54

Forschungsberichte aus dem Max-Planck-Institut
für Dynamik komplexer technischer Systeme

Philipp Schneider

Advances in Computational Strain Design with Minimal Cut Sets

Advances in Computational Strain Design with Minimal Cut Sets

Dissertation
zur Erlangung des akademischen Grades

Doktoringenieur
(Dr.-Ing.)

von Philipp Schneider
geb. am 22.10.1991 in Mainz

genehmigt durch die Fakultät für Verfahrens- und Systemtechnik der
Otto-von-Guericke-Universität Magdeburg

Promotionskommission:

Dr.-Ing. Udo Reichl (Vorsitz)

Dr.-Ing. Steffen Klamt (Gutachter)

Dr. Jürgen Zanghellini (Gutachter)

Dr.-Ing. Andreas Kremling (Gutachter)

eingereicht am: 30.09.2021
Promotionskolloquium am: 02.12.2021

Forschungsberichte aus dem Max-Planck-Institut
für Dynamik komplexer technischer Systeme

Band 54

Philipp Schneider

Advances in Computational Strain Design with Minimal Cut Sets

Shaker Verlag
Düren 2022

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Zugl.: Magdeburg, Univ., Diss., 2021

Printed in Germany.

ISBN 978-3-8440-8411-5
ISSN 1439-4804

Shaker Verlag GmbH • Am Langen Graben 15a • 52353 Düren
Phone: 0049/2421/99011-0 • Telefax: 0049/2421/99011-9
Internet: www.shaker.de • e-mail: info@shaker.de

Abstract

Modern genetic engineering tools allow the manipulation of metabolic networks for different purposes, ranging from the effective bioproduction of value-added chemicals with microorganisms to medical applications. This practice, also known as metabolic engineering, is often aided by computational design methods that predict suitable genetic intervention targets. The minimal cut set (MCS) method presents a powerful tool for the computer-aided design of genome-scale metabolic networks, and is based on mixed integer linear programming (MILP). An MCS is a minimal set of interventions that reshapes a wild type metabolism according to a template of undesired and desired metabolic functions. This work extends and generalizes the framework of MCS and presents new algorithmic and theoretical developments which largely broaden its spectrum of applications.

Many metabolic engineering approaches aim to enforce or improve the product synthesis of bioproduction hosts. A common goal is to couple the microbial growth to the synthesis of a product and make the product of interest a mandatory byproduct of growth. Although a multitude of strain design approaches employ this general principle, various and partially contradicting notions of the actual growth coupling relationship prevail. To systematize these notions, this work proposes four degrees of growth-coupled product synthesis with gradually increasing rigidity. It will be shown that an extended MCS algorithm allows the computation of strain designs for all four coupling degrees within one framework, thereby closing the gap between bilevel and MCS-based strain design approaches.

Beyond growth-coupled strain design, the MCS approach is extended by a number of functional generalizations, including a new compression routine for gene-protein-reaction associations, the support of reaction and gene additions, the possibility to specify multiple undesired and desired flux regions and the use of individual cost factors for different interventions. These generalizations allow, for example, the efficient computation of gene-based MCS and of strain designs that involve the expression of heterologous pathways or substrate co-feeding.

The MCS algorithm can be used to generate large pools of strain design candidates. To select a specific candidate for experimental implementation, one must weigh its individual strengths and weaknesses. The screening procedure presented herein uses ten criteria to assess the performance, robustness and experimental effort of a candidate and is not limited to MCS strain designs. Different intervention strategies with identical metabolic outcomes, e.g., by blocking the same pathway with the knockout of different enzymes, are gathered in equivalence classes of which only one candidate needs to be assessed. A ranked list of the candidates can be obtained by weighting the benchmarks from each criterion with suitable ranking factors. The ranking approach can be used to characterize reaction- and gene-based MCS and strain designs with different growth-coupling degrees.

All algorithmic developments of this work have been integrated into the MATLAB-based *CellNetAnalyzer* toolbox and were verified with various computation examples.

Acronyms

Notation	Description
1,4-BDO	1,4-butanediol
2,3-BDO	2,3-butanediol
ACP	ATP-coupled production
API	application programming interface
ATP	adenosine triphosphate
CBM	constrained-based modeling
CDW	cell dry weight
CNA	*CellNetAnalyzer*
dACP	directionally ATP-coupled production
dGCP	directionally growth-coupled production
DNF	disjunctive normal form
ED	Entner-Doudoroff pathway
EFM	elementary flux mode
EFV	elementary flux vector
FBA	flux balance analysis
FVA	flux variability analysis
GCP	growth-coupled production
GPR	gene-protein-reaction
GRPY	growth-rate-product-yield space
GUI	graphical user interface
IS	intervention strategy
LFP	linear-fractional program
LP	linear program
MBLP	mixed binary linear program
MCS	minimal cut set
MDF	max-min driving force
MILP	mixed integer linear program
NAD(P)H	nicotinamide adenine dinucleotide (phosphate)
NGAM	non-growth-associated ATP maintenance
optMDF	optimal max-min driving force
PE	production envelope
pGCP	potentially growth-coupled production
SL	synthetic lethal
SUCP	substrate-uptake-coupled production
wGCP	weakly growth-coupled production
YS	yield space

Contents

1 Introduction

High hopes are pinned on biotechnology to provide solutions to the manifold challenges of the 21st century. Recently, the COVID-19 pandemic showcased the pace and potential of current biotechnological research. With a few months of development only, a number of companies was able to put forth new and groundbreaking vaccines. Less than one year after the outbreak, the first products were approved by the European Union and deployed in comprehensive vaccination campaigns. The greatest challenge of this century, climate change, is yet unresolved. In 2015, the United Nations Climate Change Conference in Paris set out the goal to limit the global temperature increase to 2 °C and ideally 1.5 °C above the pre-industrial level. To reach this goal, the Paris Agreement declared that the threshold of 40 gigatons CO_2 emissions per year must not be exceeded. As of 2015, however, annual emissions were projected to reach 55 gigatons by 2030 [4], far from the set goal. To combat climate change effectively, it is hence necessary to progressively replace fossil-based technologies with carbon-neutral ones.

Bioprocesses, in contrast to traditional chemical processes, are based on renewable feedstock like cellulose or glucose, require little energy input, do not need high temperature and pressure and therefore hold great potential as an alternative and more sustainable way of chemical production [5, 6]. Nowadays, bioproduction is already possible for a plethora of chemical compounds that range from biofuels, such as ethanol and 2,3-butanediol, over biochemical building blocks like amino acids and nucleotides, to antibiotics, monoclonal antibodies, recombinant proteins and nutraceuticals [7–12] and even more complex products, like cultivated meat, seem to be in reach [13]. Additionally, biotransformation pathways, that is, sequences of regio- and stereospecific enzymatic reactions, allow the synthesis of complex compounds and single isomers that can otherwise hardly or not at all be produced by classical chemical synthesis. As the characterization of unknown and the de-novo design of artificial production pathways proceeds, new molecules become accessible to bioproduction [14–16]. In fact, a recent study was able to engineer the bioproduction of 6 out of 10 different molecules within a given time window of 90 days [17]. Another crucial factor for biotechnological processes, the design of biological production hosts, is driven by various advances in genetic engineering, for instance, by the lambda red recombineering system, which allows to modify

the microbial genome in the scope of days [18, 19] and notably by the CRISPR-cas9 system [20], whose discovery was recently honored with the Nobel Prize.

Despite these presented advantages and advances, large-scale industrial processes are a rare sight, especially for the synthesis of bulk chemicals and fuels (except for ethanol). Low titers and productivity, contamination hazards, scalability challenges and expensive downstream processing often stand in the way of commercialization [5, 21]. Even more so in the light of the currently cheap gas and oil used in chemical processes, the crucial challenge for bioproduction will hence be more frequently profitability than feasibility [5, 21–23]. Designing profitable large-scale bioprocesses is difficult but certainly not impossible. The companies Genomatica, Gevo and LanzaTech provide good examples of how processes can be streamlined for several base chemicals like ethanol, 1,4-butanediol, 1,3-butylene glycol, isobutanol and isooctane. However, the demise of their competitor Rennovia also shows how thin this margin of success is and how easily low oil prices deter investors from engaging in this sector [24]. To make bioprocesses and biorefineries a viable alternative to non-renewable technologies, continuous efforts for increasing productivity, yield, titer (TRY) and process stability are needed. There are a number of starting points for these improvements in upstream and downstream processing, for instance the selection and design of suitable production hosts and their production pathways, the choice of the substrate or the tuning of fermentation and separation process parameters, to name only a few.

One key discipline in effective biomanufacturing is metabolic engineering. To support the synthesis of a product of interest in large amounts and at high yields, the expression of (heterologous) pathways can be accompanied by a directed tailoring of the production host's metabolism [25]. Metabolic engineering aims to harness and improve the production capabilities of a strain by introducing genetic knockouts and regulatory interventions that redirect the metabolic flux towards the product. There are many laboratory-scale examples where metabolic engineering strategies were successfully implemented for the enhanced bio-based synthesis of a large variety of compounds including platform chemicals, biofuels, amino acids and precursors for bioplastics [14, 25–29]. Aside from the aforementioned experimental metabolic engineering tools, various theoretical methods have been deployed for the analysis and design of the microbial metabolism, drawing on the mathematical and computational approaches of kinetic and constraint-based modeling [30–32] or, more recently, machine learning [33]. A particular boost for constraint-based methods, has been the progress in the reconstruction of genome-scale metabolic models [34, 35]. Today, genome-scale models are available for numerous production hosts [36] and allow the in-depth analysis of metabolic networks and their capabilities but also design of networks by the means of mathematical optimization [37–40].

Optimization-based strain design methods for metabolic engineering identify interventions in the microbial metabolism to improve its production capacities. A milestone in the develop-

ment was the OptKnock algorithm [41]. Many microbial strains tune their metabolic activities to attain the highest possible growth rates. In wild types, this behavior conflicts with the metabolic engineering goal of increased product synthesis. In practice, growth is then favored because fast-growing phenotypes will simply outgrow and supersede those with production. The idea behind the OptKnock method was to find genetic knockouts that resolve this trade-off and render the product of interest a (mandatory) by-product of fast growth. Through adaptive laboratory evolution, such strains would evolve towards their growth-maximal phenotype and therefore also towards product synthesis [42]. This powerful design principle, known as *growth-coupled production* (GCP), gave rise to a variety of strain optimization methods introduced in subsequent studies [31, 43–48]. Computed strain designs for GCP could also be implemented successfully in a number of metabolic engineering studies [29, 49–56].

Two approaches dominated the further development of strain optimization methods: bilevel optimization (as used in OptKnock) and minimal cut sets (MCSs) [57]. Both prioritize different strain design goals and optimize for different mathematical objectives [58, 59]. Bilevel methods maximize product synthesis under the (optimistic) assumption that strains are able to evolve towards growing phenotypes. They are therefore sometimes called *biased* methods [58]. Bilevel algorithms usually prioritize the optimization of production capabilities and put the number of introduced interventions second. MCS methods, on the other hand, minimize the number of interventions to reach the predefined yield goal. The hereby computed strain designs have more robust production properties, as the predefined target production (yield) is guaranteed for all possible phenotypes and growth rates [3]. A previous study used the MCS framework to compute strain designs based on genome-scale metabolic networks and showed that GCP strain designs exist for of almost all small metabolites in five major production organisms [31].

This work introduces various generalizations and algorithmic improvements for the MCS framework, largely extending the scope of the MCS approach for computational strain design. An overview of the most important developments is given in Figure 1.1.

GCP has been applied in numerous strain design approaches, nevertheless, different variants and partially contradicting notions of this principle exist in the literature. Chapter 3 proposes a standardized ontology that distinguishes the following four major classes of GCP ordered by increasing coupling degree: potentially, weakly and directionally growth-coupled production (pGCP, wGCP, dGCP) and substrate-uptake-coupled production (SUCP). The relationship between the different coupling degrees is clarified, and it is shown that the MCS framework can be employed to compute strain designs for all four coupling degrees. While the formulation of MCS problems to find dGCP and SUCP strain designs is straightforward and have already been used in previous studies [60], a new feature is introduced to the MCS framework that allows the inclusion of implicit optimality constraints and thereby enables the computation of pGCP and wGCP strain designs. This extension closes the gap between

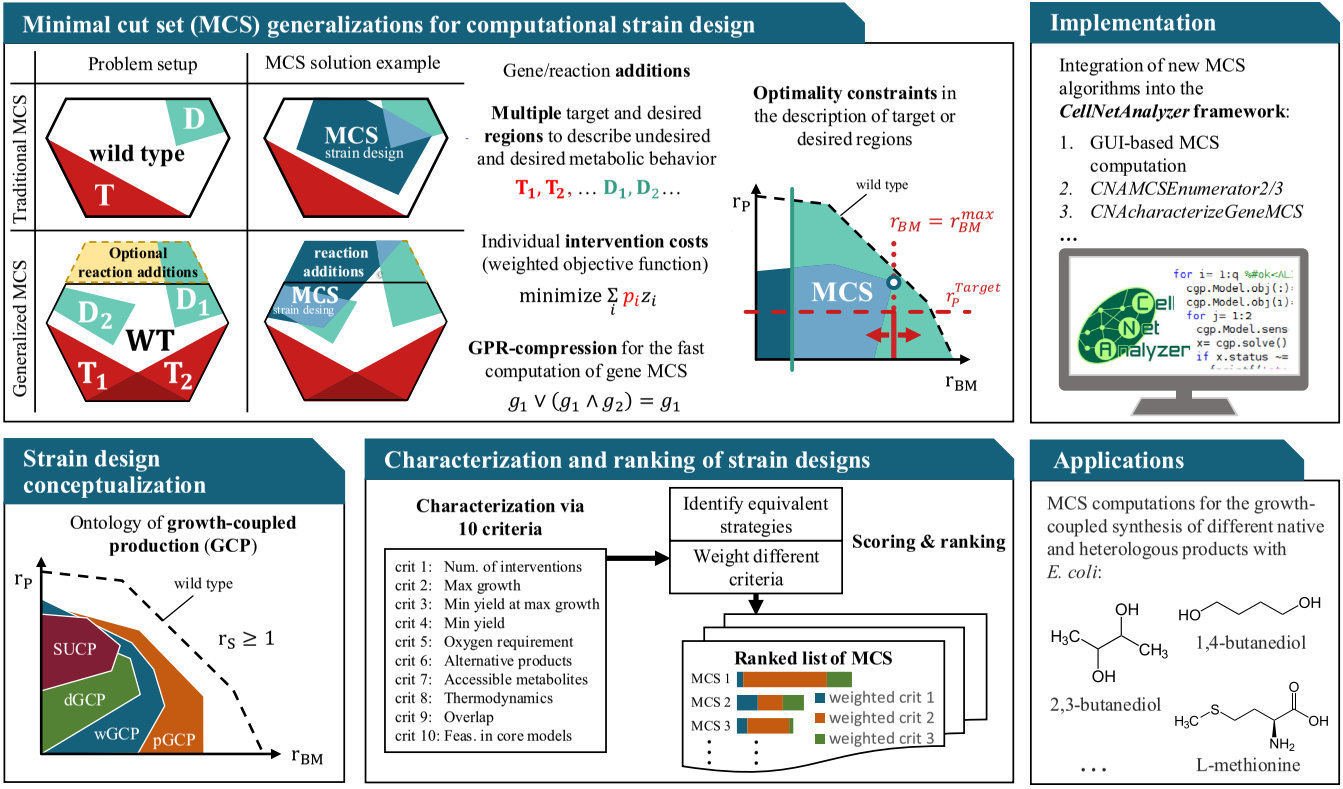


Figure 1.1: Overview of the advances in computational strain with MCS presented in this work. Chapter 3 treats the GCP-based strain design conceptualization, MCS generalizations are introduced in chapters 3 and 4, and the characterization and ranking of strain designs is described in chapter 5. Implementation details are presented in chapter 6. Results of the chapters 3, 4 and 5 have been published in the research articles [3], [2] and [1] respectively.

MCS-based and bilevel-based strain design approaches and supports the computation of all classes of GCP within a single framework. MCS enumeration in various application examples is used to demonstrate the hierarchical relationships between the different degrees of GCP. In genome-scale computations, the effect of a chosen coupling degree on the minimum number of required interventions and the needed computation time is quantified. Finally, the coupling principle is generalized to link product synthesis with other biological functions than growth. As an example, the coupling of production with net ATP formation (ATP-coupled production; ACP) is discussed.

Chapter 4 presents a number of major extensions that further generalize the existing MCS approach, broaden its scope for applications in metabolic engineering and improve algorithmic performance. Gene-protein-reaction (GPR) associations are embedded in the metabolic network structure for the computation of gene-based intervention strategies. This extension is combined with multiple novel compression rules for GPR associations that speed up the computation of gene-based MCS. Another enhancement allows the specification of strain design goals, i.e. desired and undesired flux states, via multiple target and multiple (protected) desired "regions". Together with the added support for gene (or reaction) additions and individual cost factors for interventions, a tailoring of the phenotypical metabolic space for strain design is now possible with unlimited flexibility and precision. The applicability and performance benefits of the new developments are demonstrated by computing (gene-based) *Escherichia coli* strain designs for the substrate-uptake-coupled production of 2,3-butanediol (2,3-BDO), a chemical, that has recently received much attention in the field of metabolic engineering [61, 62].

The MCS computations from chapter 4 show that one may easily generate thousands of possible intervention strategies (ISs) for a given metabolic engineering problem. For experimental implementation, the most suitable strategy must be selected. Chapter 5 focuses on how to evaluate and rank, in a meaningful way, a given pool of computed metabolic engineering strategies for growth-coupled product synthesis. Apart from straightforward criteria to select the most promising candidates, such as a preferably small number of interventions, a preferably high growth rate and a high product yield, several new criteria are specified to assess, for example, the robustness of computed ISs. In addition, "equivalence classes" are introduced for grouping intervention strategies with identical solution spaces, which effectively reduces the number of IS candidates that need to be considered. The ranking procedure involves in total ten criteria that are used for the strain design selection process. The applicability of the characterization and ranking approach is demonstrated by assessing knockout-based intervention strategies for the substrate-uptake-coupled synthesis of L-methionine and of the heterologous product 1,4-butanediol (1,4-BDO) in *E. coli* computed as MCSs in a genome-scale model.

Various algorithmic developments implementing the extensions conceived herein were integrated into functions of the publicly available *CellNetAnalyzer* framework. Details on the implementation of these functions are described in chapter 6.

Publications

The results presented in the chapters 3 to 5 have been published in the following research articles:

Schneider P., Klamt S. (2019). Characterizing and ranking computed metabolic engineering strategies. *Bioinformatics*, **35**, 17, 3063–3072, doi: 10.1093/bioinformatics/bty1065.

Schneider P., Kamp A. v., Klamt S. (2020). An extended and generalized framework for the calculation of metabolic intervention strategies based on minimal cut sets. *PLOS Computational Biology*, **16**, 7, e1008110, doi: 10.1371/journal.pcbi.1008110.

Schneider P., Mahadevan R., Klamt S. (2021). Systematizing the different notions of growth-coupled product synthesis and a single framework for computing corresponding strain designs. *Biotechnology Journal*, **16**, 12, 2100236, doi: 10.1002/biot.202100236.

These articles will not explicitly be cited in this thesis.

2 Foundations of metabolic modeling and computational strain design

2.1 Mathematical optimization

In the following, we give a brief introduction into linear and mixed integer linear programming, focusing on the theorems that are relevant for the traditional and the extended minimal cut set (MCS) approach. A more detailed introduction to linear optimization methods can be found in many textbooks [63, 64].

We consider a system of linear inequalities

$$\mathbf{A}\,\mathbf{x} \leq \mathbf{b}, \tag{2.1}$$

with a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, a vector $\mathbf{b} \in \mathbb{R}^m$ and a vector $\mathbf{x} \in \mathbb{R}^n$. The set $P = \{\mathbf{x} \in \mathbb{R}^n \,|\, \mathbf{A}\,\mathbf{x} \leq \mathbf{b}\}$ of all vectors $\mathbf{x}$ satisfying eq. (2.1) forms a convex polyhedron. In the special case that all constraints are homogeneous ($\mathbf{b} = \mathbf{0}$), the system of linear inequalities describes a (polyhedral) cone $C = \{\mathbf{x} \in \mathbb{R}^n \,|\, \mathbf{A}\,\mathbf{x} \leq \mathbf{0}\}$.

2.1.1 Linear programming (LP)

A linear program (LP) identifies a vector $\mathbf{x_{opt}} \subset P$ that maximizes or minimizes a given linear objective function defined by $\mathbf{c}^\intercal\mathbf{x}$:

$$\begin{aligned} &\text{maximize} && \mathbf{c}^\intercal\mathbf{x} \\ &\text{subject to} && \mathbf{A}\,\mathbf{x} \leq \mathbf{b}. \end{aligned} \tag{2.2}$$

Solving LPs is possible in polynomial runtime [65] and even large problems with thousands of variables and constraints can, in practice, be solved within seconds. A famous algorithm for solving LPs is the Simplex method, however, nowadays, a multitude of other algorithms, e.g., interior point methods, are used for the efficient solution of LPs. Professional optimization packages (e.g. CPLEX, gurobi, intlinprog) provide an interface to such algorithms and allow the user to fully focus on the formalization of the optimization problem.

Farkas' lemma and LP duality

Two major LP theorems are Farkas' Lemma and LP duality. Farkas' lemma defines a pair of two linear inequality systems such that one and only one of them is feasible. Because of this relationship, the feasibility of one system is a certificate for the infeasibility of the other. There exist different variants of Farkas' lemma that can be translated into each other. Here we use a version in which eq. (2.1) occurs as one of the two systems:

$$\begin{array}{c}\text{Exactly one of the following is true:}\\ \mathbf{A}\,\mathbf{x} \leq \mathbf{b} \qquad \text{or} \qquad \mathbf{A}^\mathsf{T}\mathbf{y} = \mathbf{0},\quad \mathbf{y} \geq \mathbf{0},\quad \mathbf{y}^\mathsf{T}\mathbf{b} < 0.\end{array} \tag{2.3}$$

LP duality is closely related to Farkas' Lemma. LP duality states that to any *primal* LP-problem (eq. eq. (2.2)), there exists a *dual* problem that reads:

$$\begin{array}{ll}\text{minimize} & \mathbf{b}^\mathsf{T}\mathbf{y}\\ \text{subject to} & \mathbf{A}^\mathsf{T}\,\mathbf{y} = \mathbf{c}\\ & \mathbf{y} \geq \mathbf{0},\end{array} \tag{2.4}$$

which relates to the primal problem through:

$$\mathbf{c}^\mathsf{T}\mathbf{x} = \mathbf{y}^\mathsf{T}\mathbf{A}\,\mathbf{x} \leq \mathbf{y}^\mathsf{T}\mathbf{b}. \tag{2.5}$$

Equation (2.5) implies the statement of weak duality

$$\mathbf{c}^\mathsf{T}\mathbf{x} \leq \mathbf{b}^\mathsf{T}\mathbf{y}, \tag{2.6}$$

and furthermore strong duality [66]

$$\max\, \mathbf{c}^\mathsf{T}\mathbf{x} = \min\, \mathbf{b}^\mathsf{T}\mathbf{y}. \tag{2.7}$$

Farkas' Lemma and LP duality provide valuable tools in LP, often used to bound and/or solve optimization problems. The direct relationship between LP duality and Farkas' lemma can be shown as follows: we choose the primal LP system as given in eq. (2.2) and assume $\mathbf{c} = \mathbf{0}$. We derive the dual LP system (2.4) and add the constraint $\mathbf{b}^\mathsf{T}\mathbf{y} < 0$ that contradicts eq. (2.5) ($\mathbf{b}^\mathsf{T}\mathbf{y} \geq 0$ must hold since $\mathbf{c} = \mathbf{0}$). The two systems correspond then directly with the two systems of the Farkas lemma, and exactly one of both is feasible and the other infeasible. For this reason, in the following, we will use the terms "primal" and "Farkas-dual" when referring to the two systems of the Farkas lemma. It is noteworthy that, although the trivial solution $\mathbf{y} = \mathbf{0}$ does not exist in the Farkas-dual, the system is "almost" homogeneous, since any solution of $\mathbf{y}$ remains valid when scaled with a (strictly) positive factor.

