ holds.

Blocked reactions are defined similarly to classical FCA [12]. Intuitively, these are the reactions that never participate in a steady-state flux vector.

Definition 3.1 (Blocked reaction)**.** *Given the flux polyhedron P, a reaction $i \in \{1, \ldots, n\}$ is blocked in P if $v_i = 0$, for all $v \in P$. Otherwise, i is unblocked in P.*

Next we define the constrained coupling relations..

Definition 3.2 (Constrained coupling relations)**.** *Given the flux polyhedron P, let i, j be two unblocked reactions in P. The constrained (un-)coupling relationships $\underset{c}{\longrightarrow}, \underset{c}{\longleftrightarrow}, \underset{c}{\Longleftrightarrow}, \underset{c}{\not\longrightarrow}$ and $\underset{c}{\not\longleftrightarrow}$ are defined by:*

- *$i \underset{c}{\longrightarrow} j$ (equiv. $j \underset{c}{\longleftarrow} i$) if for all $v \in P$, $v_i \neq 0$ implies $v_j \neq 0$.*
- *$i \underset{c}{\longleftrightarrow} j$ if for all $v \in P$, $v_i \neq 0$ is equivalent to $v_j \neq 0$.*
- *$i \underset{c}{\Longleftrightarrow} j$ if there exists $\lambda \neq 0$ such that for all $v \in P, v_j = \lambda v_i$.*
- *$i \underset{c}{\not\longrightarrow} j$ if there exists $v \in P$ such that $v_i \neq 0$ and $v_j = 0$.*
- *$i \underset{c}{\not\longleftrightarrow} j$ if both $i \underset{c}{\not\longrightarrow} j$ and $j \underset{c}{\not\longrightarrow} i$ hold.*

Reactions i and j are fully (resp. partially, directionally) coupled if the relation $i \underset{c}{\Longleftrightarrow} j$ (resp. $i \underset{c}{\longleftrightarrow} j$, $i \underset{c}{\longrightarrow} j$) holds. Otherwise, i and j are uncoupled, i.e., $i \underset{c}{\not\longrightarrow} j$ and $j \underset{c}{\not\longrightarrow} i$.

With $i \Longleftrightarrow j$, $i \longleftrightarrow j$, $i \longrightarrow j$, and $i \not\longrightarrow j$ we denote the corresponding (unconstrained) coupling relations in C_P, where P is replaced with C_P.

In the unconstrained case, fully coupled reactions belong to the same enzyme subset as introduced in [87].

3.2.2 Preprocessing

In a first preprocessing step, we aim to find the blocked reactions in P. As shown by computational experiments with FFCA [24] or F2C2 [66], genome-scale metabolic networks may contain several hundreds of blocked reactions. Eliminating these considerably reduces the network size.

Observation 3.3. *If a reaction is blocked in C_P, then it is also blocked in P.*

Observation 3.3 holds since $P \subseteq C_P$. Thus, non-existence of $v \in C_P$ with $v_i \neq 0$ implies the non-existence of such a vector in P. This allows us to detect and remove some blocked reactions in P by studying C_P. In particular, we can use the concept of dead-end metabolites from F2C2 [66] and perform a stoichiometric matrix-based search to find some of the blocked reactions in P.

Finding blocked reactions relates back to a more general problem. Even if a reaction is unblocked, there is no guarantee that it is able to display all the fluxes specified by its bounds. The network structure might further constrain reaction fluxes and prohibit them to attain their limit. This is different from the unconstrained case, where fluxes through unblocked reactions are scalable by any positive number. Therefore, we will compute tight bounds for every reaction, which we call the *true bounds* of the reaction.

Definition 3.4 (True bounds)**.** *$l_k^* \in \mathbb{R} \cup \{-\infty\}$ (resp. $u_k^* \in \mathbb{R} \cup \{\infty\}$) is called the true lower (resp. upper) bound of reaction k if $u_k^* \geq v_k \geq l_k^*$, for all $v \in P$, and if for all $c \in]l_k^*, u_k^*[$, there exists $v \in P$ with $v_k = c$.*

Computing the true lower and upper bound of a reaction k can be done by solving the following two linear programs:

$$\begin{aligned} l_k^* &= \min \{v_k : Sv = 0,\ v_i \geq l_i,\ \text{for all } i \in L,\ v_i \leq u_j,\ \text{for all } j \in U\}, \\ u_k^* &= \max \{v_k : Sv = 0,\ v_i \geq l_i,\ \text{for all } i \in L,\ v_i \leq u_j,\ \text{for all } j \in U\}. \end{aligned} \tag{3.3}$$

Having computed the true bounds for every reaction, we can equivalently characterize the flux polyhedron P as

$$P = \{v \in \mathbb{R}^n \mid Sv = 0,\ u_i^* \geq v_i \geq l_i^*,\ \text{for all } i \in \{1, \ldots, n\}\}. \tag{3.4}$$

Trivially, if $l_k^* = u_k^* = 0$, then k is blocked in P. Since blocked reactions can be removed from the network without altering the coupling relationships between other reactions, we will assume for the rest of this section that the constrained metabolic network does not contain such reactions.

Based on the true bounds (l_k^*, u_k^*), we will distinguish eight classes of (unblocked) reactions:

- type 1: $(-\infty, \infty)$
- type 2: $(-a, \infty)$ with $a > 0$.

- type 3: $(-a, b)$ with $b \geq a > 0$.
- type 4: $(0, \infty)$
- type 5: $(0, b)$ with $b > 0$.
- type 6: (a, ∞) with $a > 0$.
- type 7: (a, b) with $b > a > 0$.
- type 8: (a, a) with $a > 0$.

Different types could appear, but by conveniently reversing the direction of such reactions (i.e., by multiplying the corresponding column in the stoichiometric matrix with -1), all cases can be reduced to the previous eight. For example, by multiplying the column of a reaction that would be of type (-∞, 0), we get the type (0, ∞). For the remainder of this chapter we assume without loss of generality that the input network has been revised and only contains reactions of the previous eight types.

3.2.3 Algorithmic considerations

We now describe the connection between coupling relations in the flux polyhedron P and the associated flux cone C_P. As our results will show, in many cases it is enough to compute the flux coupling relations in C_P (using classical FCA) to obtain the constrained FCA relations for P.

Observation 3.5. *Consider a flux polyhedron P with no blocked reactions. If for two reactions i and j, $i \Longleftrightarrow j$ (resp. $i \longleftrightarrow j$, $i \longrightarrow j$) in C_P holds, then $i \underset{c}{\Longleftrightarrow} j$ (resp. $i \underset{c}{\longleftrightarrow} j$, $i \underset{c}{\longrightarrow} j$) also holds in P.*

Proof. The proof follows directly from $P \subseteq C_P$.

□

Obs. 3.5 asserts that if a coupling relation exists in the unbounded network, the same relation will be carried over to the constrained case. Thus, we only need to check whether $i \not\longrightarrow j$ in C_P becomes $i \underset{c}{\longrightarrow} j$ in P. If both $i \underset{c}{\longrightarrow} j$ and $j \underset{c}{\longrightarrow} i$ hold in P, then we also have to check whether $i \underset{c}{\Longleftrightarrow} j$ holds in P.

Proposition 3.6. *Consider a flux polyhedron P (with no blocked reactions), where every reaction is of type 1−7. Then there exists $v \in P$ with $l_i^* < v_i < u_i^*$, for all $i \in \{1, 2, ..., n\}$.*

Proof. We prove the statement by constructing a point that satisfies the conditions.

For all $i \in \{1, 2, ..., n\}$, let $a^i \in P$ be a vector in the flux polyhedron with $l_i^* < a_i^i < u_i^*$. By definition of l_i^* and u_i^* and since i is not of type 8, such vectors a^i always exist. Consider the convex combination of these vectors, $v := 1/n \sum_{i=1}^{n} a^i$.

Clearly, $v \in P$ and $l_i^* < v_i < u_i^*$, for all $i \in \{1, 2, ..., n\}$. □

Proposition 3.7. *Consider a flux polyhedron P with no blocked reactions. Assume every reaction is of type $1-7$. Then for any two reactions i and j, we have $i \underset{c}{\Longleftrightarrow} j$ in P if and only if $i \Longleftrightarrow j$ in C_P.*

Proof. "⇐" Observation 3.5

"⇒" Assume that $i \underset{c}{\Longleftrightarrow} j$ holds in P, and assume by contradiction that $i \Longleftrightarrow j$ does not hold in C_P. Then there $\lambda \in \mathbb{R}^*$, such that, for all $v \in P$, $v_i = \lambda v_j$ (1), and there exists $w \in C_P$ with $w_i \neq \lambda w_j$.

According to Proposition 3.6, there exists a point $z \in P$ with $l_i^* < z_i < u_i^*$ for all $i \in \{1, 2, ..., n\}$. Consider the point $\varepsilon w + z$, with $0 < \varepsilon \in \mathbb{R}$, where additionally $\varepsilon < min(\frac{l_i^* - z_i}{w_i} | \forall i, w_i < 0 \text{ and } l_i^* \neq -\infty)$ and $\varepsilon < min(\frac{u_i^* - z_i}{w_i} | \forall i, w_i > 0 \text{ and } u_i^* \neq \infty)$.

Clearly, $\varepsilon w + z$ satisfies the steady-state constraints, and by definition of ε, it satisfies the bounds for P. Thus, $\varepsilon w + z \in P$.

However, $\varepsilon w_i + z_i \neq \lambda(\varepsilon w_j + z_j)$, which is a contradiction to (1). □

Proposition 3.8. *Consider a flux polyhedron P with no blocked reactions. Assume every reaction is of type $1-5$. Then for any two reactions i and j, we have $i \underset{c}{\longrightarrow} j$ (resp. $i \underset{c}{\longleftrightarrow} j$) in P if and only if $i \longrightarrow j$ (resp. $i \longleftrightarrow j$) in C_P.*

Proof. It is enough to prove the statement for directional coupling.

"⇐" Observation 3.5

"⇒" Assume that $i \underset{c}{\longrightarrow} j$ holds in P, and assume by contradiction that $i \longrightarrow j$ does not hold in C_P. Then there exists $w \in C_P$ with $w_i \neq 0$ and $w_j = 0$. Now let $0 < \varepsilon \in \mathbb{R}$ such that $\varepsilon < min(\frac{l_i^*}{w_i} | \forall i, w_i < 0 \text{ and } l_i^* \neq -\infty)$ and $\varepsilon < min(\frac{u_i^*}{w_i} | \forall i, w_i > 0 \text{ and } u_i^* \neq \infty)$.

Since all reactions are of type $1-5$ and by definition of ε, $\varepsilon w \in P$, which is a contradiction to $i \underset{c}{\longrightarrow} j$. □

When one tries to extend Prop. 3.8 to include reactions of type 6 and 7 as well, unlike full coupling relations, directional couplings are not directly deducible from the unconstrained flux cone. That is, for two reactions i and j, it is possible to have $i \underset{c}{\longrightarrow} j$ in P and $i \not\longrightarrow j$ in C_P. This is most easily demonstrated on an example. Consider the four reactions from Fig. 3.2 with the following true bounds:

- reaction r_1: $l_1^* = 0$, $u_1^* = 1$

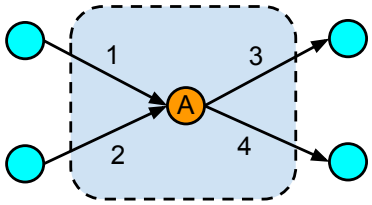


Figure 3.2: Illustrative metabolic network.

- reaction r_2: $l_2^* = 0$, $u_2^* = 3$
- reaction r_3: $l_3^* = 1$, $u_3^* = 2$
- reaction r_4: $l_4^* = 0$, $u_4^* = 1$

Notice that the lower bound of reaction r_3 is matching the upper bound of reaction r_1. Therefore, to carry flux through reaction r_4, reaction r_2 must be active. Hence $r_4 \underset{c}{\longrightarrow} r_2$ holds in P. However, in C_P every pair of reaction is uncoupled.

Together Obs. 3.5, Prop. 3.7 and Prop. 3.8 imply that if every reaction in the metabolic network is of type $1-5$, then it is enough to compute the unconstrained coupling relationships in C_P (using classical FCA). All constrained coupling relationships then can be obtained without solving additional linear programs. Additionally, if some reactions are of type 6 or 7, but none of them are type 8, we can still deduce all fully coupled relations from the unconstrained flux cone.

Thus, the only case to be further considered is the one where at least one reaction is of type $6-8$. While we can easily deduce the constrained coupling relationship between a type $6-8$ reaction and any other reaction in the network, the effect these reactions will have on the remaining constrained coupling relationships is not trivial.

Observation 3.9. *For a flux polyhedron P, let i be of type $6-8$ and let j be an unblocked reaction in P. Then $j \underset{c}{\longrightarrow} i$ holds.*

Observation 3.10. *For a flux polyhedron P, let i be of type $6-8$ and let j be of type $1-5$. Then $i \underset{c}{\not\longrightarrow} j$ holds.*

Proposition 3.11. *For a flux polyhedron P, let i and j be two reactions of type $1-3$. Then $i \underset{c}{\not\longrightarrow} j$ implies $j \underset{c}{\not\longrightarrow} i$.*

Proof. Note, the assumption that i and j are of type $1-3$ implies that $l_i^*, l_j^* < 0$ and $u_i^*, u_j^* > 0$.

Assume by contradiction that $i \underset{c}{\not\longrightarrow} j$ and $j \underset{c}{\longrightarrow} i$ both hold. Since $i \underset{c}{\not\longrightarrow} j$, then exists $v \in P$ with $v_i \neq 0$ and $v_j = 0$. Moreover, $j \underset{c}{\longrightarrow} i$ implies that there exist $w, z \in P$ with $w_i \neq 0, w_j > 0, z_i \neq 0$ and $z_j < 0$.

Assuming $sign(v_i)sign(w_i) = -1$, we can consider the point $x := \alpha v + (1-\alpha)w$ with $\alpha := \frac{w_i}{w_i - v_i}$. Since $\alpha \in]0,1[$ and $v, w \in P$, this implies $x \in P$. Moreover, $x_i = 0$ and $x_j > 0$ is contradicting $j \underset{c}{\longrightarrow} i$. Hence, $sign(v_i)sign(w_i) = 1$

A similar argument for the signs of v_i and z_i proves that $sign(v_i)sign(z_i) = 1$. Thus, the signs of v_i, w_i and z_i are all identical.

Now consider a point $y \in P$ with $sign(y_i)sign(v_i) = -1$. For y_j we distinguish two cases:

- $y_j \geq 0$: Let $x := \alpha y + (1-\alpha)w$ with $\alpha := \frac{w_i}{w_i - y_i}$. Clearly, $x_i = 0$ and $x_j > 0$, contradicting $j \underset{c}{\longrightarrow} i$
- $y_j < 0$: Let $x := \alpha y + (1-\alpha)z$ with $\alpha := \frac{z_i}{z_i - y_i}$. Clearly, $x_i = 0$ and $x_j < 0$, contradicting $j \underset{c}{\longrightarrow} i$ □

Corollary 3.12. *For a flux polyhedron P, let i and j be two reactions of type $1-3$. Then $i \underset{c}{\longrightarrow} j$ holds if and only if $j \underset{c}{\longrightarrow} i$ holds.*

Proof. The corollary follows trivially from Prop. 3.11. □

Proposition 3.13. *For a flux polyhedron P, let i and j be two reactions of type $1-3$ and assume their type is different. Then $i \underset{c}{\longleftrightarrow} j$ if and only if $i \underset{c}{\Longleftrightarrow} j$.*

Proof. "$\Leftarrow$" Trivial.

"$\Rightarrow$" Assume by contradiction that there exists $v, w \in P$ with $v_i w_j \neq v_j w_i$.

Let $x := \alpha v + (1-\alpha)w$ with $\alpha := \frac{w_i}{w_i - v_i}$. Since, $x_i = 0$ and $x_j \neq 0$, this is contradicting $i \underset{c}{\longleftrightarrow} j$. □

Proposition 3.14. *For a flux polyhedron P, let i be a reaction of type $1-3$, and j a reaction of different type. Then $j \underset{c}{\not\longrightarrow} i$ always holds.*

Proof. We prove the claim by breaking it down into several cases. Note, the assumption that i is of type $1-3$ implies that $l_i^* < 0$.

- If j is of type 6, 7 or 8 then for all $v \in P$, $v_j > 0$. However, there exists $v \in P$ with $v_i = 0$, which implies $j \underset{c}{\not\longrightarrow} i$.
- If j is of type 4 or 5, assume by contradiction that $j \underset{c}{\longrightarrow} i$. Let $v \in P$ with $v_j > 0$, then $v_i \neq 0$.

  Let $w \in P$ with $sign(w_i)sign(v_i) = -1$. Since reaction i can take both positive and negative values, such a w always exists. Moreover, since j

is of type $4-5$, $w_j \geq 0$. Let $\alpha := \frac{v_i}{v_i - w_i}$ and consider $z := \alpha w + (1-\alpha)v$. Since $\alpha \in]0,1[$ and $v, w \in P$, this implies $z \in P$. However, $z_i = 0$ and $z_j > 0$, which is contradicting $j \underset{c}{\longrightarrow} i$.

- If i is type 1 and j is of type 2 or 3, assume by contradiction that $j \underset{c}{\longrightarrow} i$.

  Let $v \in P$ with $v_j > 0$, then $v_i \neq 0$. We can assume that $v_i > 0$. The case $v_i < 0$, is almost identical to prove.

  Let $w \in P$ with $w_j < \frac{v_j}{v_i} l_i^*$.

  - If $w_i > 0$, consider $\alpha := \frac{v_j}{v_j - w_j}$ and $z := \alpha w + (1-\alpha)v$. Then $z_i > 0$ and $z_j = 0$, thus $i \underset{c}{\not\longrightarrow} j$. Based on Prop. 3.11, this in turn implies $j \underset{c}{\not\longrightarrow} i$.
  - If $w_i < 0$, consider $\alpha := \frac{v_i}{v_i - w_i}$ and $z := \alpha w + (1-\alpha)v$. Then $z_j \neq 0$ and $z_i = 0$, thus $j \underset{c}{\not\longrightarrow} i$.

- The proof of the remaining cases is similar to the previous ones.

□

Corollary 3.15. *For a flux polyhedron P, let i and j be two reactions of type $1-3$ and assume their type is different. Then $i \underset{c}{\not\longleftrightarrow} j$ holds.*

Proof. The corollary follows trivially from Prop. 3.14. □

Proposition 3.16. *For a flux polyhedron P, let i be a reaction of type 4 and j be a reaction of type 5. Then $i \underset{c}{\not\longrightarrow} j$ holds.*

Proof. Assume by contradiction that $i \underset{c}{\longrightarrow} j$ holds, and consider the following linear program:

$$\max \{v_i : Sv = 0,\ u_k^* \geq v_k \geq l_k^*, \forall k \in \{1, \ldots, n\}\} \tag{3.5}$$

Since i is of type 4, the above LP is unbounded. Hence solving it with the Simplex algorithm results in an extreme ray d. That is, for all $x \in P$ and for all $\theta > 0$ we have $x + \theta d \in P$.

Clearly, $d_i > 0$. Also, $d_j = 0$, since otherwise for a large enough θ, the feasibility in the j-th coordinate would be violated.

Now let $v \in P$, such that $v_j = 0$, and consider $z := v + d$. By definition of d, it follows that $z \in P$ with $z_i > 0$ and $z_j = 0$, thus contradicting $i \underset{c}{\longrightarrow} j$.

□

Proposition 3.17. *For a flux polyhedron P, let i and j be two unblocked reactions in P. If $i \underset{c}{\Longleftrightarrow} j$ holds, then i and j are of the same type.*

Proof.

- By Prop. 3.14, if one of the reactions is of type $1-3$ and the other is of different type, then they cannot be fully coupled.

- By Obs. 3.10, if one of the reactions is of type $6-8$ and the other is of type $1-5$, then full coupling is excluded.

- By Prop. 3.16, if one reaction is of type 4 and the other is of type 5, then they cannot be fully coupled.

The only cases not covered yet occur if both reactions are of type $6-8$, and different in type.

- If one of the reactions is of type 8, then the result is trivial.

- Here we prove the case when (w.l.o.g.) reaction i is of type 6 and j is of type 7.

  Consider solving the LP problem (3.5) with the Simplex algorithm. Again, since the LP is unbounded, the solver would stop when it finds an extreme ray d. Following the same argument as in Prop. 3.16, we get $d_i > 0$ and $d_j = 0$. Now considering any point $v \in P$ and $v + d \in P$, obviously contradicts $i \underset{c}{\Longleftrightarrow} j$.

□

The previous results are summarized in Tab. 3.1. White cells mark the cases where the constrained flux coupling relationship is uniquely determined by the reaction types, whereas gray cells represent the cases where additional linear programs have to be solved. Note that there are only two cases where any coupling can appear. In all other cases, the coupling is either uniquely determined or corresponds to one out of two options. The lower diagonal part of the table was left blank for simplicity as it is reverse symmetrical to the upper diagonal part.

For the entries in Tab. 3.1 where the coupling is not uniquely determined, solving additional linear programs (LPs) is necessary. Consider the following five LPs, where parameters i and j are two reactions and $\lambda \in \mathbb{R}^*$ is a constant

Table 3.1: Summary of the deducible constrained coupling relationships based on the reaction type

i \ j	1	2	3	4	5	6	7	8
1	$i \underset{c}{\not\longleftrightarrow} j$ $i \underset{c}{\Longleftrightarrow} j$	$i \underset{c}{\not\longleftrightarrow} j$	$i \underset{c}{\not\longleftrightarrow} j$	$i \underset{c}{\not\longleftrightarrow} j$ $i \underset{c}{\longrightarrow} j$	$i \underset{c}{\not\longleftrightarrow} j$	$i \underset{c}{\longrightarrow} j$	$i \underset{c}{\longrightarrow} j$	$i \underset{c}{\longrightarrow} j$
2	$i \underset{c}{\not\longleftrightarrow} j$ $i \underset{c}{\Longleftrightarrow} j$	$i \underset{c}{\not\longleftrightarrow} j$	$i \underset{c}{\not\longleftrightarrow} j$	$i \underset{c}{\not\longleftrightarrow} j$ $i \underset{c}{\longrightarrow} j$	$i \underset{c}{\not\longleftrightarrow} j$	$i \underset{c}{\longrightarrow} j$	$i \underset{c}{\longrightarrow} j$	$i \underset{c}{\longrightarrow} j$
3	$i \underset{c}{\not\longleftrightarrow} j$ $i \underset{c}{\Longleftrightarrow} j$	$i \underset{c}{\not\longleftrightarrow} j$ $i \underset{c}{\longrightarrow} j$	$i \underset{c}{\not\longleftrightarrow} j$ $i \underset{c}{\longrightarrow} j$	$i \underset{c}{\longrightarrow} j$	$i \underset{c}{\longrightarrow} j$	$i \underset{c}{\longrightarrow} j$	$i \underset{c}{\longrightarrow} j$	
4	$i \underset{c}{\not\longleftrightarrow} j$ $j \underset{c}{\longrightarrow} i$	$i \underset{c}{\longrightarrow} j$	$i \underset{c}{\longrightarrow} j$	$i \underset{c}{\longrightarrow} j$				
5	any	$i \underset{c}{\longrightarrow} j$	$i \underset{c}{\longrightarrow} j$	$i \underset{c}{\longrightarrow} j$				
6	$i \underset{c}{\longleftrightarrow} j$ $i \underset{c}{\Longleftrightarrow} j$	$i \underset{c}{\longleftrightarrow} j$	$i \underset{c}{\longleftrightarrow} j$					
7	$i \underset{c}{\longleftrightarrow} j$ $i \underset{c}{\Longleftrightarrow} j$	$i \underset{c}{\longleftrightarrow} j$						
8	$i \underset{c}{\Longleftrightarrow} j$							

value:

$$minLP1(i,j) = \min\{v_i : Sv = 0,\ v_j = 0,\ u_k^* \geq v_k \geq l_k^*, \forall k \in \{1,\ldots,n\}\} \tag{3.6}$$

$$maxLP1(i,j) = \max\{v_i : Sv = 0,\ v_j = 0,\ u_k^* \geq v_k \geq l_k^*, \forall k \in \{1,\ldots,n\}\} \tag{3.7}$$

$$minLP2(i,j,\lambda) = \min\{v_i - \lambda v_j : Sv = 0,\ u_k^* \geq v_k \geq l_k^*, \forall k \in \{1,\ldots,n\}\} \tag{3.8}$$

$$maxLP2(i,j,\lambda) = \max\{v_i - \lambda v_j : Sv = 0,\ u_k^* \geq v_k \geq l_k^*, \forall k \in \{1,\ldots,n\}\} \tag{3.9}$$

$$minLP3(i,j) = \min\{v_i : Sv = 0,\ v_j = 1,\ u_k^* \geq v_k \geq l_k^*, \forall k \in \{1,\ldots,n\}\} \tag{3.10}$$

Based on Tab. 3.1 we distinguish between the following cases:

1. If only $i \underset{c}{\not\longleftrightarrow} j$ or $i \underset{c}{\Longleftrightarrow} j$ is possible (entries (1, 1), (2, 2) and (3, 3)) then to find which coupling applies, it is enough to solve two LPs: $minLP1(i,j)$ and $maxLP1(i,j)$. If the optimal solution to either LP is nonzero then $i \underset{c}{\not\longleftrightarrow} j$, otherwise $i \underset{c}{\Longleftrightarrow} j$ holds. Clearly, if the first

LP already yields a nonzero optimum then solving the second LP is not necessary.

2. If $i \underset{c}{\not\longleftrightarrow} j$ or $i \underset{c}{\longrightarrow} j$ is possible (entries (1, 4), (2, 4), (3, 4), (3, 5)) then we can find out the coupling relation between the two reactions by solving the same two LPs as in the previous case. If the optimal solution to either LP is nonzero then $i \underset{c}{\not\longleftrightarrow} j$, otherwise $i \underset{c}{\longrightarrow} j$ holds.

3. If $i \underset{c}{\not\longleftrightarrow} j$ or $j \underset{c}{\longrightarrow} i$ are the possible couplings (entry (4, 5)) then solving $maxLP1(j, i)$ suffices. If the optimal solution to this LP is 0, then $j \underset{c}{\longrightarrow} i$, otherwise $i \underset{c}{\not\longleftrightarrow} j$ holds.

4. When $i \underset{c}{\longleftrightarrow} j$ or $i \underset{c}{\Longleftrightarrow} j$ are the only possible coupling relations (entries (6, 6), (7, 7)), we can decide between the two cases by solving $minLP2(i, j, \lambda)$ and $maxLP2(i, j, \lambda)$, where $\lambda := l_i^*/l_j^*$. If both LPs have zero as their optimal value then $i \underset{c}{\Longleftrightarrow} j$, otherwise $i \underset{c}{\longleftrightarrow} j$ holds. For entry (7, 7) if $l_i^*/l_j^* \neq u_i^*/u_j^*$, then no LP needs to be solved since the two reactions are partially coupled.

5. For entry (5, 5) any coupling relation is possible, thus we first solve $maxLP1(i, j)$ and $maxLP1(j, i)$. Based on the optimal solutions we proceed as follows:

   - if $maxLP1(i, j) \neq 0$ and $maxLP1(j, i) \neq 0$ then $i \underset{c}{\not\longleftrightarrow} j$ holds.
   - if $maxLP1(i, j) \neq 0$ and $maxLP1(j, i) = 0$ then $j \underset{c}{\longrightarrow} i$ holds.
   - if $maxLP1(i, j) = 0$ and $maxLP1(j, i) \neq 0$ then $i \underset{c}{\longrightarrow} j$ holds.
   - if $maxLP1(i, j) = 0$ and $maxLP1(j, i) = 0$ then either $i \underset{c}{\longleftrightarrow} j$ or $i \underset{c}{\Longleftrightarrow} j$ holds. With $\lambda := u_i^*/u_j^*$, we propose solving $minLP2(i, j, \lambda)$ and $minLP2(i, j, \lambda)$. If both LPs have zero as their optimal value then $i \underset{c}{\Longleftrightarrow} j$, otherwise $i \underset{c}{\longleftrightarrow} j$ holds.

6. In the case of entry (4, 4) we proceed similarly as in the previous case with the exception that a value for λ can't be deduced from the true bounds of the reactions. Hence, after solving the first two LPs, if the two reactions are at least partially coupled we additionally solve $minLP3(i, j)$ and use its optimal value for λ.