An important property of a primal-dual pairs in LP duality and Farkas' Lemma is that variables of one (in)equality system translate to constraints of the other. The following general

rules can be applied for LP dualization and (except for the objective function) for constructing Farkas-dual systems:

$$\begin{array}{ccc} \text{Primal objective function} & & \text{Dual objective function} \\ \text{maximize } \mathbf{c}^\mathsf{T}\mathbf{x} & & \text{minimize } \mathbf{b}^\mathsf{T}\mathbf{y} \\ \\ \text{Primal variables } (i) & & \text{Dual constraints } (i) \\ x_i \geq 0 & & \geq_i \\ x_i \leq 0 & \text{translate to} & \leq_i \\ x_i \in \mathbb{R} & \longrightarrow & =_i \\ \\ \text{Primal constraints } (j) & & \text{Dual variables } (j) \\ \geq_j & & y_j \leq 0 \\ \leq_j & & y_j \geq 0 \\ =_j & & y_j \in \mathbb{R}. \end{array} \tag{2.8}$$

Linear-fractional programming (LFP)

We write the problem of optimizing the ratio of two linear functions, subject to a set of linear constraints, as:

$$\begin{aligned} &\text{maximize} && \frac{\mathbf{c}^\mathsf{T}\mathbf{x}}{\mathbf{d}^\mathsf{T}\mathbf{x}} \\ &\text{subject to} && \mathbf{A}\,\mathbf{x} \leq \mathbf{b}. \end{aligned} \tag{2.9}$$

Under the condition that the denominator term is strictly positive ($\mathbf{A}\,\mathbf{x} \leq \mathbf{b} \Rightarrow \mathbf{d}^\mathsf{T}\mathbf{x} > 0$), eq. (2.9) may be rewritten as an LP problem, using the Charnes-Cooper transformation [67]. The formerly fixed boundaries $\mathbf{b}$ of the problem are then scaled by the auxiliary variable e while the variable $y = \dfrac{\mathbf{c}^\mathsf{T}\mathbf{x}}{\mathbf{d}^\mathsf{T}\mathbf{x}}$ expresses the original objective function:

$$\begin{aligned} &\text{maximize} && y \\ &\text{subject to} && \begin{bmatrix} \mathbf{A} & -\mathbf{b} & \mathbf{0} \\ \mathbf{d}^\mathsf{T} & 0 & 0 \\ \mathbf{c}^\mathsf{T} & 0 & -1 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ e \\ y \end{bmatrix} \begin{matrix} \leq \\ = \\ = \end{matrix} \begin{bmatrix} \mathbf{0} \\ 1 \\ 0 \end{bmatrix} \\ &&& e \geq 0. \end{aligned} \tag{2.10}$$

Solutions of $\mathbf{x}$ to the LFP stated in eq. (2.9) can be retrieved from a solution of the LP in eq. (2.10) through $\mathbf{x} = \frac{\tilde{\mathbf{x}}}{e}$.

2.1.2 Mixed integer linear programming (MILP)

Mixed integer linear programming MILP describes a class of optimization problems of the form eq. (2.2) in which some variables in the **x**-vector are restricted to be integers. This generalization can make the solution of MILP problems significantly more complex as compared to LP, and many famous MILP examples are classified as NP-hard or NP-complete [68, 69]. The MILP solution processes often involves additional techniques like the cutting-planes or the branch-and-bound method. Again, there are software packages with integrated MILP solvers and user-interfaces that accepts the input of MILP problems in a unified form (e.g., eq. (2.11), where Z is the index set of integer variables):

$$\begin{aligned} &\text{maximize} && \mathbf{c}^\top \mathbf{x} \\ &\text{subject to} && \mathbf{A}\,\mathbf{x} \leq \mathbf{b} \\ & && x_i \in \mathbb{Z}, \quad \forall i \in Z. \end{aligned} \tag{2.11}$$

We now introduce several basic formalization techniques, which will be required to state the MCS problem as a MILP. In the MCS problem, the integer variables z_i represent knockouts or additions of certain reactions or genes and are restricted to take only binary values $z_i \in \{0,1\}$. These binary variables are often used to actively control the presence or absence of a variable (x_i) or of a constraint ($\mathbf{A}_i\mathbf{x} \leq b_i$) in a linear inequality system. In the following, we explain the four cases of switching on/off variables or constraints.

Switching off constraints

Assume we aim to remove a (row) constraint $\mathbf{A}_i\mathbf{x} \leq b_i$ when a binary decision variable z_i is *true* (1) and enforce it when z_i is $false$ (0). Here, it is important to note that even when z_i is 1, the corresponding constraint *may still be fulfilled* but, in reverse, when the constraint *cannot be fulfilled*, $z_i = 1$ is *enforced*. Table 2.1 shows the intended logic.

Table 2.1: Logical relationship between a decision variable z_i and a constraint $\mathbf{A}_i\mathbf{x} \leq b_i$ when a value of $z_i = 1$ *disables* the constraint.

Indicator logic: $z_i = 0 \rightarrow \mathbf{A}_i\mathbf{x} \leq b_i$	
statement	implication
$z_i = 1$	$\mathbf{A}_i\mathbf{x} \in \mathbb{R}$
$z_i = 0$	$\mathbf{A}_i\mathbf{x} \leq b_i$
$\mathbf{A}_i\mathbf{x} \leq b_i$	$z_i \in \{0,1\}$
$\mathbf{A}_i\mathbf{x} > b_i$	$z_i = 1$

Formally, the removal of inequality constraints can be simulated by rendering it non-binding through the introduction of slack variables s_i. The activity of each slack variable (and thus the inactivity of the associated linear inequality) can be linked to the controlling variable

z_i. Some MILP solvers offer indicator constraints that can be used to conveniently switch on or off slack variables s_i according to the state of the control variable z_i:

$$\begin{aligned} \mathbf{A}_i\mathbf{x} - s_i &\leq b_i \\ z_i = 0 &\rightarrow s_i = 0 \\ z_i &\in \{0,1\}. \end{aligned} \tag{2.12}$$

In many cases, the exact value of the required slack in each constraint is irrelevant. It is thus possible to omit the slack variable and use indicator constraints to directly control the constraints:

$$\begin{aligned} z_i = 0 &\rightarrow \mathbf{A}_i\mathbf{x} \leq b_i \\ z_i &\in \{0,1\}. \end{aligned} \tag{2.13}$$

When indicator constraints are not available, the same relationship can also be implemented via the so-called big-M method:

$$\mathbf{A}_i\mathbf{x} - z_i \cdot M \leq b_i, \quad z_i \in \{0,1\}. \tag{2.14}$$

Here, M describes a sufficiently large value such that the constraint $\mathbf{A}_i\mathbf{x} \leq b_i$ becomes non-binding when $z_i = 1$.

Switching on constraints

It is also possible to simulate the *introduction* of a constraint $\mathbf{A}_i\mathbf{x} \leq b_i$ when a binary decision variable z_i is 1 (and its *absence* when $z_i = 0$). This can be achieved by inverting the previous logic, which results in the relationship shown in Table 2.2.

Table 2.2: Logical relationship between a decision variable z_i and a constraint $\mathbf{A}_i\mathbf{x} \leq b_i$ when a value of $z_i = 1$ *activates* the constraint.

Indicator logic: $z_i = 1 \rightarrow \mathbf{A}_i\mathbf{x} \leq b_i$	
statement	implication
$z_i = 1$	$\mathbf{A}_i\mathbf{x} \leq b_i$
$z_i = 0$	$\mathbf{A}_i\mathbf{x} \in \mathbb{R}$
$\mathbf{A}_i\mathbf{x} \leq b_i$	$z_i \in \{0,1\}$
$\mathbf{A}_i\mathbf{x} > b_i$	$z_i = 0$

This logic can again be implemented through indicator constraints, analogous to eq. (2.12):

$$\begin{aligned} \mathbf{A}_i\mathbf{x} - s_i &\leq b_i \\ z_i = 1 &\rightarrow s_i = 0 \\ z_i &\in \{0,1\}, \end{aligned} \tag{2.15}$$

or, in a more direct manner, without slack variables

$$\begin{aligned} z_i = 1 \rightarrow \mathbf{A}_i\mathbf{x} \leq b_i \\ z_i \in \{0,1\}, \end{aligned} \tag{2.16}$$

or using the big-M method

$$\mathbf{A}_i\mathbf{x} + z_i \cdot M \leq b_i + M. \tag{2.17}$$

Switching off variables

Given the linear inequality system (2.1), we aim to switch off a variable x_i (by forcing $x_i = 0$), when a binary decision variable z_i is 1, while it should remain unconstrained (but possibly also zero) when $z_i = 0$. It is important to note that $z_i = 0$ is enforced when $x_i \neq 0$. The switch-off-logic for variables was already used for switching on constraints by controlling their slack s_i. Table 2.3 shows the intended relationship between decision variables and continuous variables.

Table 2.3: Logical relationship between a decision variable z_i and a continuous variable x_i when $z_i = 1$ "disables" a variable x_i.

Indicator logic: $z_i = 1 \rightarrow x_i = 0$	
statement	implication
$z_i = 1$	$x_i = 0$
$z_i = 0$	$x_i \in \mathbb{R}$
$x_i = 0$	$z_i \in \{0,1\}$
$x_i \neq 0$	$z_i = 0$

An implementation of this logic is again possible via indicator constraints:

$$\begin{aligned} z_i = 1 \rightarrow x_i = 0 \\ z_i \in \{0,1\}, \end{aligned} \tag{2.18}$$

and alternatively, via the big-M method:

$$\begin{aligned} x_i + z_i \cdot M \leq M \\ -x_i + z_i \cdot M \leq M. \end{aligned} \tag{2.19}$$

If lower (lb_i) or upper (ub_i) boundaries on a variable x_i exist, they can be used instead of the value M:

$$\begin{aligned} x_i + z_i \cdot \quad ub_i \quad &\leq \quad ub_i \\ -x_i + z_i \cdot (-lb_i) \quad &\leq -lb_i. \end{aligned} \tag{2.20}$$

Switching on variables

Finally, also the *activation* of variables x_i can be linked to binary decision variables $z_i = 1$, while $z_i = 0$ implies $x_i = 0$. This can once more be achieved by inverting the logic for the

removal of variables. The resulting logic was already used to control the slack variables when switching off constraints. The relationship between the variables in shown in Table 2.4.

Table 2.4: Logical relationship between a decision variable z_i and a continuous variable x_i when $z_i = 1$ *allows for arbitrary values* of x_i

Indicator logic: $z_i = 0 \rightarrow x_i = 0$	
statement	implication
$z_i = 1$	$x_i \in \mathbb{R}$
$z_i = 0$	$x_i = 0$
$x_i = 0$	$z_i \in \{0,1\}$
$x_i \neq 0$	$z_i = 1$

An implementation of this logic is possible via indicator constraints:

$$\begin{aligned} z_i = 0 \rightarrow x_i = 0 \\ z_i \in \{0,1\}. \end{aligned} \tag{2.21}$$

Alternatively, the big-M formulation reads:

$$\begin{aligned} x_i \leq z_i \cdot M \\ -x_i \leq z_i \cdot M. \end{aligned} \tag{2.22}$$

Again, if boundaries (lb_i, ub_i) on the variables x_i exist, these can be used to fix M:

$$\begin{aligned} x_i \leq z_i \cdot \quad ub_i \\ -x_i \leq z_i \cdot (-lb_i). \end{aligned} \tag{2.23}$$

Using single binary variables to control multiple constraints

A single binary variable z_i may also be used to control multiple constraints (or variables). The previously introduced relationships also apply, for example, when switching off multiple constraints, represented by the index set K, with a single binary varialbe z_i. Importantly, a single unfulfilled constraint $j, j \in K$ suffices to impose $z_i = 1$, while z_i can only be chosen freely when all constraints $\mathbf{A}_K\mathbf{x} \leq \mathbf{b}_K$ are fulfilled. This relationship is shown in Table 2.5.

Table 2.5: Logical relationship between a decision variable z_i and multiple constraints $\mathbf{A}_K\mathbf{x} \leq \mathbf{b}_K$ when a value of $z_i = 1$ *disables* the constraints.

Indicator logic: $z_i = 0 \rightarrow \mathbf{A}_K\mathbf{x} \leq \mathbf{b}_K$	
statement	implication
$z_i = 1$	$\mathbf{A}_K\mathbf{x} \in \mathbb{R}$
$z_i = 0$	$\mathbf{A}_K\mathbf{x} \leq \mathbf{b}_K$
$\mathbf{A}_K\mathbf{x} \leq \mathbf{b}_K$	$z_i \in \{0,1\}$
$\exists j \in K : \mathbf{A}_j\mathbf{x} > b_j$	$z_i = 1$

Exclusion constraints

In the special case of a MILP, where all integers are constrained to take only the values 0 and 1 (mixed binary linear program; MBLP), the number of optimal (and suboptimal) solutions that the integer variables can take is always finite. It is possible to obtain more than one (integer) solution by excluding previously found solutions from the problem space and repeating the computation. The cycle of computing solutions and excluding them can be repeated until all solutions are found and the MILP becomes infeasible. To remove a single solution $S = \{i|z_i = 1\}$ (with the complementary set $D = \{j|z_j = 0\}$), one may add to a problem the integer cut constraint

$$\begin{aligned} \sum_{i \in S} z_i - \sum_{j \in D} z_j \leq |S| - 1 \\ z_i \in \{0,1\}. \end{aligned} \tag{2.24}$$

In the computation of MCSs, only minimal solutions are of interest. When one MCS (equivalent to a solution S) is identified, all supersets of a solution S should also be excluded. This can be achieved by the simplified integer cut constraint

$$\sum_{i \in S} z_i \leq |S| - 1. \tag{2.25}$$

2.2 Constraint-based modeling

Constrained-based modeling (CBM) approaches provide a useful and effective tool for the computer-aided analysis and redesign of metabolic networks based on genome-scale stoichiometric models that have been constructed for a plethora of different organisms over the past years [34, 35, 70, 71]. Integrated software packages, such as COBRA [30], COBRApy [72], OptFlux [73] and *CellNetAnalyzer* [74] offer a variety of CBM methods for metabolic network analysis and engineering, including flux balance analysis (FBA) [75], flux variability analysis (FVA) [76], metabolic pathway analysis based on elementary flux modes/vectors [77, 78] and different strain design algorithms (reviewed in [58, 59]).

2.2.1 Modeling metabolic networks with linear (in)equalities

We consider a metabolic network with m metabolites and n reactions whose structure is described by a stoichiometric matrix $\mathbf{N} \in \mathbb{R}^{m \times n}$. Assumption of steady state for the metabolite concentrations implies the balancing equation

$$\mathbf{N}\,\mathbf{r} = \mathbf{0}, \tag{2.26}$$

where $\mathbf{r} \in \mathbb{R}^n$ is the reaction rate vector with the unit mmol $\mathrm{g}_{\mathrm{CDW}}^{-1}\,\mathrm{h}^{-1}$. To account for the (thermodynamic) (ir)reversibility of certain reactions, a subset of the reactions Irr can be sign-restricted

$$r_i \geq 0,\ i \in Irr. \tag{2.27}$$

The set of flux vectors $\mathbf{r}$ that is subjected to the (homogeneous) constraints (2.26) and (2.27) forms a polyhedral cone, called the *flux cone* $C = \{\mathbf{r} \,|\, \mathbf{N}\mathbf{r} = \mathbf{0},\, r_i \geq 0,\ i \in Irr\}$.

Reaction rates r_i may be further constrained by a lower (lb_i) and an upper (ub_i) bounds expressing physiological flux limits (e.g. maximal substrate uptake rates or minimal ATP maintenance demand) or irreversibilities as in eq. (2.27):

$$lb_i \leq r_i \leq ub_i, \quad \forall i \in \{1, \ldots, n\}. \tag{2.28}$$

This can also be written in the vector form

$$\mathbf{lb} \leq \mathbf{r} \leq \mathbf{ub}. \tag{2.29}$$

The solutions $\mathbf{r}$ of the system, constrained by eqs. (2.26) and (2.29), span the space of feasible steady-state flux vectors. The set of vectors $\mathbf{r}$ fulfilling eq. (2.30) is a convex polyhedron $P = \{\mathbf{r} \,|\, \mathbf{N}\mathbf{r} = \mathbf{0}, \mathbf{lb} \leq \mathbf{r} \leq \mathbf{ub}\}$, also called the *flux polyhedron*. P can be analyzed by dedicated techniques of constraint-based modeling. The model (P) may be expressed in our standard form of linear inequalities (eq. (2.1)) using a single matrix $\mathbf{G}$ and a single vector $\mathbf{g}$:

$$\begin{gathered} \mathbf{G}\mathbf{r} \leq \mathbf{g} \\ \mathbf{G} = \begin{bmatrix} \mathbf{N} \\ -\mathbf{N} \\ -\mathbf{I} \\ \mathbf{I} \end{bmatrix}, \qquad \mathbf{g} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ -\mathbf{lb} \\ \mathbf{ub} \end{bmatrix}. \end{gathered} \tag{2.30}$$

Further linear constraints and variables may be added to the base model in eq. (2.30), to take into account experimental data, gene-protein-reaction (GPR) associations [2, 31, 79] and regulatory [80, 81], proteomic [82, 83] or thermodynamic [84] restrictions which improve the (predictive or descriptive) quality of the model. We will later see that such additional constraints and variables can also be used to define the sets of desired and undesired flux states for the computation of MCSs.

2.2.2 Basic constraint-based methods

A plethora of methods to analyze and utilize constraint-based models have evolved over the past decades. Detailed compendia of those methods can be found in reviews and textbooks (e.g. [37, 85, 86]). Hereafter, we list only methods that are relevant for this work.

Flux balance analysis [75] applies linear programming and enables computation of growth-maximal flux states. The LP problem reads

$$\begin{aligned} &\text{maximize} && r_{biomass} \\ &\text{subject to} && \mathbf{G}\mathbf{r} \leq \mathbf{g}. \end{aligned} \tag{2.31}$$

The objective function can also be adapted to maximize the flux through a production pathway, or to find upper and lower bounds for a single reaction in a given scenario. Flux variability analysis (FVA [76]), for instance, finds the feasible flux ranges for all reaction rates, by successively maximizing and minimizing the metabolic flux through every reaction. To determine maximal (or minimal) product yields, the ratio between two flux rates (production and substrate uptake) can be optimized by the means of linear-fractional programming [87] (cf. eqs. (2.9) and (2.10)).

Graphical projections of the metabolic flux space onto two (or more) rates are often used to unveil flux dependencies in a metabolic network, such as flux couplings or trade-offs. An important example for such a plot is the production envelope (PE, also called growth-product phase plane or biomass-vs.-product trade-off plot) which projects the flux space onto the growth rate (r_{BM}) and the synthesis rate of a product of interest (r_P) [58, 85]. Figures 2.1A and 2.1C show exemplary production envelopes that visualize the relationship of growth and ethanol production in *E. coli* under aerobic and anaerobic conditions. The PEs were computed using the *i*ML1515 model [88] with glucose as the sole substrate, limited to a maximum uptake of $10\,\mathrm{mmol\,g_{CDW}^{-1}\,h^{-1}}$ and a non-growth-associated ATP maintenance (NGAM) demand of $6.86\,\mathrm{mmol\,g_{CDW}^{-1}\,h^{-1}}$. Under aerobic conditions, there is a trade-off between growth and ethanol production, while under anaerobic conditions attaining the maximum possible growth rate is associated with ethanol production.

Similar to the PE, the ranges of two flux ratios (or yield(s)) can be mapped onto a two-dimensional yield space diagram (YS). This can be used to display the relationship between different product ($Y_{P/S}$) and biomass yields ($Y_{BM/S}$). The yield spaces for aerobic and anaerobic production of ethanol by *E. coli* are shown in Figures 2.1B and 2.1D respectively. Often, e.g., also in the presented example, the shapes of PEs and YSs for a given model are identical. However, this is not generally the case, as the introduction of inhomogeneous model constraints, such as nonzero flux bounds, can lead to different shapes in PE and YS [87].

Elementary flux modes (EFMs) [77] refer to the irreducible pathways (support-minimal vectors) of the flux cone C (cf. section 2.2.1). In turn, all vectors of the flux cone are generated by a conic combination of EFMs. Elementary flux vectors (EFVs) extend this idea to models with inhomogeneous constraints, i.e. general flux polyhedra [77]. EFVs are divided into *bounded* EFVs, some of which are *vertices* of the flux polyhedron and *unbounded* EFVs that span the recession cone of the flux polyhedron. All vectors of the flux polyhedron P can be decomposed to a convex combination of bounded EFVs and a conic combination of unbounded EFVs.

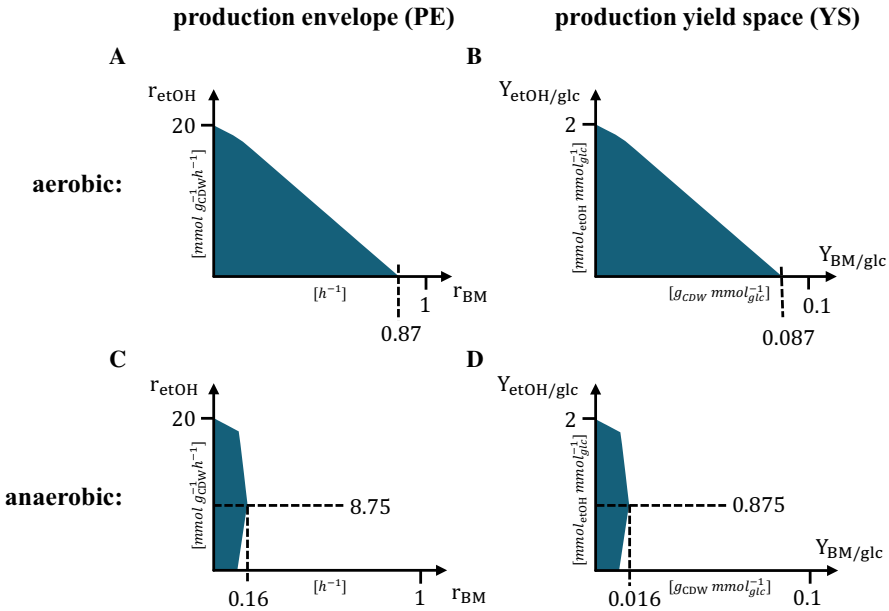


Figure 2.1: Visualization of the relationship between growth and ethanol synthesis in *E. coli* under aerobic and anaerobic process conditions using production envelopes and production yield spaces: (**A**) PE under aerobic conditions, (**B**) YS under aerobic conditions, (**C**) PE under anaerobic conditions, (**D**) YS under anaerobic conditions.

2.3 Computational strain design

Various optimization-based methods have been developed to support the rational design of metabolic networks in microbial strains (for reviews see [58, 59, 89]). These methods identify metabolic interventions (gene or reaction deletions, insertions, or up- and downregulations), often with the goal of harnessing or improving the production capabilities of microbial organisms and priming them for the use in bioproduction processes. The most common design approach for this purpose is to render the product of interest a mandatory by-product of growth. The majority of computational strain design methods for GCP are either bilevel optimization approaches or based on the framework of MCS.

2.3.1 Bilevel-based methods

OptKnock [41] was one of the first algorithms for metabolic design and introduced the principle of growth-coupled production. It identifies suitable knockouts in the metabolic network that maximize the production potential at maximal microbial growth. Using a nested MILP

optimization, the product synthesis rate is maximized (outer objective) under the premise of growth-rate optimality (inner problem):

$$\begin{array}{lll} \text{maximize} & r_P & \\ \text{subject to} & \text{maximize} & r_{BM} \\ & \text{subject to} & \mathbf{N}\,\mathbf{r} = \mathbf{0} \\ & & r_{BM} \geq r_{BM}^{min} \\ & & (1-z_i)\cdot lb_i \leq r_i \leq (1-z_i)\cdot ub_i,\ \forall i \in \{1,\ldots,n\} \\ & & \sum z_i \leq \text{maxKOs} \\ & & z_i \in \{0,1\}. \end{array} \tag{2.32}$$

This bilevel optimization problem, where the binary variables z_i represent the knockouts, can be converted to a single-level MILP. OptKnock stimulated the development of a number of related methods that use nested optimization, such as OptStrain [90], RobustKnock [44], OptORF [91], FOCAL [46, 92] or OptCouple [93].

Although OptKnock strain designs always bear production potential at maximum growth by default, the *exploitation* of that potential is not guaranteed and growth-maximal flux states without any product synthesis may also exist. Some successors of OptKnock therefore adapted the approach and used a tilted objective function [43] or introduced a third optimization level [44] to enforce product synthesis. Nevertheless, all existing nested optimization methods still only demand product synthesis in growth-maximal (or close to growth-maximal [93]) flux states.

2.3.2 Minimal cut sets

Another class of computational strain optimization methods is based on the framework of MCS [57, 60, 94, 95]. Initially, the concept of MCS was introduced to enumerate all combinations of knockouts that eliminate certain functionalities in the network, e.g., growth (synthetic lethalities) or synthesis of undesired products [94, 95]. Later, it was shown that this approach can also be harnessed for metabolic engineering and strain design. When designing a production host, one would target steady-state flux vectors that synthesize undesired byproducts or that lead to poor product yields. However, the deletions needed to eliminate the undesired (target) phenotypes can interfere with parts of the metabolic network that are essential for the desired phenotypes of the strain (e.g., growth with high product yields). Therefore, the extended framework of *constrained* MCS introduced the option to demand the conservation of *desired* behaviors [57] (the concept is illustrated in Figure 2.2). This was a vital step for establishing MCS as a strain design method in metabolic engineering.

Examples of successful MCS applications for designing microbial cell factories with excellent product yields include the growth-coupled synthesis of itaconic acid [54] by *Escherichia*

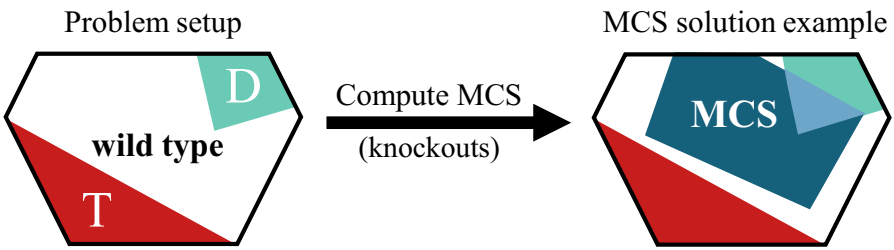


Figure 2.2: Illustration of the MCS concept. For the MCS problem setup, the user specifies the desired (D) and undesired (T) metabolic behaviors as convex regions within the solution space of steady-state flux vectors (wild type, black outline). The MCS algorithm then computes knockout sets that crop the solution space so that it no longer intersects with the region of target flux states (ensuring that the undesired metabolic behavior becomes infeasible) while it still intersects with the (protected) desired region (ensuring that desired phenotypes still exist). The MCS solution space, shown in blue, is the remaining solution space after deleting the reactions of the MCS.

coli and of a heterologous secondary metabolite (indigoidine) by *Pseudomonas putida* [29]. In the latter case, as many as 14 multiplexed gene knockdowns were implemented, as suggested by MCS analysis. A large-scale computational study based on the MCS framework also proved that growth-coupled product synthesis is, at least stoichiometrically, feasible for a broad range of metabolic products, thus demonstrated the potential of this strain design approach [31].

The computation of MCSs was originally based on EFMs. The user specified a set of undesired (target) elementary flux modes and, in the case of constrained MCS, also a set of elementary flux modes that exhibit the desired behavior (from which at least some must not be eliminated by the cuts). The MCS computation is then a search for minimal hitting sets through variants of the Berge algorithm [57, 95–98]. The bottleneck of computing MCSs with this approach is the inevitable preceding enumeration of elementary flux modes, which is infeasible in genome-scale models due to the large number of EFMs [99].

Here, the insight that MCSs in the primal network are elementary flux modes in a corresponding dual network [100, 101], as obtained from Farkas' lemma, allowed the computation of MCS via MILP [60] and helped to overcome this limitation. This approach was implemented in the *MCSEnumerator*, an algorithm that computes the shortest MCSs from specified target and desired flux regions instead of target and desired elementary flux modes [31, 60].

The Farkas-lemma-based (or duality-based) approach enabled for the first time the enumeration of thousands of MCSs in genome-scale models and initiated the developments of other variants of duality-based MCS calculations [102–104]. RegMCS, an extension of the MCS framework, allows the use of regulatory interventions, i.e. up- and down-regulation of fluxes, alongside reaction knockouts [105]. The MCS-variant MoVE enabled the strain design for two-stage processes by introducing flexible knockouts that intercept growth and support enhanced product synthesis [106]. To account for genetic interventions, Machado et al. [79] proposed to incorporate GPR associations into the metabolic network and to use the

MCSEnumerator to compute strain designs on a genetic level. Another technique, the gMCS framework, maps the relationships between sets of gene deletions and their resulting reaction deletions before the MCSs are computed. This approach was designed to find lethal gene knockouts in cancer cells [107, 108], however, the feature of conserving desired behaviors was not integrated in this approach.

Mathematical foundation of the Farkas-lemma-based MCS computation

MCSs are minimal sets of metabolic interventions that block all undesired, while keeping at least some desired (protected) behaviors feasible. Accordingly, to compute MCSs, one needs to specify the undesired (target) behavior as well as the protected (desired) functionalities by suitable systems of linear inequalities. The most generalized formulation of target behaviors uses inequalities posed by a matrix $\mathbf{T} \in \mathbb{R}^{t \times n}$ and a vector $\mathbf{t} \in \mathbb{R}^t$:

$$\mathbf{T}\,\mathbf{r} \leq \mathbf{t}. \tag{2.33}$$

In the simplest case, one demands that the knockouts of an MCS block the flux through a given undesired (target) reaction [94, 100]. If an MCS should block the operation of the reaction with the index k, one targets flux states where r_k takes on high flux rate values. For this example, a threshold of 1 is used:

$$r_k \geq 1. \tag{2.34}$$

This can be written ind the general form of eq. (2.33). $\mathbf{T}$, then, consists of a single row of zeros, except for the position k where -1 is put, while the corresponding vector $\mathbf{t}$ as the single entry -1.

The target region is then defined by the inequalities describing the target behavior (eq. (2.33)) together with the steady-state and flux bound constraints of the metabolic model (eq. (2.30)) yielding:

$$\begin{gathered} \mathbf{A_T}\,\mathbf{x} \leq \mathbf{b_T} \\ \mathbf{A_T} = \begin{bmatrix} \mathbf{G} \\ \mathbf{T} \end{bmatrix}, \quad \mathbf{x} = \mathbf{r}, \quad \mathbf{b_T} = \begin{bmatrix} \mathbf{g} \\ \mathbf{t} \end{bmatrix}. \end{gathered} \tag{2.35}$$

The target region is therefore always contained in the solution space of the full model. Importantly, the target region must not contain the zero vector because it cannot be eliminated through knockouts. This is already ensured if there is a reaction rate with a minimum flux above zero (e.g., non-growth associated ATP maintenance demand), otherwise it can be enforced by demanding some minimum substrate uptake flux through another inequality in eq. (2.33).

With the description of target flux states as linear constraints (eq. (2.33)), it is also possible to select flux vectors that operate in a specific yield range. For instance, to select flux vectors with a product yield below a threshold $Y_{P/S}^{Target} > 0$ for elimination, we first express the product yield as the ratio of the substrate uptake rate r_S and the product exchange rate r_P and then

linearize the equation under the assumption $r_S > 0$ so that it complies with the form of eq. (2.33)

$$\frac{r_P}{r_S} \leq Y_{P/S}^{Target} \quad \Leftrightarrow \quad r_P - Y_{P/S}^{Target} \cdot r_S \leq 0. \tag{2.36}$$

The definition of the desired (or protected) behavior is analogous to the target system and is specified by suitable inequalities using the matrix $\mathbf{D} \in \mathbb{R}^{d \times n}$ and a vector $\mathbf{d} \in \mathbb{R}^d$:

$$\mathbf{D}\,\mathbf{r} \leq \mathbf{d}. \tag{2.37}$$

Like the target region, the desired region is spanned by eq. (2.37) together with the metabolic system constraints shown in eq. (2.30):

$$\begin{gathered} \mathbf{A_D}\,\mathbf{x} \leq \mathbf{b_D} \\ \mathbf{A_D} = \begin{bmatrix} \mathbf{G} \\ \mathbf{D} \end{bmatrix}, \quad \mathbf{x} = \mathbf{r}, \quad \mathbf{b_D} = \begin{bmatrix} \mathbf{g} \\ \mathbf{d} \end{bmatrix}. \end{gathered} \tag{2.38}$$

As an example for computational strain design, the desired region is often used to demand strain viability. This is done by demanding the feasibility of growth rates r_{BM} above a given threshold $r_{BM}^{Desired}$ which translates to a desired constraint, as shown in eq. (2.37), easily:

$$r_{BM} \geq r_{BM}^{Desired} \quad \Leftrightarrow \quad -r_{BM} \leq -r_{BM}^{Desired}. \tag{2.39}$$

The combination of target and desired regions characterize an MCS problem. The aforementioned study [31], for instance, used one target region to eliminate all flux vectors with low product yield and one desired region to ensure that flux states with a minimal biomass yield (and thus a high product yield) are still feasible. The resulting MCS setup of this study is shown in Figure 2.3.

A desired space needs not be defined if blocking the target region is the only goal, e.g., when computing synthetic lethals [60, 107].

Use of Farkas' lemma for the computation of MCS

By definition, an MCS eliminates all undesired metabolic behaviors with a minimal set of network interventions. For the Farkas-lemma-based MCS computation, the undesired metabolic behavior is expressed by a system of linear inequalities as described above [60]. Here, we will first show how Farkas' lemma (eq. (2.3)) can be used to formalize the elimination problem. We will then add the second part of the problem to maintain the desired behavior.

The feasibility of a Farkas-dual system is a certificate of the infeasibility of the original, primal system. We therefore search for support-minimal, i.e. irreducible, sets of network interventions (e.g., reaction knockouts) that render the Farkas-dual feasible. Such an intervention set ensures the *minimal infeasibility* of the original (primal) metabolic system and thus represents an MCS.

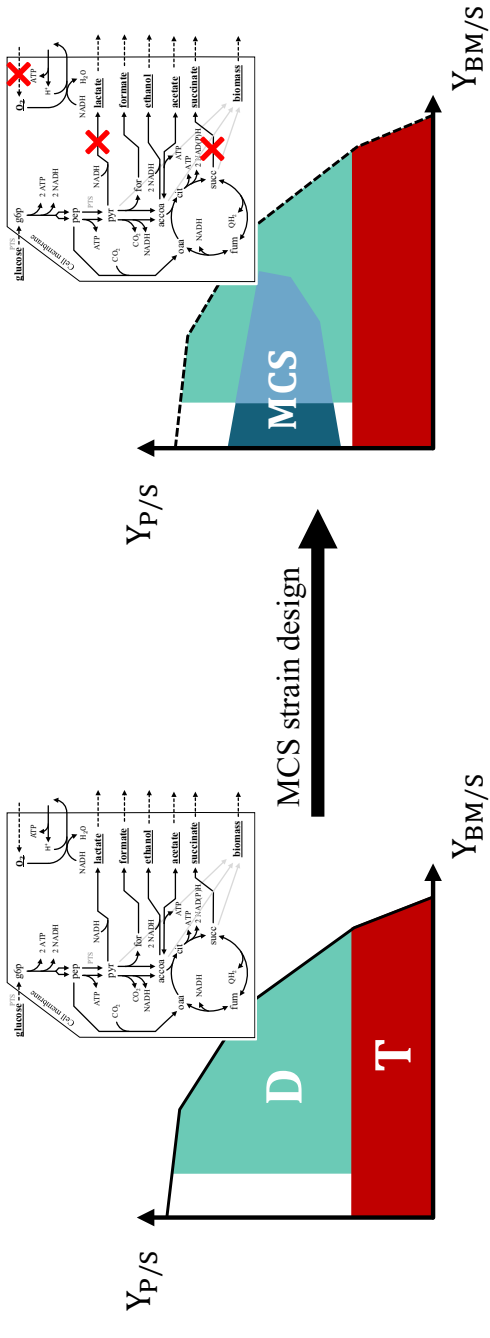


Figure 2.3: MCS approach for designing strains with growth-coupled product synthesis [31]. The MCS algorithm finds reaction knockouts that eliminate all flux states with poor product yields ($\mathbf{T}: \frac{r_P}{r_S} \leq Y^{\min}_{P/S}$), while (some) flux states with a minimum biomass yield ($\mathbf{D}: \frac{r_{BM}}{r_S} \geq Y^{\min}_{BM/S}$), and then with high product yield, remain feasible.

Here, we assume that possible network interventions are reaction knockouts that translate to single variables forced to be zero in the original (primal) system. We first apply additional constraints to the target system eq. (2.35), which permanently force all variables (reaction rates) to be zero ($\mathbf{I_{KO}}$ is the identity matrix):

$$\begin{bmatrix}\mathbf{A_T}\\ \mathbf{I_{KO}}\end{bmatrix}\mathbf{x}\begin{matrix}\leq\\ =\end{matrix}\begin{bmatrix}\mathbf{b_T}\\ \mathbf{0}\end{bmatrix}. \tag{2.40}$$

The introduced permanent knockouts will later be controlled through binary decision variables. It must be noted that the constraints only render the primal system infeasible when they *contradict* the primal system. As was mentioned above, it is therefore not possible to use undesired systems that have the zero vector $\mathbf{x}=\mathbf{0}$ as a feasible solution because then even the knockout of all reactions would not make the primal system infeasible.

With Farkas' lemma in eq. (2.3) and using the dualization rules shown in eq. (2.8), the Farkas-dual system of eq. (2.40) is given by

$$\begin{bmatrix}\mathbf{A_T^T} & \mathbf{I_{KO}}\\ \mathbf{b_T^T} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{y}\\ \mathbf{v}\end{bmatrix}\begin{matrix}=\\ \leq\end{matrix}\begin{bmatrix}\mathbf{0}\\ -1\end{bmatrix}$$
$$\mathbf{y}\geq\mathbf{0}, \tag{2.41}$$

which is, per definition, feasible. The single inequality in the last row corresponds to $\mathbf{b}^\mathsf{T}\mathbf{y}<0$ of the Farkas-dual system in eq. (2.3). The latter needs to be replaced with $\mathbf{b_T^T}\mathbf{y}\leq -1$ because the system will later be used in a MILP that cannot handle strict inequalities. The replacement is allowed because, as was mentioned before (below eq. (2.7)), any found solution of the Farkas-dual eq. (2.41) can be scaled to also fulfill $\mathbf{b_T^T}\mathbf{y}\leq -1$ without affecting the support of the solution (which makes up the MCSs).

As can be seen, the variable knockouts in the primal ($\mathbf{I_{KO}}\,\mathbf{x}=\mathbf{0}$) translate to the variables $\mathbf{v}$ in the Farkas-dual system that mime the knockout of all dual constraints by allowing for arbitrary large slack (analogous to s_i in eq. (2.12)). This system can now be used to identify MCSs, since a minimal subset of constraint relaxations (indicated by $v_i\neq 0$) that solves the Farkas-dual system corresponds directly to a minimal subset of primal knockout-constraints within $\mathbf{I_{KO}}\,\mathbf{x}=\mathbf{0}$ that keeps the primal system infeasible. Hence, every solution of eq. (2.41) with a support-minimal vector $\mathbf{v}$ represents one MCS.

As was shown in eq. (2.12), a constraint can be switched on or off by controlling its slack variable v_i by a corresponding binary variable z_i either via indicator constraints,

$$z_i=0\rightarrow v_i=0, \tag{2.42}$$

or with the big-M method (with M being a sufficiently large number)

$$-M\cdot z_i\leq v_i\leq M\cdot z_i. \tag{2.43}$$

In this case, there is a 1:1 association of metabolic knockouts, indicated by z_i and slack variables v_i. With the binary variables z_i at hand, we may now finally pose a MILP problem with an objective function that minimizes the number of interventions to block the target system (we use here the version with indicator constraints):

$$\begin{aligned} &\text{minimize} && \sum z_i \\ &\text{subject to} && \begin{bmatrix} \mathbf{A_T^\intercal} & \mathbf{I_{KO}} \\ \mathbf{b_T^\intercal} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ -1 \end{bmatrix} \\ & && \forall i : z_i = 0 \rightarrow v_i = 0 \\ & && \mathbf{y} \geq \mathbf{0}, \quad z_i \in \{0,1\}. \end{aligned} \tag{2.44}$$

This MILP finds the smallest irreducible set of interventions (support-minimal in $\mathbf{v}$) that blocks the target system, hence an MCS with the smallest possible cardinality. However, we still need to ensure that the desired system remains feasible. This can be easily achieved by including the respective constraints for the desired system in its primal representation and constraining its variables with the interventions, as shown in eq. (2.18):

$$\begin{aligned} &\text{minimize} && \sum z_i \\ &\text{subject to} && \begin{bmatrix} \mathbf{A_T^\intercal} & \mathbf{I_{KO}} & \mathbf{0} \\ \mathbf{b_T^\intercal} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A_D} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{v} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ -1 \\ \mathbf{b_D} \end{bmatrix} \\ & && \forall i : z_i = 0 \rightarrow v_i = 0 \\ & && \forall i : z_i = 1 \rightarrow x_i = 0 \\ & && \mathbf{y} \geq \mathbf{0}, \quad z_i \in \{0,1\}. \end{aligned} \tag{2.45}$$

Resubstituting all submatrices of $\mathbf{A_T^\intercal}$ and $\mathbf{A_D}$ and all subvectors of $\mathbf{b_T^\intercal}$ and $\mathbf{b_D}$ yields:

$$\begin{gathered}
\text{minimize } \sum z_i \\
\text{subject to} \\
\begin{bmatrix}
\mathbf{N}^\intercal & -\mathbf{N}^\intercal & -\mathbf{I} & \mathbf{I} & \mathbf{T}^\intercal & \mathbf{I_{KO}} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & -\mathbf{lb}^\intercal & \mathbf{ub}^\intercal & \mathbf{t}^\intercal & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{N} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{N} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D}
\end{bmatrix}
\begin{bmatrix} \mathbf{u_1} \\ \mathbf{u_2} \\ \mathbf{u_{lb}} \\ \mathbf{u_{ub}} \\ \mathbf{w} \\ \mathbf{v} \\ \mathbf{r} \end{bmatrix}
\begin{matrix} = \\ \leq \\ \leq \\ \leq \\ \leq \\ \leq \\ \leq \end{matrix}
\begin{bmatrix} \mathbf{0} \\ -1 \\ \mathbf{0} \\ \mathbf{0} \\ -\mathbf{lb} \\ \mathbf{ub} \\ \mathbf{d} \end{bmatrix} \\
\forall i : z_i = 0 \rightarrow v_i = 0 \\
\forall i : z_i = 1 \rightarrow r_i = 0 \\
\mathbf{u_1}, \mathbf{u_2} \geq \mathbf{0}, \quad \mathbf{u_{lb}}, \mathbf{u_{ub}} \geq \mathbf{0}, \quad \mathbf{w} \geq \mathbf{0}, \quad z_i \in \{0,1\}.
\end{gathered} \tag{2.46}$$

The first two columns (with $\mathbf{u}_r = \mathbf{u}_1 - \mathbf{u}_2$) and the third and fourth row of eq. (2.46) can be condensed:

$$\begin{gathered}
\text{minimize } \sum z_i \\
\text{subject to} \\
\begin{bmatrix}
\mathbf{N}^\intercal & -\mathbf{I} & \mathbf{I} & \mathbf{T}^\intercal & \mathbf{I_{KO}} & \mathbf{0} \\
\mathbf{0} & -\mathbf{lb}^\intercal & \mathbf{ub}^\intercal & \mathbf{t}^\intercal & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{N} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D}
\end{bmatrix}
\begin{bmatrix} \mathbf{u_r} \\ \mathbf{u_{lb}} \\ \mathbf{u_{ub}} \\ \mathbf{w} \\ \mathbf{v} \\ \mathbf{r} \end{bmatrix}
\begin{matrix} = \\ \leq \\ = \\ \leq \\ \leq \\ \leq \end{matrix}
\begin{bmatrix} \mathbf{0} \\ -1 \\ \mathbf{0} \\ -\mathbf{lb} \\ \mathbf{ub} \\ \mathbf{d} \end{bmatrix} \\
\forall i : z_i = 0 \rightarrow v_i = 0 \\
\forall i : z_i = 1 \rightarrow r_i = 0 \\
\mathbf{u_{lb}}, \mathbf{u_{ub}} \geq \mathbf{0}, \quad \mathbf{w} \geq \mathbf{0}, \quad z_i \in \{0,1\}.
\end{gathered} \tag{2.47}$$

We note that an even more concise formulation of eq. (2.45) can be constructed by omitting the slack-variables v_i and linking the removal of constraints directly to z_i as was described in eqs. (2.13) and (2.14):

$$\begin{aligned} \text{minimize} \quad & \sum z_i \\ \text{subject to} \quad & \begin{bmatrix} \mathbf{b_T^T} & \mathbf{0} \\ \mathbf{0} & \mathbf{A_D} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix} \leq \begin{bmatrix} -1 \\ \mathbf{b_D} \end{bmatrix} \\ & \forall i : z_i = 0 \rightarrow \mathbf{A_{T,}^T}_i \mathbf{y} = 0 \\ & \forall i : z_i = 1 \rightarrow x_i = 0 \\ & \mathbf{y} \geq \mathbf{0}, \quad z_i \in \{0,1\}. \end{aligned} \tag{2.48}$$

The MCS S (containing the indices of the knocked-out reactions) computed by the MILPs (eqs. (2.44) to (2.48)) is given by $S = \{i | z_i = 1\}$. Multiple MCS solutions (with increasing cardinality) can be found by excluding previously found solutions and their supersets through integer cut constraints (see eq. (2.25)) and solving the MILP repeatedly.

2.4 Models

2.4.1 *i*JO1366 and *EColiCore2*

*i*JO1366 is a constraint-based genome-scale metabolic model of *E. coli* [109]. Particularities of this model are the inhomogeneous flux boundaries for the glucose uptake (limited to a maximum of 10 mmol g_{CDW}^{-1} h^{-1}) and the baseline rate of ATP hydrolysis to simulate the non-growth-associated ATP maintenance (NGAM) demand (3.15 mmol g_{CDW}^{-1} h^{-1}), as well as the presence of exchange reactions for classical fermentation products and also other potential products of *E. coli*. The *i*JO1366 provides gene-protein-reaction (GPR) associations for many reactions, which may be used to study the gene activity in different metabolic phenotypes or to simulate the impact of genetic knockouts.

EColiCore2 is a metabolic core model derived from *i*JO1366 that comprises 500 reactions (about 20 % of *i*JO1366's reactions) and 686 metabolites and simulates *E. coli*'s major metabolic phenotypes [110]. The boundaries on glucose uptake and ATP demand are identical, while only standard fermentation products can be excreted.

2.4.2 *i*ML1515 and *i*ML1515core

*i*ML1515 is the successor of *i*JO1366 [88]. Notable differences between the two are the incorporation of more annotated genes (1515 instead of 1366) and hence additional pathways, an increased non-growth-associated ATP maintenance (NGAM) demand of 6.86 mmol g_{CDW}^{-1} h^{-1} and the removal of several redundant reactions.

*i*ML1515core is a metabolic core model derived herein, similar as *EColiCore2* from its parent model *i*JO1366. It contains 577 reactions from *i*ML1515 and 504 metabolites. Since the *i*ML1515core was developed herein, it can be found in the Appendix (Model A.1).

2.5 Software for constraint-based analysis and design

The integrated software frameworks *CellNetAnalyzer* (*CNA*) [74, 111], COBRA toolbox [30, 72] and *OptFlux* [73] support major constraint-based methods for the analysis of metabolic networks, such as FBA, FVA and EFM-analysis. The frameworks *CellNetAnalyzer* and COBRA require the MATLAB environment, however, also Python implementations of the two exist [72, 112].

In addition to analysis tools, these packages also offer different strain optimization methods that were (partially) first introduced as features of the individual software suits. This is the case for MCS-based methods (*CNA*) or heuristic algorithms (*OptFlux* [113]). Several methods are available in multiple packages, for instance, (basic) MCS algorithms are provided in all three packages and OptKnock is available in COBRA and *OptFlux*.

The *CellNetAnalyzer* framework has been developed at the Max Planck Institute for Dynamics of Complex Technical Systems and was initially introduced as *FluxAnalyzer* [114]. Since then, the toolbox has been further developed over two decades, resulting in many functional extensions. The computation of MCSs has been a centerpiece of *CNA* throughout its evolution [60, 94]. As this work follows up on the development of MCS (and MCS-related) algorithms, it contributed to the recent versions of *CNA*. Implementation details are given in chapter 6.

Many constraint-based analysis and design methods, including MCS, rely on MILP or LP which require additional dedicated optimization software. For *CellNetAnalyzer* and COBRA, one may choose from several commercial and free solvers: CPLEX and Gurobi, commercial solvers that are free for academic use, (int)linprog, a MILP/LP solver provided by the MATLAB Optimization Toolbox and GLPK, a free open source MILP/LP solver alternative (see section 6.3).

3 Systematizing the different notions of growth-coupled product synthesis and computing corresponding MCS strain designs

Finding the metabolic interventions that enforce growth-coupled production (GCP) is the common goal of many computational strain design methods [58, 59]. Recently, it has been shown that, in principle, almost all small metabolites in five major production organisms can be coupled with growth by suitable knockout strategies [31].

While GCP generally means that the product of interest is a by-product of growth, different degrees or types of this coupling have been used in the strain design algorithms developed in the past. As described in section 2.3.1, the first computational method for growth-coupled strain design was OptKnock [41]. One important motivation of the OptKnock method was that growth-coupled strain design can be combined with adaptive laboratory evolution strategies to evolve constructed strains towards their growth-maximal phenotype and, thus, also towards maximal product synthesis [42]. With that, it was considered to be sufficient to demand GCP only for the growth-optimal flux states, while flux distributions with sub-optimal growth can be disregarded. OptKnock implemented this through the nested (bilevel) maximization approach shown in eq. (2.32). However, strain designs found by OptKnock do not guarantee the exploitation of that production potential and solutions may exist where maximal growth is also possible without any product synthesis due to the presence of alternate optimal solutions. Such properties can be best analyzed in the production envelope (PE, see section 2.2.2), where the worst-case scenario of OptKnock shows as a vertical line touching the x-axis ($r_P = 0$) at maximum growth rate (Figure 3.1A; orange area). OptKnock served as a starting point for the development of various related methods [43, 44, 46, 48, 90, 92, 93, 115, 116]. Common to many successors of OptKnock is that they explicitly enforce some minimum product synthesis at all growth-maximal flux states while GCP for suboptimal growth rates is, as in OptKnock, usually not demanded and may occur only incidentally (Figure 3.1A; blue area). Hence, the coupling of growth with product synthesis is in most cases still only a local property and does not include the entirety of possible growth rates. Because these bilevel optimization approaches assume that maximal growth rates are attained by the mutant strains (possibly after adaptive laboratory evolution), they are sometimes called “biased” strain design methods [58].