In the F2C2 algorithm, the most important single improvement was obtained by the feasibility rule (Observation 6 in [66]). A similar observation can be made in the constrained case.

Table 2: Summary of the number of LPs to be solved in the different cases of Table 3.1

i \ j	1	2	3	4	5	6	7	8
1	1-2	0	0	1-2	0	0	0	0
2		1-2	0	1-2	0	0	0	0
3			1-2	1-2	1-2	0	0	0
4				2-5	1	0	0	0
5					2-4	0	0	0
6						1-2	0	0
7							1-2	0
8								0

Observation 3.18 (Constrained feasibility rule)**.** *For any $v \in P$ let $I = \{i \mid v_i \neq 0\}$ and $J = \{j \mid v_j = 0\}$. Then $i \underset{c}{\not\longrightarrow} j$ for all $(i, j) \in I \times J$.*

The transitive nature of flux coupling is preserved by Def. 3.2. Thus similar transitivity rules can be derived as for F2C2 [66]. For three reactions i, j and k, it is sometimes possible to derive a coupling relationship between j and k, based on the coupling relationship for reactions i and j, and reactions i and k. Table 3.3 summarizes these inference rules.

Table 3.3: Transitivity inferred constrained flux (un)coupling

Known flux coupling	$i \underset{c}{\Longleftrightarrow} j$	$i \underset{c}{\longleftrightarrow} j$	$j \underset{c}{\longrightarrow} i$	$i \underset{c}{\longrightarrow} j$
$i \underset{c}{\Longleftrightarrow} k$	$k \underset{c}{\Longleftrightarrow} j$	$k \underset{c}{\longleftrightarrow} j$	$j \underset{c}{\longrightarrow} k$	$k \underset{c}{\longrightarrow} j$
$i \underset{c}{\longleftrightarrow} k$	$j \underset{c}{\longleftrightarrow} k$	$j \underset{c}{\Longleftrightarrow} k$ $j \underset{c}{\longleftrightarrow} k$	$j \underset{c}{\longrightarrow} k$	$k \underset{c}{\longrightarrow} j$
$i \underset{c}{\longrightarrow} k$	$j \underset{c}{\longrightarrow} k$	$j \underset{c}{\longrightarrow} k$	$j \underset{c}{\longrightarrow} k$	
$k \underset{c}{\not\longrightarrow} i$	$k \underset{c}{\not\longrightarrow} j$	$k \underset{c}{\not\longrightarrow} j$		$k \underset{c}{\not\longrightarrow} j$
$i \underset{c}{\not\longrightarrow} k$	$j \underset{c}{\not\longrightarrow} k$	$j \underset{c}{\not\longrightarrow} k$	$j \underset{c}{\not\longrightarrow} k$	

The main steps of CFCA are summarized in Tab. 3.4. In the worst case, the algorithm has to solve $2n + n(n-1)$ linear programs.

Table 3.4: Main steps of the CFCA algorithm

Step	Rule
1.	Iteratively remove dead-end metabolites and incident reactions to them.
2.	Classify the reactions based on their true lower and upper bounds; remove the remaining blocked reactions.
3.	Compute the flux coupling relationships of C_P with F2C2.
4.	If every reaction is of type 1-5, use Prop. 3.5, Prop. 3.7 and Prop. 3.8 to deduce all constrained coupling relationships, STOP.
5.	If every reaction is of type 1-7, use Prop. 3.5 and Prop. 3.7 to deduce all full coupling relationships.
5.	If the constrained coupling relationship for every pair of reactions has been computed, STOP.
6.	Select a pair of reactions i and j for which a coupling has not yet been determined.
7.	Use Tab. 3.1 and solve corresponding LPs to determine the coupling between i and j.
8.	For every feasible vector computed in step 7, use the constrained feasibility rule to derive additional constrained uncoupling relationships.
9.	For every new constrained (un-)coupling relation computed in steps 7 and 8, use the transitivity rules to derive additional coupling relations.
10.	Goto step 5.

3.3 Results and conclusions

The CFCA algorithm has been implemented in Matlab [74], with CLP [18] as linear programming solver. We applied CFCA to the *E. coli* core metabolism [83], a network with 76 reactions and a biomass function. The glucose uptake was set to $10\,mmol/(gDW \cdot hr)$, and a minimum ATP production of $7.6\,mmol/(gDW \cdot hr)$ was required, which corresponds to the associated non-growth maintenance cost [83]. All other irreversible reactions were constrained to the interval $[0, 1000]$, while reversible reactions were limited to $[-1000, 1000]$.

Tab. 3.5 summarizes the coupling relations of the constrained network. For comparison, we also performed standard FCA on the unbounded network using F2C2.

Many of the new directional coupling relations result from the fact that the glucose uptake and ATP producing reactions were required to have a non-zero flux. Hence, every other reaction will be coupled to these. After subtracting the corresponding 2×76 directional couplings from the total number of 401, there still remain 249 new directional couplings between other pairs of reactions.

We conclude that CFCA is a promising new tool for studying coupling

Table 3.5: Coupling relations in the *E. coli* core metabolism

Flux coupling relations	Directional	Partial	Full	Blocked reactions
CFCA	401	4	40	2
FCA	10	0	38	0

relations in metabolic networks under more general conditions than classical FCA is able to do.

Summary of the chapter

- We introduce the concept of Constrained Flux Coupling Analysis (CFCA).
- Linear constraints on the reactions are shown to be equivalent to bounds on the reactions.
- A classification of reactions into 8 types is given.
- We show that in most cases, reaction types determine the constrained coupling relations.
- We describe mathematical results with which, under certain circumstances, one can deduce CFCA relations from FCA.
- The CFCA algorithm, to compute constrained flux coupling relations, is given.
- CFCA has been applied to a real metabolic network, and we show that more coupling relations can be computed than with unconstrained FCA.

Chapter 4

Metabolite Activity Coupling

Contents

4.1 Introduction

One of the advantages of steady-state modeling of metabolic networks is that it does not require explicit information about the concentration levels of internal metabolites. We ask the question whether without knowledge about metabolic concentrations, would it be possible to characterize the implications that exist between the activity of metabolites. In other words, if we know that a specific metabolite is actively being produced and consumed (and kept in steady-state), can we find a list of which other metabolites are necessarily produced and consumed in the system. Answers to this question are not explicitly captured by flux coupling analysis or metabolic pathway analysis. Indeed, looking at Fig. 4.1, we realize that in this example, all reactions in the network are uncoupled to each other. While studying the whole set of elementary modes could in theory answer such questions, it is often infeasible to enumerate all elementary modes, due to their exponential number. In the small artificial example in Fig. 4.1, there are 20 reactions which lead to 1024 elementary modes. Analyzing the figure by hand, intuitively one feels that there is a certain structure to the metabolites that can be captured and formalized. Any steady-state flux vector that has metabolite A as an intermediary product, also has metabolites B, C and D as intermediary products. In this section, in a similar fashion to flux coupling analysis, we will formally define this type of interdependent relationship between metabolites and we will call it *metabolite activity coupling* (MAC). We then present a linear programming based method to compute these relationships.

The term metabolic coupling appears in several different contexts in the scientific literature. Baldazzi et al. [5] discuss the importance of metabolic

coupling in the context of gene regulatory networks. In contrast, Becker et al. [7] derive metabolic coupling from the stoichiometric matrices of metabolic networks. Their method identifies pairs of metabolites that frequently co-appear in the same reaction either as reactants or products. Therefore, this method considers metabolic coupling to be a local property of the network.

The work that is closest to our intention is by Nikolaev et al. [77], who introduce the concept of *metabolite concentration coupling analysis* (MCCA). They show that in closed metabolic networks, the stoichiometric coefficients uniquely determine all conservation relationships for metabolite concentrations (Theorem 4.2).

Definition 4.1. *A metabolic network is closed if it contains no boundary reactions.*

Theorem 4.2 ([77])**.** *Consider an arbitrary closed metabolic network with stoichiometric matrix $S \in \mathbb{R}^{m\times n}$ and metabolite concentrations $C(i)$ for all $i \in M$. Any nonnegative linear combination of the system's metabolite concentrations $\sum_{i=1}^{m} \beta_i C(i)$ (with $\beta_1, ..., \beta_m \in \mathbb{R}_+$) is constant if and only if the coefficients β_i introduce linear dependencies between rows of the system's stoichiometric matrix S (i.e., satisfy linear equations $\sum_{i=1}^{m} \beta_i S_{ij} = 0$ for all $j \in \{1, 2, ..., n\}$ with some $\beta_i \neq 0$).*

Proof. See Appendix A in [77]. □

The previous theorem is closely related to the concept of *p-invariants* in Petri net theory, where p-invariants can be used to express conservations relations between the metabolites of a metabolic network [43, 131].

Based on Theorem 4.2, for every pair of metabolites m_i and m_j, Nikolaev et al. [77] propose solving a minimization and a maximization linear programming problem as shown in (LP0):

$$
\text{(LP0)} \qquad
\begin{array}{lll}
\min/\max & \beta_j & \\
\text{s.t.} & \beta_i = 1 & \\
& \sum_{k=1}^{m} \beta_k S_{kl} = 0 & \forall l \in \{1, 2, ..., n\} \\
& \beta_k \geq 0 & \forall k \in \{1, 2, ..., m\}
\end{array}
$$

Definition 4.3 ([77])**.** *Let m_i, $m_j \in M$ be two metabolites and let β_j^{min} and β_j^{max} be the optimal solutions to LP0. Then the metabolite concentration coupling (MCC) relations are defined as follows:*

- *m_i and m_j are fully coupled if and only if $\beta_j^{min} = \beta_j^{max} = c$, where c is a finite nonzero constant.*
- *m_i and m_j are partially coupled if and only if β_j^{min} and β_j^{max} are distinct finite nonzero constants.*

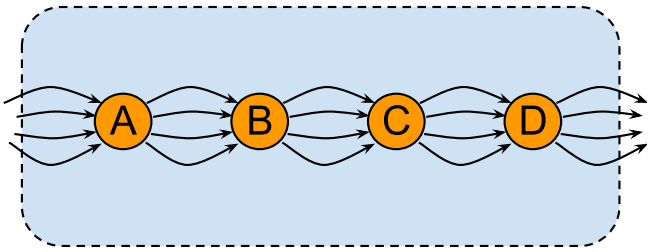


Figure 4.1: Example network

- *m_i is directionally coupled to m_j if and only if β_j^{max} is a finite non-zero constant.*
- *m_j is directionally coupled to m_i if and only if β_j^{min} is a finite non-zero constant.*

Looking at the previous definition, MCC relations are closely related to FCA relations. In fact, we can obtain all MCC relations by running an FCA tool on the modified network $N' := (S^T, \{1, 2, ..., m\})$, that is the network that we obtain by exchanging all the reactions with all the metabolites while preserving the corresponding stoichiometric values and having all new reactions as irreversible.

While metabolic concentration coupling describes a somewhat similar relationship that we intuitively intended to convey, in practice the concept that we are about to introduce is rather different. We will discuss the differences more in Section 4.3, after metabolite activity coupling has been formally introduced.

4.2 Methods

4.2.1 Definitions

For the work in this chapter we return to the unbounded steady-state flux space. Having the set of metabolites $M = \{1, 2, ..., m\}$, and the set of reactions $R = \{1, 2, ..., n\}$, we assume to know a metabolic network model, given by its stoichiometric matrix ($S \in \mathbb{R}^{m \times n}$) and the set of irreversible reactions ($Irr \subseteq R$), the two of which define the steady-state flux cone C (see Def. 1.40).

For any reaction r, we can refer to the set of incident metabolites to it, by looking for nonzero entries in the corresponding column of the stoichiometric matrix.

Definition 4.4. *For a given reaction $r \in R$, let*

- $Met^+(r) := \{i \in M \mid S_{i,r} > 0\}$.

- $Met^-(r) := \{i \in M \mid S_{i,r} < 0\}$.
- $Met(r) := \{i \in M \mid S_{i,r} \neq 0\}$.

$Met^+(r)$ is the set of metabolites produced while $Met^-(r)$ is the set of metabolites consumed by reaction r. $Met(r)$ is the total set of metabolites incident to r.

We generalize this definition to steady-state flux vectors. Any metabolite that is the intermediary product of a metabolic pathway will be called active in the pathway.

Definition 4.5. *For a given flux vector $v \in C$, let $Met(v) := \bigcup_{j \in supp(v)} Met(r_j)$. $Met(v)$ is the set of active metabolites in v.*

Conversely, we can also define the incident reactions to a metabolite. We distinguish between reactions that produce ($Reac^+(m_i)$) and consume ($Reac^-(m_i)$) said metabolite, the union of the two sets giving all incident reactions to it ($Reac(m_i)$).

Definition 4.6. *For a given metabolite $m \in M$, let*

- $Reac^+(m) := \{j \in R | S_{m,j} > 0\}$.
- $Reac^-(m) := \{j \in R | S_{m,j} < 0\}$.
- $Reac(m) := \{j \in R | S_{m,j} \neq 0\}$.

Note that reversible reactions might consume or produce a given metabolite, depending on their sign in a particular steady-state flux vector. Our definition disregards reaction reversibility, and classifies reactions purely based on stoichiometric values.

In the following we will formally define the concept of blocked metabolites and metabolite activity coupling.

Definition 4.7 (Blocked metabolite)**.** *A metabolite $m \in M$ is blocked if and only if $m \notin Met(v)$, for all $v \in C$.*

We can define the directional metabolite activity coupling as follows.

Definition 4.8. *Let m_i, $m_j \in M$ be two non-blocked metabolites. The activity of metabolite m_i is directionally coupled to the activity of metabolite m_j, if for all $v \in C$, $m_i \in Met(v) \Rightarrow m_j \in Met(v)$. We shortly say that m_i is directionally coupled to m_j.*

Partial metabolite activity coupling is defined similarly as it was for FCA, that is, partial coupling holds if and only if the two metabolites are mutually directionally coupled.

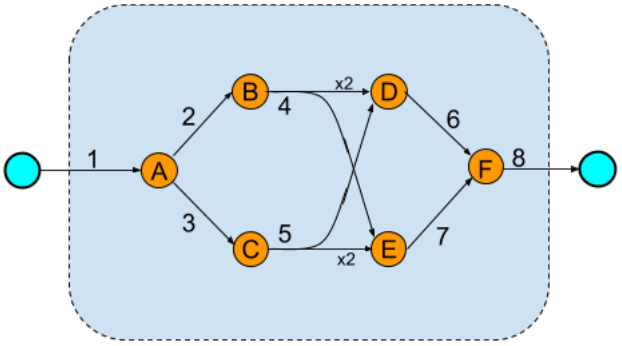


Figure 4.2: Example of the four metabolite activity coupling types. Metabolite B is *directionally coupled* to metabolite A. Metabolite A is *partially coupled* to metabolite D. Metabolite A is *fully coupled* to metabolite F. Metabolites B and C are *uncoupled*.

Definition 4.9. *Let m_i, $m_j \in M$ be two non-blocked metabolites. The activity of a metabolite m_i is partially coupled to the activity of metabolite m_j, if m_i is directionally coupled to m_j, and m_j is directionally coupled to m_i.*

Partial coupling can be extended to full coupling if we additionally require our metabolites to be produced and consumed proportionally to each other over the whole flux space.

Definition 4.10. *Let m_i, $m_j \in M$ be two non-blocked metabolites. The activity of metabolite m_i is fully coupled to the activity of metabolite m_j if and only if there exists $\lambda \in \mathbb{R}$ such that for all $v \in C$, $\sum_{k \in Reac(m_i)} |v_k S_{ik}| = \lambda \sum_{k \in Reac(m_j)} |v_k S_{jk}|$.*

If both metabolites are only incident to irreversible reactions (i.e., $Reac(m_i) \subseteq Irr$ and $Reac(m_j) \subseteq Irr$), we can give an equivalent definition for full coupling, using only the incoming (or outgoing) reactions. This is possible since $\sum_{k \in Reac^+(m_i)} |v_k S_{ik}| = \sum_{k \in Reac^-(m_i)} |v_k S_{ik}| = 0.5 \sum_{k \in Reac(m_i)} |v_k S_{ik}|$ for any metabolite m_i and flux vector $v \in C$. This definition would not work in general, when either of the metabolites is adjacent to at least one reversible reaction. This can also be exemplified in Fig. 4.3, where we see two versions of a metabolic network, one with irreversible reactions and one with reversible reactions. In both cases the two internal metabolites are fully coupled with each other. However:

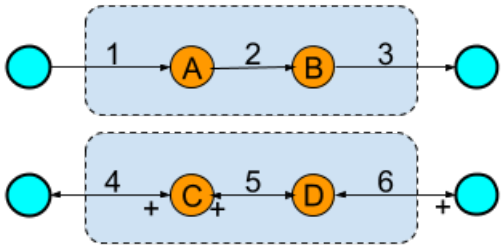


Figure 4.3: Example of two metabolic networks, where one (top) has only irreversible reactions and the second one (bottom) has only reversible reactions. The forward orientation of each reversible reaction is marked with a + sign.

- In the first network (top) $\sum_{k \in Reac(A)} |v_k S_{ik}| = 2 * \sum_{k \in Reac^+(A)} |v_k S_{ik}| = 2 * v_1$ holds, therefore we can measure the activity of metabolite A with an expression not involving absolute values.
- In the second network (bottom) $\sum_{k \in Reac(A)} |v_k S_{ik}|$ does not equal to $2 * \sum_{k \in Reac^+(A)} |v_k S_{ik}| = 2 * |v_1| + 2 * |v_2|$. In this example, choosing reaction r_5 with reverse forward direction would have alleviated the problem. However, when only the stoichiometric matrix is given, we have no ground on how to choose a 'correct' orientation for the reversible reactions.

Definition 4.11 (Metabolite activity coupling relationships)**.** *Let m_i, $m_j \in M$ be two non-blocked metabolites. We will use the following notation for the metabolite activity coupling relationships:*

- $m_i \underset{m}{\longrightarrow} m_j$ *if m_i is directionally coupled to m_j.*
- $m_i \underset{m}{\longleftrightarrow} m_j$ *if m_i and m_j are partially coupled.*
- $m_i \underset{m}{\Longleftrightarrow} m_j$ *if m_i and m_j are fully coupled.*
- $m_i \underset{m}{\not\longleftrightarrow} m_j$ *if m_i and m_j are uncoupled (i.e., neither $m_i \underset{m}{\longrightarrow} m_j$ nor $m_j \underset{m}{\longrightarrow} m_i$ holds).*

Figure 4.2 gives an example of each of the above four coupling types.

4.2.2 LP and MIP based algorithms

In the following, we present linear programming based solutions to decide which coupling type is present between two metabolites m_i and m_j. Two

cases are considered. In the first case, both metabolites will be incident to only irreversible reactions, while in the second case reversible reactions are considered as well.

Finding blocked metabolites

In the first step of the MAC algorithm, we need to find which metabolites are blocked. Using the definition of blocked metabolites, we can find these by computing the set of blocked reactions, Blk (see Chap. 2). Then, a metabolite m_i is blocked if and only if $Reac(m_i) - Blk = \emptyset$.

Finding metabolite couplings

Case 1: $Reac(m_i) \subseteq Irr$ **and** $Reac(m_j) \subseteq Irr$

Notice that for any metabolite m_i with $Reac(m_i) \subseteq Irr$ and any $v \in C$ we have $\sum_{k \in Reac(m_i)} |v_k S_{ik}| = \sum_{k \in Reac^+(m_i)} v_k S_{ik} - \sum_{k \in Reac^-(m_i)} v_k S_{ik} = 2 * \sum_{k \in Reac^+(m_i)} v_k S_{ik}$. Hence, when measuring the activity of a metabolite in this case we can omit the absolute values from the sum.

Now consider the following two linear programs.

$$\text{(LP1)} \quad \begin{array}{lrl} \min & \sum_{k \in Reac^+(m_i)} v_k S_{ik} & \\ \text{s.t.} & Sv & = 0 \\ & v_{Irr} & \geq 0 \\ & \sum_{l \in Reac^+(m_j)} v_j S_{jl} & = 1 \end{array}$$

$$\text{(LP2)} \quad \begin{array}{lrl} \min & \sum_{l \in Reac^+(m_j)} v_l S_{jl} & \\ \text{s.t.} & Sv & = 0 \\ & v_{Irr} & \geq 0 \\ & \sum_{k \in Reac^+(m_i)} v_k S_{ik} & = 1 \end{array}$$

Let o_{ij} be the optimal value of $LP1$ and o_{ji} be the optimal value of $LP2$. Based on o_{ij} and o_{ji} we can draw the following conclusions:

- $m_i \underset{m}{\longrightarrow} m_j$ holds if and only if $o_{ji} > 0$.
- $m_j \underset{m}{\longrightarrow} m_i$ holds if and only if $o_{ij} > 0$.
- $m_j \underset{m}{\longleftrightarrow} m_i$ holds if and only if $o_{ij} > 0$ and $o_{ji} > 0$.
- $m_j \underset{m}{\Longleftrightarrow} m_i$ holds if and only if $o_{ij} o_{ji} = 1$.

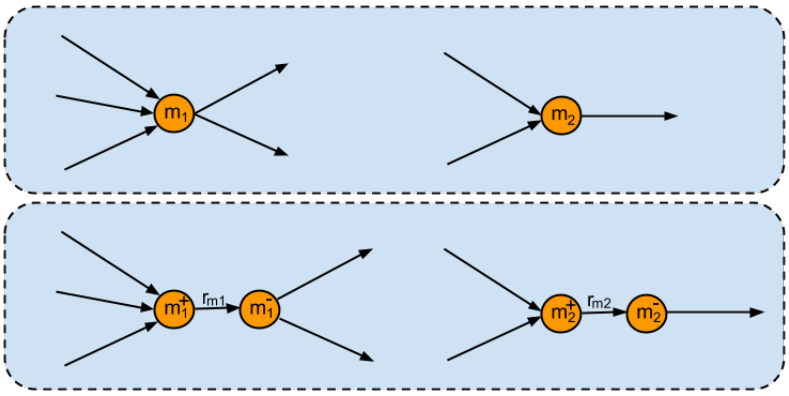


Figure 4.4: A possible interpretation of $LP1$ and $LP2$. **Top:** m_1 and m_2 are metabolites incident to only irreversible reactions. **Bottom:** Both m_1 and m_2 have been split and an artificial reaction joins the two parts.

Interestingly, $LP1$ and $LP2$ have an intuitive, alternative interpretation as well. Assume we replace metabolite m_i with two other metabolites, m_i^+ and m_i^- in such a way that $Reac(m_i^+) = Reac^+(m_i)$ and $Reac(m_i^-) = Reac^-(m_i)$. Moreover, we add an artificial reaction r_{m_i} that consumes m_i^+ and produces m_i^-. If we proceed similarly for m_j, then LP1 and LP2 are equivalent to the LPs that would be solved in a standard FCA for reactions r_{m_i} and r_{m_j}. Figure 4.4 illustrates the procedure.

Case 2: $Reac(m_i) \not\subseteq Irr$ **or** $Reac(m_j) \not\subseteq Irr$

Trying to apply $LP1$ and $LP2$ in this case is not feasible, since reversible reactions do not enforce the directionality of the fluxes producing m_i and m_j (i.e., if metabolite m_i^+ is produced, it is not forced to be consumed by reaction r_{m_i}, a reversible reaction incident to m_i^+ can consume it). Thus, an activity for the metabolite m_i^+ does not imply an activity of m_i^- and vice versa. In several other methods, a common way to deal with reversible reactions is to split them into two opposite irreversible reactions. This type of solution is not a viable option for us. As presented in Figure 4.5, assuming reversible reactions are split, these would induce three-cycles, including the artificial reaction r_{m_i} (resp. r_{m_j}). The artificial cycle then induces an uncoupling relationship between r_{m_i} and r_{m_j}, hiding the real coupling that might exist between the two.

We propose the following alternative sequence of linear programs. For metabolite m_i and every reaction $k \in Reac(m_j)$, solve $LP3$.

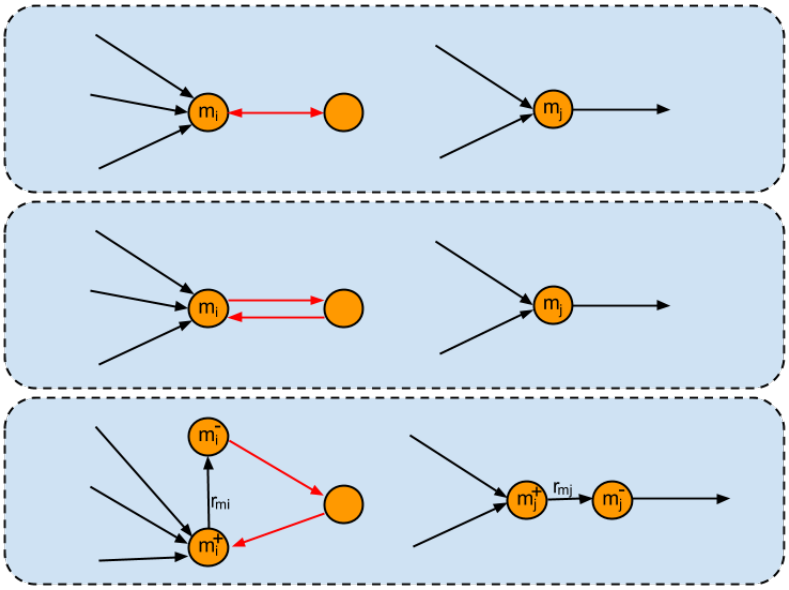


Figure 4.5: An example of how splitting reversible reactions can induce 3-cycles. **Top:** m_1 is a reaction that has an adjacent reversible reaction - red arrow. **Middle:** In the first step the reversible reaction is split into a forward and backward arc. **Bottom** Applying LP1 and LP2 would result in splitting metabolites m_i and m_j, thus introducing a three-cycle involving metabolite m_i.

$$\text{(LP3)} \qquad \begin{array}{lrll} \min/\max & v_k & & \\ \text{s.t.} & Sv & = 0 & \\ & v_{Irr} & \geq 0 & \\ & v_l & = 0 & \forall l \in Reac(m_i) \end{array}$$

We note that for $k \in Irr$, the maximization problem suffices.

Then $m_j \underset{m}{\longrightarrow} m_i$ holds if and only if every linear program above has 0 as its optimal value. We can proceed similarly to verify whether $m_i \underset{m}{\longrightarrow} m_j$ holds.

The above solution requires to solve $2*|Reac(m_i)|-|Irr \cap Reac(m_i)|$ linear programs to check whether $m_j \underset{m}{\longrightarrow} m_i$ holds. It is possible to compress these into two LPs by considering a random vector p on the n-dimensional unit sphere (U_n), chosen with uniform distribution. Let $p \in \mathbb{R}^n$ with $||p||_2 = 1$ and additionally $p_k = 0$ for all $k \notin Reac(m_j)$.

$$\text{(LP4)} \qquad \begin{array}{lrll} \min/\max & \sum_{k \in Reac(m_j)} p_k v_k & & \\ \text{s.t.} & Sv & = 0 & \\ & v_{Irr} & \geq 0 & \\ & v_l & = 0 & \forall l \in Reac(m_i) \end{array}$$

Theorem 4.12. *Let $p \in \mathbb{R}^n$ be a point chosen randomly with uniform distribution with $||p||_2 = 1$ and additionally $p_k = 0$ for all $k \notin Reac(m_j)$. Then*

- *a) if LP4 has non-zero optimal value then $m_j \underset{m}{\longrightarrow} m_i$ doesn't hold.*
- *b) if LP4 has an optimal value of 0 then $m_j \underset{m}{\longrightarrow} m_i$ holds with 'very high probability'.*

Proof. a) Assume by contradiction that $m_j \underset{m}{\longrightarrow} m_i$ holds. $m_i \notin Met(v)$ implies $m_j \notin Met(v)$ for all $v \in C$. The last set of constraints in LP4 assures that for any feasible solution v, $m_i \notin Met(v)$. Thus, for any feasible solution v and for all $k \in Reac(m_j), v_k = 0$. Hence $\sum_{k \in Reac(m_j)} p_k v_k = 0$, which is a contradiction to the non-zero optimality of LP4.
b) Assume by contradiction that the optimization problems in LP4 have an optimal value of 0, but $m_j \underset{m}{\longrightarrow} m_i$ does not hold. Let $n' := |Reac(m_j)|$ and let $C_X \in \mathbb{R}^{n'}$ be the projection (see Def. 7.1) of the steady-state flux cone C onto the subspace defined by the reactions in $Reac(m_j)$. Let p_X be the projection of p onto the same subspace. By our assumptions p is orthogonal to every point $v \in C$ (since the optimal value of LP4 is 0), hence p_X is also orthogonal to every point in C_X.