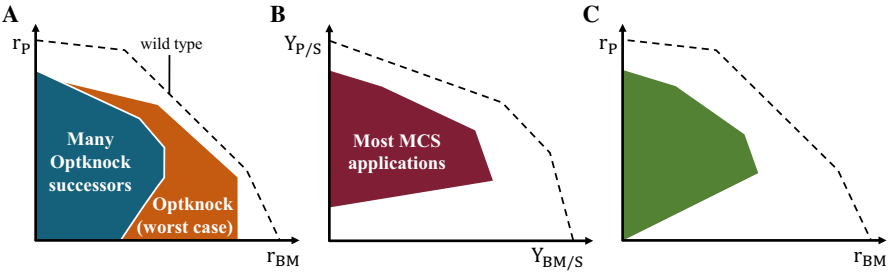


Figure 3.1: Characteristic production envelopes or yield spaces of different computational methods for growth-coupled strain design. (**A**) Typical production envelopes of biased (bilevel) optimization techniques. In the worst case, the original method OptKnock (orange) may contain flux vectors with no product synthesis at maximal growth rate which is avoided by successors of OptKnock (blue). (**B**) Typical yield space of a strain design computed with MCS, demanding a minimum product yield for all flux states. (**C**) Production envelope of a strain design with a fixed minimum ratio of product synthesis and growth rate. r_{BM}: growth rate; r_P: product synthesis rate; $Y_{P/S}$: product yield; $Y_{BM/S}$: biomass yield.

Alternative approaches developed by Trinh and co-workers [48, 50, 117, 118] as well as the framework of Minimal Cut Set (MCS) [57, 60], sought to establish a more rigid coupling (sometimes also called "strong coupling") by enforcing a certain minimum product yield in all (non-zero) flux states. The two-dimensional biomass-product-yield spaces (YS, see section 2.2.2) of such designed strains have typically a shape as shown in Figure 3.1B. Although, this stronger notion of GCP is commonly used in the context of MCS [31, 60] or for the construction of modular cells [48, 56], it seems to demand more than necessary because production is also enforced in metabolic states without growth. The relationships between computed bilevel and the strongly growth-coupled strain designs have not been studied so far.

An intermediate variant of GCP with a coupling strength between classical bilevel and strong coupling approaches can be defined by demanding a minimum ratio of product synthesis rate and growth rate. The flux spaces of the corresponding strain designs appear as areas in the PE that are entirely above a line with a certain slope (Figure 3.1C). This definition of GCP was proposed in few theoretical works, partially under different names [46, 87, 119], however, so far, it has not been used in any application and deserves further attention.

In order to characterize the different types and strengths of GCP approaches, many studies introduced attributes such as weak [46–48, 119], maximal [41], holistic [119], tight [50, 117], directional [46], partial and full [58] or strong coupling [47, 48, 119]. However, this resulted in often inconsistent or imprecise terminology in the literature, caused by the use of different mathematical approaches and their respective reference points (e.g., production envelope vs. yield space, bilevel optimization vs. MCS etc.). Occasionally, different names have been coined for equivalent definitions due to parallel developments. Moreover, some definitions appear not conclusive enough to characterize or/and distinguish different types of coupling. For example, the PE shown in Figure 3.1C may also be exhibited by a strain that has the stronger coupling

degree depicted in Figure 3.1B, hence, Figure 3.1C alone is not sufficient to distinguish these two coupling types.

This chapter systematizes and unifies the different definitions of (the degree of) GCP to obtain one consistent ontology. We will distinguish four major types of coupling, based on unambiguous definitions. We will furthermore show that the MCS framework can be used to compute strain designs for all four coupling degrees. In particular, we will present extensions of the MCS approach that now also enable the computation of classical bilevel strain designs via MCS by incorporating implicit constraints for growth-rate optimality. We then compare strain designs computed for 12 (native and heterologous) products in *Escherichia coli* for all four coupling types with respect to the number of required interventions and computation time. Finally, we also discuss a generalization of coupled production and discuss ATP-coupled product synthesis, which is relevant for other metabolic engineering strategies.

3.1 Definition of four growth-coupling degrees

The key property of GCP is the dependence of cellular growth on the synthesis of a product of interest. We propose four different degrees of this relationship, which are summarized in Figure 3.2 and detailed below.

The first coupling degree is called *potentially growth-coupled production* (pGCP) and occurs when there is *potential for product synthesis* at growth-maximal flux states. Using @ as abbreviation for "at", we express this condition by

$$r_P^{max@r_{BM}^{max}} > 0. \tag{3.1}$$

pGCP formulates the mildest condition of all coupling degrees, and OptKnock is the most popular method for computing pGCP strains designs. In contrast to the other three coupling degrees to follow, pGCP does not ensure a strict dependency between growth and production. However, it indicates that product synthesis *does not oppose* to the biological objective of growth maximization.

Clearly, definition (3.1) is only meaningful if there is at least one flux vector with a non-zero growth rate, and in the following we assume that this is always fulfilled in the system under study. Figure 3.2 provides examples of interventions that induce the respective coupling degrees in a given example network. For this network, we assume that substrate uptake is the only reaction with an upper bound for its flux and that P is our desired product. Biomass is here represented by an essential biomass precursor BM. In this network, pGCP can be induced through a single knockout: removing the reaction from A to BM ensures that the pathway with BM-optimal but product-free operation is inactive. There remain now two alternative pathways with maximal growth rates, one has P and the other Q as by-product. It is possible to shift metabolic flux between both pathways and produce the product P or Q, without a decrease in the maximum attainable growth rate.

Generally, whether a certain growth-coupling degree is prevalent in a metabolic network or not can be tested through simple flux optimizations (FBA) and yield optimizations or, alternatively, graphically from the PE and YS as shown in Figure 3.2. Regarding pGCP, its presence can be tested via FBA by first maximizing the growth rate followed by a second maximization of the production rate, with the growth rate constrained at its maximum. pGCP is present when the production rate in the second optimization takes a strictly positive value. In the PE, for pGCP it is required that there is at least one point with maximal growth rate that lies above the x-axis. In our example, we have a vertical line at maximal growth starting from the x-axis (zero production of P, see Figure 3.2), indicating that maximal growth may coincide with simultaneous production but that alternative routes with maximal BM synthesis but without production of P do exist. Importantly, even though, in the example, the YS for pGCP shows the same characteristic shape as the PE, potential coupling cannot generally be deduced from the YS because flux states with maximal growth rate may not exhibit maximal biomass yield (which is at the rightmost position in the YS) [87]. A mismatch occurs when the capacity of pathways with optimal biomass yield is reached due to inhomogeneous constraints (such as a maximal oxygen uptake rate) and suboptimal pathways become active to attain the maximal growth rate. This happens, for example, when overflow metabolism occurs as a consequence of proteome allocation constraints [120].

Next, if the dependence of growth on production is present in *all flux states with maximal growth rate* then we call it *weakly growth-coupled production* (wGCPs). We express this condition with

$$r_P^{min@r_{BM}^{max}} > 0. \tag{3.2}$$

An example of a wGCP design strategy in the toy network is also shown in Figure 3.2. After additionally knocking out the reaction from E to F, the only pathway left to offer the highest growth rate would run via E to BM and this pathway produces P as byproduct as desired. The presence of wGCP in a network can be tested via FBA by first maximizing the growth rate and subsequently minimizing the production rate, with the growth rate constrained at its maximum. If the latter minimization returns a strictly positive value, wGCP is present. Alternatively, wGCP can also be directly inferred from the PE: there must be an edge or a corner at maximal growth that does not touch the horizontal axis where the flux states without production are located (see Figure 3.2 and the blue area in Figure 3.1A). For the same reasons as for pGCP, wGCP cannot generally be deduced from the YS. In principle, as an alternative definition for weak coupling, one could demand some minimum product yield at maximum biomass yield. However, this bears multiple disadvantages. In particular, adaptive laboratory evolution favors strains that strive for the maximal growth rate instead of biomass yield. Furthermore, using a yield-based definition may significantly complicate classical bilevel-based strain design approaches. A rate-based definition of wGCP is hence preferable.

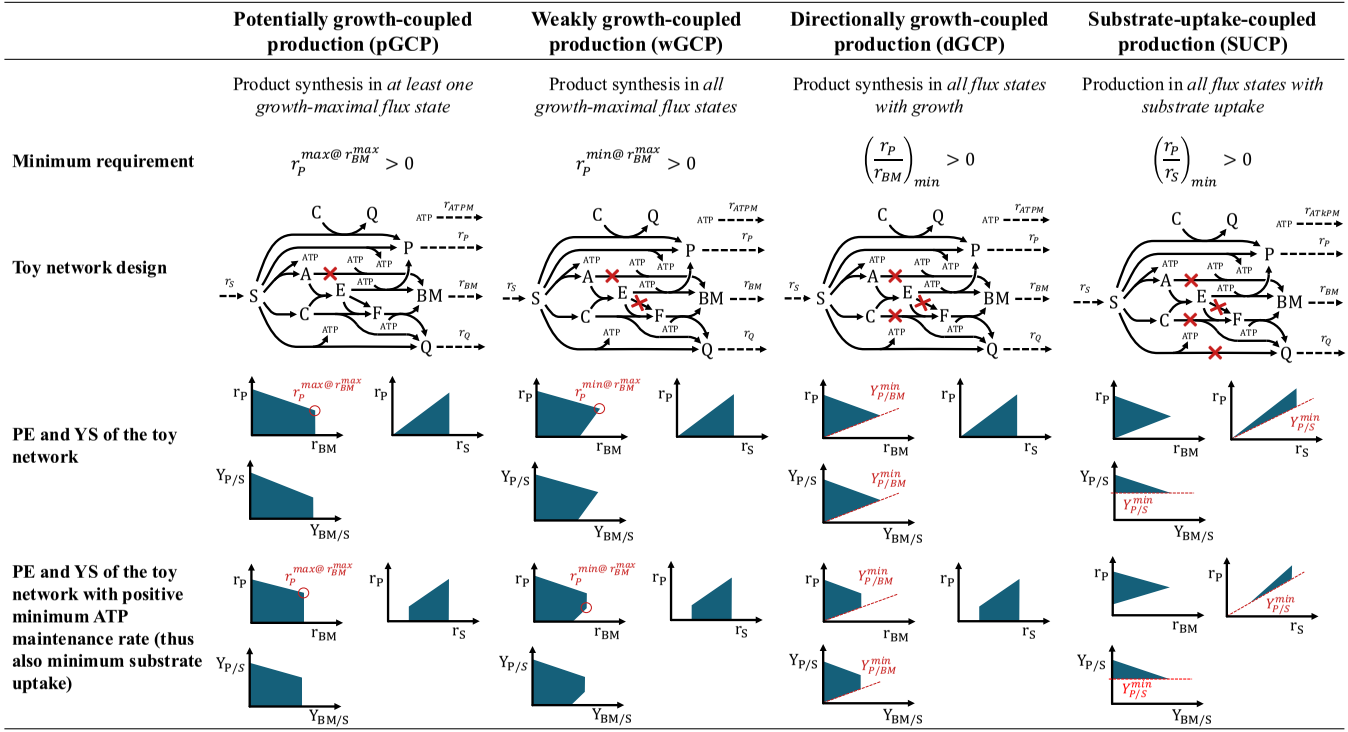


Figure 3.2: Overview of the four different degrees of growth-coupled product synthesis. The red crosses indicate suitable sets of knockouts that induce coupling between biomass (here represented by the biomass precursor BM) and product (P) synthesis with the respective coupling degree. Production envelopes and yield spaces are shown for the example network - with and without the assumption of a minimum ATP maintenance demand constraint (implying a metabolic baseline activity). r_S: substrate uptake rate; r_P: product synthesis rate; r_{BM}: biomass production (growth) rate; r_{ATPM}: rate of ATP consumed for maintenance.

There is a multitude of optimization methods developed for strain design for wGCP. These variants typically rely on bilevel optimization approaches but differ in the optimization of certain aspects under weak coupling. For example, RobustKnock maximizes the lower bound of the product synthesis rate at the state of maximum growth [44], which was alternatively achieved by using the OptKnock formulation with a tilted objective function [43]. More recent approaches, like OptORF variants [46] or OptCouple [93], allow also the definition of a broader range of growth rates for which growth-coupling is enforced. In particular, OptCouple searches for interventions to maximize the distance between (a) the maximal growth rate with guaranteed production and (b) the maximal growth rate where product synthesis is not guaranteed [93]. However, despite their specific features, all these methods compute strain designs that share the minimal property of wGCP as defined in eq. (3.2).

As the third class of growth-coupling strength, we define *directionally growth-coupled production* (dGCP) to be present when growth implies product synthesis for any positive growth rate. Formally, this means

$$Y_{P/BM}^{min} = \left(\frac{r_P}{r_{BM}} \right)_{min} > 0, \tag{3.3}$$

that is, the "yield" of product per biomass, or, more precisely, the ratio of product and biomass synthesis rates, must be strictly positive. By definition, this ratio only exists (and is thus only relevant) for flux vectors with non-zero growth rate, and we therefore demand again that at least one flux vector with non-zero growth rate exists. In the toy network, in addition to the knockouts of the pathways from A to BM and from E to F enforcing weak coupling, the pathway via F to BM also needs to be blocked. It has a reduced growth yield and might therefore not be relevant under optimal growth but directional coupling demands that this pathway is also blocked as it allows growth without product synthesis. dGCP in a network can be ascertained by minimizing the ratio ("yield") of production and growth rate. If the optimal value is greater than zero, dGCP is present. Likewise, dGCP can also be inferred graphically from the PE or YS. In both representations, the entire flux space must be located above a diagonal line that may cross the x-axis only in the point of origin. The slope of this line is the minimum amount of product synthesized per amount of produced biomass. The term "directional coupling" as defined in flux coupling analysis [121–123] designates this coupling type and was therefore also used by Tervo and Reed (2014) [46] in the context of strain design. In contrast, the terms full and partial coupling introduced in flux coupling analysis describe only special cases of the general directional coupling (see Figure 3.3). Potential and weak coupling as defined herein would be regarded as uncoupled in flux coupling analysis because the dependence between growth and product is only a local property.

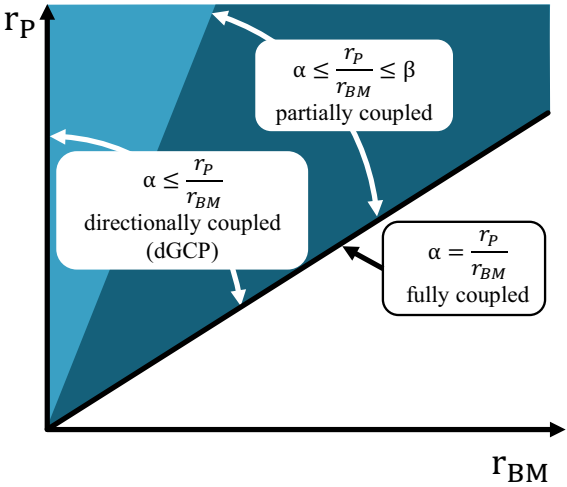


Figure 3.3: Different coupling types as defined in flux coupling analysis (cf. [121]). Directional GCP requires a strictly positive ratio of r_P and r_{BM} in all flux vectors. $\alpha = \frac{r_P}{r_{BM}} > 0$ defines the minimum of this ratio and thus the slope of the outer black line.

We call the fourth and strongest type of coupling *substrate-uptake-coupled production* (SUCP), which demands that product synthesis occurs in all flux states where substrate is taken up. This can be expressed as

$$Y_{P/S}^{min} = \left(\frac{r_P}{r_S}\right)_{min} > 0, \tag{3.4}$$

simply demanding that the yield of product per substrate is guaranteed to be above zero. In reality, this means that product synthesis occurs in all (non-zero) flux states, including those without growth. Even though the growth rate does not appear in condition (3.4), one may regard SUCP as growth-coupled production *provided that growth is feasible*: growth obviously requires substrate uptake, which in turn implies product synthesis due to eq. (3.4). This is also the reason why SUCP has sometimes been called "strong growth-coupling" [31, 47, 48, 56, 119], however, we would like to refrain from this term to make clear that the SUCP condition (3.4) as such *does not require* growth. In fact, condition (3.4) guarantees a minimum product yield even without growth and an SUCP strain design could also be used for two-stage processes with no or only little growth during the production phase (where growth is blocked, for example, by a genetic switch or nitrogen starvation; such a design approach has been called "non-growth production" by Garcia and Trinh (2019) [48]).

To achieve SUCP in the toy network, the route from substrate S to byproduct Q needs to be blocked in addition to the three knockouts of the dGCP as this pathway has a product (P) yield of zero (Figure 3.2). In the remaining network, all possible steady-state flux distributions

will now definitely produce some P. Similar to directional growth-coupling, the presence of SUCP can be confirmed through a product yield minimization. Accordingly, in the YS, the entire flux space is located above a non-zero product yield threshold and does not touch the horizontal axis (corresponding to a product yield of zero) at any point. Importantly, as shown by the example in Figure 3.2, the PE under SUCP may have the same shape as under dGCP and is thus alone not conclusive whether SUCP is present. However, apart from the YS, SUCP can also be identified in the two-dimensional projection where the axes represent substrate uptake rate (instead of growth rate used in the PE) and product synthesis rate. In this phase plane, in terms of flux coupling analysis, SUCP generally occurs as partial instead of directional coupling between r_P and r_S (Figure 3.2; cf. with Figure 3.3) because the latter would imply that substrate-independent production was possible. Furthermore, if a model carries a baseline metabolism that requires a minimum substrate uptake (e.g., due to some non-growth-associated ATP maintenance, often imposed in metabolic models) this in turn results in a minimum production rate (see Figure 3.2, last row).

As shown in the next section, the four equations (3.1) to (3.4) can be formulated with tighter constraints demanding that the product synthesis rates in equations (3.1) and (3.2), or the product-per-biomass yield (eq. (3.3)), or the product-per-substrate yield (eq. (3.4)) must not fall below a certain non-zero threshold. This is indeed often used in the problem setups of strain design methods, likewise, constraints, e.g., for a minimal possible growth rate, are often added. However, with the most relaxed versions (eqs. (3.1) to (3.4)), a clear hierarchical dependency exists between the different degrees of GCP: a network showing wGCP automatically fulfills the criterion for pGCP; if dGCP is present then the criteria of wGCP and pGCP are also naturally fulfilled, while a strain with SUCP satisfies the criteria of the other three types of GCP. In this regard, the GCP type of a particular network (or strain) design is determined by the strongest of the four conditions (3.1) to (3.4) that is fulfilled in the network (for, example, a wGCP strain design fulfills the wGCP condition (3.2) but not the dGCP condition (3.3)). Furthermore, it may happen that strain design solutions may exist for a weaker (e.g., wGCP) but not for a stronger (e.g., dGCP, SUCP) coupling type. In the opposite direction, this hierarchy also implies that methods computing strain designs for wGCP may, for instance, also return solutions that enforce dGCP or even SUCP. These hierarchical relationships will be further discussed in section 3.3.

Since previous studies used different notions and terminology in the context of growth-coupled product synthesis, in Table 3.1 we sought to semantically map these previously used terms to the definitions introduced herein. Most of the previous works distinguished only two major types of GCP. As one exception, Alter and Ebert (2019) [119] considered three different degrees of coupling (weak, holistic, strong) which are related with our definitions of wGCP, dGCP and SUCP, respectively. However, their definitions require the unnecessary assumption

of a metabolic baseline flux and are purely based on the PE, which, as shown for dGCP and SUCP, can be ambiguous.

It should be noted that the four terms potentially/weakly/directionally growth-coupled production and substrate-uptake-coupled production describe properties that are either present or absent in a given (or designed) metabolic network model. However, the coupling terms themselves do not qualify as a benchmark for the actual production performance. For example, strains with directional GCP are not per se better production hosts than strains with weak GCP, nor vice versa. For assessing the suitability and production potential of designed strains, different measures can be used [1, 43, 93, 124, 125].

Table 3.1: Mapping definitions of GCP used in previous studies to the four coupling degrees introduced in this work.

	Potentially growth-coupled production (pGCP)	Weakly growth-coupled production (wGCP)	Directionally growth-coupled production (dGCP)	Substrate-uptake-coupled production (SUCP)
Burgard et al., 2003 [41]	OptKnock / Maximally coupled objectives			
Hädicke and Klamt, 2010 [126]				Full coupling / obligatory coupling
Tepper and Shlomi, 2010 [44]	Optknock	Obligatory byproduct of biomass formation		
Feist et al., 2010 [43]	Optknock / non-unique phenotype	Coupling	Full coupling	
Tervo and Reed, 2014 [46]		Weak coupling	Directional coupling	
Machado and Herrgård, 2015 [58]	OptKnock / no (effective) growth-coupling	Partial coupling	Full coupling	Full coupling
Klamt and Mahadevan, 2015 [47]		Weak coupling		Strong coupling
Garcia and Trinh, 2019, 2020 [48, 56]		Weak coupling (wGCP)		Strong coupling
Alter and Ebert, 2019 [119]	OptKnock	Weak coupling	Holistic coupling	Strong coupling
Vieira et al., 2019 [125]		Weak coupling (MCSw)		MCSf, MCSe

3.2 Computing minimal cut sets for all four coupling degrees

The vast majority of constraint-based metabolic design algorithms use growth-coupled product synthesis as the underlying design principle. However, so far, there is no single computational framework that allows the computation of intervention strategies for all four introduced coupling strengths in genome-scale networks. As already mentioned in the introduction section,

bilevel (biased) optimization approaches naturally focus on (different variants of) pGCP and wGCP designs while MCS-related strain design calculations predominantly demanded SUCP. Surprisingly, explicit computation of dGCP, as a medium coupling strength, has rarely been considered with the two exceptions of purely theoretical studies [46, 87]. In the following, we will explain how the MCS framework can be used to compute metabolic designs for all four types of coupling. This will require an extension of the current MCS problem formulation to also allow the computation of pGCP and wGCP designs in large-scale networks.

As introduced in section 2.3.2, MCSs are defined as minimal (irreducible) sets of interventions that block all undesired and preserve at least some of the desired behaviors. Interventions are typically reaction or gene knockouts, but reaction/gene additions or overexpression [105] can also be considered (see chapter 4). Initially, MCSs were calculated from a given set of undesired and desired elementary modes. In fact, this approach allows the computation of metabolic designs for all four coupling types by selecting appropriate sets of desired and undesired elementary modes (for example, wGCP and SUCP were computed in [57]). However, this approach becomes quickly infeasible in large-scale networks where elementary modes cannot be fully enumerated. As reviewed in section 2.3.2, Farkas-lemma-based approaches allow the computation of shortest MCSs also in genome-scale networks in one step via mixed integer linear programming (MILP), without computation of elementary modes in a preprocessing step [60]. Again, as a common principle for MCS-based methods, desired and undesired (target) behaviors must be specified, which, for the dual approach, is done in the form of linear inequalities, as shown in eqs. (2.33) and (2.37). The computation of MCSs for growth-coupled design thus requires the proper translation of the demands of a particular GCP type into the linear inequality systems $\mathbf{D}\,\mathbf{r} \leq \mathbf{d}$ and $\mathbf{T}\,\mathbf{r} \leq \mathbf{t}$ that describe the desired and undesired flux states, respectively. For the MCS computation, $\mathbf{T}, \mathbf{t}$ and $\mathbf{D}, \mathbf{d}$ are inserted in the MCS MILP as shown in section 2.3.2 (eq. (2.47)).

3.2.1 MCS setup for substrate-uptake-coupled production (SUCP) and directionally growth-coupled production (dGCP)

For the case of SUCP, the undesired flux states are straightforward to describe. According to eq. (3.4), the MCSs need to block all flux solutions that have a positive substrate uptake rate but a zero product synthesis rate, i.e. a zero product yield:

$$\frac{r_P}{r_S} = 0. \tag{3.5}$$

The yield term (eq. (3.4)) is actually non-linear and is only defined for strictly positive substrate uptake. To integrate this constraint in the MCS framework, it needs to be linearized with two linear inequalities

$$r_P = 0 \tag{3.6}$$

$$r_S > 0. \tag{3.7}$$

Strict inequalities as in eq. (3.7) cannot be used in linear programming, and so we replace eq. (3.7) with:

$$r_S \geq \varepsilon, \tag{3.8}$$

with a sufficiently small number $\varepsilon > 0$ as a lower threshold for r_S. Due to numerical tolerances used by most solvers, too small numbers for ε should be avoided, otherwise the MCS algorithm will seek to also eliminate flux vectors of zero growth with no production. Importantly, if a metabolic network has some basal activity (e.g. due to a non-zero ATP maintenance demand), eqs. (3.7) and (3.8) are automatically fulfilled and can be dropped, hence, only eq. (3.6) would remain.