Since $C \neq \{0\}$, it implies that $C_X \neq \{0\}$, hence $dim(C_X) \geq 1$. It follows that $dim({C_X}^{\perp}) \leq n' - 1$. We can measure the probabilty of p_X being an element of ${C_X}^{\perp} \cap U_{n'}$ by dividing the surface area of ${C_X}^{\perp} \cap U_{n'}$ in $\mathbb{R}^{n'}$ with

the surface area of $U_{n'}$. The surface area of ${C_X}^{\perp} \cap U_{n'}$ in $\mathbb{R}^{n'}$ is 0, due to the fact $dim({C_X}^{\perp} \leq n' - 1)$. We conclude that with 'very high probability' p is not in $C^{\perp}$ which contradicts the optimal value of LP4 being 0. □

Distinguishing between partial and full coupling

While with $LP3$ and $LP4$ we can detect directional coupling, and consequently partial coupling, the question of differentiating between fully coupled and partially coupled metabolite pairs is more difficult. We cannot answer it based on these LPs. Moreover, we do not know a polynomial-time algorithm to solve this problem. The difficulty lies in the exponential number of combinations available for the orientation of reversible reactions incident to m_i and m_j. I.e., assuming there are k reversible reactions adjacent to metabolite m_i results in 2^k possible combinations for forward and backward orientation.

Regarding the complexity of the decision problem to tell whether two partially coupled metabolites are also fully coupled, we know that it is clearly in *co-NP*. If the two metabolites are not fully coupled, then there exists a flux vector v that does not satisfy the definition of full coupling. Having such a vector v allows disproving the full coupling relationship in polynomial time. Intuitively, the following stronger conjecture may hold.

Conjecture 4.13. *Give two metabolites m_i and m_j with $m_i \underset{m}{\longleftrightarrow} m_j$. Deciding whether $m_i \underset{m}{\Longleftrightarrow} m_j$ holds is a co-NP-complete problem.*

We propose *MIP5* to distinguish between partial and full coupling.

$$
\begin{array}{lrll}
\min/\max & \sum_{k \in Reac(m_j) \cap Irr} v_k |S_{jk}| + \sum_{k \in Reac(m_j) - Irr} (v_k^+ + v_k^-) |S_{jk}| & & \\
\text{s.t.} & Sv & = 0 & \\
 & v_{Irr} & \geq 0 & \\
 & \sum_{l \in Reac(m_i) \cap Irr} v_l |S_{il}| + & & \\
 & \sum_{l \in Reac(m_i) - Irr} (v_l^+ + v_l^-) |S_{il}| & = 1 & \\
 & v_k - v_k^+ + v_k^- & = 0 & \forall k \in T \\
 & v_i \neq 0 \implies (v_i^+ > 0) \oplus (v_i^- > 0) & & \forall k \in T
\end{array}
\tag{MIP5}
$$

where $T := Reac(m_j) \cup Reac(m_i) - Irr$. The $\oplus$ sign in the last constraint represents the exclusive or operator, which in the context of a mixed integer program can be realized by introducing binary variables.

Then $m_j \underset{m}{\Longleftrightarrow} m_i$ holds if and only if the optimal values of the minimization and maximization problems of *MIP5* are identical.

The idea of *MIP5* is to evaluate the absolute value of the reversible reactions incident to m_i and m_j. We do this by introducing a variable v_k^+ and v_k^- for every such reaction. The fourth set of constraints is for reversible reactions and enforces the link between v_k, v_k^+ and v_k^-, that is $v_k = v_k^+ - v_k^-$. The last

constraint asserts that if v_k is nonzero, then exactly one of v_k^+ and v_k^- will be nonzero as well. The third set of constraints ensures there is activity around metabolite m_i. This constraint is a reduced form of $\sum_{l \in Reac(m_i)} |v_l S_{il}| = 1$, where the variables are not within absolute values anymore. This was made possible by splitting the reversible reactions, which assured that the newly introduced variables are non-negative. A similar formulation is used in the objective function to measure activity around metabolite m_j.

4.3 Results and discussion

In a first step we have implemented the metabolite activity coupling (MAC) method presented in this chapter and applied it on networks of various sizes. The implementation was done in Matlab [74] and the LP solver of choice was CLP [18]. For solving MIP5 we have used Gurobi [41].

The following networks were used:

- ILLUSNET network from [64]
- RBC : metabolic network of red blood cell [134]
- EC core: central metabolic network of *E. coli* [83]
- *H. pylori* genome-scale metabolic network [121]
- *S. cerevisiae* (yeast) genome-scale metabolic network [29]
- The 2003 genome-scale metabolic network of *E. coli* (iJR904) [90]
- The metabolic network of human hepatocyte (HepatoNet1) [38]

Table 4.1 summarizes the number of metabolite activity couplings for each network. Additionally, for comparison purposes we have included the number of flux coupling relationships as well. The "number of elements" column therefore represents the number of reactions for FCA and number of metabolites for MAC. It is interesting to note that in all cases, FCA finds more fully coupled pairs, while MAC always finds more partially and directionally coupled pairs. The difference is highest in the number of directional couplings, especially for large-scale networks. This observation hints that FCA and MAC are not competing or alternative methods, but they should be used to complement each other.

Regarding the running time of MAC, it is in a similar time-scale as FFCA, and thus it is much slower than F2C2. This difference was expected, since LPs are solved for every pair of metabolites. Metabolites do not have a property that is similar to the reversibility of reactions, a metabolite's turnover is always a nonnegative value. Thus, similar rules as the ones based on the reversibility

of reactions in F2C2 are not available. It is important to note that, even though MIPs have to be solved in some cases, due to the fact that typically there are less than a dozen binary variables in each MIP, these do not slow down the algorithm. The complexity of the MIPs allows them to be solved in a matter of seconds.

Fully coupled reaction pairs were important in FCA because they hint at structural features of the network. The enzymes catalyzing these reactions often make part of the same enzyme complex and are referred to as enzyme subsets [87]. Moreover, these pairs of reactions can be lumped together without irreversibly modifying the steady-state flux space, therefore reducing the complexity of the network. At this point it is unclear whether a similar reduction is possible or not with fully coupled metabolites. But it is certainly an interesting direction that should be studied more.

From Table 4.1 it appears that the strength of MAC is computing directional couplings. A possible application of these couplings could be the following. Assuming the goal is to inhibit the turnover of a given metabolite m_i, then the set of reactions incident to any directionally coupled metabolite is a candidate for drug target.

Table 4.1: Comparison of the couplings in FCA and MAC.

Network name	Method	No. of elements	No. of blocked	Full coupling	Partial coupling	Directional coupling
ILLUSNET	FCA	19	1	6	0	9
	MCA	18	1	1	1	10
RBC	FCA	42	0	21	0	15
	MCA	30	0	9	1	63
EC Core	FCA	63	0	19	0	9
	MCA	63	0	13	8	417
H. pylori	FCA	480	263	164	3	309
	MCA	414	248	122	7	1260
S. cerevisiae	FCA	1144	505	692	90	2214
	MCA	945	497	633	236	7263
E. Coli	FCA	930	165	2567	68	6208
	MCA	618	162	2116	131	12987
Human hepatocyte	FCA	2309	0	1463	201	1593
	MCA	1040	0	978	223	5365

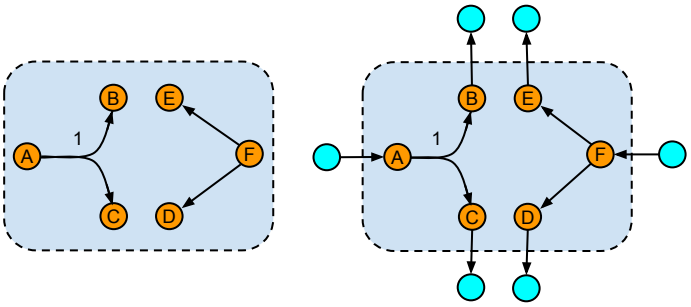


Figure 4.6: Example of a closed (left) and an open (right) system.

Finally we will compare metabolite concentration coupling with metabolite activity coupling. A main difference between the two methods is the assumptions they start with. In MCCA the metabolic system considered has to be in closed form, i.e., there should be no exchange reactions. Applying steady-state analysis to such a system reduces the flux-space to the conic combination of internal cycles. Theoretically, one could look at the same system in both closed and open forms (e.g. Fig. 4.6) and apply the corresponding coupling analysis method. Table 4.2 summarizes the coupling relations for the example network from Fig. 4.6.

Table 4.2: Comparison of the couplings in MCCA resp. MAC in the networks from Fig. 4.6 left resp. Fig. 4.6 right.

Coupling relation	MCCA	MCA
Full couplings	$E \underset{m}{\Longleftrightarrow} F$	$A \underset{m}{\Longleftrightarrow} B$
	$E \underset{m}{\Longleftrightarrow} D$	$A \underset{m}{\Longleftrightarrow} C$
	$F \underset{m}{\Longleftrightarrow} D$	$B \underset{m}{\Longleftrightarrow} C$
Directional couplings	$B \underset{m}{\longrightarrow} A$	$D \underset{m}{\longrightarrow} F$
	$C \underset{m}{\longrightarrow} A$	$E \underset{m}{\longrightarrow} F$

As it results from Table 4.2, the two methods indeed find very different coupling relations.

Summary of the chapter

- We have introduced the novel concept of metabolite activity coupling analysis.
- Directional coupling types can be derived by means of solving linear programs.
- To distinguish between partial and fully coupled metabolite pairs, a MIP approach has been presented.
- Based on computational analysis, metabolite activity coupling is shown to be complementary to flux coupling analysis.

Chapter 5

Computing Elementary Flux Modes Involving a Set of Target Reactions

Contents

Remark: The content of this chapter was originally published in [23] and is independent from the related work [86].

5.1 Introduction

Elementary (flux) modes (EMs) [105, 101, 104, 106, 84] are an important concept for the structural analysis of metabolic networks, with many practical applications (see e.g. [137] and references therein). As a consequence, the development of methods for the computation of EMs has become an active research area over the past years [34, 130, 118, 115, 25, 91, 92]. The computational complexity of enumerating all EMs is not known [1]. However, there exist several algorithms and software packages for an exhaustive enumeration in a given metabolic network [34, 118, 130, 115]. While these methods work very well for small networks, due to the possibly exponential number

of EMs, they may fail for medium or large genome-scale networks. With the ever increasing size of genome-scale metabolic network reconstructions, EM analysis nowadays can often be used only under additional assumptions (e.g., modifying the system boundary of the network, blocking a large number of uptake reactions etc.). These extra assumptions may have the bad side effect of changing the structure of the network, sometimes introducing artificial pathways [73].

One way to deal with genome-scale networks is to define a subset of interest $\mathcal{R}_I$ of the full reaction set $\mathcal{R}$, without altering the network topology. Kaleta et al. [50] look for sets of reactions in $\mathcal{R}_I$, called flux patterns, which indicate the existence of an EM having those reactions in its support. They enumerate a basis for this set, the elementary flux patterns. [126] and [73] project the steady-state flux cone onto the subspace defined by $\mathcal{R}_I$, and enumerate the partial EMs for this subspace.

Given the difficulty of computing and analyzing the full set of all EMs, recent research has focused on finding a special subset of EMs [25, 91, 92]. De Figueiredo et al. [25] describe a mixed-integer programming method to enumerate the k shortest EMs ($k \geq 1$). Their method has been extended to find shortest EMs involving one reaction of choice and also to enumerate a minimal generating set of EMs [91]. In the last paper, the authors note that their method cannot be applied to find elementary modes involving two predefined reactions.

The problem of finding an EM involving two or more given reactions was also considered by [1]. The authors give a more general formulation, in that they not only want their EM to involve a certain set of target reactions $T \subset \mathcal{R}$ (with $|T| = t > 0$), but also to avoid another set of reactions $F \subset \mathcal{R}$ (with $F \cap T = \emptyset$). In other words, the goal is to find an EM e, with $e_i \neq 0$ for $i \in T$, and $e_i = 0$ for $i \in F$. The authors show that defining a set of reactions F to be avoided does not add to the difficulty of the problem, and in fact reactions belonging to F can simply be removed from the network. [2] study the complexity and prove that the decision problem, whether an EM involving two or more target reactions exists, is not solvable in polynomial time, unless $P = NP$.

To the best of our knowledge, no algorithm for this problem has been published so far. Here, we develop a mixed-integer programming approach for the more general problem of computing k elementary modes ($k \geq 1$) involving a given set of target reactions T, for $|T| \geq 2$. Computational experiments show that the method can be applied even to large genome-scale networks.

5.2 Definition of the problem

The goal of this chapter is to study the following *problem* ($\mathscr{P}$):

> Given a metabolic network $N = (S, Irr, \mathcal{M}, \mathcal{R})$, a set of target reactions $T \subseteq \mathcal{R}$ and $k \geq 1$, compute a set E of EMs in N, $|E| = k$, such that $supp(e) \supseteq T$, for all $e \in E$.

For the rest of this chapter, we assume that all reactions are irreversible, i.e., $Irr = \mathcal{R}$, and that none of the reactions is blocked. These assumptions do not limit the applicability of our methods. Reversible reactions can be split into a forward and backward reaction. When $|T| > 1$ and $l \in \{1, ..., |T|\}$ reactions in T are reversible, then the original problem can be reduced to 2^l subproblems by considering every combination of forward and backward arcs for the reversible reactions in T. Each of these subproblems can be solved independently of the others.

In general, the splitting operation may induce a number of artificial EMs in the form of two-cycles. However, these two-cycles in most cases do not increase the set of solutions for problem ($\mathscr{P}$). In fact, the only case where such two-cycles satisfy the conditions of problem ($\mathscr{P}$) is for $|T| = 1$. In this case, there is exactly one artificial EM that needs to be filtered out from the final solution set. For $|T| > 1$ no pairs of split reactions will be part of the same EM.

All blocked reactions can be identified in polynomial time, by solving a linear number of linear programs. Afterwards, they can be removed from the network without altering the underlying flux cone.

5.3 Methods

In the following, we divide the general problem ($\mathscr{P}$) into three subproblems and discuss them individually: the one-reaction case, the two-reaction case and the general t-reaction case (where $t = |T| > 2$). The underlying details vary in each case, but there is a general concept followed by all three methods. In every case, we aim to incrementally find an alternating sequence N^1, e^1, N^2, e^2, ..., N^k, e^k, of subnetworks and EMs of the input network N with the following properties for all $i \in \{1, ..., k\}$:

1. The target reactions r_1, ..., r_t are part of every subnetwork N^i.
2. In every subnetwork N^i, reaction r_1 is directionally coupled to the reactions r_2, ..., r_t.
3. No subnetwork N^i has as flux mode any of the EMs e^l for $l < i$.
4. e^i is an EM in N^i involving r_1.

Clearly, the main difficulty in finding such a sequence of subnetworks is in imposing condition (2). In turn, once a new subnetwork $N^i = (S^i, \mathcal{R}^i, \mathcal{M}, \mathcal{R}^i)$ has been found, a corresponding EM e^i can be computed by solving the linear program LP(N^i) [1].

$$\begin{array}{lrlll} \mathrm{LP}(N^i)\text{: } \min & 0 & & & \\ \text{s.t.} & S^i v & = & 0, & \\ & v_{r_1} & \geq & 1, & \\ & v_j & \geq & 0, & \forall j \in \mathcal{R}^i. \end{array}$$

Using a Simplex-based method, we can compute a basic feasible solution e^i of $\mathrm{LP}(N^i)$, which corresponds to a vertex of the truncated flux cone of N^i (resp. an extreme ray of the flux cone of N^i), and thus defines an EM in N^i [34], which involves r_1. Due to the conservation property of EMs (see e.g. Lemma 1 in [72]), e^i is also an EM in N. The set $E^i := \{e^1, ..., e^i\}$, $i \geq 1$, contains EMs of the original network N that involve every target reaction. Terminating the search after k EMs have been found provides a solution to problem $\mathscr{P}$.

5.3.1 The one-reaction problem

If the set of target reactions consists of only one reaction, condition (2) is trivially satisfied for every subnetwork satisfying condition (1). Assuming N^1, e^1, N^2, e^2, ... N^i, e^i have already been computed, we can determine a new subnetwork N^{i+1} by solving the following mixed-integer linear program (MILP1).

$$\begin{array}{lrlll} \mathrm{MILP1}(E^i)\text{:} & & & & \\ \min & 0 & & & \\ \text{s.t.} & Sv & = & 0, & \\ & v_{r_1} & \geq & 1, & \\ a_l \leq & v_l & \leq & Ma_l, & \forall l \in \mathcal{R}, \\ \sum_{l \in supp(e^u)} & a_l & \leq & |supp(e^u)| - 1, & \forall u \in [1..i], \\ & v_l & \geq & 0, & \forall l \in \mathcal{R}, \\ & a_l & \in & \{0,1\}, & \forall l \in \mathcal{R}. \end{array}$$

The constraints in (MILP1) are the same as for the computation of the so-called shortest EMs in [25]. There are two groups of variables: v represents the steady-state flux values of the reactions, while the 0-1 vector a models the support of v (i.e., $v_l > 0 \Leftrightarrow a_l = 1$). The variables v_l and a_l are linked by the 3rd constraint, using a suitably large constant $M > 0$. An important difference to [25] is the objective function (see also Sect. 5.4.2). We do not try to find the smallest set of reactions satisfying the constraints of (MILP1). Instead computing any feasible solution is sufficient. This turns out to be enough to derive a subnetwork that satisfies conditions (1-3). Indeed, given a feasible solution (v', a') of (MILP1), define N^{i+1} by the set of reactions $\mathcal{R}^{i+1} := \{l \in \mathcal{R} \mid a'_l = 1\}$. This subnetwork N^{i+1} clearly satisfies conditions (1) and (2), while the 4th constraint in (MILP1), the so-called *no-good constraints*, guarantees condition (3). By solving $\mathrm{LP}(N^{i+1})$, we obtain an EM e^{i+1}.

Table 5.1 summarizes the method for the one-reaction case. The two conditional exit points of the algorithm are Step 2 and 6. If the algorithm terminates at Step 2, Prop. 5.1 assures all EMs will be found. In contrast, if the exit occurs at Step 6, we enumerate k EMs.

Table 5.1: Algorithm 1 for the one-reaction case.

Step	Action
0.	Initialize $i := 1, E := \emptyset$.
1.	Try to find a feasible solution (v', a') of MILP1(E).
2.	If MILP1(E) is infeasible, then STOP.
3.	Otherwise, use (v', a') to derive subnetwork N^i.
4.	Find a basic feasible solution e^i of LP(N^i).
5.	Let $E := E \cup \{e^i\}$ and $i := i + 1$.
6.	If $i > k$ then STOP.
7.	Go to Step 1.

Proposition 5.1. *For any EM $e \notin E$ and sufficiently large $M > 0$, there is a flux mode v' in N such that $(v', supp(e))$ is a feasible solution for MILP1(E).*

A formal proof of Prop. 5.1 is omitted here. It can be easily obtained by suitably scaling the vector e.

We note that it is possible to initialize $N^1 := N$, thus avoiding the need to solve the very first mixed-integer program. The computationally hard part is Step 1 of the algorithm, while the other steps can be done in polynomial time.

5.3.2 The two-reaction problem

A natural idea to find a shortest EM involving a pair of reactions $\{r_1, r_2\} \subseteq \mathcal{R}$ would be to extend the previous method by forcing both r_1 and r_2 to be active, while minimizing the total number of active reactions:

$$
\begin{aligned}
(\text{MILP2}) : \min \quad & \textstyle\sum_{l \in \mathcal{R}} a_l & & & & \\
\text{s.t.} \quad & Sv & = & 0, & & \\
& v_{r_1} & \geq & 1, & & \\
& v_{r_2} & \geq & 1, & & \\
a_l \leq \; & v_l & \leq & M a_l, & \forall l \in \mathcal{R}, & \\
& v_l & \geq & 0, & \forall l \in \mathcal{R}, & \\
& a_l & \in & \{0, 1\}, & \forall l \in \mathcal{R}. &
\end{aligned}
$$

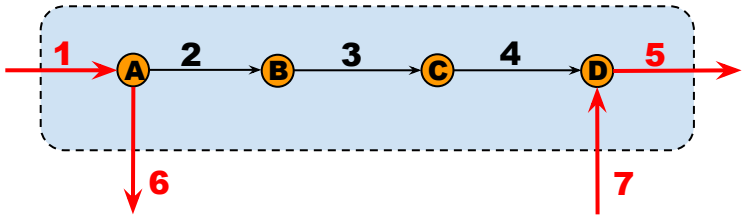


Figure 5.1: Example network. Thick arrows represent the optimal solution of (MILP2)

(MILP2) indeed finds a shortest flux mode containing r_1 and r_2. However, it may fail to produce an EM. As illustrated by Fig. 5.1, if we apply (MILP2) for reactions 1 and 5, a non-elementary flux mode will be found, involving the reactions 1,6,7,5. However, an optimal solution (v^*, a^*) of (MILP2) still has interesting properties, which will turn out to be useful for refining our method. These are described in the following propositions.

Proposition 5.2. *Let $v^* = \sum_{i=1}^{s} \lambda_i e^i$, with $\lambda_i > 0$, be any decomposition of an optimal solution (v^*, a^*) of (MILP2) into s EMs in N. Then for all $i \in [1..s]$, $e^i_{r_1} > 0$ or $e^i_{r_2} > 0$. There exists $i \in [1..s]$ with $e^i_{r_1} > 0$ and $e^i_{r_2} > 0$, if and only if $s = 1$.*

Proof. Suppose there exists $i \in [1..s]$ such that $e^i_{r_1} = 0$ and $e^i_{r_2} = 0$. Let $p := \min\{v^*_j / e^i_j \mid j \in supp(e^i)\}$ and $v' := v^* - pe^i$. Then there exist $\lambda > 0$ and $a' \in \{0,1\}^{\mathcal{R}}$ such that $(\lambda v', a')$ is a feasible solution of (MILP2), with $\sum_{l \in \mathcal{R}} a'_l < \sum_{l \in \mathcal{R}} a^*_l$, in contradiction to the optimality of v^*.

From $s = 1$, we get $v^* = \lambda_1 e^1, \lambda_1 > 0$, and therefore $e^i_{r_1} > 0$ and $e^i_{r_2} > 0$. Conversely, suppose there exists $i \in [1..s]$ such that $e^i_{r_1} > 0$ and $e^i_{r_2} > 0$. Up to scaling, e^i is a feasible solution of (MILP2) with $supp(e^i) \subseteq supp(v^*)$. From the optimality of v^*, we get $supp(e^i) = supp(v^*)$, which implies $s = 1$. □

Prop. 5.2 shows that any EM participating in a decomposition of v^* must contain at least one of the two target reactions. The only case when an EM contains both r_1 and r_2 is when the optimal solution v^* itself is an EM.

The result can be formulated in a stronger form, extending it to the whole subnetwork $N^* = (S_{\mathcal{MR}^*}, \mathcal{R}^*, \mathcal{M}, \mathcal{R}^*)$ defined by v^* (with $\mathcal{R}^* = supp(v^*)$). Every EM e in this subnetwork will satisfy $e_{r_1} > 0$ or $e_{r_2} > 0$. If v^* is itself an EM, then also the subnetwork N^* will have only one EM, namely v^*. The following corollary summarizes our previous results.

Corollary 5.3. *In the subnetwork N^* defined by v^*, the reactions r_1 and r_2 are either fully coupled or mutually exclusive.*

Our next proposition shows that any flux mode using the reactions r_1 and

r_2 can be scaled by a positive factor so that it becomes a feasible solution of (MILP2).

Proposition 5.4. *Let $v \in C$ with $v_{r_1} > 0$ and $v_{r_2} > 0$. Then for sufficiently large $M > 0$, there exists a feasible solution (v', a') of (MILP2), such that $v' = \lambda v$, for some $\lambda > 0$.*

Proof. Let $\lambda := 1/\min\{v_i \mid i \in \mathit{supp}(v)\}$ and $M := \max\{v_i/v_j \mid i,j \in \mathit{supp}(v)\}$. Now consider $v' := \lambda v$ and let $a'_l := 1$, for all $l \in \mathit{supp}(v)$, and $a'_l := 0$, for all $l \in \mathcal{R} \setminus \mathit{supp}(v)$. For all $l \in \mathit{supp}(v)$, $v'_l = v_l/\min\{v_i \mid i \in \mathit{supp}(v)\} \geq 1$. Thus, $v'_{r_1} \geq 1$ and $v'_{r_2} \geq 1$. By definition of a', for all $l \in \mathit{supp}(v)$, $a'_l = 1 \leq v'_l = v_l/\min\{v_i \mid i \in \mathit{supp}(v)\} \leq \max\{v_i/v_j \mid i,j \in \mathit{supp}(v)\} = M = Ma'_l$. Since $v' \in C$ also holds, we conclude that (v', a') is a feasible solution of (MILP2). □

In general, when decomposing an arbitrary steady-state flux vector into EMs, the number of participating EMs ranges from only one to many. Moreover, a decomposition does not necessarily have to be unique. It turns out that in our special case, a decomposition is much more constrained. In fact, no decomposition can contain more than two EMs. As direct corollary of this result, we will also get the uniqueness of the decomposition.

Proposition 5.5. *Let (v^*, a^*) be an optimal solution of (MILP2) and let $v^* = \sum_{i=1}^{s} \lambda_i e^i$, with $\lambda_i > 0$, for all $i \in [1..s]$, be a decomposition of v^* into s EMs with pairwise different support. Then $s \leq 2$.*

Proof. Assume $s > 2$. From Prop. 5.2 it follows that there exist $i,j \in [1..s]$ such that e^i and e^j both contain r_1 or both contain r_2. Without loss of generality, we assume that e^i and e^j both contain r_1. Then, since $s \neq 1$, both e^i and e^j do not contain r_2. Let $\mathcal{R}_{\mathit{diff}} := \mathit{supp}(e^j) \setminus \mathit{supp}(e^i)$. Since e^i is an EM, we have $\mathcal{R}_{\mathit{diff}} \neq \emptyset$. Let $p := \min\{v^*_r/e^j_r \mid r \in \mathcal{R}_{\mathit{diff}}\}$. The vector $v' := v^* - pe^j$ satisfies $\mathit{supp}(v') \subsetneq \mathit{supp}(v^*)$. However, it need not be feasible for (MILP2) because it may violate some constraint $v_r \geq 0$. Let $V := \{r \in \mathcal{R} \mid v'_r < 0\} \subseteq \mathit{supp}(e^j) \cap \mathit{supp}(e^i)$. If $V \neq \emptyset$, define $q := \min\{v'_r/e^i_r \mid r \in V\} < 0$, otherwise $q := 0$. Then $v'' := v' + (1-q)e^i \geq 0, v''_{r_1} > 0, v''_{r_2} > 0$, and we still have $\mathit{supp}(v'') \subsetneq \mathit{supp}(v^*)$. By Prop. 5.4, this implies the existence of a feasible solution for (MILP2) with a smaller objective function value than v^*, which is a contradiction. □

Corollary 5.6. *Any decomposition of v^* into EMs is unique.*

Proof. The result is trivial if v^* is itself an EM. Thus we only have to consider the case $s = 2$. Assume by contradiction that $v^* = \lambda_1 e^1 + \lambda_1 e^2 = \mu_1 e^3 + \mu_2 e^4$, where at least three of e^1, e^2, e^3, e^4 have pairwise different support. Clearly, then $v^* = (\lambda_1 e^1 + \lambda_1 e^2 + \mu_1 e^3 + \mu_2 e^4)/2$, which contradicts the result of Prop. 5.5. □

The previous results show that an optimal solution v^* of (MILP2) either is an EM (good case), or the sum of two EMs (bad case). Of special interest is Cor. 5.3, which asserts that in the subnetwork N^* defined by v^*, r_1 and r_2 are either fully coupled or mutually exclusive. Thus, to make the optimal solution of (MILP2) an EM, it is enough to add constraints that exclude the second case. These additional constraints must forbid the existence of an EM in N^* involving exactly one of r_1 and r_2. This can be achieved by requiring that r_1 should be directionally coupled to r_2 in N^* (or alternatively that r_2 is directionally coupled to r_1). To formulate this mathematically, we delete r_2 from N^* and require that r_1 is blocked in the resulting subnetwork N'. More formally, if $S' := S_{*,supp(v^*)\setminus\{r_2\}}$ is the stoichiometric matrix of N', then the following system should be infeasible:

$$\begin{aligned} S'z &= 0, \\ z_{r_1} &= 1, \\ z &\geq 0. \end{aligned}$$

By applying Farkas' Lemma (see e.g. [100]), this infeasibility requirement can be turned into a feasibility condition in the dual space. Let $y \in \mathbb{R}^m$, $x \in \mathbb{R}$. Then the following system in y and x should be feasible:

$$\begin{aligned} (S')^T y + u^{r_1} x &\geq 0, \\ x &\leq -1. \end{aligned}$$

Here u^r is the r-th unit vector (with an entry 1 for component r, and 0 otherwise) and $.^T$ denotes transposition of a matrix.

This formulation inherently uses S', the stoichiometric matrix of the new subnetwork. Naturally, information about it is not derivable independently. Thus the first constraint set needs to be reformulated to dynamically adjust itself according to the current solution (v, a) of (MILP2). This leads to the following constraints:

$$\begin{aligned} S^T y + u^{r_1} x &\geq -M(\mathbf{1} - a + u^{r_2}), \\ x &\leq -1, \end{aligned} \qquad \text{DirC}(r_1, r_2)$$

where $\mathbf{1}$ denotes a vector all whose components are 1.

By using a large enough constant M, the first constraint becomes trivially satisfiable for inactive reactions ($a_l = 0$) and for r_2, where the right-hand side simplifies to $-M$. In contrast, for active reactions ($a_l = 1$) different from r_2, the right-hand side sums up to 0, thus effectively activating the constraint. The inequalities DirC(r_1, r_2) are called *directional coupling constraints for* r_1 *implying* r_2. Extending (MILP2) with DirC(r_1, r_2), allows computing a shortest EM through r_1 and r_2. The following Prop. 5.7 summarizes our construction. It guarantees that by adding the constraints DirC(r_1, r_2), any feasible solution of (MILP2) defines a subnetwork in which condition (2) is satisfied.

Proposition 5.7. *Let (v', a') be any feasible solution of (MILP2) augmented with the directional coupling constraints DirC(r_1, r_2). Then in the subnetwork N^* defined by supp(v'), reaction r_1 is directionally coupled to reaction r_2.*

Proof. Let (v', a') be a feasible solution of (MILP2) augmented with DirC(r_1, r_2). Assume by contradiction that in the subnetwork N^* defined by *supp*(v'), r_1 is not directionally coupled to r_2. Then there exists a flux mode w in N^* with $w_{r_1} > 0$ and $w_{r_2} = 0$. Since (v', a') is feasible, there exist $y' \in \mathbb{R}^m$, $x' \in \mathbb{R}$ such that

$$\begin{aligned} S^T y' + u^{r_1} x' &\geq -M(\mathbf{1} - a' + u^{r_2}), \\ x' &\leq -1. \end{aligned}$$

Let N' be the subnetwork obtained from N^* by deleting reaction r_2, with corresponding stoichiometric matrix S'. Removing the inequality corresponding to r_2, we get the feasible system

$$\begin{aligned} (S')^T y' + u^{r_1} x' &\geq 0, \\ x' &\leq -1. \end{aligned}$$

Applying the Farkas' Lemma, we can now derive that the system

$$\begin{aligned} S' z &= 0, \\ z_{r_1} &= 1, \\ z &\geq 0. \end{aligned}$$

is infeasible. This is a contradiction to the existence of w. We conclude that in the subnetwork N^* defined by *supp*(v'), r_1 is directionally coupled to r_2. □

By iteratively adding no-good constraints corresponding to already found EMs, we are able to enumerate any number of EMs in an increasing order of length. The resulting mixed-integer linear program can be expected to work for smaller-scale network models, but due to the difficulty of proving optimality in mixed-integer linear programs, the algorithm will most likely turn impractical for larger models. The reason for the bottle-neck is clearly that in every iteration we aim to find the shortest EM not yet discovered. Similar to the one-reaction case, we next trade the optimality condition on the length for an easier to solve program. The final form of our method is given in the mixed-integer linear program (MILP3). Tab. 5.2 summarizes the algorithm in the two-reaction case.

$$
\begin{array}{rrcll}
\text{MILP3}(E^i): \min & 0 & & & \\
\text{s.t.} & Sv & = & 0, & \\
& v_{r_1} & \geq & 1, & \\
& v_{r_2} & \geq & 1, & \\
a_l \leq & v_l & \leq & M_0\, a_l, & \forall l \in \mathcal{R}, \\
& S^T y + u^{r_1} x & \geq & M_1(a - \mathbf{1} - u^{r_2}), & \\
& -x & \geq & 1, & \\
& \sum_{l \in supp(e^q)} a_l & \leq & |supp(e^q)| - 1, & \forall q < i, \\
& v_l & \geq & 0, & \forall l \in \mathcal{R}, \\
& a_l & \in & \{0,1\}, & \forall l \in \mathcal{R}, \\
& x, y_m & \in & \mathbb{R}, & \forall m \in \mathcal{M}.
\end{array}
$$

Table 5.2: Algorithm 2 for the two-reaction case.

Step	Action
0.	Initialize $i := 1, E := \emptyset$.
1.	Try to find a feasible solution (v', a') of MILP3(E).
2.	If MILP3(E) is infeasible, then STOP.
3.	From (v', a') derive subnetwork N^i.
4.	Find a basic feasible solution e^i of LP(N^i).
5.	Let $E := E \cup \{e^i\}$ and $i := i + 1$.
6.	If $i > k$ then STOP.
7.	Go to Step 1.

5.3.3 The general t-reaction case

Although this problem seems to be much harder at first sight, it turns out that the previous results provide all the ingredients necessary to tackle this general case. We propose two strategies for building a mixed-integer linear program that can be used in a similar fashion to (MILP3).

In the *cascade* method, we extend (MILP2) with DirC(r_1, r_2), DirC(r_2, r_3), ..., DirC(r_{t-1}, r_t). Based on Prop. 5.7 and the transitivity of directional coupling [66], in any feasible solution of this new MILP, reaction r_1 will imply reactions r_2, ..., r_t, thus satisfying condition (2). Similarly, in the *hub* method, DirC(r_1, r_2), DirC(r_1, r_3), ..., DirC(r_1, r_t) are added to (MILP2).

Alternative coupling strategies can be thought of. Indeed, by constructing any spanning tree on the vertices r_1, r_2,... r_t, with r_1 being the root of the tree, and taking the union of the directionality constraints corresponding to each edge, we create the conditions for a subnetwork where every desired reaction

is directionally coupled to r_1. At this time, it is unclear whether there is a practical advantage in choosing one of these methods compared to the others. In all cases, the total number of constraints (and variables) is the same and grows linearly with t.

5.3.4 Flux uncoupling

The computational complexity of the problem to decide for a pair of uncoupled reactions whether they are sometimes coupled or mutually exclusive [72] is NP-complete [2]. However, this problem can be seen as a special case of Sect. 5.3.2. Indeed, rather than enumerating some (or all) EMs involving these two reactions, we are merely interested in the existence of any. For this purpose, it is enough to execute Step 1 of Algorithm 2. If MILP3($\emptyset$) is feasible (resp. infeasible), we can conclude that our two reactions are sometimes coupled (resp. mutually exclusive).

While this works for any pair of reactions, the above solution might not be optimal if our goal is to find all uncoupling relations. For a given pair of reactions, rather than stopping after Step 1 of Algorithm 2, one could continue executing Steps 2-7 and potentially compute an EM, if one exists. This EM can then be used to decide not only about the uncoupling relation for the current pair of reactions, but for other pairs as well. More specifically, once an EM e has been computed for two reactions i and j, this EM can also be used to show that other pairs of reactions in $supp(e)$ are sometimes coupled. This may greatly reduce the number of pairs for which solving an MILP is necessary. Moreover, by performing FCA [24, 66] as a heuristic presolving step, many uncoupling pairs can be deduced without having to solve an MILP.

5.3.5 Choosing big-M values

One issue that we have not addressed so far is the importance of choosing right values for M_0 and M_1. From the theoretical side, we are assured about the existence of M_0 and M_1 values for which the algorithm behaves as intended. However, when implementing these algorithms, it becomes crucial to choose correct values. On one hand, we are inclined to select large constants to guarantee the correctness of the solutions. If we select constants that are not large enough, we risk cutting off feasible solutions corresponding to EMs we may interested in. On the other hand, the larger these values are, the less numerically stable the MILPs become. In the following, we analyze how setting M_0 and M_1 affects the output of the algorithms.

We note that every feasible solution (v, a) of (MILP3) satisfies $v_i \in [1, M_0]$ for $a_i = 1$. Thus, M_0 represents the largest ratio v_i/v_j that can occur for non-zero fluxes v_i and v_j.

M_1 appears only in the directional coupling constraints. For any flux vector that contains r_1 but not r_2, the directional coupling constraints will be infeasible, regardless of the choice of M_1. Thus the only bad case occurs when an EM containing both r_1 and r_2 that should be feasible is rendered infeasible by the directional coupling constraints. This happens if M_1 is chosen not large enough, such that some of the constraints that should be trivially satisfied become unsatisfiable. Let v' be an EM for which this case occurs, i.e., $v'_{r_1} \geq 0$, $v'_{r_2} \geq 0$, with infeasible $DirC(r_1, r_2)$ constraints. Let $a \in \{0,1\}^n$ be a vector corresponding to the support of v'. We investigate the following LP-relaxation of (MILP3).

$$\begin{array}{rrcl} \min & 0 & & \\ & S^T y + u^{r_1} x & \geq & M_1(a - \mathbf{1} - u^{r_2}), \\ & -x & \geq & 1. \end{array}$$

Its corresponding dual LP reads

$$\begin{array}{rrcl} \max & w - M_1 \sum_{i \in \{i | a_i = 0\} \cup \{r_2\}} v_i & & \\ & Sv & = & 0, \\ & v_{r_1} - w & = & 0, \\ & v, w & \geq & 0. \end{array}$$

Note that the null vector 0 is always dual feasible. Since in our assumption the primal is infeasible, the dual must be unbounded. We conclude that the primal is infeasible because $w = v_{r_1} > M_1 \sum_{i \in \{i | a_i = 0\} \cup \{r_2\}} v_i$. In order to avoid this problem, we recommend over-approximating M_1 with $M_1 \geq v_{r_1}/v_i, i \in \mathcal{R} \setminus \{r_1\}$.

5.4 Results and Discussion

The above described methods have been implemented in Matlab. The MILP solver of choice was Gurobi 5.0. In the following, we present several use-case scenarios that have been performed on different real-world networks. These tests were aimed at validating the correctness of the methods, and also to motivate their existence, by applying them on networks where an exhaustive EM enumeration would fail. Furthermore, we ran benchmarking tests to measure the running time of the algorithms and other statistical properties. All computations were performed using a single Intel T2600 (2.16 GHz) processor on a 32-bit Windows 7 system, with a maximum memory of 640MB allocated to Matlab.

Checking the correctness of the resulting flux vectors is easy. A simple rank test as in [52] can prove the EM property, while checking for non-zero entries in the target reactions assures that we are indeed using them. Due to the nature of the algorithms, the EM property never gets violated. Indeed,

because the last step of every method presented involves solving an LP to optimality using a Simplex-based method, the result will be a vertex of the truncated flux cone.

5.4.1 Use-case scenario 1

In [26], the authors compare graph-based pathway enumeration with EM analysis in the context of discovering pathways producing G6P (KEGG entry C00668) from AcCoA (KEGG entry C00024). The underlying network was based on the human Krebs-cycle with two possible configurations, one of which was able to display the required phenotype, while the other was not. Answering this type of questions is a perfect use-case scenario for our method. Indeed, we ran our algorithms, trying to find EMs containing the required reactions, and in both configurations we were able to replicate the answers presented by the authors.

It is important to mention that the network used in the previous study is quite small (20 metabolites and 26 reactions). EM enumeration tools work very well for models of this size. Thus, we performed the same analysis for a genome-scale reconstruction of the human metabolic model [28]. In [51] a similar analysis has been performed with the use of elementary flux patterns. A crucial difference between analyzing the small and the genome-scale models, beside their size, is that in the latter network, the two metabolites we were interested in are internal to the model. This is important for two reasons. First, the number of adjacent reactions to each metabolite can be more than one, making the selection of a pair of reactions involving G6P and AcCoA non-unique. Indeed, after eliminating the blocked reactions from the network, we still had 63 viable pairs of reactions.

Secondly, and more importantly, for two internal reactions r_1 and r_2, one cannot easily find out whether reaction r_1 is a predecessor of r_2 or vice-versa. In the case of boundary reactions, it is very natural to claim that in an EM, uptake reactions precede outgoing reactions. We could say that the products of the outgoing reactions are obtained from the substrates of the uptake reactions. A similar statement for internal reactions is not trivial. Deciding whether one internal reaction precedes another one is an unsolved problem. While our method can still identify EMs with internal reactions, the interpretation of these EMs has to be done with care.

For these reasons, two artificial transport reactions were added to the network, one importing AcCoA and the other secreting G6P. The biological interpretation of our method applied to this modified system would be the following. "Assuming we can inject AcCoA into the network and have a method to secret G6P. Can the underlying network convert AcCoA into G6P?". Allowing one hour of running time, we could identify 6 EMs with the desired property, which can be found in the supplementary material. The number of

reactions taking part in these EMs ranged from 53 to 267.

In a similar way, a possible usage of the tool would be to enumerate EMs that produce biomass, allowing the system to grow, but at the same time also produce one or more by-products (e.g. biofuels or toxic compounds). Such a set of EMs gives us insight into the growth-coupled production capabilities of a microorganism. Alternatively, these EMs could also be used as an input for other methods, such as constrained cut set computation [42].

5.4.2 Use-case scenario 2

While the main novelty of the chapter is Algorithm 2, Algorithm 1 provides an alternative method for computing EMs through a single given reaction. Compared to methods like [25], our algorithm improves running time at the cost of computing not necessarily the shortest EMs. While shorter EMs are more likely to represent biological phenotypes that occur in practice, it is unclear if only the shortest ones should be of interest. Rezola et al. [91] introduced the concept of *generating flux modes* (GFMs) and compared the 100 shortest EMs with the 100 shortest GFMs producing lysine in the *E. coli* iAF1260 model [30]. The authors argued that even though GMFs are longer and take more time to compute, they present a more varied description of the total solution set. We extended their experiment by also considering Algorithm 1. We enumerated 100 EMs producing lysine and compared their statistical properties with the other two methods. The results are summarized in Table 5.3.

Table 5.3: Comparison between shortest EMs, shortest GFMs and Algorithm 1. (NoR) - total number of reactions involved in any of the computed modes. (LI) - length interval of the modes. (AHD) - Average Hamming distance. '*' indicates numbers taken from [91].

Method	NoR	LI	AHD
shortest EM*	54	25-26	12.79
shortest GFMs*	132	25-37	16.21
Algorithm 1	272	25-57	26.082

It becomes clear from Table 5.3 that Algorithm 1 computes a broader range of pathways that can be used for lysine production. While it still finds some representatives of the shortest EMs, it also detects more advanced EMs. Fig. 5.2 presents the length distribution of the EMs obtained by Algorithm 1.

Even if we extend our interest beyond the set of shortest EMs, one might still want to avoid unreasonably long ones (i.e., those involving more than L reactions, for some $L \geq 1$). This can easily be achieved by adding the

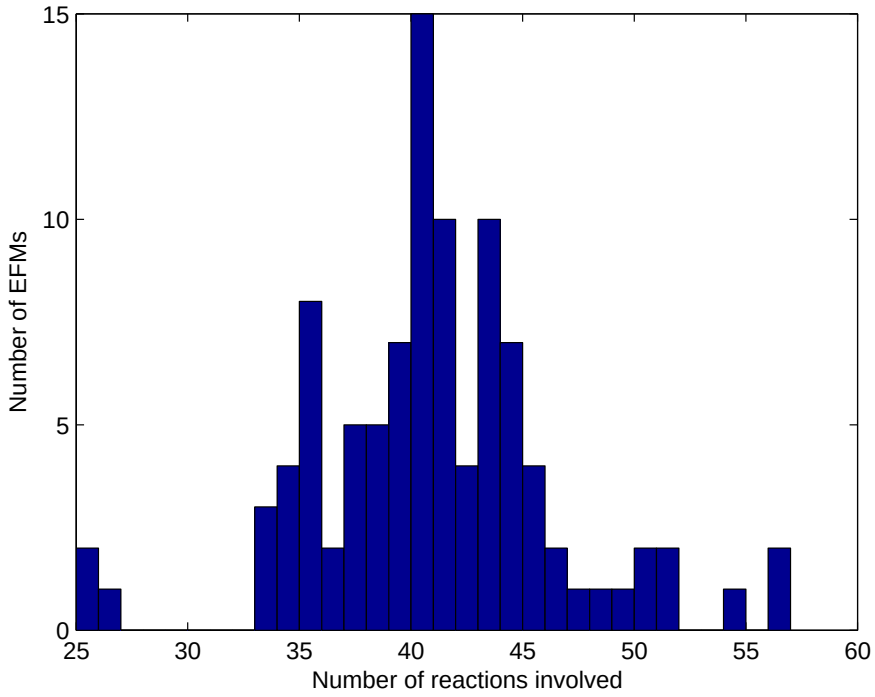


Figure 5.2: Histogram of the length of EMs found by Algorithm 1

constraint $\sum_{l\in\mathcal{R}} a_l \leq L$ to the MILP formulation.

In our next experiment, we compared Algorithm 1 with the method of [25] for computing shortest EMs. As input we used the metabolic network of *S. cerevisiae iND750* [29] with ethanol (R_ETOH) being the target reaction, required to be active in all EMs. For various M values and choosing the flux variables to be either integer or continuous, 20 EMs were computed in each case. Table 5.4 summarizes the results.

Table 5.4: Comparing Algorithm 1 with the computation of shortest EMs (Running time in secs).

		Integer variables		Continuous variables	
	Method	Length	Time	Length	Time
M = 10	Shortest	6-10	1719s	6-10	2074s
	Algo. 1	6-15	16s	6-23	15s
M = 100	Shortest	6-10	8158s	6-10	3421s
	Algo. 1	6-21	21s	6-31	18s
M = 1000	Shortest	6-10	14362s	6-10	7780s
	Algo. 1	6-31	16s	6-50	29s

Table 5.4 highlights the main differences in the two methods. Algorithm 1 finds EMs much quicker at the expense of individual EMs' length. As expected, in almost every case, the higher the constant M was chosen, the more expensive the computation becomes. De Figueiredo et al. [25] remark that using continuous flux variables, the MILPs might be more time consuming to solve. However, when computing the shortest EMs in this particular network, we do not consistently observe this behavior. Indeed, while for $M = 10$, choosing the flux variables as continuous proved to be slightly more expensive, in all other cases continuous variables led to shorter running times.

In every iteration of Algorithm 1, the MILPs to be solved become more complex, since they contain additional constraints. Over time, these constraints are expected to slow down the solving of (MILP2). Hence, the rate at which new EMs are computed is expected to decrease, the more EMs we enumerate. To study this effect, we enumerated 1000 EMs in the network of *S. cerevisiae iND750* [29], with ethanol (R_ETOH) being the target reaction. The flux variables were chosen as continuous, while a value for M of 1000 was used. The total running time was 4350 seconds. For reference, in the same time frame we could compute only 16 EMs with the method presented by [25]. We measured the time t_i required for the computation of the i-th EM, for all $i \in \{1, ..., 1000\}$. In order to study how the lengths of consecutive EMs vary, we also measured the length l_i of the i-th EM, for all $i \in \{1, ..., 1000\}$. In Fig. 5.3, we display the evolution of the mean times $\bar{t}_i := (\sum_{j=1}^{i} t_j)/i$ together with the mean lengths $\bar{l}_i := (\sum_{j=1}^{i} l_j)/i$.

5.4.3 Statistical analysis and flux uncoupling

The main bottleneck of the method are cases when EMs of the required type do not exist. While finding an EM if it exists seemed to work well in practice, proving their non-existence (i.e., showing (MILP3) to be infeasible) is rather hard and time-consuming. Intuitively, one can think of the MILP search-tree. Depending on how many feasible solutions there are, the solver might find one without traversing the whole tree. For proving the non-existence, the whole search-tree must be traversed.

In the next experiment, we tried to compute an EM for every pair of reactions. A maximum time of 60 seconds was allocated for each pair. Reaching the timeout meant that we were unable to compute if this pair shared an EM or not. The test has been performed on two small to medium-sized real world networks, the central metabolism of *E. coli* (ECC) [83] and the *H. pylori* (HP) [121] genome-scale metabolic network. For the constants M_0 and M_1, the value of 10000 was chosen, while the feasibility tolerance was set to 10^{-6}. The total running time in the case of ECC was approximately 3 hours, while the algorithm took close to 3 days in the case of HP.

Table 5.5 empirically sheds light on the nature of real-world networks.

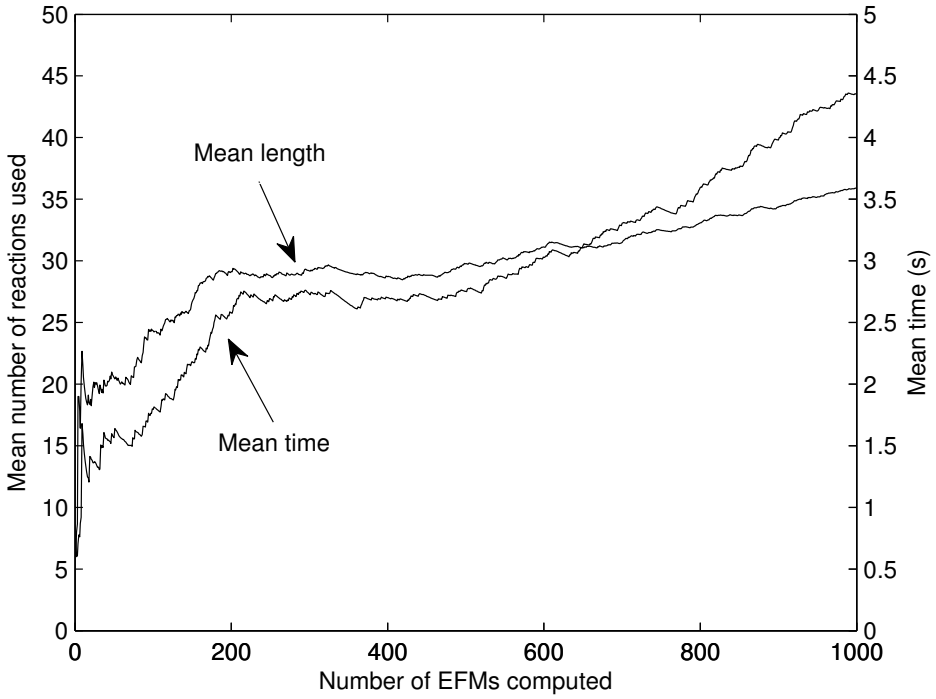


Figure 5.3: Mean time and mean length of the EMs computed.

Table 5.5: Summary of the correct computational results

Netw.	Reac.	Pairs	EM found	Mean length	No EM
ECC	90	8010	7691	24.36	176
HP	269	72092	66749	46.57	1862

One can observe in these networks that most pairs of reactions share at least one EM, and the algorithm presented in this chapter is able to find them.

Table 5.6: Summary of the bad instances

Network	Timed out		Numerical error	
	Absolute	Relative	Absolute	Relative
ECC	78	0.97%	65	0.81%
HP	2078	2.88%	1403	1.94%

From Table 5.6, it becomes clear that (although not many) there are cases where the algorithm does not produce a relevant result, either by not finishing before its time-out, or by producing an erroneous result. The latter cases are attributed to the incorrect choice of M_0 and M_1 and numerical imprecisions in the MILP solver.

Seeing the high probability for the existence of an EM containing two randomly chosen reactions, the question arises whether solving (MILP2) would suffice. If it correctly finds an EM in most cases, this would motivate using (MILP2) as a heuristic approach to the problem. Unless the two reactions are blocked, (MILP2) is always feasible, and the optimal solution is characterized by Cor. 5.3 which asserts that we either get an EM or a false positive. We performed the same experiment as before, and attempted to compute a flux vector for every pair of reactions. We decided about the EM property of the computed flux vectors by applying the rank test [34, Lemma 2]. For every instance, a maximum of 60 seconds was allowed. Table 5.7 summarizes these results.

Table 5.7: The performance of (MILP2) on metabolic networks

Network	EM found	False positive	Timed out
ECC	5212	2686	202
HP	206	9213	62673

It turns out that in the case of the small *E. coli* network, (MILP2) performs reasonably well. In 67% of the cases where EMs exist, it correctly finds one. However, for the medium-scale *H. pylori* network, solving (MILP2) to optimality almost never terminates in the allocated time. Based on this empirical evidence we conclude that (MILP2) may not be a viable approach for medium- to large-scale metabolic networks.

5.5 Conclusion

We have presented novel methods to compute EMs involving any number of predefined target reactions. These algorithms can also be used to distinguish between mutually exclusive and sometimes coupled reactions. From the application on genome-scale metabolic networks, we conclude that the methods work as intended and are fast enough for practical use. They should become a valuable asset for constraint-based analysis of metabolic networks.

A prototype implementation in Matlab is available for download at `https://sourceforge.net/projects/caefm`.

Summary of the chapter

- We presented several novel methods for computing EMs involving a predefined set of terget reactions.
- The methods developed within the chapter were successfully benchmarket against real-world genome-scale metabolic networks.
- A number of use-case scenarios were discussed offering potential practical usage for the methods.
- We analyzed how to choose suitable parameters for the MILP methods.
- Different statistical analyses have been performed in order to gain insight about the properties of the method, as well as about EMs in general.

Chapter 6

On the Order of Reactions in Elementary Modes

Contents

6.1 Introduction

In the previous chapter we have seen that one of the limits for the practical usability of finding elementary modes with certain predefined reactions comes from the fact that there is no inherent order among the reactions participating in an elementary mode. From a very superficial perspective, elementary modes are vectors, and on their own they carry little explicit information about the topology of the flux they describe. In directed graphs, elementary modes would correspond to simple paths and minimal cycles, thus every acyclic elementary mode has an ordering of the reactions by definition. In contrast, a similar notion of ordering in hypergraphs is not trivial. Elementary modes that are not an internal cycle nor a simple path as defined in graphs will have branching traits (e.g. Fig. 6.2) or cyclic traits (e.g. Fig. 6.1). In such elementary modes not every two reactions can be compared and decided which needs to 'happen first'.

As shown in Figure 6.1, reactions 2 and 3 both need products of eachother in order to take place, thus at the first look they are incomparable. By individually studying elementary modes, in some cases it is possible to deduce a partial or even full ordering between the reactions. For example, in the figure before we can claim that reaction 1 definitely occurs before reaction 4. We aim to devise a general method that can analyse any elementary mode and extract ordering information, with some or all properties of a partial ordering (see

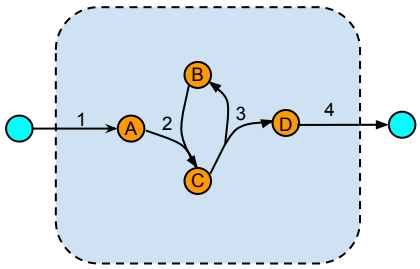


Figure 6.1: A small example where reactions 2 and 3 both need products produced by the other reaction. Hence, an ordering is not trivial to make.

Def. 1.15), by creating an ordering matrix. An ordering matrix is similar to the adjacency matrix of a graph in that the reactions (or edges) will represent the rows and columns of the matrix, and an entry (i, j) in the matrix is equal to 1 if and only if reaction i can be said to occur before reaction j. Having such an ordering, not only would lead to better understanding of individual elementary modes, but it would also aid in creating a visual representation of them. In this chapter, we will explore different notions of ordering and present methods to compute them for arbitrary elementary modes.