As was already mentioned before, stronger constraints for the product yield are often demanded by specifying a minimum product yield threshold $Y_{P/S}^{Target} > 0$ for the right-hand side in eq. (3.4). In this case, eq. (3.5) would read

$$\frac{r_P}{r_S} \leq Y_{P/S}^{Target}, \tag{3.9}$$

which, under the constraint eq. (3.7) (or eq. (3.8)), can be safely rewritten to

$$r_P - Y_{P/S}^{Target} r_S \leq 0. \tag{3.10}$$

As desired (protected) region for SUCP, we demand that growth with a given threshold for a (desired) minimal growth rate should be possible:

$$r_{BM} \geq r_{BM}^{Desired}. \tag{3.11}$$

By demanding eq. (3.11) we do not need to explicitly demand the fulfillment of eq. (3.4) because the computed MCSs will anyway ensure that all flux vectors with zero product yield (as defined by eqs. (3.6) and (3.8)) will be blocked, hence, if there remains a feasible flux vector obeying eq. (3.11), it will automatically satisfy eq. (3.4). In many applications (also shown in Figure 2.3), a minimal biomass yield is used. This case can be treated similar as in eqs. (3.9) and (3.10). To summarize, as also graphically illustrated in Figure 3.4, an MCS problem for enforcing SUCP is defined by the undesired flux vectors described by eq. (3.6) or (3.10) and eq. (3.8) and the desired flux vectors specified by eq. (3.11).

The procedure for directional GCP is analogous to SUCP (Figure 3.4). According to condition (3.3), for dGCP, all flux states with growth but zero product synthesis must be blocked. Equations (3.5) and (3.7) now translate to:

$$r_P = 0 \tag{3.12}$$

$$r_{BM} \geq \varepsilon. \tag{3.13}$$

As a more general version, we can again replace eq. (3.12) with a threshold for the "product-per-biomass yield":

$$r_P - Y_{P/BM}^{Target} r_{BM} \leq 0. \tag{3.14}$$

For the protected flux states, we can again use eq. (3.11).

3.2.2 MCS setup for weakly growth-coupled production (wGCP)

For the MCS computation of wGCP strain designs, we may initially proceed in the same way as for dGCP and SUCP. As the desired system for wGCP, defined by $\mathbf{D}$ and $\mathbf{d}$, we again choose a minimal demanded growth rate (see eq. (3.11)). To formulate the undesired space, we use the coupling condition for wGCP in eq. (3.2) analogous to SUCP and dGCP. For the target flux vectors $\mathbf{T}$ and $\mathbf{t}$ we retrieve the following two inequalities

$$r_P = 0 \tag{3.15}$$

$$r_{BM} = r_{BM}^{max}, \tag{3.16}$$

which demand that all flux vectors need to be blocked that simultaneously have no production and maximal growth. The problem here is that r_{BM}^{max}, in turn, depends on the knockouts yet to be identified by the MCS algorithm and can thus not be defined a priori. This cyclic dependency can be avoided by implicitly demanding growth optimality in the MCS problem formulation. In the following, we will explain how strong LP duality can be applied to translate eq. (3.16) into a set of inequality constraints that may be used with the MCS framework. Importantly, as before, the interventions introduced by the optimization problem must be considered, in the actual target system but now also in the dual LP system to account for growth-optimality in the modified network.

Concretely, the target system for wGCP is constructed as follows: First, we demand the standard metabolic network constraints as usual: $\mathbf{G}\,\mathbf{r} \leq \mathbf{g}$ (cf. chapter 2, eq. (2.30)). Second, the target constraints are formulated via $\mathbf{T}\,\mathbf{r} \leq \mathbf{t}$. For wGCP, this is used to specify the undesired behavior $r_p = 0$ (no production). Finally, the constraint of growth-optimality (eq. (3.16)) in the target system is formulated via strong duality. For this purpose, we consider the primal LP system $\mathbf{G}\,\mathbf{r} \leq \mathbf{g}$ with the objective function (maximize $\mathbf{c}^\intercal\mathbf{r}$), demanding growth rate optimality. According to eq. (2.4), the dual LP system reads:

$$\begin{aligned} &\text{minimize} && \mathbf{g}^\intercal\mathbf{y} \\ &\text{subject to} && \mathbf{G}^\intercal\mathbf{y} = \mathbf{c} \\ & && \mathbf{y} \geq \mathbf{0}. \end{aligned} \tag{3.17}$$

Due to (strong) duality (eqs. (2.6) and (2.7)), introducing the constraint $\mathbf{c}^\intercal\mathbf{r} \geq \mathbf{g}^\intercal\mathbf{y}$ (or equivalently $\mathbf{g}^\intercal\mathbf{y} - \mathbf{c}^\intercal\mathbf{r} \leq \mathbf{0}$) ensures the demanded optimality of the growth rate. Taken together, the target system reads

$$\mathbf{A_T}\,\mathbf{x} \leq \mathbf{b_T}$$

$$\mathbf{A_T} = \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{T} & \mathbf{0} \\ -\mathbf{c}^\intercal & \mathbf{g}^\intercal \\ \mathbf{0} & \mathbf{G}^\intercal \\ \mathbf{0} & -\mathbf{G}^\intercal \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \mathbf{y} \end{bmatrix}, \qquad \mathbf{g} = \begin{bmatrix} \mathbf{g} \\ \mathbf{t} \\ 0 \\ \mathbf{c} \\ -\mathbf{c} \end{bmatrix}, \qquad \mathbf{y} \geq \mathbf{0}. \tag{3.18}$$

The equality $\mathbf{G}^\intercal\,\mathbf{y} = \mathbf{c}$ has been split up into two inequalities $\mathbf{G}^\intercal\,\mathbf{y} \leq \mathbf{c}$ and $-\mathbf{G}^\intercal\,\mathbf{y} \leq -\mathbf{c}$ to meet the general (inequality) form of target systems.

Before we create the Farkas-dual of eq. (3.18) and prepare it for MCS computations, we condense the system and add trivial constraints for the knockouts of all reactions ($\mathbf{I_{KO}}\,\mathbf{r} = \mathbf{0}$) in the original system. These knockouts must also be introduced in the dual LP system to correctly describe the growth-optimal state under these interventions. We therefore obtain:

$$\begin{bmatrix} \mathbf{G} & \mathbf{0} & \mathbf{0} \\ \mathbf{T} & \mathbf{0} & \mathbf{0} \\ -\mathbf{c}^\intercal & \mathbf{g}^\intercal & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^\intercal & \mathbf{I_{KO}} \\ \mathbf{I_{KO}} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{y} \\ \mathbf{s} \end{bmatrix} \begin{matrix} \leq \\ \leq \\ \leq \\ = \\ = \end{matrix} \begin{bmatrix} \mathbf{g} \\ \mathbf{t} \\ 0 \\ \mathbf{c} \\ \mathbf{0} \end{bmatrix} \tag{3.19}$$

$$\mathbf{y} \geq \mathbf{0}.$$

The second $\mathbf{I_{KO}}$ (third column) and the associated variables $\mathbf{s}$ ensure that the reactions are also "removed" in the dual LP system. To compute the MCSs, we use the Farkas-dual of the target system (eq. (3.19)):

$$\begin{bmatrix} \mathbf{G}^\intercal & \mathbf{T}^\intercal & -\mathbf{c} & \mathbf{0} & \mathbf{I_{KO}} \\ \mathbf{0} & \mathbf{0} & \mathbf{g} & \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I_{KO}} & \mathbf{0} \\ \mathbf{g}^\intercal & \mathbf{t}^\intercal & \mathbf{0} & \mathbf{c}^\intercal & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \\ h \\ \mathbf{q} \\ \mathbf{v} \end{bmatrix} \begin{matrix} = \\ \geq \\ = \\ < \end{matrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ 0 \end{bmatrix} \tag{3.20}$$

$$\mathbf{u} \geq \mathbf{0}, \quad \mathbf{w} \geq \mathbf{0}, \quad h \geq 0.$$

As before, the knockouts of the computed MCSs will be represented by the nonzero values in $\mathbf{v}$, which can again be represented through binary decision variables z_i and indicator constraints. It is also important to keep in mind that the eventually introduced knockouts in the original and the dual LP system must be identical. The knockouts z_i introduced in the network are linked, therefore, not only to the target system by $z_i = 0 \rightarrow v_i = 0$ but also affect the dual inequality system by $z_i = 1 \rightarrow q_i = 0$.

Adding the desired region, replacing the strict "< 0" inequality with "≤ -1" and minimizing the number of interventions, we obtain the following MILP for computing MCSs that enforce wGCP:

$$\text{minimize } \sum z_i$$

$$\text{subject to}$$

$$\begin{bmatrix} \mathbf{G}^\intercal & \mathbf{T}^\intercal & -\mathbf{c} & \mathbf{0} & \mathbf{I_{KO}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{g} & \mathbf{G} & \mathbf{0} & \mathbf{0} \\ \mathbf{g}^\intercal & \mathbf{t}^\intercal & \mathbf{0} & \mathbf{c}^\intercal & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \\ h \\ \mathbf{q} \\ \mathbf{v} \\ \mathbf{r} \end{bmatrix} \begin{matrix} = \\ = \\ \geq \\ \leq \\ \leq \\ \leq \end{matrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ -1 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{3.21}$$

$$\forall i : z_i = 0 \rightarrow v_i = 0$$

$$\forall i : z_i = 1 \rightarrow q_i = 0$$

$$\forall i : z_i = 1 \rightarrow r_i = 0$$

$$\mathbf{u} \geq \mathbf{0}, \quad \mathbf{w} \geq \mathbf{0}, \quad h \geq 0.$$

As can be seen, the $\mathbf{I_{KO}}$ matrix that represented the (permanent) knockouts in the dual LP system (associated with the variables $\mathbf{q}$) can be removed, as the knockouts in $\mathbf{q}$ are now globally linked to the binary variables z_i.

Resubstituting all submatrices of $\mathbf{G}$ and condensing columns/rows, we obtain the system:

$$\text{minimize } \sum z_i$$

$$\text{subject to}$$

$$\begin{bmatrix} \mathbf{N}^\intercal & -\mathbf{I} & \mathbf{I} & \mathbf{T}^\intercal & -\mathbf{c} & \mathbf{0} & \mathbf{I_{KO}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{N} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{lb} & -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{ub} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{lb}^\intercal & \mathbf{ub}^\intercal & \mathbf{t}^\intercal & \mathbf{0} & \mathbf{c}^\intercal & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{N} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{u_r} \\ \mathbf{u_{lb}} \\ \mathbf{u_{ub}} \\ \mathbf{w_t} \\ h \\ \mathbf{q} \\ \mathbf{v} \\ \mathbf{r} \end{bmatrix} \begin{matrix} = \\ = \\ \geq \\ \geq \\ \leq \\ = \\ \leq \\ \leq \\ \leq \end{matrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ -1 \\ \mathbf{0} \\ -\mathbf{lb} \\ \mathbf{ub} \\ \mathbf{d} \end{bmatrix} \tag{3.22}$$

$$\forall i : z_i = 0 \rightarrow v_i = 0$$

$$\forall i : z_i = 1 \rightarrow q_i = 0$$

$$\forall i : z_i = 1 \rightarrow r_i = 0$$

$$\mathbf{u_{lb}}, \mathbf{u_{ub}} \geq \mathbf{0}, \quad \mathbf{w_t} \geq \mathbf{0}, \quad h \geq 0, \quad z_i \in \{0,1\}.$$

An alternative (more compact) formulation of the MILP (3.21) can again be achieved by omitting the slack variables **v**:

$$\begin{gathered}
\text{minimize } \sum z_i \\
\text{subject to} \\
\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{g} & \mathbf{G} & \mathbf{0} \\ \mathbf{g}^\mathsf{T} & \mathbf{t}^\mathsf{T} & \mathbf{0} & \mathbf{c}^\mathsf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \\ h \\ \mathbf{q} \\ \mathbf{r} \end{bmatrix} \begin{matrix} \geq \\ \leq \\ \leq \\ \leq \end{matrix} \begin{bmatrix} \mathbf{0} \\ -1 \\ \mathbf{g} \\ \mathbf{d} \end{bmatrix} \\
\forall i : z_i = 0 \rightarrow \begin{bmatrix} \mathbf{G}_i^\mathsf{T} & \mathbf{T}_i^\mathsf{T} & -c_i \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \\ h \end{bmatrix} = 0 \\
\forall i : z_i = 1 \rightarrow q_i = 0 \\
\forall i : z_i = 1 \rightarrow r_i = 0 \\
\mathbf{u} \geq \mathbf{0}, \quad \mathbf{w} \geq \mathbf{0}, \quad h \geq 0, \quad z_i \in \{0,1\}.
\end{gathered} \tag{3.23}$$

The resulting MCS setup for wGCP (also for the more general case with thresholds) is visualized in Figure 3.4.

3.2.3 MCS setup for potentially growth-coupled production (pGCP) and OptKnock-like strain designs

pGCP demands that production is *possible* in (some) flux states of maximal growth. As in the previous cases, the formal coupling definition (here eq. (3.1)) is the starting point for specifying the MCS problem for pGCP. As a peculiarity, it turns out that pGCP does not require the formulation of an undesired behavior and only needs the specification of a *desired* behavior. The latter is expressed by three constraints (Figure 3.4): we demand (1) that the considered flux states are at maximum growth rate

$$r_{BM} = r_{BM}^{max}, \tag{3.24}$$

that (2) this rate must lie above a minimum growth rate threshold (eq. (3.11))

$$r_{BM} \geq r_{BM}^{Desired}, \tag{3.25}$$

and that (3) the product synthesis rate is non-zero

$$r_P > 0. \tag{3.26}$$

The strict inequality in eq. (3.26) can be approximated with a sufficiently small number $\varepsilon > 0$:

$$r_P \geq \varepsilon. \tag{3.27}$$

The two constraints (3.25) and (3.27) (and possibly others) are again integrated in the matrix form $\mathbf{D}\,\mathbf{r} \leq \mathbf{d}$. We formalize the third constraint ($r_{BM} = r_{BM}^{max}$) using strong LP duality, analogous to the wGCP case. The following inequality system describes the desired flux space:

$$\begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{D} & \mathbf{0} \\ -\mathbf{c}^\mathsf{T} & \mathbf{g}^\mathsf{T} \\ \mathbf{0} & \mathbf{G}^\mathsf{T} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{y} \end{bmatrix} \begin{matrix} \leq \\ \leq \\ \leq \\ = \end{matrix} \begin{bmatrix} \mathbf{g} \\ \mathbf{d} \\ 0 \\ \mathbf{c} \end{bmatrix} \qquad \mathbf{y} \geq \mathbf{0}. \tag{3.28}$$

Interestingly, a target region is not needed to search for MCSs enforcing pGCP (Figure 3.4). In contrast to the other coupling degrees, the desired system in pGCP is in most cases initially infeasible (as there is normally no production at maximal growth) and with the optimization problem formulated below we thus look for a minimal set of interventions that make the desired system feasible. Since we only need a desired and no target system, Farkas dualization is not required, however, to properly account for the knockouts (eventually rendering the desired system feasible) we need to introduce a knockout matrix $\mathbf{I_{KO}}$ with corresponding (slack) variables $\mathbf{s}$. The activity of the latter is again indicated by binary variables $\mathbf{z}$ marking the introduced knockouts. The resulting MILP reads as follows:

$$\begin{gathered} \text{minimize } \sum z_i \\ \text{subject to} \\ \begin{bmatrix} \mathbf{G} & \mathbf{0} & \mathbf{0} \\ \mathbf{D} & \mathbf{0} & \mathbf{0} \\ -\mathbf{c}^\mathsf{T} & \mathbf{g}^\mathsf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^\mathsf{T} & \mathbf{I_{KO}} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{y} \\ \mathbf{s} \end{bmatrix} \begin{matrix} \leq \\ \leq \\ \leq \\ = \end{matrix} \begin{bmatrix} \mathbf{g} \\ \mathbf{d} \\ 0 \\ \mathbf{c} \end{bmatrix} \\ \forall i : z_i = 1 \rightarrow r_i = 0 \\ \forall i : z_i = 0 \rightarrow s_i = 0 \\ \mathbf{y} \geq \mathbf{0}, \quad z \in \{0,1\}. \end{gathered} \tag{3.29}$$

Again, eq. (3.29) can be condensed by using indicator constraints directly and omitting the slack variables $\mathbf{s}$:

$$\begin{gathered} \text{minimize } \sum z_i \\ \text{subject to} \\ \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{D} & \mathbf{0} \\ -\mathbf{c}^\mathsf{T} & \mathbf{g}^\mathsf{T} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{y} \end{bmatrix} \begin{matrix} \leq \\ \leq \\ \leq \end{matrix} \begin{bmatrix} \mathbf{g} \\ \mathbf{d} \\ 0 \end{bmatrix} \\ \forall i : z_i = 1 \rightarrow r_i = 0 \\ \forall i : z_i = 0 \rightarrow \mathbf{G}_i^\mathsf{T}\mathbf{y} = c_i \\ \mathbf{y} \geq \mathbf{0}, \quad z \in \{0,1\}. \end{gathered} \tag{3.30}$$

The solutions (MCSs) of the MILPs (eqs. (3.29) and (3.30)) deliver strain designs that enforce pGCP. They are closely related to (and could indeed coincide with) the strain designs found by the original OptKnock formulation [41, 85]. However, the MCS MILP still differs from the OptKnock setup in the chosen objective function. While OptKnock maximizes the (potential) production rate, which is a continuous variable, with a limited number of interventions, an MCS MILP typically minimizes the number of interventions required to reach predefined strain design goals. However, the objective function in eq. (3.29) can be easily adapted in this way, yielding an OptKnock equivalent MILP:

$$
\begin{gathered}
\text{maximize } r_P \\
\text{subject to} \\
\begin{bmatrix} \mathbf{G} & \mathbf{0} & \mathbf{0} \\ \mathbf{D} & \mathbf{0} & \mathbf{0} \\ -\mathbf{c}^\mathsf{T} & \mathbf{g}^\mathsf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^\mathsf{T} & \mathbf{I_{KO}} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{y} \\ \mathbf{s} \end{bmatrix} \begin{matrix} \leq \\ \leq \\ \leq \\ = \end{matrix} \begin{bmatrix} \mathbf{g} \\ \mathbf{d} \\ 0 \\ \mathbf{c} \end{bmatrix} \\
\forall i : z_i = 1 \rightarrow r_i = 0 \\
\forall i : z_i = 0 \rightarrow s_i = 0 \\
\sum z_i \leq MaxNoKO \\
\mathbf{y} \geq \mathbf{0}, \quad z \in \{0,1\}.
\end{gathered}
\tag{3.31}
$$

We note that eq. (3.31) is the most general representation of a MILP computing OptKnock solutions, which, in contrast to the original formulation, allows consideration of arbitrary (side) constraints in the metabolic system (represented by $\mathbf{G}\,\mathbf{r} \leq \mathbf{g}$) as well as for the desired fluxes (represented by $\mathbf{D}\,\mathbf{r} \leq \mathbf{d}$).

With the implicit integration of optimality constraints for wGCP and pGCP, all four coupling types can now be handled with the MCS framework as summarized in Figure 3.4 (for the simple conditions (3.1) to (3.4) as well as for the more general case with arbitrary thresholds for production rates and yields). In the application examples in the sections 3.3 and 3.4 we will make use of this scheme to compute and compare MCSs for all four coupling strengths.

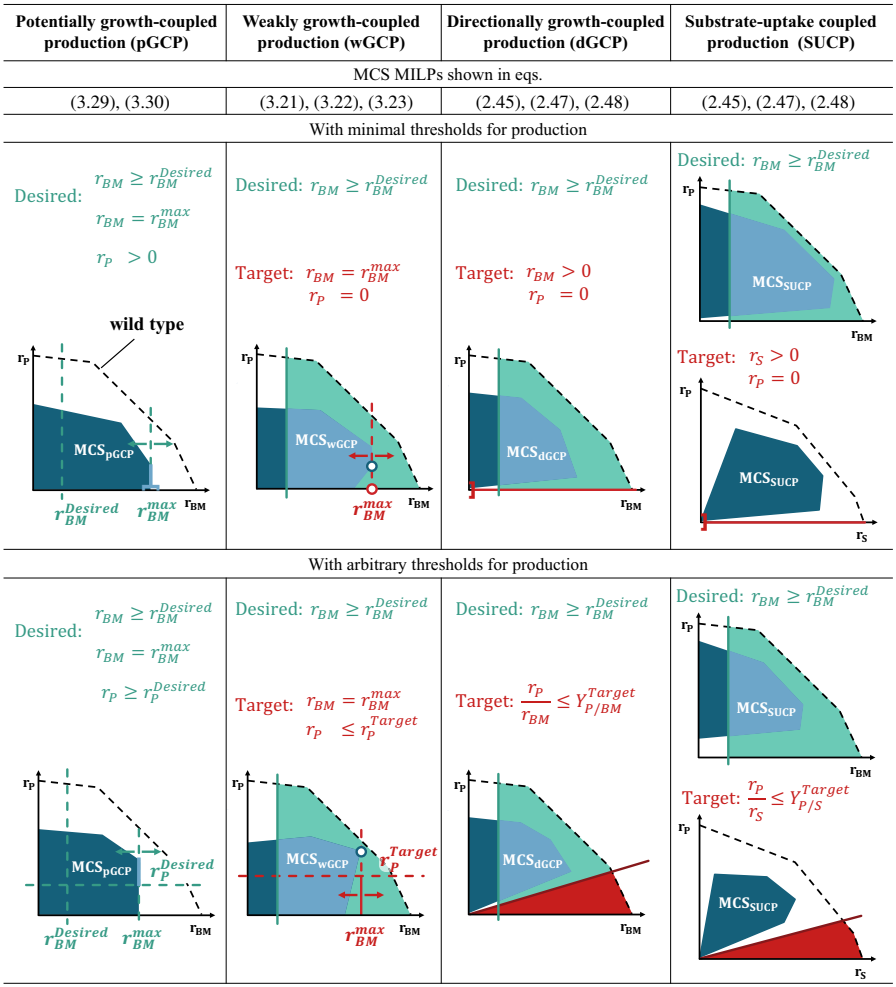


Figure 3.4: The MCS approach can be used to compute strain designs for all four coupling types by specifying suitable desired and undesired flux regions. For the computation of MCSs strain designs inducing pGCP and wGCP, growth optimality in the desired/target regions is implicitly demanded in the optimization problems. Exemplary wild type flux spaces are outlined with a dashed black line, desired flux spaces are shown in green, target flux spaces in red and the designed flux spaces (after implementing the MCSs) in dark blue. The upper row shows the case with minimal coupling requirements and the lower row the most general case with arbitrary thresholds for $r_P^{Desired}$ (pGCP), r_P^{Target} (wGCP), $Y_{BM/S}^{Target}$ (dGCP) and $Y_{P/S}^{Target}$ (SUCP). Note that all inequalities with a strict smaller sign (e.g.,$r_P > 0$) will be approximated by non-strict inequalities (e.g., $r_P \geq \varepsilon$) for the MCS computation.

3.3 Comparing strain designs for the growth-coupled production of ethanol at all coupling degrees

In the first computation, we used an *E. coli* core model and enumerated gene MCS (using gene-protein-reaction associations, see section 4.5) for the GCP of ethanol with all four coupling degrees. The core model was derived from the genome-scale model *i*ML1515 [88] (see section 2.4.2 and Model A.1). For the respective MCS problems, we used the most relaxed formulations (upper row in Figure 3.4; the resulting full problem setups are shown in Table A.1). In order to compute comparable sets of MCSs, we used identical thresholds for the demanded minimum growth rate ($r_{BM}^{Desired} \geq 0.05\,\mathrm{h}^{-1}$) in the desired regions. In this way, MCSs for the four coupling types are expected to exhibit the aforementioned nested hierarchy. In addition to gene knockouts, the MCSs were allowed to block oxygen supply as an intervention. We considered two scenarios (cf. also Figure 3.2): scenario (A), where a minimum ATP demand ($r_{ATPM} \geq 6.86\,\mathrm{mmol\,g_{CDW}^{-1}\,h^{-1}}$) for non-growth-associated ATP maintenance (NGAM) processes was included (which is standard in most models) and scenario (B) without a minimum NGAM demand for ATP ($r_{ATPM} \geq 0$).