6.1.1 Hyperpaths versus elementary modes

An alternative way of modeling metabolic networks is through disregarding the stoichiometric coefficients and treating the metabolic model as a directed hypergraph [13, 93]. This modeling is not limited by the binary edges of a directed graph modeling (i.e., as in [16]), while at the same time preserves some of its advantages. In specific, the notion of paths in directed hypergraphs is similar to that of paths in a directed graph.

Let $V = \{v_1, v_2, ..., v_n\}$ be the set of vertices (or equivalently, metabolites).

Definition 6.1 (Hyperarc)**.** *An ordered pair $e := (X, Y)$ is called a hyperarc, where $X, Y \subseteq V$ and $X \cap Y = \emptyset$. X is the set of reactants and Y is the set of products for reaction e.*

For a hyperarc e, X_e will represent its set of reactants while Y_e will denote its set of products.

Definition 6.2 (Hypergraph)**.** *$H := (V, E)$ with E a set of hyperarcs, $E := \{e_1, e_2, ..., e_m\}$ is called a hypergraph.*

Next, we give a slightly modified definition of hyperpaths to the one defined in [13]. In both the original and our definition, the hyperpath defines a total

ordering (see Def. 1.16) of the hyperedges participating in the hyperpath. The difference is that in [13] the authors allow any starting set of metabolites for the hyperpath to begin with and any terminal set of metabolites to end with. In the definition we are about to present, a hyperpath always has to start with boundary import reactions and end with boundary export reactions. Therefore the definition in [13] can be thought of as a generalization of the definition we are using.

Definition 6.3 (Hyperpath)**.** *A non-empty set of hyperarcs $P \subseteq E$ with $p := |P| > 0$ is a hyperpath if the following two conditions hold*

- *there is a bijective mapping $\sigma : \{1, ..., p\} \to P$ such that for $\forall k \in \{1, ..., p\}, X_{\sigma(k)} \subseteq \cup_{j<k} Y_{\sigma(j)}$.*
- $\cup_{i \in P} Y_i \subseteq \cup_{i \in P} X_i$

In the previous definition, the first condition assures that we can order the reactions in such a way that for any reaction its predecessors will produce all of its reactants, allowing the reaction to take place. The second condition is necessary to guarantee that all the produced metabolites are consumed as well. In fact, as a consequence of the two conditions, $\cup_{i \in P} Y_i = \cup_{i \in P} X_i$ will always hold, while $X_{\sigma(1)} = \emptyset$ is also satisfied. It is possible to generalize the above definition for hyperpaths producing certain sets of metabolites [13].

Definition 6.4 (Minimal hyperpath)**.** *A hyperpath P is said to be minimal if no proper subset $P' \subset P$ is a hyperpath.*

In [13] the authors present a fast algorithm to compute hyperpaths producing given desired metabolites.

While this modeling does provide an inherent order inside every hyperpath, questions arise about the practical meaning of it. The definition of a hyperpath presumes the existence of an ordering, but does not require the uniqueness of it. Indeed, the same hyperpath may admit several different orders for the reactions. This proves that one cannot pick a single ordering as it would lead to contradicting and meaningless results. For example, in Figure 6.2 both $[1, 2, 3, 4]$ and $[1, 2, 4, 3]$ are valid orderings. If we would only know about one, e.g. the first ordering, then claiming that reaction 3 occurs before reaction 4 would be incorrect. In order to derive meaningful results, one would need to study all possible orderings.

Minimal hyperpaths and elementary modes intuitively seem closely related. However, these two concepts are completely independent. As it is demonstrated by the figures, there exist sets of reactions that are elementary modes but not hyperpaths (e.g. Figure 6.1), hyperpaths that are not elementary modes (e.g. Figure 6.3), and there are sets of reactions that are both (e.g. Figure 6.2).

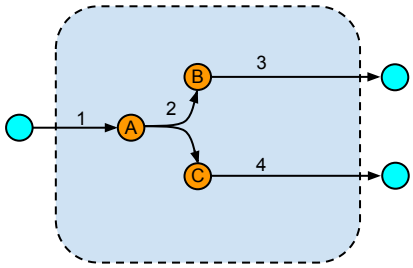


Figure 6.2: An example of an elementary mode that is also a hyperpath.

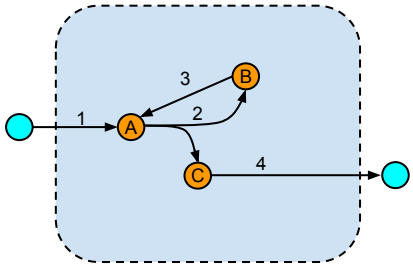


Figure 6.3: Example of 4 reactions that form a hyperpath, but not an elementary mode. In fact, from the steady-state condition of metabolites A and B it follows that reaction 1 is a blocked reaction.

In the following we will focus on defining an ordering relationship, similar to the one for hypergraphs, but for a pair of reactions within an elementary mode.

6.2 Methods

One of the advantages of studying an ordering within an elementary mode as opposed to within a genome-scale metabolic network is that in an elementary mode, if the flux rate of a reaction is given, then all other flux-rates are uniquely determined. For this reason, we can study elementary modes as stand-alone metabolic networks where every reaction is irreversible, even if some of the originating reactions were reversible. For the remainder of this chapter, we will assume that our input is a metabolic network $N := (S, Irr, R, M)$ (with $m = |M|$ metabolites, $n = |R|$ reactions) that is a single elementary mode (i.e. $dim(kern(S)) = 1$) and a flux vector e^* representing the elementary mode in N. Note, that unless $Irr = \emptyset$, one can compute an equivalent elementary mode to e^* directly from S [52]. If $Irr = \emptyset$ (i.e., there are only reversible reactions), the method in [52] computes an elementary mode e that is equivalent to either e^* or $-e^*$. However, since e^* is given and it uniquely determines the direction of all reversible reactions, we can consider all reactions to be irreversible. Under these assumptions, the only necessary input is the stoichiometric matrix S.

6.2.1 Ordering based on simple reachability

The intuitive idea of ordering that we will try to capture in our definition is similar to the one for hypergraphs. Two reactions are ordered if one is reachable from the other, but not vice-versa. First, we formally define reachability.

Definition 6.5 (Reachable)**.** *Let r_1, $r_2 \in R$. Then reaction r_2 is reachable from reaction r_1 in N if and only if there exists a bijective mapping of $\sigma : \{1, ..., n\} \to R$ such that:*

- $\sigma(1) = r_1$
- *For every $i < \sigma^{-1}(r_2), Y_{\sigma(i)} \cap X_{\sigma(i+1)} \neq \emptyset$*

Intuitively, one can think of reachability as the existence of a directed pathway over the branches of reactions. This is ensured by the second condition which requires that any two consecutive reactions in the pathway must share at least one metabolite in such a way that the first reaction produces it, while the following reaction consumes it.

As an example. let us have a look at Fig. 6.3. Here reaction r_4 is reachable from reaction r_3 since $\sigma(1) = r_3, \sigma(2) = r_2, \sigma(3) = r_4, \sigma(4) = r_1$ satisfies both

required conditions. However, reaction r_3 is not reachable from reaction r_4 since there is no σ that would satisfy the second condition. In fact, no other reaction is reachable from r_4.

Checking whether one reaction is reachable from another can be computed in linear time in R. In fact, Algorithm *Reachable* computes all the reactions that are reachable from a given reaction r_1. Note that, for simplicity, in the algorithm we have reused the definitions of Met (see Def. 4.4) and *Reac* (see Def. 4.6) from Chapter 4. In the previous algorithm, the while loop is executed at most n times, thus the algorithm terminates.

Algorithm Reachable

Input

r - a reaction of interest

S -the stoichiometric matrix of the network

Output

V - the set of reachable reactions from r

1. $V := \{r\}$
2. $V' := \emptyset$
3. **while** $V \neq V'$
4. $\quad V' := V$
5. $\quad V := V \cup Reac^{-}(Met^{+}(V))$
6. **end while**

Based on the reachability definition, we distinguish the following three cases of ordering.

Definition 6.6 (Ordering)**.** *Two reactions r_1 and r_2:*

- *directionally ordered if $r_1 = r_2$ or if r_2 is reachable from r_1 but not vice-versa. We will note this relation as $r_1 \leq r_2$.*
- *ambiguously ordered if both r_2 is reachable from r_1 and r_1 is reachable from r_2.*
- *unordered if neither r_2 is reachable from r_1 nor r_1 is reachable from r_2.*

Executing Algorithm Reachable once for r_1 and once for r_2 gives us the necessary information to classify the order relationship between them according to the previous three cases. In general, performing the algorithm n times, once for every reaction, is sufficient to classify every pair of reactions. While this would be a valid solution to determine all the orderings between reactions, it is not the most efficient way to do it.

In the following we will show an alternative way of computing the same information, based on the Levi-graph [67] and computing its strongly connected components.

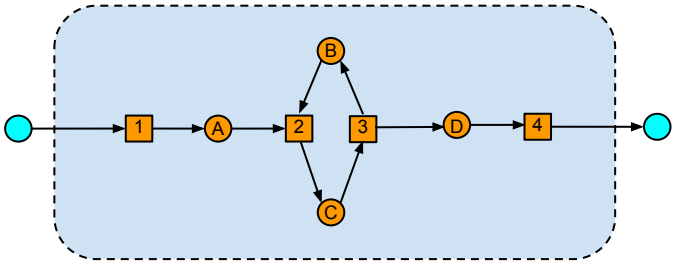


Figure 6.4: Example of the Levi-graph corresponding to Figure 6.1. Note that the blue nodes, originating from external metabolites, are part of the Levi-graph.

Definition 6.7 (Incidence structure)**.** *Let P and L be two sets of entities (e.g. P are points and L are lines). Then $I \subseteq P \times L$ is an incidence relation between the two sets of entities and $C := (P, L, I)$ is called an incidence structure.*

Clearly, a hypergraph or a metabolic network can be looked at as an incidence structure between the metabolites M and the reactions R. In specific, let $I := \{(i, j) \mid \text{with } S_{ij} \neq 0\}$. Then $C = (M, R, I)$ is an incident structure.

Definition 6.8 (Levi-graph)**.** *For an incident structure $C = (P, L, I)$, the graph $G := (P \cup L, I)$ is called the Levi-graph of C [67].*

The Levi-graph is a bipartite graph and it is closely related to Petri nets [85]. Given a metabolic network, the size of its Levi-graph will be $|M| + |R| + |B(R)|$ nodes and $NNZ(S) + |B(R)|$ edges. Here $NNZ(S)$ denotes the number of nonzero elements of S and $B(R)$ represents the set of boundary reactions. Note the importance of adding |B(R)| to both the set of nodes and the set of edges. This results from the fact that in the stoichiometric matrix boundary reactions have a 'hidden' branch. The 'part' of the reaction that crosses the system boundary is not represented by any value in S. Hence we include the external metabolite corresponding to each boundary reaction into new set of nodes, which also allows including all the edges by the Levi-graph. Transforming a metabolic network into its Levi-graph representation can be done in polynomial time.

The advantage of having the Levi-graph representation is that classical graph-theoretical algorithms can be applied to it. In specific, the strongly connected components (SCC) of this graph are computable in polynomial time with a number of existing algorithms (i.e., Tarjan's algorithm [114] or

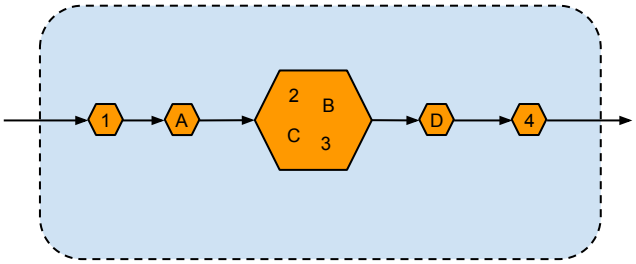


Figure 6.5: Strongly connected components of the graph from Figure 6.4

Kosaraju's algorithm [19]). The strongly connected components form a directed acyclic graph (DAG). Since we were starting with a metabolic network that is an elementary mode, this guarantees that the DAG will be connected.

The following observation provides the connection between the strongly connected components and the ordering of reactions.

Observation 6.9. *The strongly connected components of the Levi-graph obtained from a metabolic network corresponding to an elementary mode uniquely determine the orderings for every distinct pair of reactions. In specific, two reactions r_1 and r_2 with $r_1 \neq r_2$ are:*

- *directionally ordered if there is a directed path from the SCC containing r_1 to the SCC containing r_2.*
- *ambiguously ordered if r_1 and r_2 belong to the same SCC.*
- *unordered if there is no directed path from the SCC containing r_1 to the SCC containing r_2.*

We note that Obs. 6.9 cannot be extended to the case when $r_1 = r_2$, since in this case the 'two' reactions belong to the same SCC. Nevertheless, this is not a problem since by using Def. 6.6 we don't need to compute the ordering in such cases by an algorithm. Once the strongly connected components are computed, a topological sorting algorithm [19] can be applied to find the ordering between the different SCCs. As an example, Figure 6.5 depicts the DAG obtained after computing the SCCs of Figure 6.4. The strongly connected components include both the reactions and metabolites (since both of these entities were represented by nodes in the Levi-graph). As such, a post-processing step is needed to remove the metabolites. Generally, the extra

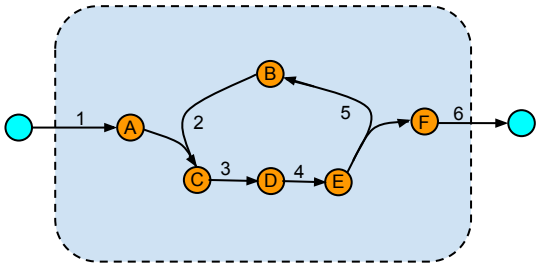


Figure 6.6: Example network

preprocessing step, transforming the hypergraph into a Levi-graph as well as the post-processing step required by this approach are not necessary and can be eliminated. Algorithm *computeSCCs* presents a modified Tarjan's algorithm [114] to compute the SCCs directly from the stoichiometric matrix.

Observation 6.10. *The set of reactions R with relation $<$ (directionally ordered) forms a partially ordered set.*

The reflexivity and antisymmetry follow directly from Definition 6.6. Transitivity follows from the fact that if r_2 is reachable from r_1, and r_3 is reachable from r_2, then we can construct a directed path from r_1 to r_3 by concatenating the two paths. The concatenated path might contain duplicate nodes, which we can lump together and delete any edges between them. Repeating this step until all duplicate nodes are removed results in a directed path from r_1 to r_3 satisfying the requirements, thus $r_1 \leq r_3$.

The above definition and algorithm of ordering perform well for reactions that do not belong to the same SCC. The question arises whether it is possible to define a weaker ordering within the reactions of an SCC. This becomes particularly important since we will show that, in some cases, almost all reactions of an elementary mode belong to one SCC.

6.2.2 Ordering based on pathway reachability

Analysing the elementary mode from Figure 6.6 with the previous method, one would get three SCCs. There would be one containing reaction $\{r_1\}$, one containing reactions $\{r_2, r_3, r_4, r_5\}$ and lastly, one containing reaction $\{r_6\}$. Intuitively however, one could argue that there is still some sense of ordering not captured by the method. In specific there is a weak ordering $\leq_w$ between the reactions of the second SCC in the form of $r_2 \leq_w r_3 \leq_w r_4 \leq_w r_5$. This information 'gets destroyed' by the feedback arcs of reaction r_2 and r_5 through metabolite B. In this subsection we will extend the previous method to account for these subtle ordering relations.

Algorithm computeSCCs

Input

S -the stoichiometric matrix of the network

Output

SCC - the set of strongly connected components in the Levi-graph corresponding to the network

```
1.  depth := 0, U := ∅, SCC := ∅
2.  visited[R] := false
3.  for each r ∈ R do
4.      if not visited[r] then
5.          visit(r)
6.      end if
7.  end for
8.
9.  function visit(r)
10.     visited[r] := true
11.     depth[r] := depth
12.     ancestor[r] := depth
13.     depth := depth +1
14.     push(U, r)
15.     N := getNeighbors(r)
16.     for each r' ∈ N do
17.         if not visited[r'] then
18.             visit(r')
19.             ancestor[r] := min(ancestor[r], ancestor[r'])
20.         else if r' ∈ U then
21.             ancestor[r] := min(ancestor[r], depth[r'])
22.         end if
23.     end for
24.     if ancestor[r] = depth[r] then
25.         C := ∅
26.         repeat
27.             s := pop(U)
28.             push(C, s)
29.         until s = r
30.         SCC := SCC ∪ {C}
31.     end if
32. end function
33.
34. function getNeighbors(r)
35.     N := ∅
```

(to be continued)

Algorithm computeSCCs (continued)

```
36.             for each i ∈ M with S_ir > 0 do
37.                 for each j ∈ R with S_ij < 0 do
38.                     N := N ∪ {j}
39.                 end for
40.             end for
41.             return N
42.         end function
```

Clearly, the reason why the described problem occurs is because cycles induce SCCs. Thus, we will modify our previous definition of reachability so that it reduces the effect imposed by cycles. Rather than defining reachability as the existence of an arbitrary directed path between two reactions, we want this directed path to be extensible into a larger pathway that has boundary reactions at its two ends. That is, the extended pathway should start with an incoming boundary arc of any reaction, then somewhere inbetween visit the two reactions we are interested in and finally end with an outgoing arc of a boundary export reaction. Note, that the extended larger path is not allowed to have duplicate nodes or edges. We will give an example after the definition.

Definition 6.11 (Pathway reachability)**.** *A reaction r_2 is pathway reachable from a reaction r_1 if there exists a bijective mapping $\sigma : \{1, ..., n\} \to R$ and a bijective mapping $\pi : \{1, ..., m\} \to M$ such that:*

- $\sigma(1) \in B^+(R)$
- $\sigma^{-1}(r_1) < \sigma^{-1}(r_2) \leq \sigma^{-1}(r_k) \leq m$
- *For every* $i < \sigma^{-1}(r_k), \pi(i) \in Y_{\sigma(i)} \cap X_{\sigma(i+1)}$

with $r_k = argmin\{\sigma^{-1}(r_j) \mid r_j \in B^-(R)\}$

In the previous definition, $B^+(R) := \{j \in R \mid \forall i \in M, S_{ij} \geq 0\}$ represents the set of boundary incoming, while $B^-(R) := \{j \in R \mid \forall i \in M, S_{ij} \leq 0\}$ the set of boundary outgoing reactions. The reaction r_k is the first outgoing boundary reaction in the ordering of σ. The first condition in the definition assures that we start the pathway with an incoming boundary reaction. The second condition guarantees that we will visit reactions r_1, r_2 and r_k (the first outgoing boundary reaction) in the proper order. The third condition enforces the pathway's connectedness as well as the existence of a unique metabolite between each two reactions (i.e. excludes the existence of an internal loop).

Using Fig. 6.6 as an example, we note that r_5 is reachable from r_2, and r_2 is reachable from r_5. Now, r_5 is also pathway reachable from r_2 since $\sigma = (1, 2, 3, 4, 5, 6)$ satisfies the conditions of Def. 6.11. In specific, r_1 is a boundary incoming reaction. Moreover $\sigma^{-1}(r_1) = 2$, $\sigma^{-1}(r_2) = 5$ and $\sigma^{-1}(r_k) = 6$,

thus the second condition of the definition is also satisfied. Lastly, using $\pi = (A, C, D, E, F, B)$ satisfies the third condition. In specific, metabolite A is produced by reaction $\sigma(1) = 1$ and consumed by reaction $\sigma(2) = 2$. Then metabolite C is produced by reaction $\sigma(2)$ and consumed by reaction $\sigma(3) = 3$ and so on.

In contrast, r_2 is not pathway reachable from r_5 as no σ and π exist that satisfy Def. 6.11.

Based on pathway reachability we can define the pathway ordering very similarly to the previous ordering.

Definition 6.12 (Pathway ordering)**.** *Two reactions r_1 and r_2 are directionally pathway ordered if $r_1 = r_2$ or if r_2 is pathway reachable from r_1 but not vice-versa. We will note this relation as $r_1 \leq_w r_2$.*

By definition, pathway ordering is reflexive and antisymmetric, however, as we will see it later it is not a transitive relation.

There are several ways to compute the pathway ordering between two reactions. In the following we will present two methods for this purpose. The basis of both methods is the enumeration of all transport pathways (cf. p.7).

We start with the Levi-graph of an elementary mode. If we choose not to distinguish the nodes corresponding to reactions and the nodes corresponding to metabolites, we can look at this graph as an artificial metabolic network. This metabolic network has the property that it contains no hyperedges, but we can still apply the methods developed for general metabolic networks. In specific, we can enumerate elementary modes. Acyclic elementary modes will correspond to the transport pathways that we are seeking for. Several methods exist in the literature for enumerating EMs (i.e., Metatool [130] or EFMTool [118]). Once the artificial elementary modes are computed, finding the pathway ordering is a simple exercise of individually ordering each of these acyclic EMs and using them to extract the pathway reachability information.

Algorithm *pathwayReachableEM* with input argument E_G (the set of elementary modes of the Levi-graph) describes the procedure. The constants n and m used in the method represent the size (number of reactions and metabolites) of the elementary mode. The algorithm implicitly assumes that, in the Levi-graph, nodes corresponding to metabolites preserve their numbering, while nodes corresponding reactions will be numbered $m + 1$ to $m + n$ (the assumption is used on line 8 of the algorithm).

Having performed *pathwaysReachableEM*, deducing the pathway ordering becomes trivial. In specific, $r_i \leq_w r_j$ holds if and only if $O_{ij} = true$ and $O_{ji} = false$.

With the method described two concerns are raised. In general, computing all elementary modes of a metabolic network is considered to be a difficult task to perform in practice. However, having as input the Levi-graphs of elementary modes greatly reduces the number of edges, thus making this task

Algorithm pathwayReachableEM(E_G)

Input

E_G -the set of EMs of the Levi-graph associated with the network

S -the stoichiometric matrix of the network

Output

O - the pathway reachability table

```
1.    O := {false}^{n×n}
2.    for each e ∈ E_G do
3.        o := findFirst(e)
4.        if o ≠ ∅ then
5.            o := findRest(e, o)
6.        end if
7.        for each (i, j) ∈ [1..|o|]^2 with i < j do
8.            O(Met_G^+(o(i)) − m, Met_G^+(o(j)) − m) := true
9.        end for
10.   end for
11.
12.   function findFirst(e)
13.       for each r in Reac(e) do
14.           if Met^−(r) = ∅ then
15.               return r
16.           end if
17.       end for
18.       return ∅
19.   end function
20.
21.   function findRest(e, o)
22.       for each r in Reac(e)
23.           if Met_G^−(r) = Met^+o(|o|) then
24.               return findRest(e, [o, r])
25.           end if
26.       end for
27.       return 0
28.   end function
```

feasible. The second concern relates to the memory usage of the method, as storing potentially millions of EMs can be challenging. These two issues motivate an alternative way of computing the pathway reachability table O. Rather than computing all transport pathways at once, one should enumerate them one-by-one in a linear way. Each pathway computed can be used as soon as it is computed, and discarded right after use, thus dramatically reducing the total memory need of the algorithm at the expense of potentially longer computation time. The algorithm *pathwayReachableDFS* presents the alter-

native procedure of computing the pathway orderings. The method is based on depth-first search (DFS) technique [19] and can be applied directly on the hypergraph, skipping the need for the Levi-graph transformation.

Algorithm pathwayReachableDFS

Input

S -the stoichiometric matrix of the network

Output

O - the pathway reachability table

```
1.   O := {false}^{n×n}
2.   U := ∅
3.   for each r ∈ R do
4.       if Met⁻(r) = ∅ then
5.           visit(r)
6.       end if
7.   end for
8.   return O
9.
10.  function visit(r)
11.      push(U, r)
12.      if Met⁺(r) = ∅ then
13.          for each (i, j) ∈ [1..|U|]² with i < j do
14.              O(U(i), U(j)) := true
15.          end for
16.      else
17.          for each p ∈ Met⁺(r) do
18.              for each r' ∈ Reac⁻(p) do
19.                  if r' ∉ U then
20.                      visit(r')
21.                  end if
22.              end for
23.          end for
24.      end if
25.      pop(U)
26.  end function
```

Applying either *pathwayReachableEM* or *pathwayReachableDFS* for Figure 6.7, the resulting pathway reachability table O is given in Table 6.1. Based on this table we can conclude the following pathway ordering relationships in the network: $r_1 \leq_w r_2$, $r_1 \leq_w r_3$, $r_1 \leq_w r_4$, $r_1 \leq_w r_5$, $r_1 \leq_w r_8$, $r_2 \leq_w r_3$, $r_2 \leq_w r_4$, $r_2 \leq_w r_8$, $r_4 \leq_w r_5$, $r_4 \leq_w r_8$, $r_5 \leq_w r_3$, $r_5 \leq_w r_6$, $r_5 \leq_w r_8$, $r_6 \leq_w r_2$, $r_6 \leq_w r_3$ and $r_7 \leq_w r_8$. For comparison purposes, we reference Table 6.2 as well, which displays the standard reachability.

Unlike the ordering defined in the previous subsection, pathway ordering is

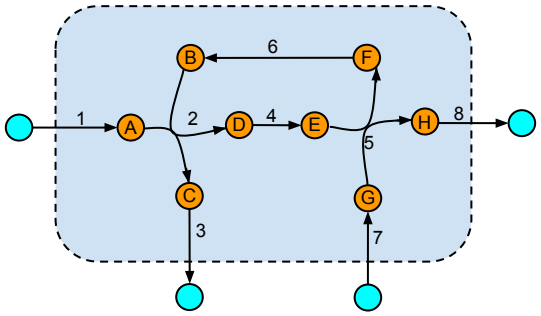


Figure 6.7: Example network

not transitive, thus $(R, \leq_w)$ is not a partially ordered set. We can demonstrate this with a counterexample from Figure 6.7. Here $r_4 \leq_w r_5$ holds and so does $r_5 \leq_w r_6$. However $r_4 \leq_w r_6$ does not hold.

Table 6.1: Pathway reachability table for the network in Figure 6.7. Bullet points represent values of 'true'.

Reaction	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
r_1	•	•	•	•	•			•
r_2		•	•	•	•			•
r_3			•					
r_4				•	•			•
r_5		•	•		•	•		•
r_6		•	•			•		
r_7		•	•		•	•	•	•
r_8								•

6.3 Results and discussion

When studying the relationship between the two described orderings, the question arises whether one is a subset of the other. Clearly, since the pathway ordering can find relationships for reactions in the same SCC, it will contain some pairs that the simple ordering does not. Moreover, the inclusion in the opposite sense is also not true. In Figure 6.7, $r_1 \leq r_6$ holds, but $r_1 \leq_w r_6$ does not. Hence the two orderings are unrelated.

The lack of transitivity certainly casts some doubt over the usefulness of the pathway ordering. The lack of this feature is attributed to the local nature of the pathway ordering (i.e., we order two reactions based only on a subset of the artificial pathways). A possible way to deal with this would be to combine

Table 6.2: Standard reachability table for the network in Figure 6.7. Bullet points represent values of 'true'.

Reaction	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
r_1	•	•	•	•	•	•		•
r_2		•	•					•
r_3			•					
r_4			•	•				•
r_5			•		•			•
r_6			•			•		•
r_7		•	•	•	•	•	•	•
r_8								•

the two orderings. That is, one could use the standard ordering for reactions belonging to different SCCs in the Levi-graph, and use the pathway ordering for reactions making part of the same SCC. The two ordering algorithms were implemented in Matlab [74] and combined according to the aforementioned idea. We applied the method on several elementary modes computed in Chapter 5. A graphical representation of each resulting ordering was automatically generated with the *biograph* function of the Bioinformatics Toolbox [74]. The first two EMs we have selected to illustrate were computed in the *E. Coli Core* [83], while the next two in the *E. Coli iJR904* [90] network.

- Fig. 6.8 illustrates the simple ordering of an elementary mode with 26 reactions that produces ethanol from glucose. *Cluster 1* represents a set of 16 reactions that could not be ordered with the first method. However, applying the pathway ordering on this subset of reactions finds additional ordering relations as displayed in Fig. 6.9.
- Fig. 6.10 depicts the simple ordering of an elementary mode with 17 reactions that produces lactate from glucose. The set of reactions in *Cluster 1* are further ordered with the pathway ordering and displayed in Fig. 6.11.
- Fig. 6.12 is an example of an elementary mode that produces lysine. The two clusters of reactions that could not be ordered with simple ordering are presented in Fig. 6.13 and Fig. 6.14.
- Fig. 6.15 shows again an EM that is producing lysine. Most reactions in *Cluster 1* could not be ordered even with the pathway ordering (Fig. 6.16).

The method can be extended to deal with general steady-state flux vectors that are not necessarily elementary modes. This is straight-forward to imple-

ment as in the algorithms themselves, the requirement of having an EM as the input network was seldom used. In fact, using the presented algorithms with general steady-state flux vectors is feasible as is. The only minor difference that should be considered is that in this case the steady-state flux vector should also be an explicitly stated input parameter. This was not necessary in the case when the input network was an EM as the stoichiometric matrix could be used to compute the EM up to a scalar multiple. Going one step further, extending the methods for general metabolic networks with no steady-state flux vector defined is a more difficult task. The main difficulty comes from the undecided orientation of reversible reactions. Splitting the reversible reactions into a forward and a backward arc might offer a starting point. However, this operation induces a number of two-cycles that might be counter-productive. This problem is proposed for further research.

The main motivation for finding an order between reactions was stemming from the deficiency of the method described in Chapter 5. While the current method is able to order the reactions of an already computed elementary mode, it does not help in finding EMs with a specific ordering. For solving this problem one would need to formulate the presented ordering definitions as linear constraints with possibly integer variables. At this point it is not clear whether this is a feasible task, hence this problem should be further investigated.

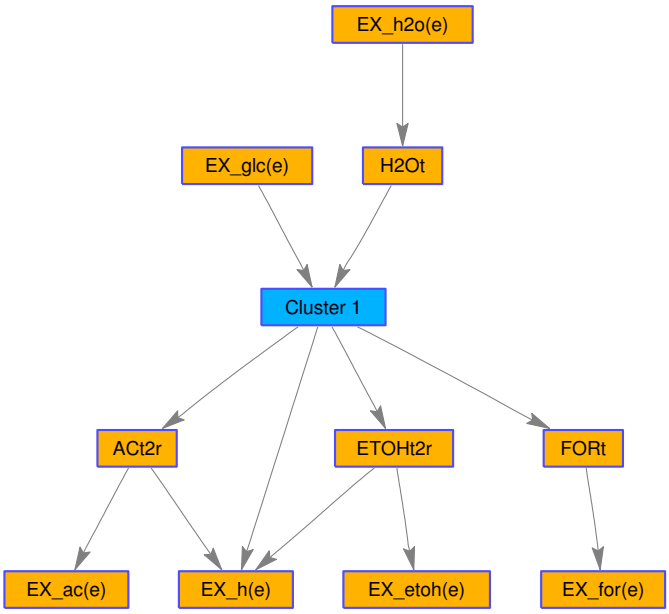


Figure 6.8: Graphical representation of the simple ordering of an elementary mode that produces ethanol from glucose. *Cluster 1* contains 17 reactions that cannot be ordered.

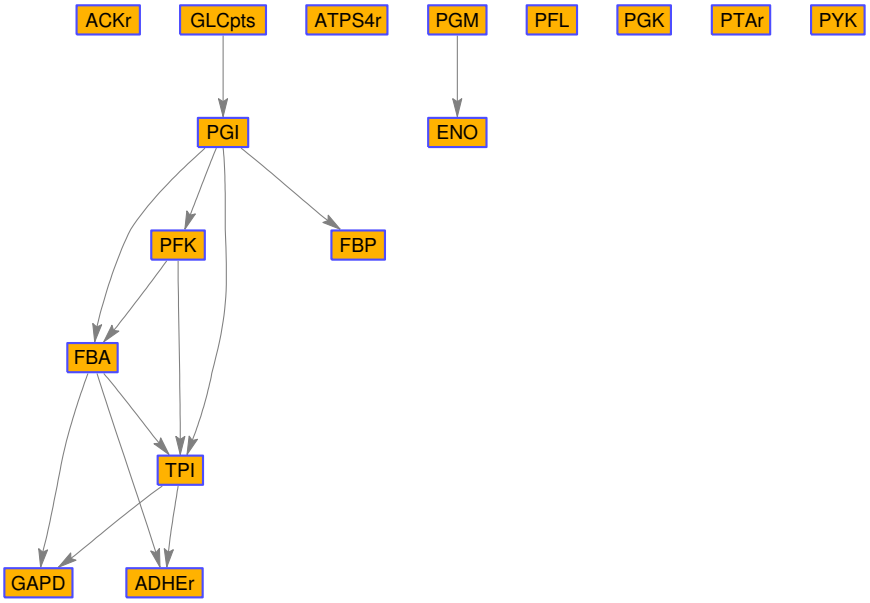


Figure 6.9: Pathway ordering of the reactions in Cluster 1 from Fig. 6.8

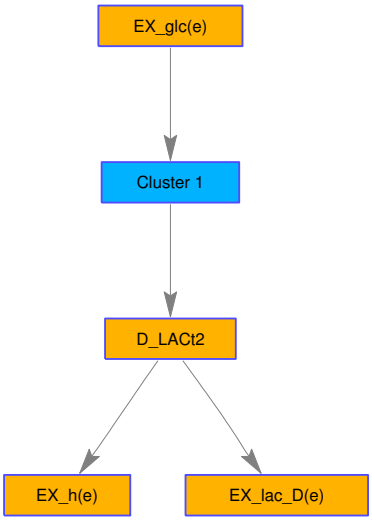


Figure 6.10: Graphical representation of the simple ordering of an elementary mode that produces lactate from glucose. *Cluster 1* is a set of 13 reactions that cannot be ordered.

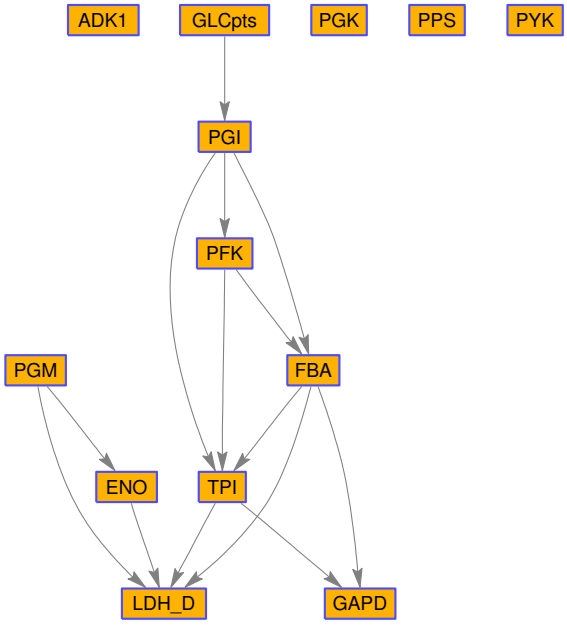


Figure 6.11: Graphical representation of the pathway ordering for the reactions in Cluster 1 from Fig. 6.10

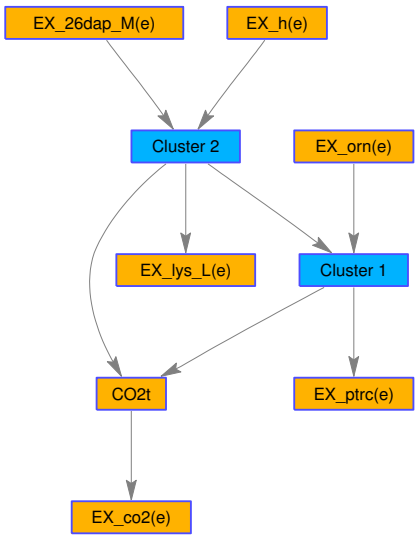


Figure 6.12: Graphical representation of the simple ordering of an elementary mode that produces lysine. *Cluster 1* is a set of 2, while *Cluster 2* is a set of 4 reactions that cannot be ordered.

Figure 6.13: Graphical representation of the pathway ordering for the reactions in Cluster 1 from Fig. 6.12. No ordering is possible between the two reactions.

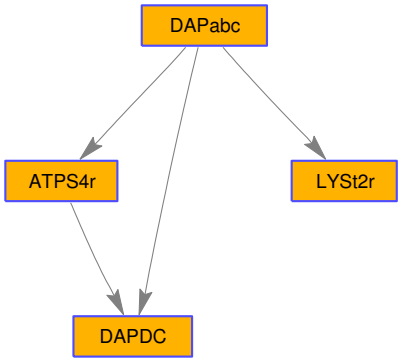


Figure 6.14: Graphical representation of the pathway ordering for the reactions in Cluster 2 from Fig. 6.12

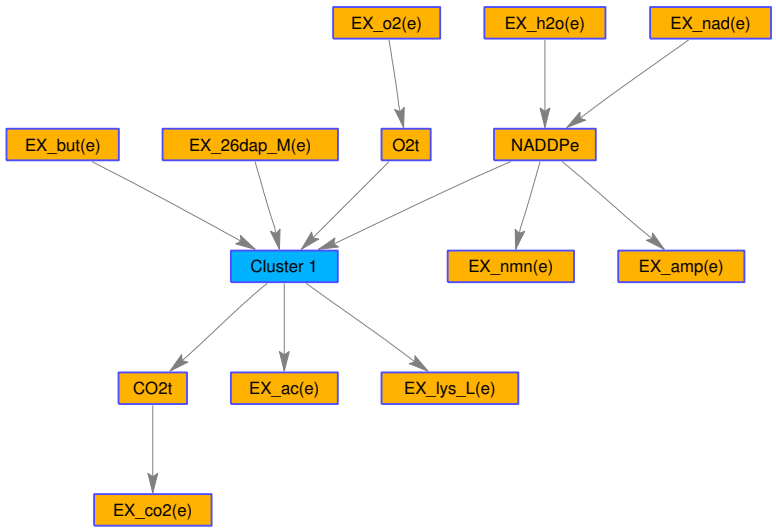


Figure 6.15: Graphical representation of the simple ordering of an elementary mode that produces lysine. *Cluster 1* is a set of 15 reactions that cannot be ordered.

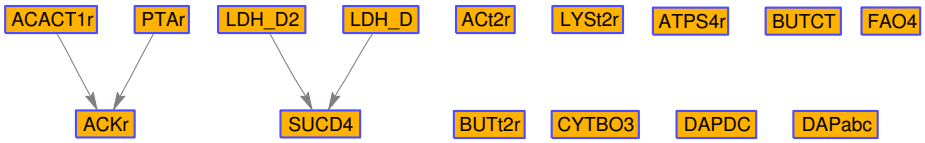


Figure 6.16: Graphical representation of the pathway ordering for the reactions in *Cluster 1* from Fig. 6.15. No ordering is possible between most reactions in this set.

Summary of the chapter

- We compared hyperpaths with elementary modes and argued why ordering the reactions is necessary.
- Two distinct definitions of ordering are given, the simple ordering and the pathway ordering.
- Graph-based algorithms are presented to compute both the simple ordering and the pathway ordering of reactions.
- The ordering algorithms have been successfully applied on elementary modes of real metabolic networks.

Chapter 7

Analysis of Metabolic Subnetworks by Projection

Contents

Remark: This chapter summarizes and extends results originally published in [73].

7.1 Introduction

Metabolic pathway analysis is the study of meaningful minimal pathways or routes of connected reactions in metabolic network models [61, 117]. Two closely related concepts are often used for explaining such pathways: elementary modes (EMs) [105, 103] and extreme pathways (EXPAs) [96]. From a mathematic perspective, EMs and EXPAs are generating sets of the flux cone [61, 46]. Several approaches have been proposed for the computation of such pathways [87, 132, 35, 57, 8, 130, 118, 119].

While EM and EXPA analysis are promising approaches for studying metabolic networks [98, 123], due to the combinatorial explosion of the number of such pathways [60, 135], this kind of analysis cannot be performed for "large" networks. Recent advances in the computation of EMs and extreme rays of polyhedral cones [118, 119] have made it possible to compute tens of millions of EMs. However, computing all of them for large genome-scale networks may still be impossible. Additionally, as we have seen in Chapter 5, one is often only interested in the behavior of a subset of reactions, and not all of them. Therefore, even if the EMs are computable, possibly many of them are not relevant because they are unrelated to the reactions of interest.

The goal of this chapter is to introduce the new concept of *Projected Cone Elementary Modes* (ProCEMs) for the analysis of substructures of metabolic networks. The organization is as follows. Firstly, the mathematical concepts used in the text are formally defined. Secondly, we review the studies which have tried to investigate subnetwork of large-scale networks. In the next step, we present the concept of ProCEMs and propose a method to compute them. Finally, we compare ProCEMs with elementary flux patterns (EFPs) from the mathematical and computational point of view, and analyse some concrete biological networks.

7.2 Definitions

Definition 7.1 (Projection)**.** *Given a set $Q \subseteq \mathbb{X} \times \mathbb{Y} = \mathbb{R}^n$, where $\mathbb{X}$ resp. $\mathbb{Y}$ are subspaces of $\mathbb{R}^n$ of dimension p resp. q with $p + q = n$, the projection of Q onto $\mathbb{X}$ is defined as*

$$\mathcal{P}_{\mathbb{X}}(Q) := \{x \in \mathbb{X} \mid \exists y \in \mathbb{Y}, (x, y) \in Q\}. \tag{7.1}$$

In the special case, when $Q = \{v\}$, we will simply write $\mathcal{P}_{\mathbb{X}}(v)$ instead of $\mathcal{P}_{\mathbb{X}}(\{v\})$.

Proposition 7.2. *Let Q be a polyhedral cone. Then the projection $\mathcal{P}_{\mathbb{X}}(Q)$ is also a polyhedral cone.*

Proof. Let $\{a_1, a_2, ..., a_z\}$ be a finite conic generating set of Q. We will prove that $\{\mathcal{P}_{\mathbb{X}}(a_1), \mathcal{P}_{\mathbb{X}}(a_2), ..., \mathcal{P}_{\mathbb{X}}(a_z)\}$ is a finite generating set of $\mathcal{P}_{\mathbb{X}}(Q)$.

Clearly, every $\mathcal{P}_{\mathbb{X}}(a_i) \in \mathcal{P}_{\mathbb{X}}(Q)$ for all $i \in \{1, 2, ..., z\}$. Let $x \in \mathcal{P}_{\mathbb{X}}(Q)$. Then there exists $y \in \mathbb{Y}$ with $(x, y) \in Q$. Thus, there exists $\alpha_1, \alpha_2, ..., \alpha_z \in \mathbb{R}^+$ with $(x, y) = \sum_{i=1}^{z} \alpha_i a_i$.

Looking at the coordinates of $\mathbb{X}$, we get $x = \sum_{i=1}^{z} \alpha_i \mathcal{P}_{\mathbb{X}}(a_i)$. □

Remark: The inverse of Prop. 7.2 is not true. Indeed, it is possible to project an infinitely generated cone such that the result is a polyhedral cone

(e.g. any two-dimensional projection of an ice-cream cone is always finitely generated).

Now consider a metabolic network $\mathcal{N} = (S, Irr)$ with $p + q$ reactions and a subnetwork $\mathcal{N}' = (S', Irr')$ given by a subset of p "interesting" reactions. For the flux cone C of $\mathcal{N}$ we assume $C \subseteq \mathbb{X} \times \mathbb{Y}$, where the reactions of $\mathcal{N}'$ correspond to the subspace $\mathbb{X}$. Prop. 7.2 assures that the projection of C onto the subspace defined by the set of interesting reactions, $\mathcal{P}_{\mathbb{X}}(C)$, is again a polyhedral cone. We will refer to this as the *projected flux cone* on $\mathbb{X}$.

Definition 7.3 (ProCEM and PEM)**.** *Consider a steady-state flux cone C, and $\mathbb{X}$ the subspace of interesting reactions.*

- *An elementary mode f of the projected cone $\mathcal{P}_{\mathbb{X}}(C)$ will be called a projected cone elementary mode (ProCEM).*
- *For an elementary mode e of the flux cone C, the projection $\mathcal{P}_{\mathbb{X}}(e)$ of e to $\mathbb{X}$ will be called a projected elementary mode (PEM).*

As we will see in Subsection 7.5.1, the two concepts of PEM and ProCEM are closely related but different.

If the subnetwork $\mathcal{N}'$ has to be analysed, PEMs might be more relevant than EMs, as they are in lower dimension and easier to study. PEMs can be enumerated easily if the set of all EMs is given, by following Def. 7.3. However, continuing with the assumption that the complete set of EMs is impractical (or impossible) to enumerate with currently existing methods, the computation of PEMs proves to be an infeasible approach for genome-scale networks.. As we will see in the following sections, ProCEMs provide an interesting alternative.

7.3 An overview of related methods

As we have seen already, the set of EMs of a genome-scale network may be large, and in general, cannot be computed with the available tools. Even if this is possible, one cannot easily extract interesting information from it. Therefore, a subset of EMs (or in case that we are interested in a subset of reactions, the set of PEMs) should be computed to reduce the running time and/or output size of the algorithm.

When the set of reactions of interest contains only a handful of reactions, the method presented in Chapter 5 provides an interesting route to compute corresponding EMs. However, this is only possible when such elementary modes, containing all reactions of interest exist. With an increased number of reactions of interest, the likelihood for the existence of such a specific EM dramatically decreases (e.g. it is enough for the subnetwork to contain two parallel reactions for no EM to exist). A workaround would be to individually

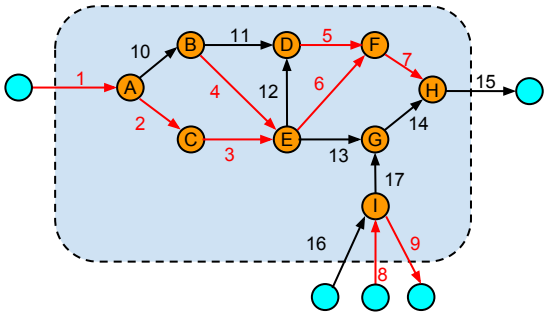


Figure 7.1: A small metabolic network with 17 reactions. We might be interested only in a subnetwork containing reactions $\{1, \ldots, 9\}$, which are shown by red arrows. This subnetwork will be called SuN.

study every subset of the subnetwork, which would lead to an exponential number of MILPs to solve.

Several alternative approaches to this problem have been proposed in the literature. These strategies can be classified into four main categories:

7.3.1 Computation of a subset of EMs

The first strategy is to constrain the complete set of EMs (or EXPAs) to a subset describing a phenotype space or a set of phenotypic data. For example, Covert and Palsson [21] showed that consideration of regulatory constraints in the analysis of a small "core metabolism" model can reduce the set of 80 EXPAs to a set of 2 to 26 EXPAs, depending on the applied regulatory constraints.

Urbanczik [125] suggested to compute "constrained" elementary modes which satisfy certain optimality criteria. As a result, instead of a full enumeration of EMs, only a subset of them should be computed, which results in a big computational gain. The idea of reducing the set of EMs has been used recently in an approach called *yield analysis* [110]. In this approach, the yield space (or solution space) is defined as a bounded convex hull. Then, the minimal generating set spanning the yield space is recalculated, and therefore, all EMs with negligible contribution to the yield space can be excluded. The authors show that their method results in 91% reduction of the EM set for glucose/xylose-fermenting yeast.

7.3.2 Computation of EMs in isolated subsystems

A second strategy to focus on the EMs (or EXPAs) of interest is to select a (possibly disconnected) subnetwork, rather than the complete metabolic model, by assuming all other reactions and metabolites to be "external", and

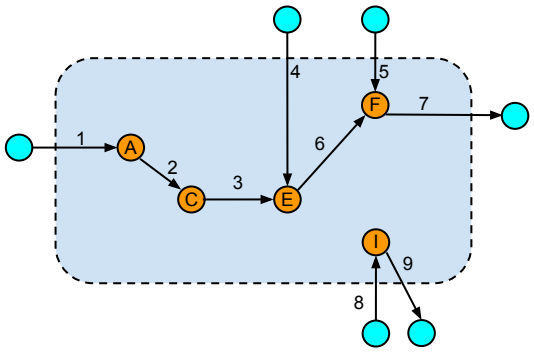


Figure 7.2: The reduced subnetwork comprising only the nine interesting reactions of interest from Fig. 7.1. Metabolites in the subnetwork, that do not have both an adjacent producing and consuming reaction, are treated as external.

computing the EMs (or EXPAs) of this selected subsystem. This idea, i.e., cutting out subsystems or splitting big networks into several subsystems, is broadly used in the literature (e.g., see [80, 102, 97, 95, 108, 111, 15, 109, 128, 129, 55, 120, 54]). In some of these studies, not only the system boundary is redrawn, but also some reactions may be removed for further simplifying the network.

Although this strategy is useful, it can result in serious errors in the computational analysis of network properties [50]. For example, dependencies and coupling relationships between reactions can be influenced by redrawing the system boundaries. Burgard et al. [12] showed that subsystem-based flux coupling analysis of the *H. pylori* network [95] results in an incomplete detection of coupled reactions. Kaleta et al. [50] suggest that neglecting such a coupling can lead to fluxes which are not part of any feasible EM in the original complete network. Existence of such infeasible "pathway fragments" [44] can result in incorrect conclusions.

To better understand this problem, we consider Fig. 7.1 as an example. Let us assume that we are interested in a subnetwork composed of reactions $1, \ldots, 9$. This subnetwork is called SuN. If we simply assume the "uninteresting" reactions and metabolites to be the external reactions and metabolites, we will obtain the subsystem shown in Fig. 7.2. This subnetwork has only four EMs, two of which are not part of any feasible steady-state flux vector in the complete network. For example, the EM composed of reactions 5 and 7 in Fig. 7.2 cannot appear at steady-state in the original complete network, because the coupling between reaction 1 and reaction 5 is broken. Therefore, analysing this subnetwork instead of the original network can result in false conclusions.

7.3.3 Computation of elementary flux patterns

We observed that some errors may appear in the analysis of isolated subsystems. One possible solution to this problem is to compute a "large" subset of PEMs, or alternatively, as suggested by Kaleta et al. [50], to compute the support of a subset of PEMs. These authors proposed a procedure to compute the elementary flux patterns (EFPs) of a subnetwork within a genome-scale network. A *flux pattern* is defined as a set of reactions in a subnetwork that is included in the support of some steady-state flux vector of the entire network [50]. A flux pattern is called an *elementary flux pattern* if it is not the union of two or more different flux patterns (see Subsection 1.3.6 for more details). Each EFP is the support of (at least) one PEM. It is suggested that in many applications, the set of EFPs can be used instead of EMs [50].

Although EFPs are promising tools for the analysis of metabolic pathways, they also have their own shortcomings. The first important drawback of EFPs is that they cannot be used in place of EMs in certain applications [35], where the precise flux values are required. For example, in the identification of all pathways with optimal yield [102, 107] and in the analysis of control-effective fluxes [111, 15, 138], the flux values of the respective reactions in the EMs should be taken into account.

Another important shortcoming of EFP analysis is that it is possible to have very different EMs represented by the same EFP, since flux values are ignored in EFPs. For example, consider the case that two reactions i and j are partially coupled [12]. This means that there exist at least two EMs, say e and f, such that $e_i/e_j \neq f_i/f_j$ [72]. However, if we consider a subnetwork composed of these two reactions, then we will only have one EFP, namely $\{i, j\}$. From the theoretical point of view, finding all EMs that correspond to a certain EFP is computationally hard (see Chapter 5).

Every EFP is related to at least one EM in the original metabolic network. However, one of the limitations of EFP analysis is that EFPs are activity patterns of some EMs, not necessarily all of them. We will show this by an example. In Fig. 7.1, the flux cone is a subset of $\mathbb{R}^{17}$, while the subnetwork SuN induces a 9-dimensional subspace $\mathbb{X} = \mathbb{R}^9$. If G is the set of EMs in Fig. 7.1, then the set of PEMs can be computed as $P = \{\mathcal{P}_{\mathbb{X}}(e) \mid e \in G\}$. The set of the 10 PEMs of SuN in Fig. 7.1 is shown in Table 7.1.

For the same network and subnetwork, we used `EFPTools` [49] to compute the set of EFPs. The resulting 7 EFPs are also presented in Table 7.1. If we compare the PEMs and EFPs, we find out that the support of each of the first 7 PEMs is equal to one of the EFPs. However, for the last three PEMs no corresponding EFP can be found in Table 7.1. This is due to the fact that $supp(p8) = E4 \cup E5$, $supp(p9) = E3 \cup E5$, and $supp(p10) = E1 \cup E2$. Hence, the flux patterns corresponding to these PEMs are not elementary. Therefore,

Table 7.1: List of elementary flux patterns, projected cone elementary modes and projected elementary modes of SuN. Flux through reactions $1, \ldots, 9$, respectively, are the elements of the shown vectors. Zero vector and also the empty set are excluded.

EFPs	EFP set	ProCEM	PEM	vector
$E1$	$\{9\}$	$u1$	$p1$	(0, 0, 0, 0, 0, 0, 0, 0, 1)
$E2$	$\{8\}$	$u2$	$p2$	(0, 0, 0, 0, 0, 0, 0, 1, 0)
$E3$	$\{1, 4\}$	$u3$	$p3$	(1, 0, 0, 1, 0, 0, 0, 0, 0)
$E4$	$\{1, 2, 3\}$	$u4$	$p4$	(1, 1, 1, 0, 0, 0, 0, 0, 0)
$E5$	$\{1, 5, 7\}$	$u5$	$p5$	(1, 0, 0, 0, 1, 0, 1, 0, 0)
$E6$	$\{1, 4, 6, 7\}$	$u6$	$p6$	(1, 0, 0, 1, 0, 1, 1, 0, 0)
$E7$	$\{1, 2, 3, 6, 7\}$	$u7$	$p7$	(1, 1, 1, 0, 0, 1, 1, 0, 0)
–	–	$u8$	$p8$	(1, 1, 1, 0, 1, 0, 1, 0, 0)
–	–	$u9$	$p9$	(1, 0, 0, 1, 1, 0, 1, 0, 0)
–	–	–	$p10$	(0, 0, 0, 0, 0, 0, 0, 1, 1)

some EMs may exist in the network which have no corresponding EFP on a certain subnetwork. This means that by EFP analysis possibly many EMs of the original network cannot be recovered. Informally speaking, we may ask whether the set of EFPs is the largest set of PEM supports which can be computed without enumerating all EMs.

7.3.4 Projection methods

A possible strategy to simplify the network analysis is to project the flux cone down to a lower-dimensional space of interest. In other words, if we are interested in a subnetwork, we may project the flux cone onto the lower-dimensional subspace defined by the "interesting" reactions. Note that projecting the flux cone is in general different from removing reactions from the network.

Consider the simple network shown in Fig. 7.3A and a graphical representation of its corresponding flux space in Fig. 7.3C (here, the axes r_1, r_2, r_3 correspond to reactions $1, 2, 3$, thus the flux cone is the open triangle shown in light green). This network has two EMs, which are the generating vectors of the flux cone, g_1 and g_2. Now, if we are interested in a subnetwork composed of reactions 1 and 2, then we can project the flux cone to the 2D subspace produced by these two reactions. The projected cone is shown in dark green. When the flux cone is projected onto the lower-dimensional space, new generating vectors may appear. In this example, g_1 and g_3 (in 2D space) are the generating vectors of the projected cone. Intuitively, one can think about g_3 as the projected flux vector through reaction 1 and 3. This projected flux

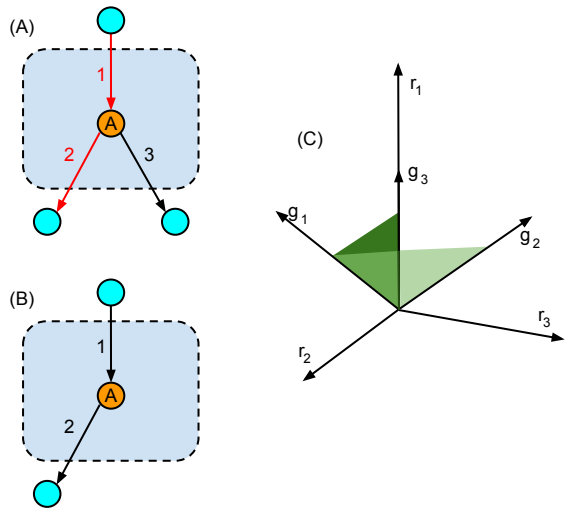


Figure 7.3: A small metabolic network. The reactions in the interesting subnetwork are shown as red arrows. (B): the same metabolic network as in (A), but with reaction 3 removed. The flux cone of this network is generated by only one vector, namely g_1. (C): The flux cone of the network in (A), shown in light green, can be generated by vectors g_1 and g_2. The projected flux cone is shown in dark green and is generated by g_1 and g_3 in a two dimensional subspace.

cone is certainly different from the flux cone of a network made by deleting reaction 3 (Fig. 7.3B). Such a network has only one EM, and its corresponding flux cone can be generated by only one vector, namely, g_1.