For simplicity, Figure 3.5 shows the results for the MCSs up to size 3 for both scenarios. The chosen representation highlights the hierarchical relationships between the MCSs for the different coupling types. This hierarchy implies, for example, that an MCS computed for pGCP might simultaneously fulfill the requirements for the stronger coupling types wGCP, dGCP, or SUCP. Likewise, an MCS for wGCP may also be a valid MCS enforcing dGCP and SUCP and an MCS computed for dGCP may also ensure SUCP. On the other hand, if, for example, an MCS calculated for wGCP does not imply dGCP (or SUCP) then it might be extendable, with additional knockouts to obtain a valid dGCP MCS. Yet, in the reverse direction, every MCS for dGCP is inherited from a wGCP MCS, i.e., it is either identical to or a superset (with additional knockouts) of an MCS found for wGCP. The same hierarchical relationship holds for any pair of coupling degrees. For example, wGCP and pGCP for ethanol synthesis can be established in both scenarios by blocking only the oxygen supply, while for dGCP and SUCP additional knockouts are needed. In fact, with at least one additional knockout (e.g., *ldhA*, the lactate dehydrogenase gene), it is possible to extend the pGCP- and wGCP-related MCS (O2) to an dGCP MCS (O2, *ldhA*). In turn, such an dGCP MCS might also be a valid SUCP MCS (as is the case for (O2, *ldhA*)) and if not, then it might be extendable to an SUCP MCS. For example, in scenario (B) (with zero ATP maintenance demand), the dGCP MCS (O2, *atpS*) is not sufficient for SUCP, but it can be extended with a knockout of the glucose PTS (via the genes of *ptsI* or *ptsH*) to reach SUCP. Interestingly, these two SUCP MCSs do not exist in scenario (A) because here knocking out the PTS would reduce the maximal growth rate below the given minimum threshold due to the posed NGAM demand in scenario (A). For the same reason, there are several dGCP MCSs with two knockouts in scenario (B) which are not applicable for scenario (A). This example shows that the specified minimum NGAM

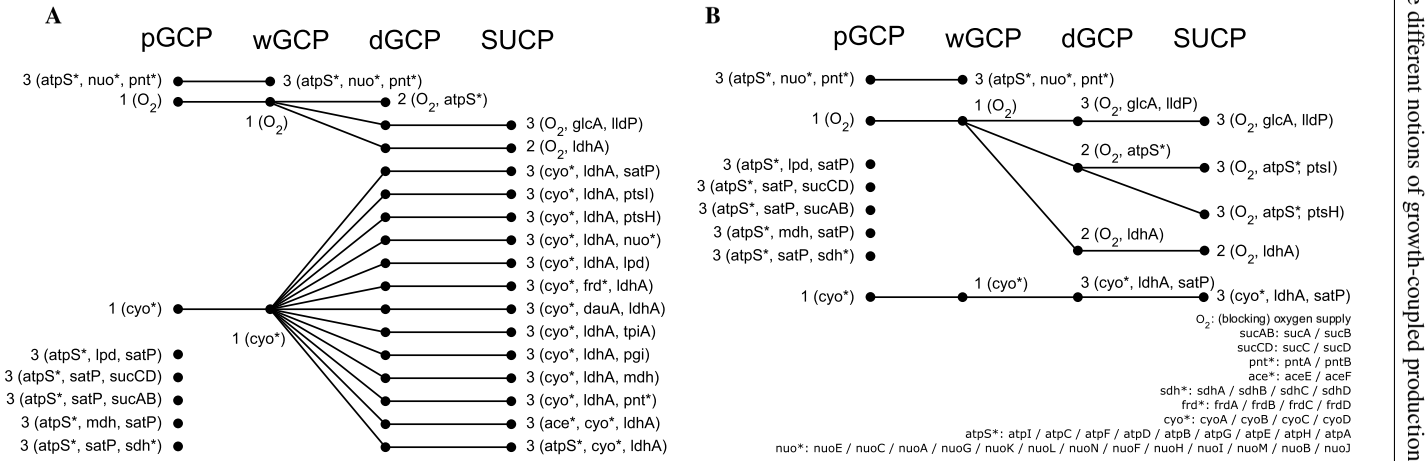


Figure 3.5: Summary of all gene MCSs up to size 3 for growth-coupled production of ethanol in a metabolic core model of *E. coli*, generated from two different scenarios: (**A**) with and (**B**) without a minimum NGAM ATP demand. Each black dot represents an MCS, the numbers indicate the size of the MCSs and the lines the hierarchical relationships between the MCSs of the different coupling degrees.

demand in metabolic models may have a significant effect on the MCSs and thus needs careful consideration.

Except for the single case described above, we found that all other dGCP MCSs are also valid SUCP MCSs in both scenarios. In contrast, not a single pGCP or wGCP MCS was a valid dGCP or even SUCP MCS. Moreover, there is one MCS valid for pGCP and wGCP (*atpS*, *nuo*, *pnt*), which is not extendable to an MCS for dGCP, thus occurring as 'dead end' in Figure 3.5A and B. Likewise, a number of MCSs with three knockouts exist for pGCP that do not reappear, neither identically nor as supersets, for stricter coupling types. Generally, there are three possible reasons why supersets of such MCS may not be found as a solution for stricter coupling. First, supersets may exist but only with a higher cardinality than allowed in the MCS enumeration. Second, knockouts, which would increase the coupling strength, might be prohibited in the computation, as is the case for spontaneous reactions. Lastly, an MCS for stronger coupling may simply not exist due to the network topology and constraints, for example, when a reaction with a strictly positive lower bounded for its rate, like ATP maintenance in many models, cannot be coupled with production.

3.4 Genome-scale calculations for selected products

Next, we applied the extended MCS framework to compute realistic growth-coupled strain designs in a genome-scale model of *E. coli* (*i*ML1515 [88]) for the production of 12 relevant products. The goal was to compare the number of required interventions and the computational effort to determine strain designs for the four different coupling degrees. We considered native, as well as heterologous products. For heterologous products, the required pathways were added to the model (detailed pathways listed in Table A.2). In contrast to the calculations in the previous section, where the strain design only had to exclude zero production, we here demanded higher minimum production thresholds, since predominantly poorly performing strain designs are typically found when using low production minima [93, 127]. We used a reference production rate $r_{P,ref}$ to define the setups for each coupling degree. $r_{P,ref}$ was determined as 20 % of the maximum possible production when the wild type strain, extended with heterologous pathways if applicable, grows at 20 % of its maximal growth rate. For pGCP, we demanded that this reference production rate ($r_{P,ref}$) would be attainable together with the maximum growth rate of the MCS strain. For wGCP, we defined the undesired fluxes as those with production inferior to the threshold $r_P \leq r_{P,ref}$ at maximum growth rate. For dGCP, we demanded a minimum ratio of growth and production rate by targeting all fluxes below the threshold $\frac{r_P}{r_{BM}} \leq \frac{r_{P,ref}}{0.2\ r_{BM,max\ (WT)}}$. Finally, in the case of SUCP we targeted fluxes with $\frac{r_P}{r_S} \leq \frac{r_{P,ref}}{r_{S,max}}$. In all setups, we additionally specified a desired region to guarantee that found strain designs still allow for growth rates greater than $0.05\,\mathrm{h}^{-1}$. For the NGAM demand of ATP, we used $r_{ATPM} \geq 6.86\,\mathrm{mmol\,g_{CDW}^{-1}\,h^{-1}}$. The setups for all computation are shown in Table A.1. The computations were performed with MATLAB 2020b, together with IBM ILOG® CPLEX®

12.10. Each computation was performed on single nodes of a high-performance cluster with two 8-core Intel Xeon Skylake Silver 4110 and 192 GB memory per node.

In contrast to the full enumeration, used in the *E. coli* core model above, we applied the following two procedures for finding single MCSs, which have been implemented in *CellNetAnalyzer* (see chapter 6): In the first computation we aimed to find (quickly) just a single MCS, without the necessity to obtain the MCS with the smallest number of knockouts (Table 3.2), while in the second procedure we searched for an MCS with a low (ideally minimum number) of knockouts (Table 3.3). We emphasize that the performance of these calculations depends on many factors, such as choice of the MILP solver (here CPLEX) and its parameters. The presented results of the computations provide a small sample that only exposes tendencies. To at least reduce dependency of the results on the chosen seed, we performed 12 runs in the first scenario with a runtime limit of 2 h and 6 runs in the second procedure with a limit of 4 h (all with identical solver parameters but starting the computation from different randomly generated seeds) for each combination of product and growth-coupling degree to obtain averaged values for runtimes and MCS sizes (Tables 3.2 and 3.3). The code that was used for the computations is available in the Appendix (see Source Code A.1).

We first analyze the data for the procedure that quickly searches for any MCS (without the necessity to find one with smallest cardinality; (Table 3.2)). With the exception of lysine under wGCP, at least one MCS for all combinations of products and coupling degrees could be found. However, a more differentiated view can be obtained by looking at the number of successful runs finishing before the timeout of two hours, among the 12 different seeds for each product. For every coupling degree, we can find products, where at least one run did not succeed. Furthermore, it can be seen that the mildest (pGCP) and the strongest (SUCP) coupling degree have the highest success rates. For the successful runs, we can see that, in average, SUCP was the fastest followed by pGCP and dGCP (similar) and with wGCP being the slowest. Finally, the computations also show a hierarchy in the sense that the average size of the MCSs increases with coupling strengths. However, it should be noted that this cannot be generalized, since minimum cardinality for the MCS computations was here not demanded and there are indeed cases where the MCSs of a stronger coupling degree have lower average size.

When searching for MCSs with low (ideally minimum) cardinality shown in Table 3.3, we can see that only in relatively few cases MCSs with guaranteed minimum number of knockouts could be found. A minimum solution could be found for all coupling degrees for the products ethanol, isobutanol, 1,4-butanediol and 2,3-BDO. pGCP and SUCP show again a slightly better performance with respect to successful finishing full minimization. On the other hand, dGCP and SUCP calculations were the only ones that could find for each product at least one MCS with (relatively) low number of interventions (e.g., pGCP and wGCP did not find solutions for lysine, wGCP furthermore missed resveratrol). On the other hand, pGCP was the fastest in cases where it finished the computation. wGCP and dGCP again had a higher number of

Table 3.2: MCS computations for designing strains with growth-coupled (pGCP, wGCP, dGCP, SUCP) production of 12 different products. In these calculations, we aimed to find just a single MCS, without the necessity to obtain the smallest one. For each combination of product and growth-coupling degree, 12 computations were started with different seeds and with a time limit of two hours each (details of the computation setups for each run are shown in Table A.1 and the source code is provided in Source Code A.1 in the Appendix). The last row contains the mean values over all scenarios with the exception of lysine where a meaningful comparison is not possible because no MCS could be found for wGCP. Abbreviations: av. size: average number of knockouts per MCS; suc/tot: number of successful computations (that returned an MCS) per total runs; av. runt: average runtime of successful computations; (h) next to product name marks heterologous products.

	pGCP			wGCP			dGCP			SUCP		
Product	av. size	suc/tot	av. runt	av. size	suc/tot	av. runt	av. size	suc/tot	av. runt	av. size	suc/tot	av. runt
Ethanol	1.0	12/12	4 min	1.0	12/12	4 min	2.6	11/12	4 min	4.3	12/12	4 min
Lysine	13.0	2/12	10 min		0/12	timeout	13.0	1/12	9 min	25.1	8/12	14 min
Glutamate	4.5	12/12	4 min	7.0	4/12	5 min	12.0	1/12	7 min	17.3	8/12	6 min
Isobutanol (h)	7.2	10/12	5 min	4.9	7/12	8 min	7.3	6/12	13 min	10.1	12/12	4 min
1,4-BDO (h)	5.4	9/12	18 min	4.5	4/12	4 min	10.8	5/12	4 min	8.3	12/12	5 min
2,3-BDO (h)	6.3	12/12	4 min	7.2	10/12	8 min	8.9	7/12	19 min	8.4	12/12	4 min
Itaconic acid (h)	5.6	12/12	4 min	9.0	3/12	9 min	9.9	8/12	10 min	12.8	12/12	4 min
Isoprene (h)	12.1	9/12	14 min	11.3	3/12	41 min	10.7	3/12	13 min	19.9	8/12	11 min
Butane (h)	6.5	10/12	11 min	6.3	4/12	7 min	11.0	4/12	5 min	24.0	3/12	14 min
Methacrylic acid (h)	8.2	12/12	4 min	12.0	5/12	7 min	10.3	3/12	5 min	10.6	12/12	6 min
Resveratrol (h)	8.5	11/12	10 min	9.0	2/12	60 min	14.2	5/12	22 min	18.8	6/12	8 min
Bisabolene (h)	10.9	8/12	20 min	10.9	7/12	26 min	11.8	5/12	4 min	12.6	11/12	10 min
Mean (all rows except lysine)	**6.9**	**10.6/12**	**9 min**	**7.6**	**5.5/12**	**16 min**	**10.5**	**5.3/12**	**10 min**	**13.4**	**9.8/12**	**7 min**

Table 3.3: MCS computations for designing strains with growth-coupled (pGCP, wGCP, dGCP, SUCP) production of 12 different products. In these calculations, we aimed to find a single MCS with a low number of knockouts (ideally with the minimum number of knockouts). For each combination of product and growth-coupling strength, six computations were started with different seeds and with a time limit of four hours each (details of the computation setups for each run are shown in Table A.1 and the source code is provided in Source Code A.1 in the Appendix). The last row contains the mean values of all scenarios except lysine and resveratrol, for which pGCP (and wGCP) MCSs could not be found. Due to the large number of timeouts, the runtimes are not considered. Abbreviations: m. size: minimum size of all found MCSs (smallest MCSs); opt/suc/tot: number of computations that returned the smallest MCS / number of computations that returned an MCS without guaranteeing minimality / total runs; av. runt: average runtime of computations with assured smallest MCSs; (h) next to product name marks heterologous products.

	pGCP			wGCP			dGCP			SUCP		
Product	m.size	opt/suc/tot	av.runt	m.size	opt/suc/tot	av.runt	m.size	opt/suc/tot	av.runt	m.size	opt/suc/tot	av.runt
Ethanol	1	6/6/6	4 min	1	6/6/6	4 min	2	6/6/6	5 min	3	6/6/6	5 min
Lysine		0/0/6	timeout		0/0/6	timeout	11	0/4/6	timeout	11	0/6/6	timeout
Glutamate	3	6/6/6	38 min	7	0/1/6	timeout	8	0/2/6	timeout	8	0/6/6	timeout
Isobutanol (h)	2	6/6/6	6 min	4	1/2/6	3 h	4	2/2/6	3 h	5	2/6/6	4 h
1,4-BDO (h)	2	6/6/6	7 min	2	6/6/6	7 min	4	6/6/6	2 h	5	3/6/6	3 h
2,3-BDO (h)	3	6/6/6	31 min	4	1/2/6	86 min	4	2/6/6	77 min	4	6/6/6	30 min
Itaconic acid (h)	3	6/6/6	14 min	7	0/1/6	timeout	8	0/3/6	timeout	8	0/6/6	timeout
Isoprene (h)	6	0/4/6	timeout	6	0/2/6	timeout	5	0/5/6	timeout	7	0/6/6	timeout
Butane (h)	5	0/5/6	timeout	5	0/1/6	timeout	5	0/6/6	timeout	5	2/6/6	4 h
Methacr. acid (h)	3	6/6/6	33 min	6	0/2/6	timeout	6	0/5/6	timeout	6	0/6/6	timeout
Resveratrol (h)		0/0/6	timeout	7	0/2/6	timeout	8	0/4/6	timeout	8	0/6/6	timeout
Bisabolene (h)	6	0/4/6	timeout	5	0/5/6	timeout	6	0/6/6	timeout	7	0/6/6	timeout
Mean (all rows except lysine and resveratr.)	**3.4**	**4.2/5.5/6**		**4.7**	**1.4/2.8/6**		**5.2**	**1.6/4.7/6**		**5.8**	**1.9/6/6**	

timeouts (unsuccessful runs). In cases where the minimum MCS was found the monotone increase in MCS size with stronger coupling degrees can again be observed (but notice the larger MCSs for isoprene for pGCP and wGCP (6) compared to dGCP (5) which occurs because the minimization could not be completed in those cases).

3.5 Generalization of coupled production: from growth-coupled to ATP-coupled production

So far, we focused on growth-coupled production of a target metabolite, the most common principle for strain design. In the following, we will show that this coupling principle can be generalized to other biological functions than growth. One relevant example is to couple ATP synthesis with product formation. Such a coupling might be particularly relevant for the idea of enforced ATP wasting as a metabolic engineering strategy, which received increased attention in recent literature [62, 128–134]. The idea is that introduction of an ATP wasting mechanism, such as artificial futile cycles [128, 129, 132] or, more directly, via the ATP-hydrolyzing F_1-portion of the ATPase [62, 134, 135], may boost substrate uptake and product synthesis if ATP production is coupled with the synthesis of the target metabolite. This effect could be harnessed for improving the performance of two-stage processes [136].

It is now straightforward to consider the same four coupling strengths as for growth-coupled production. Potentially ATP-coupled production (pACP) would demand that product synthesis is possible under maximal ATP production. Weakly ATP-coupled production (wACP) means that product synthesis is mandatory under maximal ATP production. Directionally ATP-coupled production requires product synthesis whenever ATP is generated and substrate-uptake-coupled production again implies product formation whenever substrate is taken up. For meaningful definitions, we would demand in all four cases that ATP production is feasible in the network (similar to growth in the growth-coupled case). ATP-coupled strain designs can then be calculated in an analogous way to growth-coupled strain designs, e.g., via MCS, by replacing terms with the growth rate with terms for ATP production. Net ATP production (and consumption) can be simulated via the NGAM reaction (r_{ATPM}) included in most models.

While this generalization can directly be applied, as it could be for coupling any other flux with product formation, the case of ATP synthesis requires special attention. While the consideration of pGCP and wGCP is meaningful in the context of (laboratory) evolution, it is less reasonable to assume that the cell strives to maximize ATP synthesis for non-growth-associated processes. For this reason, we will not further consider pACP and wACP. Second, under the premise that feasibility of growth and net ATP synthesis is ensured, the definition of SUCP is identical for growth- and ATP coupling as it implies product synthesis when substrate is taken up, which is a requirement for growth and ATP synthesis. This has some important consequences. In particular, the existence of SUCP-like growth-coupled strain designs proven for a wide range of products and production hosts in [31] now also implies wide feasibility of

ATP-coupled strain designs under SUCP because a minimum ATP maintenance constraint for the r_{ATPM} reaction was used in these calculations.

For the reasons given above, we need not consider pACP, wACP and SUCP for ATP-coupled product synthesis and focus now on directional ACP. Generally, since ATP is required for growth, dACP implies almost always dGCP. Only in "pathological" examples, growth may abolish ATP-coupled product synthesis. In fact, this can only happen if growth consumes ATP and the product (or a product precursor) in the same fixed ratio as both are generated. Such a case is extremely unlikely, and we did not observe this in our computation examples. We therefore assume in the following that dGCP is present in a strain whenever dACP is. However, the other way around does not necessarily hold. For example, a dGCP strain might be established because a target metabolite becomes, by suitable interventions, a necessary byproduct of a biosynthetic pathway and must then be excreted. This strain would not show dACP. Here we can again distinguish the two cases, with and without a given demand of ATP for NGAM. Importantly, if the NGAM demand is not zero, all resulting strain designs computed for dACP will also fulfill the condition of SUCP: then, a minimum amount of substrate must be taken up to produce the demanded amount of ATP and since dACP demands some product synthesis whenever ATP is synthesized, dACP implies product synthesis whenever substrate is taken up. We can therefore concentrate on the case for dACP without a minimum NGAM demand.

For a comparison with the respective dGCP and SUCP strains, we fully enumerated MCSs up to the size of 4 in the core model of *i*ML1515 for dACP of ethanol by *E. coli*. In analogy to the eqs. (3.12) and (3.13) used for dGCP, the target system for dACP reads

$$r_P = 0 \tag{3.32}$$

$$r_{ATPM} \geq \varepsilon, \tag{3.33}$$

and we also demanded feasibility of growth (eq. (3.11)).

The results of these computations are shown in Figure 3.6 and indicate that dACP takes an intermediate role within the (hierarchical) tree of MCSs connecting dGCP with SUCP strain designs: all MCSs for SUCP imply dACP which in turn implies dGCP. With the explanations given above, this was to be expected, first, because SUCP with feasible growth and net ATP formation implies both dACP and dGCP and, second, because directional coupling of ATP synthesis with ethanol formation should also (directionally) couple growth with ethanol production since ATP is required for growth. In the reverse direction, Figure 3.6 suggests that many dGCP strategies are induced by directional coupling of ATP synthesis with ethanol formation and that some dACP strategies in turn even fulfill SUCP. An example of an MCS that is valid for all strategies is to block supply of oxygen and to knockout the pathway to the alternative fermentation product lactate. However, there are some MCSs for dGCP that are not valid dACP strategies (e.g., fourth MCS and last MCS in Figure 3.6) indicating that

the mechanism of growth-coupling induced by these MCSs must be ATP-independent. For example, as suggested by other studies, branching points in the metabolism, known as anchor reactions, may serve as suitable targets to make a compound a mandatory by-product of essential biomass precursors [119, 137]. The fourth dGCP MCS in Figure 3.6 is a particularly interesting case as this MCS can be extended to (directionally) couple both biomass and ATP synthesis with product formation and that some (but not all) of these dACP strategies directly support SUCP.

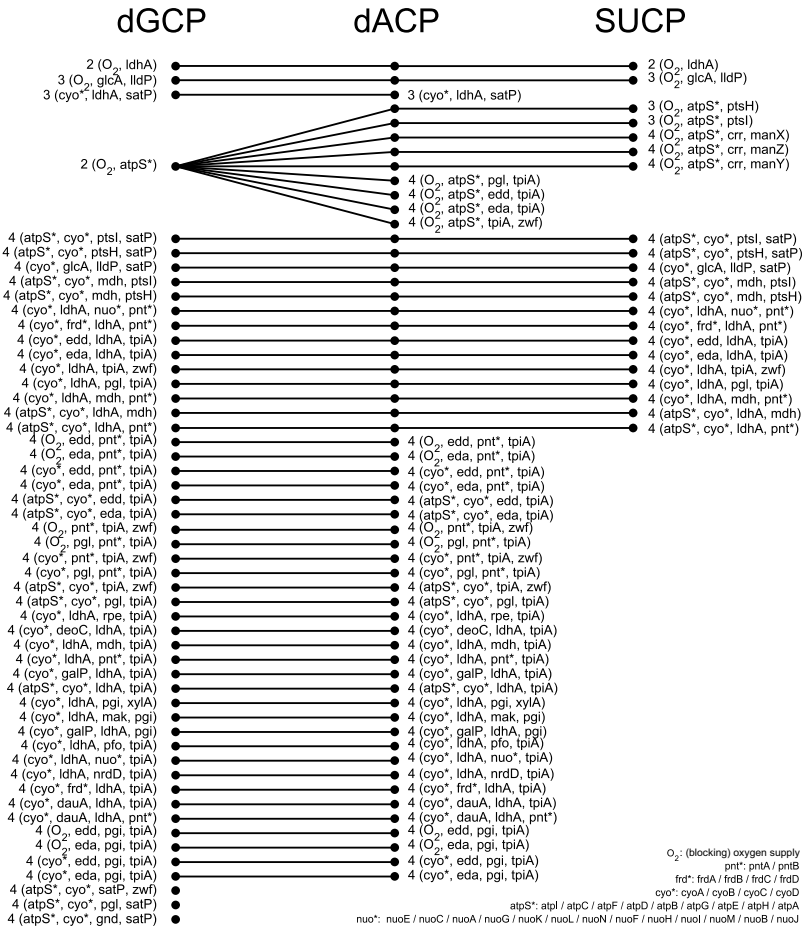


Figure 3.6: Comparison of the MCSs for dGCP, dACP and SUCP of ethanol in the *i*ML1515 core model. In all cases, no NGAM was demanded. The MCSs of size 2 and 3 for dGCP are identical with those from Figure 3.5B. ATP-coupled production takes an intermediate position between dGCP and SUCP. All SUCP-MCSs also have dACP, which in turn have dGCP. Some MCSs for dGCP are not ATP-coupled. In these cases, production is forced through alternative mechanisms.

3.6 Discussion

Coupling growth with product synthesis is the most common design principle used in computational strain design for metabolic engineering. In this chapter, we reviewed and systematized the existing notions, resulting in four unambiguous classes. All existing methods for computing growth-coupled strategies can be assigned to one of these four categories. We based each coupling type on simple mathematical definitions implying a clear hierarchical order, such that a strain design strategy for a stronger coupling type automatically implies all weaker ones. Simple constraint-based modeling techniques such as FBA, production envelopes, and yield spaces can be used to test whether coupling of a certain type is present in a given network (design).

We clarified how the framework of MCS can be employed to compute strain design for each of the four classes. While the formulation of specific MCS problems to find dGCP and SUCP strain designs is straightforward and has, for the case of SUCP, been used in previous work, strategies for pGCP and wGCP could so far not be computed with MCS. We therefore extended the existing MCS framework with implicit constraints for the maximal growth rate, so that product synthesis can now be demanded for the special case of optimal growth. This extension closes the gap between MCS-based and bilevel-based strain design approaches. The new optimality constraint feature for (reaction or gene) MCS was also implemented in *CellNetAnalyzer* (see section 6.1.4) and is compatible with the MCS developments introduced in chapter 4, such as the definition of multiple target and desired regions as well as substrate and heterologous pathway additions [2].

We illustrated the hierarchical dependency of the four coupling types by exemplarily computing and analyzing MCSs for growth-coupled production of ethanol in *E. coli*. The results confirmed that an MCS found for a demanded stronger coupling degree is always a superset of (or identical to) at least one MCS computed for weaker coupling. In the reverse direction, searching for MCSs of a weaker coupling type (e.g., dGCP) may deliver strain design strategies that are also valid for stronger coupling (e.g., SUCP). If a valid MCS found for a given coupling type (e.g., dGCP) does not imply stronger coupling (e.g., SUCP), then it might be extensible to an MCS of stronger coupling by adding further interventions. In a second computation, we used the MCS algorithm to compute realistic gene-knockout strategies that prime *E. coli* for the production of 12 native and heterologous products, again using all four coupling types. We found that in most cases where MCSs with a minimum number of cuts could be found, demanding weaker coupling strengths may be beneficial in terms of required interventions, especially when comparing pGCP against the other three coupling strengths. However, much less is saved (typically only one or two interventions) when comparing the smallest number of interventions in wGCP and dGCP against SUCP (see last row in Table 3.3). Moreover, although the respective runs always depend on the problem setup and the chosen solver (parameters) and seeds, we noted a tendency that computing the respective MCSs for pGCP and SUCP had

higher success rates compared to wGCP and dGCP and was also faster (in case of SUCP at least when searching for any MCS; Table 3.2). For wGCP, this can be explained by the fact that the strain design optimization problem has the largest size of all coupling degrees. Also, we did not encounter any case where a solution for the weakest coupling degree (pGCP) existed but not for the strongest (SUCP). This confirms results of a recent study showing that SUCP strain designs exist for almost all potential products (metabolites) in five important production hosts [31]. Altogether, from the perspective of computability, pGCP and SUCP seem to be preferable, however, since pGCP cannot guarantee product synthesis (not even at optimal growth) this could be an argument for favoring SUCP strategies. Generally, we recommend assessing the feasibility of each coupling type by computing (single) smallest or even random MCSs before searching for an MCS with a minimum number of interventions.