Historically, the idea of flux cone projection has already been used in some papers. Wiback and Palsson [134] suggested that the space of cofactor production of the red blood cell can be studied by projecting the cell-scale metabolic network onto a 2D subspace corresponding to ATP and NADPH production. A similar approach was used by Covert et al. [21] and also by Wagner and Urbanczik [133] to analyze the relationship between carbon uptake, oxygen uptake and biomass production. All the above studies considered very small networks. Therefore, the authors computed the extreme rays of the flux cone and then projected them onto the subspace of interest, without really projecting the flux cone. Urbanczik and Wagner [126] later introduced the concept of *elementary conversion modes* (ECMs), which are in principle the extreme rays of the cone obtained by projecting the original flux cone onto the subspace of boundary reactions. They suggest that the extreme rays of this "conversion" cone, i.e., the ECMs, can be computed even for large networks [124].

Following this idea, we introduce the ProCEM set ("Projected Cone Elementary Mode" set), which is the set of EMs of the projected flux cone. In

contrast to [126], we formulate the problem in a way that any subnetwork can be chosen, not only the boundary reactions. Additionally, we compare the closely related concepts of ProCEMs, PEMs and EFPs.

7.4 An algorithm for computing the set of ProCEMs

7.4.1 Computational procedure

For computing ProCEMs of a given subnetwork, we use the block elimination algorithm [4], which is based on the "projection lemma" (see Section I.4 in [76]).

Our algorithm needs three input objects: the stoichiometric matrix $S \in \mathbb{R}^{m\times n}$ of the network $\mathcal{N}$, the set of irreversible reactions $Irr \subseteq \{1,\dots,n\}$, and the set of reactions $\Sigma \subseteq \{1,\dots,n\}$ in the subnetwork of interest, while as an output it will return the complete set of ProCEMs. The computation of ProCEMs is achieved in three main consecutive steps.

Step 1 - Preprocessing: The aim of this step is to remove inconsistencies from the metabolic network and to transform it into a form suitable for the projection in Step 2. First, based on Σ we sort the columns of S in the form:

$$\bar{S} = \begin{pmatrix} \bar{A} & \bar{B} \end{pmatrix} \tag{7.2}$$

where the reaction corresponding to the i-th column belongs to Σ iff the i-th column is in $\bar{A}$. Next, the blocked reactions [12] are removed. Finally, each of the reversible reactions is split into two irreversible "forward" and "backward" reactions. The final stoichiometric matrix will be in the form:

$$S' = \begin{pmatrix} A & B \end{pmatrix} \tag{7.3}$$

where the columns of A represent the "interesting" reactions after splitting reversible reactions and removing the blocked reactions. In the following, we assume that A (resp. B) has p (resp. q) columns.

Given S', the steady-state flux cone in canonical form will look as follows

$$C = \{(x,y) \in \mathbb{R}^{p+q} \mid G\cdot x + H\cdot y \leq 0\}, \tag{7.4}$$

where the matrices G (resp. H) represent the columns to be kept (resp. eliminated):

$$G = \begin{pmatrix} -A \\ A \\ -I_p \\ 0_{q,p} \end{pmatrix}, \quad H = \begin{pmatrix} -B \\ B \\ 0_{p,q} \\ -I_q \end{pmatrix} \tag{7.5}$$

Here I_p denotes the $p\times p$ identity matrix, and $0_{p,q}$ the $p\times q$ zero matrix.

Step 2 - Cone Projection: In this step, the flux cone is projected, eliminating the reactions corresponding to columns in H. Several methods have been proposed in the literature for the projection of polyhedra [48]. For our purpose we chose the *block elimination method* [4]. This method allows us to find an inequality description of the projected cone by enumerating the extreme rays of an intermediary cone called the *projection cone*. In our case, the projection cone is defined as

$$W = \{w \in \mathbb{R}^{2m+p+q} \mid H^T \cdot w = 0, w \geq 0\}, \tag{7.6}$$

where H^T denotes the transpose of H.

We enumerate the extreme rays $\{r^1, r^2, ..., r^k\}$ of W using the double description method [33]. The projected cone is given by

$$\mathcal{P}_{\mathbb{X}}(C) = \{x \in \mathbb{R}^p \mid R \cdot G \cdot x \leq 0\}, \tag{7.7}$$

where

$$R = (r^1 \ \ldots \ r^k)^T. \tag{7.8}$$

This representation of the projected cone contains as many inequalities as there are extreme rays in W, thus a large number of them might be redundant [48]. These redundant inequalities are removed next (see below).

Step 3 - Finding ProCEMs: In the final step, the extreme rays of the projected cone, i.e., the ProCEMs, are enumerated. Similarly as in Step 2, the double description method is employed to enumerate the extreme rays of $\mathcal{P}_{\mathbb{X}}(C)$.

With the block elimination algorithm, it is also possible to perform the projection in an iterative manner. This means that rather than eliminating all the "uninteresting" reactions in one step, we can partition these in t subsets and then iteratively execute Step 2, eliminating every subset of reactions one by one. By proceeding in this fashion, the intermediary projection cones, $W^1, W^2, ..., W^t$ get typically smaller, thus enumerating their extreme rays requires less memory. On the other side, the more sets we partition into, the slower the projection algorithm usually gets.

7.4.2 Implementation and computational experiments

The ProCEM enumeration algorithm has been implemented in Matlab [74]. In our implementation, `polco` tool v4.7.1 [118, 119] is used for the enumeration of extreme rays (both in Step 2 and 3). For removing redundant inequalities in Step 2, the `redund` method from the `lrslib` package v4.2 is applied [3]. All computations were performed on a 64-bit Debian Linux system with Intel Core 2 Duo 3.0 GHz processor. A prototype implementation is available on request from the authors.

7.4.3 Dataset

The metabolic network model of red blood cell (RBC) [134] was used in this study. The network was taken from the example metabolic networks associated with CellNetAnalyzer [59] and differs slightly from the original model. Additionally, we studied the plastid metabolic network of *Arabidopsis thaliana* (see Additional file 1 in [73]). This network was extracted from the genome-scale metabolic network of *A. thaliana* [27] by focusing on those metabolites which appeared in the plastid compartment. Then, the subsystem of "sugar and starch metabolism" was selected as the interesting subnetwork of the plastid metabolic network.

7.5 Results and discussion

7.5.1 Mathematical relationships among PEMs, EFPs and ProCEMs

From Table 7.1, one can observe that the set of ProCEMs in Fig. 7.1 is included in the set of PEMs. Additionally, the set of EFPs is included in the set of ProCEM supports. Here, we prove that these two properties are true in general. This means that the analysis of ProCEMs has at least two advantages compared to the analysis of EFPs. Firstly, ProCEMs can tell us about the flux ratio of different reactions in an elementary mode, while EFPs can only tell us whether the reaction has a non-zero value in that mode. Secondly, enumeration of ProCEMs may result in modes which cannot be obtained by EFP analysis.

Theorem 7.4. *In a metabolic network $\mathcal{N}$ with irreversible reactions only, let J (resp. P) be the set of ProCEMs (resp. PEMs) for a given set of "interesting" reactions. Then $J \subseteq P$.*

Proof. We have to show that for every $u \in J$ there exists an elementary mode $e \in C$ in $\mathcal{N}$, such that $\mathcal{P}_{\mathbb{X}}(e) \cong u$.

We know that for any $u \in J$, there exists $v \in C$, such that $\mathcal{P}_{\mathbb{X}}(v) = u$.

Any $v \in C$ can be written in the form $v = \sum_{k=1}^{r} c_k \cdot e^k$, where $e^1, \ldots, e^r$ are some elementary modes of $\mathcal{N}$ and $c_1, \ldots, c_r > 0$. Thus, $u = \mathcal{P}_{\mathbb{X}}(v) = \sum_{k=1}^{r} c_k \cdot \mathcal{P}_{\mathbb{X}}(e^k)$.

If all the vectors $\mathcal{P}_{\mathbb{X}}(e^k)$ are pairwise equivalent, u is a PEM. Otherwise, u is a linear combination of at least two non-equivalent PEMs, which are vectors in $\mathcal{P}_{\mathbb{X}}(C)$. Thus, u is not an extreme ray of $\mathcal{P}_{\mathbb{X}}(C)$, in contradiction with Lemma 1 in [35] stating that in a metabolic network with irreversible reactions only, the EMs are exactly the extreme rays. □

Theorem 7.5. *In a metabolic network $\mathcal{N}$ with irreversible reactions only, let E (resp. J) be the set of EFPs (resp. ProCEMs) for a given set of interesting reactions. Then, $E \subseteq \{supp(u) \mid u \in J\}$.*

Proof. Assume by contradiction, that for some $f \in E$, there exists no $v \in J$ such that $f = supp(v)$.

Since f is an EFP, then there exists $p \in P$ (the set of PEMs), such that $f = supp(p)$. Thus, we have $p \notin J$, but $p \in \mathcal{P}_{\mathbb{X}}(C)$. There exist $r \geq 2$ different ProCEMs, say $u^1, \ldots, u^r \in J$, such that $p = \sum_{k=1}^{r} c_k \cdot u^k$, with $c_k > 0$ for all k.

Since $u^k \geq 0$, for all k, we have $supp(p) = \bigcup_{k=1}^{r} supp(u^k)$, with $supp(u^k) \neq supp(p)$ for all k. Since $supp(u^k)$ is a flux pattern for all k, this is a contradiction with f being an EFP. Hence, the statement follows. □

7.5.2 ProCEMs inherit existing coupling relations

In Subsection 7.3.2 we have questioned whether studying isolated subsystems produces relevant results, as it does not preserve the coupling relations from the original network. In the following, we prove that projection, and in specific the set of ProCEMs, preserves the already existing relations, while possibly introducing new ones.

Proposition 7.6. *Given a metabolic network $\mathcal{N} = (S, Irr)$ with an associated steady-state flux cone C, let i and j be two reactions present both in the original network and the subnetwork of interest. Then $i \longrightarrow j$ (resp.$i \longleftrightarrow j$ or $i \Longleftrightarrow j$) in C implies, $i \longrightarrow j$ (resp.$i \longleftrightarrow j$ or $i \Longleftrightarrow j$) in $\mathcal{P}_{\mathbb{X}}(C)$.*

Proof. We will prove the statement for directional coupling. The other two cases follow similarly.

Assume by contradiction that $i \not\longrightarrow j$ in $\mathcal{P}_{\mathbb{X}}(C)$. Then there exists $x \in \mathcal{P}_{\mathbb{X}}(C)$ with $x_i \neq 0$ and $x_j = 0$. Thus, there also exists $v := (x, y) \in C$ with $v_i \neq 0$ and $v_j = 0$. This is a contradiction to $i \longrightarrow j$ in C.

□

The reverse of the statements is not true. Since ProCEMs represent a compressed set of descriptors for the steady-state flux cone, new coupling relations can appear. To exemplify this behavior, we refer the reader to Fig. 7.2. Here, reactions r_4 and r_6 are directionally coupled, although in the originating network (Fig. 7.1) the same two reactions were uncoupled.

7.5.3 Computing the set of EFPs from the set of ProCEMs

Here, we present a simple algorithm to show that it is possible to compute the set of EFPs when the set of ProCEMs is known. Algorithm *computeEFP* summarizes this procedure.

Algorithm computeEFP

Input

J - the complete set of ProCEMs

Output

E - the complete set of EFPs

```
1.  E := ∅
2.  for each u ∈ J do
3.      Z := supp(u)
4.      for each v ∈ J do
5.          if supp(v) ⊊ supp(u) then
6.              Z := Z − supp(v)
7.          end if
8.      end for
9.      if Z ≠ ∅ then
10.         E := E ∪ {supp(u)}
11.     end if
12. end for
```

We know that the support of every ProCEM u is a flux pattern Z. Moreover, Theorem 7.5 asserts that every EFP has a corresponding ProCEM. Hence, in the procedure, we only need to check which flux patterns are elementary. If Z is *not* elementary, then it is equal to the union of some other flux patterns. Therefore, to check the elementarity of a flux pattern $supp(u)$, one can iteratively subtract from it all other flux patterns, which are subsets of $supp(u)$. Once all flux patterns have been subtracted (excluding itself), the result becomes the empty-set if and only if $supp(u)$ was an EFP. This algorithm has the complexity $\mathcal{O}(nq^2)$, where q is the number of ProCEMs and n is the number of reactions.

7.5.4 Comparing EFPs and ProCEMs

Analysis of subnetworks in the metabolic network of red blood cell

In order to compare our approach (computation of ProCEMs) with the enumeration of EFPs, we tested these methods for analysing subnetworks of the RBC model [134]. Again, we split every reversible reaction into one forward and one backward irreversible reaction. The resulting network contains 67

reactions, including 20 boundary reactions, and a total number of 811 EMs. For comparing the methods, the set of all boundary reactions was considered as the interesting subsystem.

- Computing PEMs from EMs: A total set of 502 PEMs was computed.
- Computing ProCEMs: Using our method 252 ProCEMs were computed.
- Computing EFPs: Using both `EFPTools` [49] and *computeEFP*, a complete set of 90 EFPs is determined.

The above results imply that the ProCEMs cover more than half of the PEMs, while the EFPs cover less than one fifth of the PEMs. This provides empirical evidence for the relevance of using ProCEMs for the analysis of subnetworks.

In order to compare the computational runtime of EFPs and ProCEMs, as well as the number of elements in both sets, the following task was performed on the RBC model [134]. In each iteration, a random subnetwork containing r reactions was selected, for which the EFPs and ProCEMs were computed. The task was repeated for different subnetwork sizes. The computational results can be found in Fig. 7.4.

From Fig. 7.4, it results that EFP computation is faster than ProCEM computation for small subnetworks. However, when the subnetwork size r increases, computation of ProCEMs does not become slower, while computation of EFPs significantly slows down. This is an important observation, because the difference between the number of EFPs and ProCEMs also increases with r.

Analysis of subnetworks in the plastid metabolic network of *Arabidopsis thaliana*

ProCEM analysis becomes important when PEMs cannot be computed. This may happen frequently in the analysis of large-scale metabolic networks, as memory consumption is a major challenge in computation of EMs [118]. In such cases, cone projection might still be feasible.

As an example, the metabolic network of *A. thaliana* plastid was studied (Additional file 1 in [73]). This network contains 102 metabolites and 123 reactions (205 reactions after splitting reversible reactions). The reactions we were interested in are the 57 reactions involved in sugar and starch metabolism (see Additional file 1 in [73]). Applying the presented methods, gave the following results:

- Computing PEMs from EMs: Using `efmtool` (and also `polco`) [118], the computation of EMs was not possible due to running out of memory, even with 2 GB of allocated memory. Therefore, for no subnetwork of the plastid network, PEMs could be computed.
- Computing ProCEMs: We have computed the ProCEMs as described in the Method and Implementation section, using a projection step size of 5 reactions. The complete set of 1310 ProCEMs was computed in approximately 15 minutes.
- Computing EFPs using `EFPTools` [50, 49]: The tool iteratively found 279 EFPs, after 4 days of running time, after which it was manually stopped. Interestingly, already 270 EFPs were computed in the first two days, proving the significant slowdown effect the program displayed.
- Computing EFPs using *computeEFP*: The complete set of 1054 EFPs was obtained in 30 seconds.

We conclude that in metabolic networks for which the set of EMs cannot be enumerated, ProCEMs prove to be a useful concept to get insight into reaction activities.

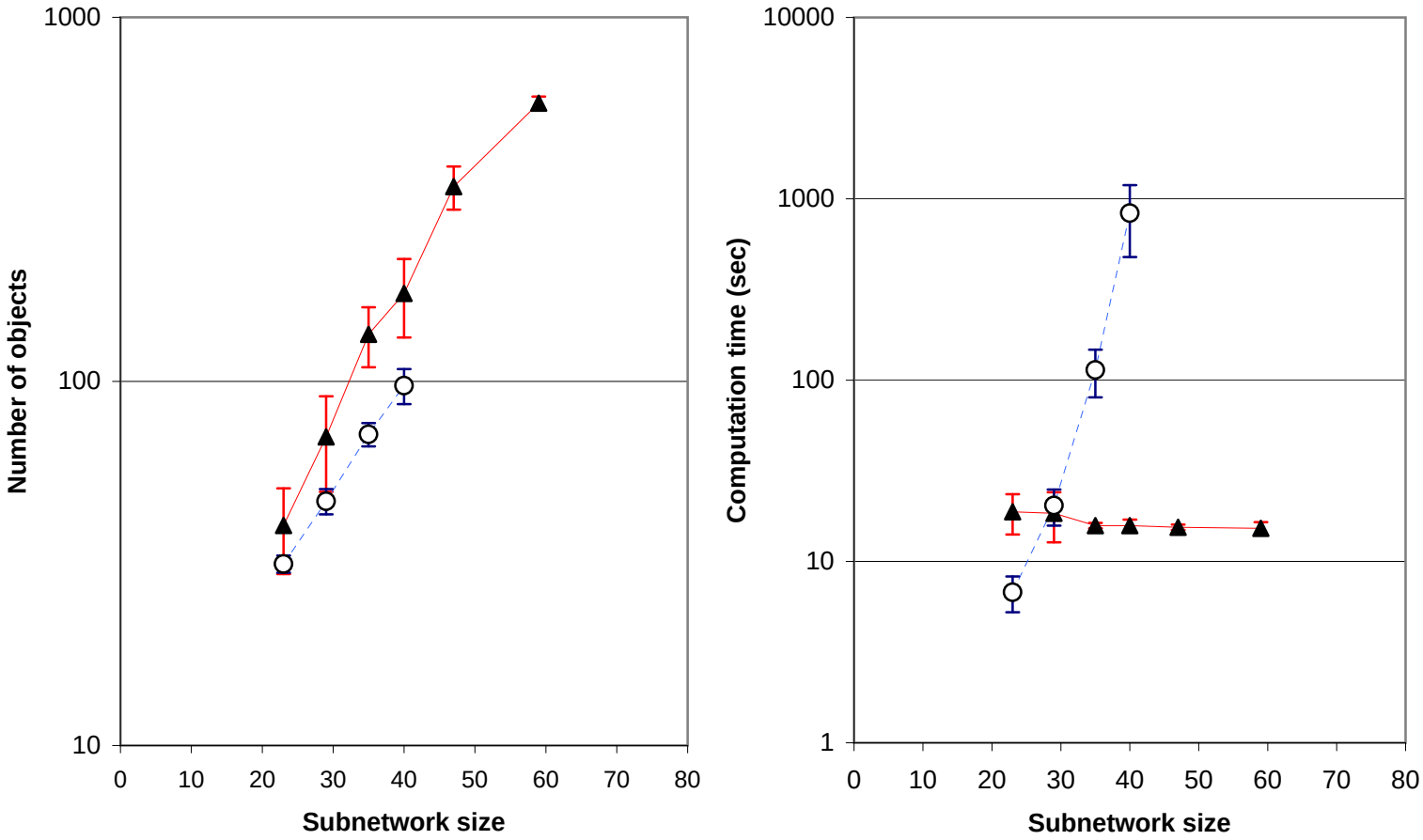


Figure 7.4: ProCEM vs. EFP computation Left: Number of ProCEMs and EFPs computed for random subnetworks of different sizes. Right: The computation times (per second) required for computing the ProCEMs and EFPs in the left chart. ▲: ProCEMs; ○: EFPs. Confidence intervals in this plot are based on one-sample t-test (95% c.i.). For large subnetworks ($r > 40$), we could not compute the EFPs because the program was very slow.

Summary of the chapter

- For studying the behaviour of a subnetwork or a subset of reactions of interest, we introduce the concept of ProCEMs.
- We prove that the set of ProCEMs is "inbetween" the set of PEMs and the set of EFPs.
- A possible implementation to compute ProCEMs is outlined.
- We show on real metabolic networks that in some cases it is possible to compute all ProCEMs, where enumerating the complete set of PEMs or EFPs is not practical.

Bibliography

[1] V. Acuña, F. Chierichetti, V. Lacroix, A. Marchetti-Spaccamela, M. Sagot, and L. Stougie. Modes and cuts in metabolic networks: Complexity and algorithms. *BioSystems*, 95:51–60, 2009. (Cited on pages 14, 79, 80 and 81.)

[2] V. Acuña, A. Marchetti-Spaccamela, M. Sagot, and L. Stougie. A note on the complexity of finding and enumerating elementary modes. *BioSystems*, 99:210–214, 2010. (Cited on pages 16, 80 and 89.)

[3] D. Avis. *lrs*: A revised implementation of the reverse search vertex enumeration algorithm. In G. Kalai and G. Ziegler, editors, *Polytopes - Combinatorics and Computation, Oberwolfach Seminars*, pages 177–198. Birkhäuser-Verlag, 2000. (Cited on page 130.)

[4] E. Balas and W. Pulleyblank. The perfectly matchable subgraph polytope of a bipartite graph. *Networks*, 13:495–516, 1983. (Cited on pages 129 and 130.)

[5] V. Baldazzi, D. Ropers, J. Geiselmann, D. Kahn, and H. de Jong. Importance of metabolic coupling for the dynamics of gene expression following a diauxic shift in escherichia coli. *Journal of Theoretical Biology*, 295:100–115, Feb. 2012. (Cited on page 63.)

[6] D. A. Beard and H. Qian. *Constraint-based modeling of metabolomic systems. Encyclopedia of Genetics, Genomics, Proteomics and Bioinformatics.* 2005. (Cited on page 9.)

[7] S. A. Becker, N. D. Price, and B. O. Palsson. Metabolite coupling in genome-scale metabolic networks. *BMC Bioinformatics*, 7:111, 2006. (Cited on page 64.)

[8] S. L. Bell and B. O. Palsson. Expa: a program for calculating extreme pathways in biochemical reaction networks. *Bioinformatics*, 21:1739–1740, 2005. (Cited on page 121.)

[9] A. Bordbar, J. M. Monk, Z. A. King, and B. O. Palsson. Constraint-based models predict metabolic and associated cellular functions. *Nature Reviews Genetics*, 15(2):107–120, 2014. (Cited on page 9.)

[10] A. Bordbar, J. M. Monk, Z. A. King, and B. O. Palsson. Constraint-based models predict metabolic and associated cellular functions. *Nature Reviews Genetics*, 15(2):107–120, 2014. (Cited on page 17.)

[11] J. Bundy, B. Papp, R. Harmston, R. Browne, E. Clayson, N. Burton, R. Reece, S. Oliver, and K. Brindle. Evaluation of predicted network modules in yeast metabolism using nmr-based metabolite profiling. *Genome Res*, 2007. (Cited on pages 16, 22 and 23.)

[12] A. P. Burgard, E. V. Nikolaev, C. H. Schilling, and C. D. Maranas. Flux coupling analysis of genome-scale metabolic network reconstructions. *Genome Res.*, 14(2):301–312, 2004. (Cited on pages 15, 21, 22, 23, 49, 125, 126 and 129.)

[13] P. Carbonell, D. Fichera, S. Pandit, and J. L. Faulon. Enumerating metabolic pathways for the production of heterologous target chemicals in chassis organisms. *BMC Systems Biology*, 6(1):10+, 2012. (Cited on pages 100 and 101.)

[14] R. Caspi, T. Altman, J. M. Dale, K. Dreher, C. A. Fulcher, F. Gilham, P. Kaipa, A. S. Karthikeyan, A. Kothari, M. Krummenacker, et al. The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic acids research*, 38(suppl 1):D473–D479, 2010. (Cited on page 10.)

[15] T. Çakır, B. Kırdar, and K. O. Ülgen. Metabolic pathway analysis of yeast strengthens the bridge between transcriptomics and metabolic networks. *Biotechnology and Bioengineering*, 86:251–260, 2004. (Cited on pages 14, 125 and 126.)

[16] A. Cho, H. Yun, J. Park, S. Lee, and S. Park. Prediction of novel synthetic pathways for the production of desired chemicals. *BMC Systems Biology*, 4(1):35+, 2010. (Cited on page 100.)

[17] B. L. Clarke. Stoichiometric network analysis. *Cell biophysics*, 12(1):237–253, 1988. (Cited on page 14.)

[18] COIN-OR. Computational Infrastructure for Operations Research. `http://www.coin-or.org/`. (Cited on pages 32, 40, 60 and 74.)

[19] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition*. The MIT Press, 2nd edition, Sept. 2001. (Cited on pages 106 and 112.)

[20] M. W. Covert, I. Famili, and B. O. Palsson. Identifying constraints that govern cell behavior: a key to converting conceptual to computational models in biology? *Biotechnology and bioengineering*, 84(7):763–772, 2003. (Cited on page 9.)

[21] M. W. Covert and B. O. Palsson. Constraints-based models: regulation of gene expression reduces the steady-state solution space. *Journal of Theoretical Biology*, 221:309–325, 2003. (Cited on pages 124 and 128.)

[22] L. David and A. Bockmayr. Constrained flux coupling analysis. In *Proceedings of WCB13 Workshop on Constraint Based Methods for Bioinformatics*, page 75. (Cited on page 47.)

[23] L. David and A. Bockmayr. Computing elementary flux modes involving a set of target reactions. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 99(PrePrints):1, 2014. (Cited on pages 18 and 79.)

[24] L. David, S.-A. Marashi, A. Larhlimi, B. Mieth, and A. Bockmayr. FFCA: a feasibility-based method for flux coupling analysis of metabolic networks. *BMC bioinformatics*, 12(1):236, Jan. 2011. (Cited on pages 12, 16, 17, 21, 22, 23, 25, 26, 34, 41, 50 and 89.)

[25] L. F. de Figueiredo, A. Podhorski, A. Rubio, C. Kaleta, J. E. Beasley, S. Schuster, and F. J. Planes. Computing the shortest elementary flux modes in genome-scale metabolic networks. *Bioinformatics*, 25(23):3158–3165, 2009. (Cited on pages 14, 79, 80, 82, 92, 93 and 94.)

[26] L. F. de Figueiredo, S. Schuster, C. Kaleta, and D. A. Fell. Can sugars be produced from fatty acids? a test case for pathway analysis tools. *Bioinformatics*, 24(22):2615–2621, 2008. (Cited on page 91.)

[27] C. G. de Oliveira Dal'Molin, L.-E. Quek, R. W. Palfreyman, S. M. Brumbley, and L. K. Nielsen. AraGEM, a genome-scale reconstruction of the primary metabolic network in Arabidopsis. *Plant Physiology*, 152:579–589, 2010. (Cited on page 131.)

[28] N. C. Duarte, S. A. Becker, N. Jamshidi, I. Thiele, M. L. Mo, T. D. Vo, R. Srivas, and B. O. Palsson. Global reconstruction of the human metabolic network based on genomic and bibliomic data. *Proceedings of the National Academy of Sciences*, 104(6):1777–1782, 2007. (Cited on pages 10, 31, 41 and 91.)

[29] N. C. Duarte, M. J. Herrgård, and B. Ø. Palsson. Reconstruction and validation of saccharomyces cerevisiae iND750, a fully compartmentalized genome-scale metabolic model. *Genome research*, 14(7):1298–1309, July 2004. (Cited on pages 31, 40, 74, 93 and 94.)

[30] A. M. Feist, C. S. Henry, J. L. Reed, M. Krummenacker, A. R. Joyce, P. D. Karp, L. J. Broadbelt, V. Hatzimanikatis, and B. Ø. Palsson. A genome-scale metabolic reconstruction for Escherichia coli K-12

MG1655 that accounts for 1260 ORFs and thermodynamic information. *Mol. Syst. Biol.*, 3(121), JUN 2007. (Cited on pages 31, 41 and 92.)

[31] A. M. Feist, J. C. M. Scholten, B. O. Palsson, F. J. Brockman, and T. Ideker. Modeling methanogenesis with a genome-scale metabolic reconstruction of methanosarcina barkeri. *Mol Syst Biol*, 2(1744-4292 (Electronic)), 2006. (Cited on page 40.)