As a last methodological development, we showed how product synthesis can be coupled to other biological functions than growth and that the notion of coupling strengths as introduced herein for growth-coupling can be naturally generalized for those cases. As a relevant example for metabolic engineering, especially in the context of enforced ATP wasting strategies, we discussed the design principle of ATP-coupled production (ACP). Since ATP synthesis is directly relevant for growth, there are several relationships between growth-coupled and ATP-coupled strain designs. In particular, intervention strategies inducing strongly ATP-coupled product synthesis imply practically in all cases also dGCP while the converse is not necessarily true.

4 Extensions and generalizations of the MCS framework

In the last chapter, the MCS framework was already extended to allow the computation of weaker (bilevel-like) strain designs for growth-coupled product synthesis. In the following, several new features are introduced to the MCS algorithm that improve the algorithmic performance and further broaden the scope of applications of the MCS framework. The main extensions include (1) the specification of multiple target and desired regions; (2) the possibility to consider combinations of reaction deletions and additions; (3) the computation of substrate co-feeding strategies; (4) an improved technique for integrating gene-protein-reaction (GPR) associations into the metabolic models together with a number of effective preprocessing and compression steps to reduce the GPR rules and thus to accelerate the computation of gene MCS; and (5) the possibility to associate individual cost factors to each intervention. The application of the new developments is exemplified in realistic example computations for the growth-coupled production of 2,3-butanediol with *Escherichia coli*, using substrate co-feeding. Finally, also the performance gain from GPR compression is benchmarked.

4.1 Multiple desired or undesired flux spaces

So far, only one target and one desired region were considered in the computation of MCS (eq. (2.45)). This is sufficient for many applications. However, there are cases where multiple undesired and desired flux spaces must be defined to properly formulate a strain or network design problem. The idea of multiple desired regions was already discussed in [57]. However, applications with multiple target regions were not considered and, most importantly, the calculation of MCS for multiple desired regions was so far only possible for the classical way of MCS calculation via elementary modes, which, in contrast to the Farkas-lemma-based approach, is not applicable to genome-scale models.

To motivate the use of multiple target regions, we consider an example for the design of a two-stage process. The cells should grow aerobically at a high growth rate in the first phase. For the anaerobic production phase we demand inhibition of growth and a minimum product per substrate yield ($Y^{min}_{P/S}$ e.g., ethanol per glucose). As a result, we have two target regions that need to be blocked. The first is used to remove flux vectors of growth above a threshold of

0.01 under anaerobic conditions (the growth rate r_{BM} is given in h^{-1}, all flux rates are given in mmol $g_{CDW}^{-1}\,h^{-1}$; units will be omitted in the following equations):

$$r_{BM} \geq 0.01 \quad \Leftrightarrow \quad -r_{BM} \leq -0.01, \tag{4.1}$$

which can be written in standard form as

$$\mathbf{T_1 r} \leq \mathbf{t_1}, \tag{4.2}$$

and be combined with the constraints of the original metabolic model as shown in eq. (2.35), to describe the undesired steady-state flux space

$$\mathbf{A_{T1} x} \leq \mathbf{b_{T1}}. \tag{4.3}$$

The second target (undesired) region comprises anaerobic flux vectors with product per substrate yields below a threshold $Y_{P/S}^{min}$. As explained in section 2.3.2, the trivial (zero) flux vector must not be part of the target region because this flux distribution can never be eliminated through knockouts. Therefore, the zero vector is excluded in the first target region, by selecting only flux states above a minimum growth rate. Constraint-based models often hold constraints that already exclude the zero-flux vector as feasible solution (e.g., a minimum non-growth associated ATP maintenance demand). If this is not the case, we must explicitly exclude the trivial flux vector, e.g. by demanding a minimum substrate uptake rate for the target region. In the second target region, we therefore set a lower bound of 0.1 to the substrate uptake reaction (r_s). Using the linearization from eq. (2.36), the second target region can thus be described by a set of three inequalities

$$\begin{aligned} r_{O_2,up} &\leq 0 \\ r_P - Y_{P/S}^{min}\, r_s &\leq 0 \\ -r_S &\leq -0.1, \end{aligned} \tag{4.4}$$

which can again be expressed in matrix notation:

$$\mathbf{T_2 r} \leq \mathbf{t_2}, \tag{4.5}$$

and be extended with the other metabolic network constraints (2.30) to the form of eq. (2.35)

$$\mathbf{A_{T2} x} \leq \mathbf{b_{T2}}. \tag{4.6}$$

For the example, we also need to specify two desired regions: The first expresses that a high growth rate, above a specified threshold r_{BM}^{min}, must be attainable under aerobic conditions:

$$-r_{BM} \leq -0.2. \tag{4.7}$$

We do not need to specify the aerobic conditions explicitly in the desired region, as oxygen uptake is generally possible in the original model. Equation (4.7) makes up the first desired system

$$\mathbf{D_1 r} \leq \mathbf{d_1}, \tag{4.8}$$

that is again combined with the original model as in eq. (2.38) to

$$\mathbf{A_{D1} x} \leq \mathbf{b_{D1}}. \tag{4.9}$$

We then also need to ensure that, under anaerobic conditions, when low-yield solutions from the second target region are eliminated, some production flux states (with then high-yield) remain feasible. This gives rise to the second desired region:

$$\begin{aligned} -r_P &\leq -5 \\ r_{O_2,up} &\leq \quad 0, \end{aligned} \tag{4.10}$$

which reads

$$\mathbf{D_2 r} \leq \mathbf{d_2}, \tag{4.11}$$

in the compact description and is combined with the original model to

$$\mathbf{A_{D2} x} \leq \mathbf{b_{D2}}. \tag{4.12}$$

It is important to notice that the desired systems 1 and 2 cannot be jointly represented by a single desired system because their union is not convex (the same holds for the two target regions).

As a generalization, we allow in the following the definition of arbitrary many target regions, each defined by inequalities posed by an appropriate pair of matrix and vector ($\mathbf{A_{T1}}, \mathbf{b_{T1}}$), ($\mathbf{A_{T2}}, \mathbf{b_{T2}}$), ..., ($\mathbf{A_{Tj}}, \mathbf{b_{Tj}}$). Likewise, we allow the definition of arbitrary many desired regions represented by ($\mathbf{A_{D1}}, \mathbf{b_{D1}}$), ($\mathbf{A_{D2}}, \mathbf{b_{D2}}$), ..., ($\mathbf{A_{Dk}}, \mathbf{b_{Dk}}$). In the MILP formulation (eq. (2.45)), each target region and each desired region needs to be integrated as separate blocks in the formulation, analogous to the single regions in the original formulation in eq. (2.44). The variables and constraints of each system are linked to a single global set of binary variables z_i

that indicate reaction knockouts. For example, an integrated MILP with two target ($\mathbf{A_{T1}}, \mathbf{b_{T1}}$), ($\mathbf{A_{T2}}, \mathbf{b_{T2}}$) and two desired ($\mathbf{A_{D1}}, \mathbf{b_{D1}}$), ($\mathbf{A_{D2}}, \mathbf{b_{D2}}$) regions reads:

$$
\begin{aligned}
&\text{minimize} && \sum z_i \\
&\text{subject to} && \begin{bmatrix} \mathbf{A_{T1}^\top} & \mathbf{I_{KO}} & & & & \\ \mathbf{b_{T1}^\top} & & & & & \\ & & \mathbf{A_{T2}^\top} & \mathbf{I_{KO}} & & \\ & & \mathbf{b_{T2}^\top} & & & \\ & & & & \mathbf{A_{D1}} & \\ & & & & & \mathbf{A_{D2}} \end{bmatrix} \begin{bmatrix} \mathbf{y_1} \\ \mathbf{v_1} \\ \mathbf{y_2} \\ \mathbf{v_2} \\ \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} \begin{matrix} = \\ \leq \\ = \\ \leq \\ \leq \\ \leq \end{matrix} \begin{bmatrix} \mathbf{0} \\ -1 \\ \mathbf{0} \\ -1 \\ \mathbf{b_{D1}} \\ \mathbf{b_{D2}} \end{bmatrix} \\
& && \forall i : z_i = 0 \rightarrow v_{1,i} = 0 \\
& && \forall i : z_i = 0 \rightarrow v_{2,i} = 0 \\
& && \forall i : z_i = 1 \rightarrow x_{1,i} = 0 \\
& && \forall i : z_i = 1 \rightarrow x_{2,i} = 0 \\
& && \mathbf{y_1} \geq \mathbf{0}, \quad \mathbf{y_2} \geq \mathbf{0}, \quad z_i \in \{0,1\}.
\end{aligned}
\tag{4.13}
$$

Another application of multiple target and desired regions will be presented later when searching for MCSs for substrate co-feeding strategies.

4.2 Weighting interventions with cost factors

A further generalization of the MCS MILP problem is to replace the original objective function

$$\text{minimize} \sum_i z_i, \tag{4.14}$$

that simply counts the number of interventions, with a weighted cost function:

$$\text{minimize} \sum_i p_i z_i. \tag{4.15}$$

The weights p_i can be specified by the user and reflect the experimental costs of implementing the respective intervention. For example, blocking oxygen uptake to enforce anaerobic conditions is a frequent intervention in many MCSs when searching for strain designs with high product yields. Clearly, this "intervention" is experimentally easy to implement and in many biotechnological applications even the preferred process condition. The weight factor for the oxygen uptake reaction could thus be set to zero.

There is one important change in the interpretation of MCS when using the generalized objective function (eq. (4.15)). Minimization of the original (simpler) objective function (eq. (4.14)) (which is equivalent to set all p_i to unity in eq. (4.15)) always delivers MCSs with a minimum number of knock-outs, hence, of MCSs with minimal support. With the generalized objective function (eq. (4.15)), the MCSs still ensure that the cost function is minimized (which

still justifies the use of the term "minimal" cut set). However, if some p_i are set to 0, then MCSs may be found where the set of reaction knockouts is not support-minimal since some deletions may be added "for free". In those cases, there can be MCSs with identical costs that differ only in the presence/absence of these cost-free reaction deletions. It is easy to filter out equivalent MCSs that are supersets of others in a postprocessing step, however, in some cases it might just be the goal to find all these alternatives. Generally, even if the p_i of a reaction is zero, it might not be removable as the desired region may require the presence of this reaction.

4.3 Reaction additions by inverting the knockout-logic

Most computational strain optimization algorithms use combinations of (gene or reaction) knockouts to enforce desired phenotypes. However, some dedicated methods have been developed for biased strain design techniques to also allow addition of heterologous reactions alongside reaction deletions, to further enlarge the search space for network design [90, 92, 115]. So far, this feature is not available in the MCS approach but will be developed in the following. As one particular application, we will describe how substrate co-feeding strategies can be computed with this extended MCS approach.

For combining reaction deletions and additions in the MCS framework, the (new) addable reactions are included in the original network and every reaction is then marked as either "deletable", "addable" or "non-targetable". The *deletion* of a reaction i, marked by $z_i = 1$, implies costs in the objective function (eq. (4.15)) and translates directly to the deactivation of associated constraint(s) in the target system(s) and variable(s) in the desired system(s) as explained in section 2.1.2 (see Tables 2.2 and 2.4). Addable reactions are treated inversely so that the indicator constraints in the shown example MILP (eq. (4.13)) change to:

$$\begin{aligned} &\forall i : z_i = 1 \rightarrow v_i = 0 \\ &\forall i : z_i = 0 \rightarrow x_i = 0. \end{aligned} \tag{4.16}$$

$z_i = 1$ then marks the *addition* of a reaction i which leads to increased costs in the objective function (eq. 4.15) and $z_i = 0$ marks the standard case where reaction i is not used. This relationship is expressed through indicator constraints in eq. (4.16) but can also be formulated with big-M constraints (see eqs. (2.22) and (2.23)).

4.4 Combining reaction deletions and additions to find substrate co-feeding strategies

Figure 4.1 illustrates that, compared to the traditional MCS approach, the generalized design specifications (multiple target and desired regions) and the extended search space (by reaction additions) enormously increase flexibility in formulating and solving complex network and strain design problems. The dark blue shapes mark the solution spaces of steady-state flux vectors of the different mutants after applying an MCS. While each target and each desired

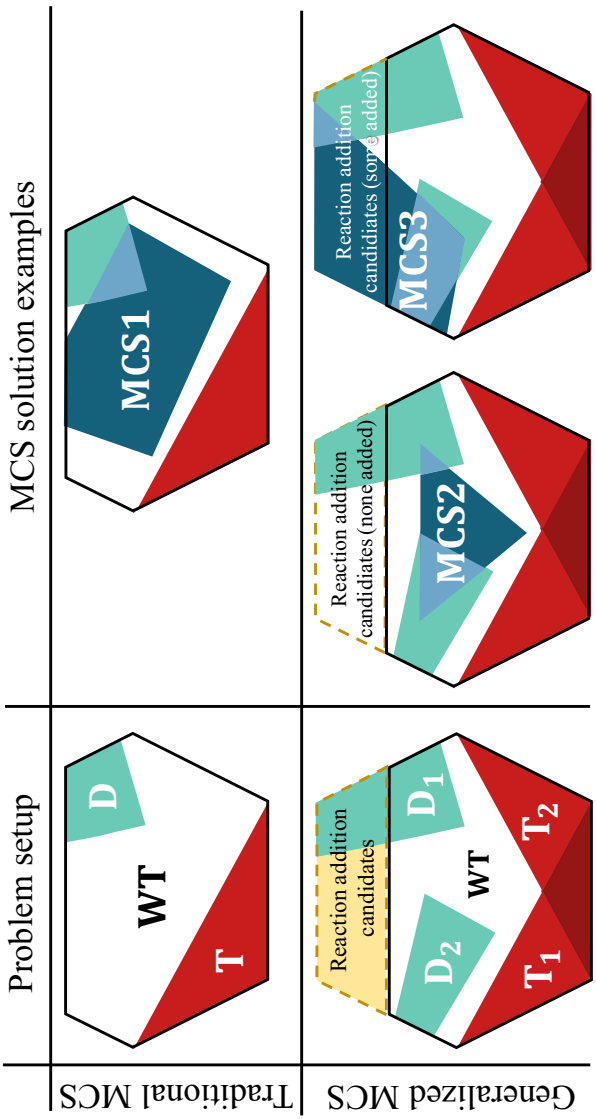


Figure 4.1: Exemplary illustration of metabolic solution spaces of wild type (solid black outline) and strains designed with the traditional and extended/generalized MCS approach (dark blue shapes). Red areas: target regions; green areas: desired regions; yellow area in the problem setup for generalized MCS: flux states accessible through the addition of reactions.

polyhedron is always convex (as are the solution spaces of the wild type and of the MCS mutant), the union of *multiple target* or of *multiple desired* regions is, in general, not.

With the extensions of multiple target regions (section 4.1) and possible addition of reactions (section 4.3), we are now able to compute suitable MCSs that exploit substrate co-feeding, which adds another dimension for strain design strategies. Co-feeding of substrates has been used in metabolic engineering, for example, to enhance the productivity of a designed strain [138–140], to cope with genetic knockouts that induce auxotrophies [54], or to specifically design auxotrophic [141] and biosensor strains [142]. Substrate co-feeding can also be used to effectively regenerate or balance cofactors [143, 144] or to provide product precursors [145]. Despite these applications, as pointed out by Liu et al. [145], directed strain design harnessing substrate co-feeding opportunities has rarely been employed so far. Figure 4.2 shows an example network holding potential for different strain design strategies that rely either on single substrate feeding or co-feeding. The main (standard) substrate S can be metabolized to biomass (BM) and four different products (P (desired product), Q, R, U). Dashed reaction arrows with yellow captions indicate exchange reactions (the latter are assumed to happen spontaneous, i.e., they are non-targetable). The blue dashed reaction indicates that U could potentially also be taken up and thus serve as a substrate. As we will see later, especially treating the case of a metabolite either being an excreted product or serving as a potential substrate is technically challenging. A realistic example would be acetate that can often be excreted as product but may also serve as relatively cheap (co-)substrate in bioprocesses. We assume that P is the product of interest. As a typical strain design strategy, we may enforce growth-coupled product synthesis where the target region contains flux vectors with low product yield (e.g., $Y_{P/S} = \frac{r_{P,ex}}{r_{S,up}} \leq 0.4$; see eq. (2.36)), while the desired region demands that biomass synthesis is feasible (e.g., $r_{BM} \geq 0.1$; see eq. (4.7)). A traditional MCS-based strategy delivered by the algorithm is indicated in Figure 4.2 (knockouts marked in orange): S is used as the sole substrate and r_3, r_7, r_9 and r_{12} are deleted. In the remaining network, ATP production entails the generation of Z, a metabolite that can only be drained by synthesizing and excreting P. The option to co-feed metabolite U offers now another coupling strategy by dissecting the metabolism into an upper and a lower part (interventions marked in blue in Figure 4.2). The removal of r_6 and the addition of the uptake reaction for U ($r_{U,up}$) leads to a functional separation of the two substrates: S is then used for ATP and product synthesis while U is needed as a biomass precursor.

Generally, scenarios with multiple substrates require a suitable redefinition of the yield to properly account for both substrates. In this example, we would consider the combined yield $Y_{P/(S+U)} = \frac{r_{P,ex}}{r_{S,up}+r_{U,up}}$ and specify the target region by $Y_{P/(S+U)} \leq 0.4$. We could then mark $r_{U,up}$ as addable reaction and try to search for both classical one-substrate strain designs (using only S) as well as substrate co-feeding strategies (using S and U). If U would only act as optional substrate and the excretion of U was not possible, then co-feeding strategies could be straightforwardly identified by marking the uptake of U as addable reaction. However, in

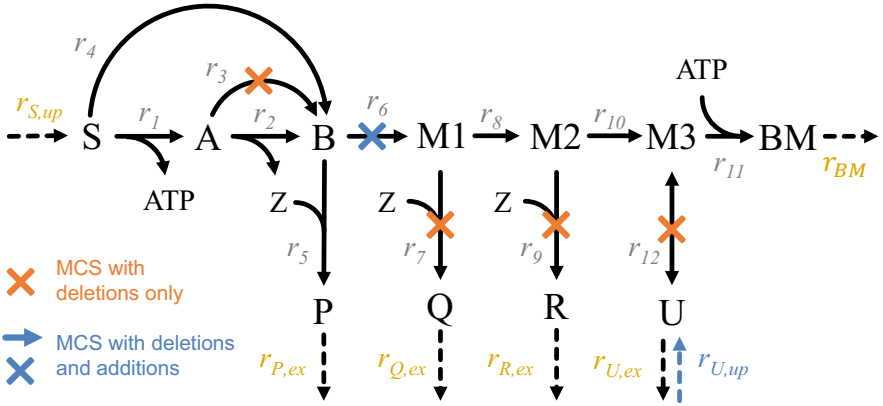


Figure 4.2: Example network of a wild-type strain that can be turned into a production host for the target product P, either by using substrate S and four knockouts (orange MCS) or by a co-utilization of S and U , which requires one deletion and the addition of the uptake reaction for U (blue MCS). Reactions with dashed arrows and yellow names indicate non-targetable exchange reactions.

this example, with U also being a potential product, this alone would not be sufficient to find suitable co-feeding solutions. The MCS algorithm would seek to eliminate all flux states with yields inferior to $Y_{P/(S+U)}$. Yet, the cycle of the sink pseudo-reaction $r_{U,ex}$ and an activated uptake reaction $r_{U,up}$ can carry high fluxes and inflate the denominator of the yield function, making it impossible to disrupt all flux distributions with yields below the threshold. Allowing the deletion of $r_{U,ex}$ would affect the correct identification of single substrate strategies because the algorithm would then be able to generate solutions that delete the secretion reaction $r_{U,ex}$ although it is non-targetable.

A solution to this dilemma is the definition of two target regions. The first target flux polyhedron describes undesired solutions for the "classical" one-substrate usage where U is not taken up:

$$\mathbf{T1}: \qquad Y_{P/S} = \frac{r_{P,ex}}{r_{S,up}} \leq 0.4, \quad r_{S,up} \geq 0.1, \quad r_{U,up} = 0. \tag{4.17}$$

These constraints can be reformulated to bring them into the required form $\mathbf{A_{T1}x} \leq \mathbf{b_{T1}}$. The second target region specifies undesired solutions for the case where U is taken up, using the combined yield term. Importantly, in the definition of this second target region, only the relevant cases where U is not excreted are considered. Hence, with the defined target regions 1 and 2, the irrelevant case with an active cycle in $r_{U,up}$ and $r_{U,ex}$ is simply not included in the set of target vectors:

$$\mathbf{T2}: \qquad Y_{P/(S+U)} = \frac{r_{P,ex}}{r_{S,up} + r_{U,up}} \leq 0.4, \quad r_{S,up} \geq 0.1, \quad r_{U,ex} = 0. \tag{4.18}$$

Together with the desired region ($r_{BM} \geq 0.1$), the enumeration of strain designs with the introduced MCS algorithm extended with multiple target regions and addition of reactions will return both solutions (blue and orange in Figure 4.2). This MCS computation example is also available in the Appendix (see Source Code A.2).

4.5 Compressing GPR rules for the efficient computation of MCSs with genetic interventions

The MCS approach was originally constructed to find suitable combinations of *reaction* deletions that block a certain metabolic phenotype. However, when using it for strain design in metabolic engineering, in most cases only *genes* can be targeted directly (exceptions are pseudo-reactions for uptake of oxygen or substrates, which can be switched off by removing the respective compound from the medium). A translation of reaction deletions to corresponding gene deletions is often possible, however, there are also complicated relationships, where

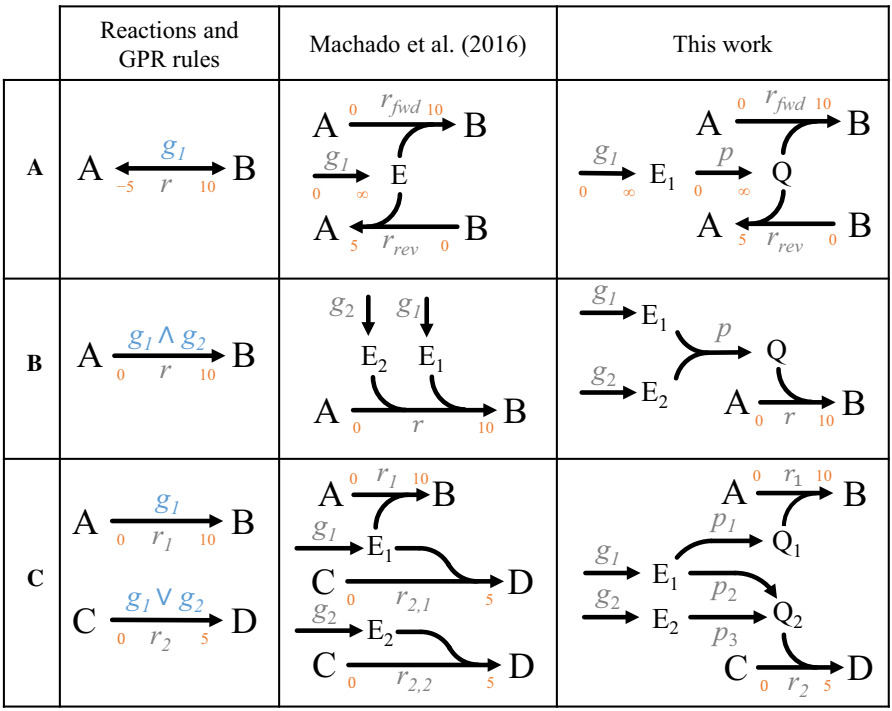


Figure 4.3: Integration of GPR rules into the metabolic network structure. In contrast to the approach of Machado et al. [79], in the variant introduced herein, every reaction with a GPR rule consumes a pseudo-metabolite Q that represents a pool of enzymes catalyzing this reaction. Example cases: (**A**) reversible reactions; (**B**) enzymes with subunits; (**C**) promiscuous enzymes (g_1) and isoenzymes (g_1, g_2).

this translation is not straightforward, for example, when an enzyme or enzyme subunit is involved in several reactions. Hence, for its application in strain design, MCSs should deliver suitable combinations of gene interventions (gene MCSs), which induce the desired metabolic phenotype by indirectly targeting the operation of certain reactions [79]. This requires the incorporation of gene-protein-reaction (GPR) rules, which has been done in several strain optimization methods [31, 79, 91, 108, 146, 147]. GPR rules are typically Boolean functions in disjunctive normal form (DNF) describing how the enzymes catalyzing the metabolic reactions in the network are made from combinations of gene products. Two major approaches for integrating GPR rules in the MCS framework have been suggested [79, 108]. Herein we adopt the approach of Machado et al. [79] where the GPR rules are translated to pseudo-reactions and pseudo-metabolites, which are seamlessly integrated into the metabolic network. The reaction-based MCS algorithm can then remove or add genes via their associated pseudo-reactions. In the original approach of Machado et al., each gene product (i.e., either an enzyme or an enzyme subunit denoted by (E, E_1, E_2...) in Figure 4.3) is introduced as a pseudo-metabolite that is generated by an individual "enzyme synthesis" pseudo-reaction (denoted by $g_1, g_2 \ldots$ in Figure 4.3) from its corresponding gene. (Note that, depending on the context, we will use the symbols $g_1, g_2 \ldots$ interchangeably either as the enzyme synthesis pseudo-reaction (as in Figure 4.3) or as the actual gene from which the enzymes are made (see e.g. step 0 in Figure 4.4). This is justified because there is a 1:1 relationship between both.) Each enzyme pseudo-metabolite E, E_1, E_2... is then integrated as a reactant in the respective reaction(s) it catalyzes, and it thus becomes essential for the operation of the(se) reaction(s). Reversible reactions need to be split and considered as separate reactions for both directions. Depending on the GPR rules, it can happen that (1) one enzyme catalyzes multiple reactions, (2) a reaction requires several enzyme subunits, or (3) one reaction is catalyzed by multiple (iso)enzymes (see Figure 4.3). In the latter case, Machado et al. split the reaction to account separately for each isoenzyme [79]. A disadvantage of this approach is that copying an isozyme reaction with a finite and non-zero flux bound may result in an effective reaction flux that is multiple times higher than intended by the original model (see Figure 4.3C). We therefore choose a slightly different representation of the GPR rules, which guarantees that the flux ranges from the original model are conserved for all reactions (see right column in Figure 4.3). The key difference of this approach is that we introduce for each reaction a separate pseudo-metabolite (Q) representing the pool of all enzymes (including those arising by the combination of enzyme subunits) that can catalyze this reaction. This reaction-specific enzyme pool can be filled by pseudo reactions ($p_1, p_2 \ldots$ in Figure 4.3) according to the respective GPR rules. The enzyme pool Q is then "consumed" in its associated reaction. Hence, reactions are only split in the case of reversibility, where the flux boundaries can easily be mapped from the original reaction. Auxiliary pseudo-reactions are all unidirectional and unbounded. The new approach adds more pseudo-metabolites and pseudo-reactions than the original approach, yet, many of these

additional elements disappear again when compressing the network with integrated GPR rules (see below).