[32] D. A. Fell and J. R. Small. Fat synthesis in adipose tissue. an examination of stoichiometric constraints. *Biochem. J*, 238:781–786, 1986. (Cited on page 17.)

[33] K. Fukuda and A. Prodon. Double description method revisited. In *Combinatorics and Computer Science: 8th Franco-Japanese and 4th Franco-Chinese Conference. Brest, France*, volume 1120 of *Lecture Notes in Computer Science*, pages 91–111, 1996. (Cited on pages 14, 15 and 130.)

[34] J. Gagneur and S. Klamt. Computation of elementary modes: a unifying framework and the new binary approach. *BMC Bioinformatics*, 5:175, 2004. (Cited on pages 14, 79, 82 and 96.)

[35] J. Gagneur and S. Klamt. Computation of elementary modes: a unifying framework and the new binary approach. *BMC Bioinformatics*, 5:175, 2004. (Cited on pages 34, 121, 126 and 131.)

[36] M. R. Garey and D. S. Johnson. A guide to the theory of np-completeness. *San Francisco*, 1979. (Cited on page 5.)

[37] M. A. Gibson. *Computational methods for stochastic biological systems.* PhD thesis, California Institute of Technology, 2000. (Cited on page 9.)

[38] C. Gille, C. Bolling, A. Hoppe, S. Bulik, S. Hoffmann, K. Hubner, A. Karlstadt, R. Ganeshan, M. Konig, K. Rother, M. Weidlich, J. Behre, and H.-G. Holzhutter. HepatoNet1: a comprehensive metabolic reconstruction of the human hepatocyte for the analysis of liver physiology. *Molecular Systems Biology*, 6(1), Sept. 2010. (Cited on pages 31 and 74.)

[39] Y. Goldstein. *Zur mathematischen Theorie elementarer Flussmoden: Diplomarbeit.* PhD thesis, Freie Universitaet Berlin, 2009. (Cited on page 13.)

[40] Y. A. Goldstein and A. Bockmayr. A lattice-theoretic framework for metabolic pathway analysis. In *Computational Methods in Systems Biology*, pages 178–191. Springer, 2013. (Cited on page 47.)

[41] I. Gurobi Optimization. Gurobi optimizer reference manual, 2013. (Cited on pages 5, 27 and 74.)

[42] O. Hädicke and S. Klamt. Computing complex metabolic intervention strategies using constrained minimal cut sets. *Metabolic Engineering*, 13(2):204 – 213, 2011. (Cited on pages 17 and 92.)

[43] S. S. I. Zevedei-Oancea. Topological analysis of metabolic networks based on petri net theory. *In Silico Biology*, 3(3):323–345, 2003. (Cited on page 64.)

[44] M. Imielinski and C. Belta. Exploiting the pathway structure of metabolism to reveal high-order epistasis. *BMC Systems Biology*, 2:40, 2008. (Cited on page 125.)

[45] N. Jamshidi and B. Palsson. Investigating the metabolic capabilities of mycobacterium tuberculosis H37Rv using the in silico strain iNJ661 and proposing alternative drug targets. *BMC systems biology*, 1(1):26+, 2007. (Cited on page 41.)

[46] D. Jevremovic, C. T. Trinh, F. Srienc, and D. Boley. On algebraic properties of extreme pathways in metabolic networks. *Journal of Computational Biology*, 17:107–119, 2010. (Cited on page 121.)

[47] M. John, M. Nebut, J. Niehren, L. (cnrs Umr, and I. Lille. Knockout prediction for reaction networks with partial kinetic information, 2012. (Cited on page 9.)

[48] C. N. Jones, E. C. Kerrigan, and J. M. Maciejowski. On polyhedral projection and parametric programming. *Journal of Optimization Theory and Applications*, 138:207–220, 2008. (Cited on page 130.)

[49] C. Kaleta. EFPTools, for computing elementary flux patterns (EFPs). `http://users.minet.uni-jena.de/~m3kach/EFPA/`, 2009. (Cited on pages 126, 134 and 135.)

[50] C. Kaleta, L. F. de Figueiredo, and S. Schuster. Can the whole be less than the sum of its parts? Pathway analysis in genome-scale metabolic networks using elementary flux patterns. *Genome Research*, 19:1872–1883, 2009. (Cited on pages 16, 17, 24, 80, 125, 126 and 135.)

[51] C. Kaleta, L. F. de Figueiredo, S. Werner, R. Guthke, M. Ristow, and S. Schuster. In silico evidence for gluconeogenesis from fatty acids in humans. *PLoS Computational Biology*, 7(7), 2011. (Cited on page 91.)

[52] A. v. Kamp and S. Schuster. Metatool 5.0: fast and flexible elementary modes analysis. *Bioinformatics*, 22(15):1930–1931, 2006. (Cited on pages 90 and 103.)

[53] S. M. Kelk, B. G. Olivier, L. Stougie, and F. J. Bruggeman. Optimal flux spaces of genome-scale stoichiometric models are determined by a few subnetworks. *Scientific reports*, 2:580, 2012. (Cited on page 47.)

[54] D. Kenanov, C. Kaleta, A. Petzold, C. Hoischen, S. Diekmann, R. A. Siddiqui, and S. Schuster. Theoretical study of lipid biosynthesis in wild-type *Escherichia coli* and in a protoplast-type L-form using elementary flux mode analysis. *FEBS Journal*, 277:1023–1034, 2010. (Cited on pages 14 and 125.)

[55] J. I. Kim, J. D. Varner, and D. Ramkrishna. A hybrid model of anaerobic *E. coli* GJT001: combination of elementary flux modes and cybernetic variables. *Biotechnology Progress*, 24:993–1006, 2008. (Cited on pages 14 and 125.)

[56] T. Y. Kim, S. B. Sohn, Y. B. Kim, W. J. Kim, and S. Y. Lee. Recent advances in reconstruction and applications of genome-scale metabolic models. *Current opinion in biotechnology*, 23(4):617–623, 2012. (Cited on page 17.)

[57] S. Klamt, J. Gagneur, and A. von Kamp. Algorithmic approaches for computing elementary modes in large biochemical reaction networks. *Systems biology*, 152(4):249–255, Dec. 2005. (Cited on page 121.)

[58] S. Klamt and E. D. Gilles. Minimal cut sets in biochemical reaction networks. *Bioinformatics*, 20(2):226–234, 2004. (Cited on page 17.)

[59] S. Klamt, J. Saez-Rodriguez, and E. D. Gilles. Structural and functional analysis of cellular networks with CellNetAnalyzer. *BMC Systems Biology*, 1:2, 2007. (Cited on page 131.)

[60] S. Klamt and J. Stelling. Combinatorial complexity of pathway analysis in metabolic networks. *Molecular Biology Reports*, 29:233–236, 2002. (Cited on page 122.)

[61] S. Klamt and J. Stelling. Two approaches for metabolic pathway analysis? *Trends in Biotechnology*, 21:64–69, 2003. (Cited on pages 11 and 121.)

[62] S. Klamt, J. Stelling, M. Ginkel, and E. D. Gilles. FluxAnalyzer: exploring structure, pathways, and flux distributions in metabolic networks on interactive flux maps. *Bioinformatics*, 19(2):261–269, Jan. 2003. (Cited on page 34.)

[63] A. Larhlimi. New concepts and tools in constraint-based analysis of metabolic networks. Doctoral thesis, Freie Universität Berlin, 2008. (Cited on pages 14, 15, 26, 27, 28 and 30.)

[64] A. Larhlimi and A. Bockmayr. A new approach to flux coupling analysis of metabolic networks. In *Proceedings of the Second international conference on Computational Life Sciences*, CompLife'06, pages 205–215, Berlin, Heidelberg, 2006. Springer-Verlag. (Cited on pages 12, 25, 26, 31, 37 and 74.)

[65] A. Larhlimi and A. Bockmayr. A new constraint-based description of the steady-state flux cone of metabolic networks. *Discrete Applied Mathematics*, 157(10):2257–2266, May 2009. (Cited on pages 14, 15 and 26.)

[66] A. Larhlimi, L. David, J. Selbig, and A. Bockmayr. F2C2: a fast tool for the computation of flux coupling in genome-scale metabolic networks. *BMC bioinformatics*, 13(1):57, Jan. 2012. (Cited on pages 17, 21, 50, 58, 59, 88 and 89.)

[67] F. W. Levi. *Finite geometrical systems: six public lectues delivered in February, 1940, at the University of Calcutta*. 1942. (Cited on pages 104 and 105.)

[68] N. E. Lewis, H. Nagarajan, and B. O. Palsson. Constraining the metabolic genotype–phenotype relationship using a phylogeny of in silico methods. *Nature Reviews Microbiology*, 10(4):291–305, 2012. (Cited on page 17.)

[69] H. Lodhi and S. Muggleton. Modelling metabolic pathways using stochastic logic programs-based ensemble methods. In *Proceedings of the 20 International Conference on Computational Methods in Systems Biology*, CMSB'04, pages 119–133, Berlin, Heidelberg, 2005. Springer-Verlag. (Cited on page 9.)

[70] J. Löfberg. Mexclp. `http://control.ee.ethz.ch/~johanl/clp.php`. (Cited on page 40.)

[71] R. Mahadevan and C. Schilling. The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metabolic engineering*, 5(4):264–276, 2003. (Cited on page 17.)

[72] S.-A. Marashi and A. Bockmayr. Flux coupling analysis of metabolic networks is sensitive to missing reactions. *Biosystems*, 103(1):57 – 66, 2011. (Cited on pages 15, 82, 89 and 126.)

[73] S.-A. Marashi, L. David, and A. Bockmayr. Analysis of metabolic subnetworks by flux cone projection. *Algorithms for molecular biology : AMB*, 7(1):17, Jan. 2012. (Cited on pages 14, 80, 121, 131 and 134.)

[74] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010. (Cited on pages 30, 40, 60, 74, 114 and 130.)

[75] A. Montagud, A. Zelezniak, E. Navarro, P. F. F. de Córdoba, J. F. Urchueguía, and K. R. R. Patil. Flux coupling and transcriptional regulation within the metabolic network of the photosynthetic bacterium synechocystis sp. PCC6803. *Biotechnology journal*, 6(3):330–342, Mar. 2011. (Cited on pages 16, 22 and 23.)

[76] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, Inc., New York, 1988. (Cited on page 129.)

[77] E. V. Nikolaev, A. P. Burgard, and C. D. Maranas. Elucidation and structural analysis of conserved pools for genome-scale metabolic reconstructions. *Biophysical journal*, 88(1):37–49, Jan. 2005. (Cited on page 64.)

[78] R. Notebaart, P. Kensche, M. Huynen, and B. Dutilh. Asymmetric relationships between proteins shape genome evolution. *Genome Biology*, 10(2), 2009. (Cited on pages 16, 21, 22 and 23.)

[79] R. A. Notebaart, B. Teusink, R. J. Siezen, and B. Papp. Co-Regulation of metabolic genes is better explained by flux coupling than by network distance. *PLoS Comput Biol*, 4(1):e26+, Jan. 2008. (Cited on pages 16, 22 and 23.)

[80] J. C. Nuño, I. Sánchez-Valdenebro, C. Pérez-Iratxeta, E. Meléndez-Hevia, and F. Montero. Network organization of cell metabolism: monosaccharide interconversion. *Biochemical Journal*, 324:103–111, 1997. (Cited on pages 14 and 125.)

[81] J. D. Orth, T. M. Conrad, J. Na, J. A. Lerman, H. Nam, A. M. Feist, and B. O. Palsson. A comprehensive genome-scale reconstruction of escherichia coli metabolism-2011. *Molecular Systems Biology*, 7:53+, Oct. 2011. (Cited on page 41.)

[82] C. Pal, B. Papp, and M. J. Lercher. Adaptive evolution of bacterial metabolic networks by horizontal gene transfer. *Nature Genetics*, 37(12):1372–1375, Nov. 2005. (Cited on pages 16, 21 and 23.)

[83] B. O. Palsson. *Systems Biology: Properties of Reconstructed Networks*. Cambridge University Press, New York, 2006. (Cited on pages 31, 60, 74, 94 and 114.)

[84] A. J. Papin, J. Stelling, N. D. Price, S. Klamt, S. Schuster, and B. O. Palsson. Comparison of network-based pathway analysis methods. *TRENDS in Biotechnology*, 22(8):400–405, 2004. (Cited on pages 12 and 79.)

[85] J. L. Peterson. Petri nets. *ACM Computing Surveys*, 9:223–252, 1977. (Cited on page 105.)

[86] J. Pey and F. J. Planes. Direct calculation of elementary flux modes satisfying several biological constraints in genome-scale metabolic networks. *Bioinformatics*, page btu193, 2014. (Cited on page 79.)

[87] T. Pfeiffer, I. Sánchez-Valdenebro, J. C. Nuño, F. Montero, and S. Schuster. METATOOL: for studying metabolic networks. *Bioinformatics*, 15:251–257, 1999. (Cited on pages 9, 29, 34, 49, 75 and 121.)

[88] N. D. Price, J. L. Reed, and B. Ø. Palsson. Genome-scale models of microbial cells: evaluating the consequences of constraints. *Nature Reviews Microbiology*, 2(11):886–897, 2004. (Cited on page 9.)

[89] N. D. Price, J. Schellenberger, and B. O. Palsson. Uniform sampling of steady-state flux spaces: means to design experiments and to interpret enzymopathies. *Biophysical journal*, 87(4):2172–2186, 2004. (Cited on pages 10 and 17.)

[90] J. L. Reed, T. D. Vo, C. H. Schilling, and B. O. Palsson. An expanded genome-scale model of *Escherichia coli* k-12 (ijr904 gsm/gpr). *Genome Biology*, 4(9):R54+, 2003. (Cited on pages 31, 40, 74 and 114.)

[91] A. Rezola, L. F. de Figueiredo, M. Brock, J. Pey, A. Podhorski, C. Wittmann, S. Schuster, A. Bockmayr, and F. J. Planes. Exploring metabolic pathways in genome-scale networks via generating flux modes. *Bioinformatics*, 27(4):534–540, 2011. (Cited on pages 14, 79, 80 and 92.)

[92] A. Rezola, J. Pey, L. F. de Figueiredo, A. Podhorski, S. Schuster, A. Rubio, and F. J. Planes. Selection of human tissue-specific elementary flux modes using gene expression data. *Bioinformatics*, 29(16):2009–16, 2013. (Cited on pages 14, 79 and 80.)

[93] P. R. Romero and P. Karp. Nutrient-related analysis of pathway/genome databases. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 471–482, 2001. (Cited on page 100.)

[94] J. Schellenberger, J. O. Park, T. M. Conrad, and B. Ø. Palsson. Bigg: a biochemical genetic and genomic knowledgebase of large scale metabolic reconstructions. *BMC bioinformatics*, 11(1):213, 2010. (Cited on page 10.)

[95] C. H. Schilling, M. W. Covert, I. Famili, G. M. Church, J. S. Edwards, and B. O. Palsson. Genome-scale metabolic model of *Helicobacter pylori*

26695. *Journal of Bacteriology*, 184:4582–4593, 2002. (Cited on pages 14 and 125.)

[96] C. H. Schilling, D. Letscher, and B. O. Palsson. Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *Journal of Theoretical Biology*, 203:229–248, 2000. (Cited on pages 14 and 121.)

[97] C. H. Schilling and B. O. Palsson. Assessment of the metabolic capabilities of *Haemophilus influenzae* Rd through a genome-scale pathway analysis. *Journal of Theoretical Biology*, 203:249–283, 2000. (Cited on pages 14 and 125.)

[98] C. H. Schilling, S. Schuster, B. O. Palsson, and R. Heinrich. Metabolic pathway analysis: Basic concepts and scientific applications in the post-genomic era. *Biotechnology Progress*, 15:296–303, 1999. (Cited on page 122.)

[99] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Chichester, 1986. (Cited on page 27.)

[100] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1998. (Cited on pages 5 and 86.)

[101] S. Schuster, T. Dandekar, and A. Fell. Detection of elementary flux modes in biochemical networks : a promising tool for pathway analysis and metabolic engineering. *Trends Biotechnol.*, 17:53–60, 1999. (Cited on pages 12 and 79.)

[102] S. Schuster, T. Dandekar, and D. A. Fell. Detection of elementary flux modes in biochemical networks: A promising tool for pathway analysis and metabolic engineering. *Trends in Biotechnology*, 17:53–60, 1999. (Cited on pages 14, 125 and 126.)

[103] S. Schuster, D. A. Fell, and T. Dandekar. A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature Biotechnology*, 18:326–332, 2000. (Cited on page 121.)

[104] S. Schuster, D. A. Fell, and T. Dandekar. A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature Biotechnology*, 18:326–332, 2000. (Cited on pages 12 and 79.)

[105] S. Schuster and C. Hilgetag. On elementary flux modes in biochemical reaction systems at steady state. *Journal of Biological Systems*, 2:165–182, 1994. (Cited on pages 12, 79 and 121.)

[106] S. Schuster, C. Hilgetag, J. H. Woods, and P. A. Fell. Reaction routes in biochemical reaction systems: Algebraic properties , validated calculation procedure and example from nucleotide metabolism. *J Math Biol*, 45(2):153–181, 2002. (Cited on pages 12, 14 and 79.)

[107] S. Schuster, S. Klamt, W. Weckwerth, F. Moldenhauer, and T. Pfeiffer. Use of network analysis of metabolic systems in bioengineering. *Bioprocess and Biosystems Engineering*, 24:363–372, 2001. (Cited on page 126.)

[108] S. Schuster, T. Pfeiffer, F. Moldenhauer, I. Koch, and T. Dandekar. Exploring the pathway structure of metabolism: decomposition into subnetworks and application to *Mycoplasma pneumoniae*. *Bioinformatics*, 18:351–361, 2002. (Cited on pages 14 and 125.)

[109] R. Schwarz, P. Musch, A. von Kamp, B. Engels, H. Schirmer, S. Schuster, and T. Dandekar. YANA - a software tool for analyzing flux modes, gene-expression and enzyme activities. *BMC Bioinformatics*, 6:135, 2005. (Cited on pages 14 and 125.)

[110] H.-S. Song and D. Ramkrishna. Reduction of a set of elementary modes using yield analysis. *Biotechnology and Bioengineering*, 102:554–568, 2009. (Cited on page 124.)

[111] J. Stelling, S. Klamt, K. Bettenbrock, S. Schuster, and E. D. Gilles. Metabolic network structure determines key aspects of functionality and regulation. *Nature*, 420:190–193, 2002. (Cited on pages 14, 125 and 126.)

[112] R. Steuer, T. Gross, J. Selbig, and B. Blasius. Structural kinetic modeling of metabolic networks, 2008. (Cited on page 9.)

[113] P. F. Suthers, Y. Jung, and C. D. Maranas. Improved computational performance of MFA using elementary metabolite units and flux coupling. *Metabolic Engineering*, Oct. 2009. (Cited on pages 16, 22 and 23.)

[114] R. Tarjan. Depth-First Search and Linear Graph Algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972. (Cited on pages 105 and 107.)

[115] M. Terzer. *Large Scale Methods to Enumerate Extreme Rays and Elementary Modes*. PhD thesis, Swiss Federal Institute of Technology, Zurich, 2009. (Cited on pages 13, 14 and 79.)

[116] M. Terzer, N. D. Maynard, M. W. Covert, and J. Stelling. Genome-scale metabolic networks. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 1(3):285–297, 2009. (Cited on page 9.)

[117] M. Terzer, N. D. Maynard, M. W. Covert, and J. Stelling. Genome-scale metabolic networks. *WIREs Systems Biology and Medicine*, 1:285–297, 2009. (Cited on pages 11 and 121.)

[118] M. Terzer and J. Stelling. Large-scale computation of elementary flux modes with bit pattern trees. *Bioinformatics*, 24:2229–2235, 2008. (Cited on pages 14, 79, 110, 121, 122, 130, 134 and 135.)

[119] M. Terzer and J. Stelling. Parallel extreme ray and pathway computation. In *Proceedings of the 8th International Conference on Parallel Processing and Applied Mathematics (PPAM 2009)*, volume 6068 of *Lecture Notes in Computer Science*, pages 300–309, Wroclaw, Poland, 2010. (Cited on pages 121, 122 and 130.)

[120] B. Teusink, A. Wiersma, L. Jacobs, R. A. Notebaart, and E. J. Smid. Understanding the adaptive growth strategy of *Lactobacillus plantarum* by in silico optimisation. *PLoS Computational Biology*, 5:e1000410, 2009. (Cited on pages 14 and 125.)

[121] I. Thiele, T. D. Vo, N. D. Price, and B. O. Palsson. Expanded metabolic reconstruction of *Helicobacter pylori* (iIT341 GSM/GPR): an in silico genome-scale characterization of single- and double-deletion mutants. *Journal of Bacteriology*, 187:5818–5830, 2005. (Cited on pages 31, 74 and 94.)

[122] N. Tomar and R. K. De. Comparing methods for metabolic network analysis and an application to metabolic engineering. *Gene*, 521(1):1–14, 2013. (Cited on page 17.)

[123] C. T. Trinh, A. Wlaschin, and F. Srienc. Elementary mode analysis: a useful metabolic pathway analysis tool for characterizing cellular metabolism. *Applied Microbiology and Biotechnology*, 81:813–826, 2009. (Cited on page 122.)

[124] R. Urbanczik. SNA–a toolbox for the stoichiometric analysis of metabolic networks. *BMC Bioinformatics*, 7:129, 2006. (Cited on page 128.)

[125] R. Urbanczik. Enumerating constrained elementary flux vectors of metabolic networks. *IET Systems Biology*, 1(5):274–279, 2007. (Cited on page 124.)

[126] R. Urbanczik and C. Wagner. Functional stoichiometric analysis of metabolic networks. *Bioinformatics*, 21:4176–4180, 2005. (Cited on pages 34, 80, 128 and 129.)

[127] A. Varma and B. O. Palsson. Metabolic capabilities of< i> escherichia coli</i>: I. synthesis of biosynthetic precursors and cofactors. *Journal of theoretical biology*, 165(4):477–502, 1993. (Cited on page 17.)

[128] W. Verwoerd. Identifying coherent subnetworks in genome scale metabolic networks. In *MODSIM07*, pages 2013–2019, Christchurch, New Zealand, 2007. (Cited on pages 14 and 125.)

[129] W. S. Verwoerd. A new computational method to split large biochemical networks into coherent subnets. *BMC Systems Biology*, 5:25, 2011. (Cited on pages 14 and 125.)

[130] A. von Kamp and S. Schuster. Metatool 5.0: fast and flexible elementary modes analysis. *Bioinformatics*, 22:1930–1931, 2006. (Cited on pages 14, 79, 110 and 121.)

[131] K. Voss, M. Heiner, and I. Koch. Steady state analysis of metabolic pathways using petri nets. *In Silico Biology*, 3:31, 2003. (Cited on page 64.)

[132] C. Wagner. Nullspace Approach to Determine the Elementary Modes of Chemical Reaction Systems. *Journal of Physical Chemistry B*, 108:2425–2431, 2004. (Cited on page 121.)

[133] C. Wagner and R. Urbanczik. The geometry of the flux cone of a metabolic network. *Biophysical Journal*, 89:3837–3845, 2005. (Cited on page 128.)

[134] S. J. Wiback and B. O. Palsson. Extreme pathway analysis of human red blood cell metabolism. *Biophysical Journal*, 83:808–818, 2002. (Cited on pages 31, 74, 128, 131, 133 and 134.)

[135] M. Yeung, I. Thiele, and B. O. Palsson. Estimation of the number of extreme pathways for metabolic networks. *BMC Bioinformatics*, 8:363, 2007. (Cited on page 122.)

[136] K. Yizhak, T. Tuller, B. Papp, and E. Ruppin. Metabolic modeling of endosymbiont genome reduction on a temporal scale. *Molecular Systems Biology*, 7(1), Mar. 2011. (Cited on pages 16, 21, 23 and 31.)

[137] J. Zanghellini, D. E. Ruckerbauer, M. Hanscho, and C. Jungreuthmayer. Elementary flux modes in a nutshell: Properties, calculation and applications. *Biotechnology Journal*, page n/a, June 2013. (Cited on page 79.)

[138] Q. Zhao and H. Kurata. Genetic modification of flux for flux prediction of mutants. *Bioinformatics*, 25:1702–1708, 2009. (Cited on page 126.)

[139] A. R. Zomorrodi, P. F. Suthers, S. Ranganathan, and C. D. Maranas. Mathematical optimization applications in metabolic networks. *Metabolic engineering*, 14(6):672–686, 2012. (Cited on page 17.)

Appendix A

Abbreviations

CBM - constraint-based modeling

CFCA - constrained flux coupling analysis

DAG - directed acyclic graph

EM - elementary (flux) mode

EFM - elementary flux mode

EFP - elementary flux pattern

F2C2 - fast flux coupling calculator

FBA - flux balance analysis

FCA - flux coupling analysis

FCF - flux coupling finder

FFCA - feasibility-based flux coupling analysis

FVA - flux variability analysis

LP - linear program

MAC - metabolite activity coupling

MCC - metabolite concentration coupling

MCCA - metabolite concentration coupling analysis

MCS - minimal cut sets

MILP - mixed integer linear program

MIP - mixed integer (linear) program

MMB - minimal metabolic behavior

NP - non-deterministic polynomial (time)

ODE - ordinary differential equation

P - polynomial (time)

PEM - projected elementary mode

ProCEM - projected cone elementary mode

SCC - strongly connected component

Appendix B

Curriculum Vitae

Education

Current May 2009	Studying towards Ph.D. at Freie Universität Berlin Research area: Systems biology Thesis title: Algorithms for the constraint-based analysis of metabolic networks Supervisor: Prof. Dr. Alexander Bockmayr
Sep 2007-May 2009	Phase I student at Berlin Mathematical School Mentor: Prof. Dr. Alexander Bockmayr
Sep 2005-Jun 2006	MSc in Computer Science from Babes - Bolyai University, Cluj-Napoca, Romania Research area: Component based programming Thesis title: Cleanroom methods Supervisor: Prof. Dr. Militon Frentiu
Sep 2001-Jun 2007	BSc in Computer Science from Petru Maior University, Targu Mures, Romania Thesis title: A universal voice controlled remote control on a PDA device Supervisor: Dr. Haller Piroska
Sep 2001-Jun 2005	BSc in Mathematics & Informatics from Sapientia - Hungarian University of Transylvania, Targu Mures, Romania Thesis title: Person recognition and identification based on voice patterns Supervisor: Dr. Antal Margit

Teaching Experience

May 2009-Sep 2013	Teaching Assistant at Freie Universität Berlin Tutored courses: Discrete mathematics I & II, Topology, Project management in software development.
Sep 2005-Sep 2007	Teaching Assistant at Sapientia - Hungarian University of Transylvania, Targu Mures, Romania Tutored courses: Discrete mathematics I, Databases

Publications

- ⊗ László Dávid, Alexander Bockmayr: Computing elementary flux modes involving a set of target reactions. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, August 2014
- ⊗ László Dávid, Alexander Bockmayr: Constrained Flux Coupling Analysis. *Workshop on Constraint based Methods for Bioinformatics*, WCB'13, Uppsala, 75-83, Sep 2013
- ⊗ Sayed-Amir Marashi, László Dávid, Alexander Bockmayr: Analysis of Metabolic Subnetworks by Flux Cone Projection. Algorithms for Molecular Biology, 7:17, 2012
- ⊗ Abdelhalim Larhlimi, László Dávid, Joachim Selbig, Alexander Bockmayr: F2C2: a fast tool for the computation of flux coupling in genome-scale metabolic networks. BMC Bioinformatics 13:57, 2012
- Sayed-Amir Marashi, László Dávid, Alexander Bockmayr: On flux coupling analysis of metabolic subsystems. Journal of Theoretical Biology, 302, 62-69, 2012
- ⊗ László Dávid, Sayed-Amir Marashi, Abdelhalim Larhlimi, Bettina Mieth, Alexander Bockmayr: FFCA: a feasibility-based method for flux coupling analysis of metabolic networks. BMC Bioinformatics 12:236, 2011

⊗ : Publications with first authorship or joint first authorship.

Appendix C

Declaration

I assert to have written this PhD-thesis on my own, that I cited all the sources I used, and that I did not use any illegitimate tools.

30 September, 2014

László Dávid