Once the GPR rules have been added to the metabolic network, the MCS algorithm can be used to generate gene intervention-based strain designs. Here, metabolic reactions remain protected from knockouts (non-targetable) while the manipulation of the pseudo-reactions yielding the enzymes or enzyme subunits from the genes (reactions $g_1, g_2 \ldots$ in Figure 4.3) is allowed. Reactions such as oxygen or substrate uptake, which have no associated genes but can be controlled externally, can also be kept targetable. In this way, the MCS algorithm can identify suitable combinations of genetic interventions and process conditions to meet the strain design specifications.

The extension of metabolic networks with GPR rules as described above may significantly increase the network size and thus the complexity of strain design methods. Generally, compression techniques and preprocessing steps to cope with the complexity of (classical) genome-scale MCS calculations are essential and integrated in some strain design (including MCS) algorithms. These methods include, for example, the removal of conservation relations and blocked reactions as well as lumping of coupled reactions (see e.g. [148, 149]). In addition to these network compression techniques already in use, we here propose a set of preprocessing steps to also reduce and simplify GPR rules before integrating them in the metabolic network. For gene-based MCS, every gene is simulated as a targetable reaction that has an associated boolean variable in the underlying MILP indicating whether it is active or not. Boolean variables are computationally more expensive than continuous variables. The compression steps introduced below seek to reduce the number of targetable reactions (here corresponding to targetable genes) without losing any MCS solution.

A first step for the reduction of network and GPR rules is the identification and removal of blocked reactions and their genes. More advanced compression steps described below exploit the fact that the *desired region(s)* render several reactions essential. This can be used to discard certain targetable genes from the model, as their knockout would block essential or at least never affect non-essential reactions. For example, if some minimum growth rate has to be maintained in the designed strains, any gene being essential for growth needs not be considered as knockout candidate and can be removed from the GPR rules. In practice, essential reactions can be identified by FVA, performed on each desired region. If, for any desired region, the upper and lower bound of a reaction are non-zero and have the same sign, then the reaction is essential and must not be blocked by any set of gene deletions.

Apart from essential genes, the compression steps introduced below also exploit the occurrence of "equivalent genes" which can be lumped. In total, we use the following seven compression steps to reduce the number of genes and GPR rules added to the model:

(1) Remove GPR rules of blocked reactions. Deletion (or addition) of genes will have no effect on these reactions; hence their rules are not relevant for the MCS computation.

(2) Mark a gene g_p as protected ($g_p = 1$) and thus non-targetable if it occurs exclusively in GPR rules for essential reactions. Knocking out such a gene can never contribute in blocking non-essential reactions.

(3) Mark a gene as protected if it is essential for at least one essential reaction, since deleting this gene will inevitably result in disrupting the desired behavior.

(4) Identify GPR terms (disjunctions) that cannot be targeted due to protected genes. If one of the conjunctions of a DNF consist exclusively of protected genes, then the respective reaction cannot be knocked out by any combination of gene deletions (e.g., if gene g_p is protected ($g_p = 1$) then $g \vee g_p = 1$). This makes the entire GPR rule of the reaction irrelevant for MCS computation and it can be discarded.

(5) Discard all protected genes. After step 4, all protected genes occur in conjunctions with unprotected genes and their consideration is no longer necessary ($g_p = 1 \rightarrow g \wedge g_p = g$).

(6) Reduce non-minimal conjunctions in the GPR rule that arose by the previous steps (e.g., $g_1 \vee (g_1 \wedge g_2) = g_1$).

(7) Lump equivalent gene deletion candidates that always occur together in conjunctions (e.g., $g_1 \wedge g_2$) and lump gene addition candidates likewise. In some cases, another lumping can be performed inside single GPR rules if the genes do not occur in any other context. Finally, repeat the substitution as well for the disjunctions of type ($g_{1\wedge 2} \vee g_3$). To map the original intervention costs, each lumped gene carries the minimum intervention costs of the gene rule that was substituted. Computed MCSs involving lumped genes must be expanded in a post-processing step.

We added the GPR rule compression and integration as a building block in the pipeline for MCS computation. This pipeline now involves problem setup, GPR rule compression and integration, network compression and post-processing steps for expanding computed MCSs (see also the example in Figure 4.4 and Figure 4.5 discussed below; a more detailed description of the computation pipeline is given in section 6.1.3):

(0) Model setup, definition of target and desired regions(s) and of addable/deletable/non-targetable reactions/genes.

→ User calls *CNAgeneMCSEnumerator2* (or *3*).

(1) FVA of the full model to find blocked reactions and FVA of the desired regions to identify essential reactions.

(2) Compression of GPR rules (making use of the FVA results from step (1))

(3) Integration of compressed GPR rules in the network with *CNAintegrateGPRrules* (see section 6.1.3).

→ Function calls *CNAMCSEnumerator2* (or *3*).

(4) Network compression (removal of blocked reactions and conservation relations, lumping reactions, protection of essential reactions, determination of effective flux bounds etc.) with *CNAcompressMFNetwork* (see section 6.1).

(5) MCS computation using the MCS core algorithm.

(6) Decompression of lumped reactions.

(7) Decompression of lumped genes.

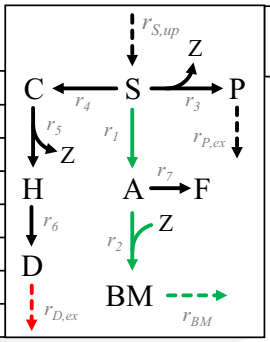
(8) Characterizing and Ranking MCS (see chapter 5).

Note that the steps (4)-(6) (leaving out steps (1)-(3) and (7)) can still be used to compute classical MCSs with reaction knockouts. In *CellNetAnalyzer* (see below), these steps are conducted by *MCSEnumerator* functions (that is, *CNAMCSEnumerator2* or *3*). The steps (1)-(3) and (7) are specific for the calculation of gene MCSs, and in *CellNetAnalyzer* they are performed by the functions (*CNAgeneMCSEnumerator2* or *3*) which calls *CNAMCSEnumerator2* or *3* between step (3) and (7) as a sub-module. In this way, the actual gene MCSs are computed as reaction knockouts of the enzyme synthesis reactions, as explained above.

Figure 4.4 and Figure 4.5 show an example of the whole pipeline, demonstrating how the compression of the GPR rules and subsequently of the network structure reduces the dimension of the problem. In the presented example network, the substrate S can be turned into biomass and the two products P and D. The strain design goal is to inhibit the production of D by suitable genetic knockouts, while maintaining the ability to grow. An FVA shows that the reactions r_1 and r_2 are essential for the desired flux states. This information is used to compress the original

0 **Network definition and design specifications**

Original GPR rule (DNF)
r_1: $(g_1 \wedge g_2) \vee (g_1 \wedge g_3)$
r_2: $g_2 \vee g_4$
r_3: $(g_3 \wedge g_6) \vee g_8$
r_4: $(g_4 \wedge g_8) \vee (g_1 \wedge g_8)$
r_5: $(g_1 \wedge g_5 \wedge g_9) \vee (g_1 \wedge g_7)$
r_6: $g_4 \wedge g_1$
r_7: $g_6 \vee g_7$



Target Region: $r_{D,ex} \geq 1$
Desired Region: $r_{BM} \geq 1$

1 **Results of FVA:**

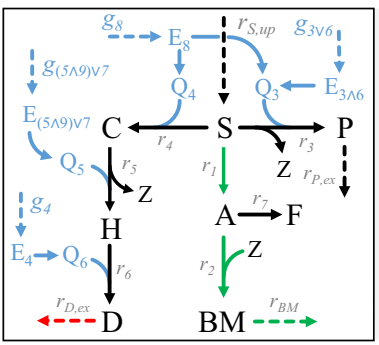
Blocked reaction: r_7

Essential reactions for the desired region: r_1, r_2

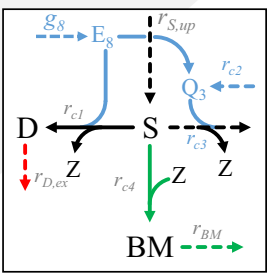
2 **Compression of GPR rules**

	1) Remove blocked reactions (r_7) **2)** Protect genes that occur only in essential reacions (g_2)	**3)** Protect genes that are essential for essential reactions (g_1)	**4)** Discard GPR rules that cannot be knocked out **5)** Discard protected genes	**6)** Remove non-minimal GPR rules **7)** Lump equivalent genes
r_1	$(g_1 \wedge \boldsymbol{g_2}) \vee (g_1 \wedge g_3)$	$(\boldsymbol{g_1} \wedge \boldsymbol{g_2}) \vee (\boldsymbol{g_1} \wedge g_3)$		
r_2	$\boldsymbol{g_2} \vee g_4$	$\boldsymbol{g_2} \vee g_4$		
r_3	$(g_3 \wedge g_6) \vee g_8$	$(g_3 \wedge g_6) \vee g_8$	$(g_3 \wedge g_6) \vee g_8$	$g_{3\wedge 6} \vee g_8$
r_4	$(g_4 \wedge g_8) \vee (g_1 \wedge g_8)$	$(g_4 \wedge g_8) \vee (\boldsymbol{g_1} \wedge g_8)$	$(g_4 \wedge g_8) \vee g_8$	g_8
r_5	$(g_1 \wedge g_5 \wedge g_9) \vee (g_1 \wedge g_7)$	$(\boldsymbol{g_1} \wedge g_5 \wedge g_9) \vee (\boldsymbol{g_1} \wedge g_7)$	$(g_5 \wedge g_9) \vee g_7$	$g_{(5\wedge 9)\vee 7}$
r_6	$g_4 \wedge g_1$	$g_4 \wedge \boldsymbol{g_1}$	g_4	g_4
r_7	~~$g_6 \vee g_7$~~			

3 **Network extension with compressed GPR rules**



4 **Network compression**



r_{c1}: $g_{(5\wedge 9)\vee 7}$, g_4, r_4, r_5, r_6, $pg_{4\rightarrow Q6}$, $p_{(g5\wedge g9)\vee 7\rightarrow Q5}$
r_{c2}: $g_{3\wedge 6}$, $pg_{3\wedge g6\rightarrow Q3}$
r_{c3}: r_3,$rP_{,ex}$
r_{c4}: r_1,r_2

Figure 4.4: Example illustrating the 7 steps of gene MCS calculation (part 1). Red arrow: targeted reaction; green arrows: protected reactions; blue: GPR rules; dashed blue arrows: switchable gene pseudo-reactions. For the sake of readability, the protein pool pseudo-reactions (p) connecting the enzymes (E) with the enzyme pools (Q) were not labeled in the network in step 3 and 4. The reactions r_{c1} to r_{c4} represent lumped reactions resulting from network compression (the contained reactions are shown below the box of step 4). The subsequent steps 5-7 of this example are shown in Figure 4.5.

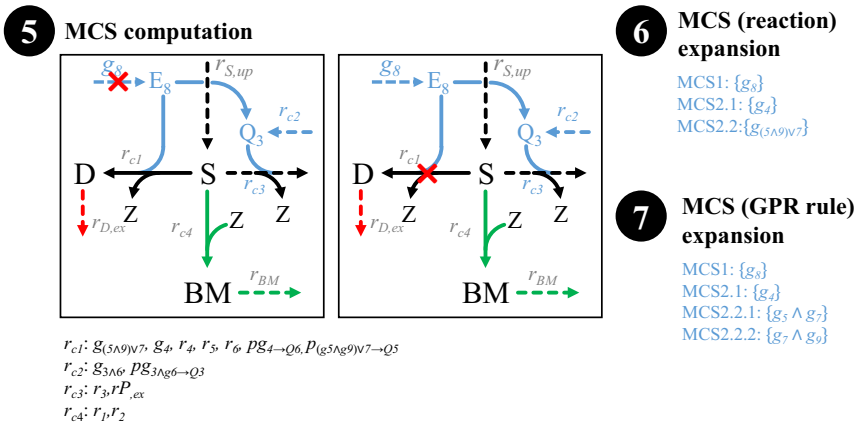


Figure 4.5: Continuation of the exemplary gene MCS calculation from Figure 4.4 (part 2): MCS computation (step 5), expansion/decompression of MCSs at the reaction (step 6) and finally at the gene (step 7) level. For better readability, the protein pool pseudo-reactions (p) connecting the enzymes (E) with the enzyme pools (Q) were not labeled in the network in step 5.

GPR-rule set along the presented steps (Figure 4). The reconfigured network with integrated compressed GPR rules is presented in Figure 4.4, step 3, showing that from the nine genes only four (partially lumped) genes remain in the network after GPR compression. Next, classical network compression (Figure 4.4, step 4) lumps coupled reactions, not distinguishing between "real" metabolic reactions and GPR pseudo-reactions. Analogous to lumped gene interventions, for each lumped reaction (r_{c1}, r_{c2}, r_{c3} and r_{c4} obtained in step 4), an individual cost-factor is generated that equals the minimal intervention cost for the ensemble of the original reactions. Applied to the compressed network, the MCS algorithm will deliver two solutions to suppress the undesired behavior: either the knockout of g_8 or of the lumped reaction r_{c1} (Figure 4.5, step 5). In order to match the original model, these solutions need to be expanded by reversing the previous compression steps, first at the reaction (Figure 4.5, step 6) and then at the gene level (Figure 4.5, step 7). In the example, decompressing the MCS results in four final solutions: A knockout of g_8, a knockout of g_4 or a knockout combination of either g_5 and g_7 or g_7 and g_9.

As shown in this example, decompression of lumped genes and reactions will not only result in a higher number of MCSs, it may also lead to a higher number of interventions per MCS or more expensive MCSs. Yet, the cost-reattribution during GPR compression (Figure 4.5, step 7) ensures that any MCS found in a compressed network still expands to at least one MCS in the original network with identical costs (e.g. $g_{((5\wedge 9)\vee 7)}$ would have deletion costs of 2 in the compressed network). In more complex cases where MCSs have multiple expansions with different costs, all MCSs that exceed the MCS cost limit are discarded.

4.6 Example strain design for the production of 2,3-butanediol using substrate co-feeding

Earlier, we introduced several generalizations and new algorithmic developments for the computation of MCSs, which are especially (but not exclusively) useful for strain design applications. In the following, we will demonstrate the applicability and the performance benefits of these extensions by realistic example calculations of relevant *E. coli* strain designs for the production of 2,3-BDO. 2,3-BDO is a bulk chemical whose bio-based production received much attention in the field of metabolic engineering [61, 62, 150–152]. The spectrum of its industrial application covers a broad range from cosmetics and nutrition to bioplastic [150]. Previous studies could increase the productivity of natural producers like *Klebsiella pneumonia* or *Bacillus licheniformis*, however, the use of these species has several disadvantages due to requirement of complex and expensive medium components and potential pathogenicity [153]. Other studies therefore aimed to synthesize 2,3-BDO with more established production organisms such as *E. coli* via heterologous pathways [62, 151, 153, 154]. For the computation examples presented below, we therefore consider *E. coli*, equipped with the heterologous linear pathway for 2,3-BDO synthesis via α-acetolactate and (R)-acetoin [153] and computed different sets of MCSs to design strains with growth-coupled 2,3-BDO synthesis. Using different scenarios, we first benchmarked the performance gain in the MCS computation through the introduced compression rules for GPR associations against the conventional approach in both a core and a genome-scale model of *E. coli*. Afterwards, we exemplify the use of multiple target and desired regions and the possibility to consider combinations of gene deletions and additions to obtain different strain designs including substrate co-feeding strategies for 2,3-BDO synthesis.

4.6.1 Effect of GPR rule compression on the MCS computation performance

MILPs used in the context of different strain design techniques are often very large and complex, especially when applied in genome-scale networks. Loss-free compression techniques may enhance the performance of these MILPs and are often even essential to reach acceptable runtimes and to obtain a significant pool of solutions. Network compression has become a standard tool in many constraint-based calculations [148, 149], including MCS approaches [31]. In addition, we herein introduced a set of rules for compressing GPR associations, which are relevant when metabolic interventions are directly calculated at the gene instead of the reaction level. Thus, GPR and network compression tackle different model redundancies.

To benchmark the effectiveness of GPR and/or network compression, we compare the impact of both compression routines on the final problem size, the MILP runtime and the computational overhead in a small-scale (a slightly adapted version of *EColiCore2*, a model of the central metabolism of *E. coli* presented by Hädicke and Klamt [110]) as well as in a genome-scale (*i*ML1515 model by Monk et al. [88]) MCS computation setup (Table 4.1). Within these

two models, which were both extended with GPR rules as described in section 4.5, we fully enumerate gene MCSs up to the size of 9 (core) and 7 (genome-scale) gene knockouts to obtain growth-coupled strain design that produce 2,3-BDO from glucose with a minimum product yield of at least 30 % of the theoretical (stoichiometric) maximum under a minimum attainable growth rate of $0.05\,h^{-1}$. As explained in section 4.1, these constraints can be expressed by one target and one desired region. We assume that all genes are targetable. Furthermore, the oxygen uptake reaction remains targetable so that the algorithm may also find anaerobic strain designs. Although (non-exhaustive) sampling of single or multiple MCSs is possible and sometimes preferred in practice [31], we here perform a full enumeration up to the given maximum MCS size to ensure that identical pools of solutions are delivered for the different runs. For the MCS computation benchmarks with core and genome-scale setups, we used MATLAB 2019b, together with IBM ILOG® CPLEX® 12.10. The source code for these computations is provided in Source Code A.2 in the Appendix. With the specified target and desired regions, there exist 6025 solutions (MCSs) in the core and 2632 solutions in the genome-scale setup (Table 4.1).

A comparison of the MILP runtimes for the four different runs (with/without GPR rule compression and with/without network compression) reveals the effectiveness of the state-of-the-art network compression procedure but also demonstrates the advantages of an extended compression by the herein presented GPR compression procedure. In fully uncompressed models, the computation time for MCS lies in the range of days even in the core model, while the genome-scale computation is not possible at all. The benchmarked scenarios show that at least one sort of problem compression, network or GPR rule, is necessary to achieve reasonable runtimes of the MCS algorithm. The different compression steps reduce the number of reactions and metabolites by up to more than 90 %, which in turn also leads to an enormous reduction in the number of MCSs found in the compressed network (e.g., 89 vs. 6025 in the core network), which are expanded to the full set of MCS in a postprocessing step. Each MCS of the compressed network is a representative of an MCS equivalence class (see chapter 5). As expected, MCS computation with combined network and GPR compression performs best: compared to the traditional pure network compression, it runs about 15 times faster in the core setup and reduces the runtime of the genome-scale setup by a factor of about 4. The overhead of the compression routines is comparably small (especially in genome-scale networks ($< 7\,\%$)) and pre- and postprocessing times shorten when GPR- and/or network compression are used. The additional speedup achieved by the GPR rule compression is vital for the genome-scale computation of gene MCSs, since the runtimes can range from hours to days and up to weeks. It should also be noted that we considered all 1515 genes in the genome-scale model as potential targets, in contrast to many other studies where the set of targetable genes (or reactions) is often manually reduced to shrink the problem size to manageable dimensions.

The new approach for calculating MCSs with integrated GPR associations was then compared with the gMCS approach of Apaolaza et al. [107, 108]. The latter determines in a

preprocessing step a mapping of minimal gene knockout sets required to block at least one reaction. As the gMCS method can so far only deal with target reactions but not with desired regions, we used the calculation of synthetic lethals (SLs; combinations of reaction knockouts that block growth) in the genome-scale *i*ML1515 network as benchmark. We enumerated all SLs up to size 4 and found with both methods the identical set of 889 MCS, which confirms their consistency. For the runtimes, we determined 65 minutes with our methods and 163 minutes with gMCS.

Table 4.1: Benchmark results of different compression scenarios in two MCS computation setups (core and genome-scale model of *E. coli*) for growth-coupled 2,3-BDO synthesis. The rows "#Genes" and "#GPR rules" refer to the respective number of genes and GPR rules eventually integrated in the network after GPR rules compression (as long as the latter is conducted; otherwise it refers to the original number of genes and GPR rules); likewise, the rows "#Species", "#Reactions", "#Targetable reactions/genes" refer to the numbers of species, reactions, and targetable genes/reactions, respectively, after integration of the (compressed/non-compressed) GPR rules and/or after network compression and before setting up the MCS MILP. The computed MCSs are listed in Table A.5.

Target: $Y_{2,3\text{-}BDO/glc} \leq 0.3\, Y^{max}_{2,3\text{-}BDO/glc}$, Desired: $r_{BM} \geq 0.05\,\mathrm{h}^{-1}$								
	***EColiCore2* (max MCS size: 9)** 502 reactions, 486 species, 508 genes, 683 GPR rules				***i*ML1515 (max MCS size: 7)** 2715 reactions, 1879 species, 1516 genes, 3828 GPR rules			
Compression(s):	No compression	Network compression	GPR compression	Network + GPR compression	No compression	Network compression	GPR compression	Network + GPR compression
#Genes (after compression)	508	508	114	114	1516	1516	649	649
#GPR rules	683	683	163	163	3828	3828	1531	1531
#Species	1435	244	682	107	5616	1207	3529	1068
#Reactions	1690	500	799	233	8103	2513	4995	2245
#Targetable reactions/genes	251	181	64	58	1320	631	574	528
#MCSs found	6025	92	169	89		177	190	177
#MCSs found after decompression	6025	6025	6025	6025	**Not finished (>100 h)**	2632	2632	2632
Runtime [min]	**5199.7**	**178.9**	**41.0**	**11.5**		**707.8**	**846.9**	**180.3**
Pre- and postprocessing overhead [min]	3.8	6.4	8.4	6.2		18.8	12.3	11.7

4.6.2 Applying the new MCS features for strain design

Next, we used the 2632 strain designs found for 2,3-BDO synthesis from glucose in the genome-scale model as a reference to compare it with the new features of the MCS framework such as multiple target and desired regions, gene/reaction additions and substrate co-feeding. The different computation setups are listed in Table 4.2. As a first extended setup, we constrained the genome-scale search of MCSs for growth-coupled 2,3-BDO synthesis by introducing a second desired region that requires any mutant to cope with an increased ATP maintenance rate ($r_{ATPM} \geq 18\,\mathrm{mmol\,g_{CDW}^{-1}\,h^{-1}}$; scenario 2 in Table 4.2). This supplementary constraint could be used for developing more stress resistant strains or to prime strains for ATP-wasting boosted production [62, 130, 133, 136]. The enumeration of gene MCSs for this problem up to size 7 returned 544 strain designs, expanded from 25 MCSs found by the MILP in the compressed network. As expected, for reasons of consistency, all 544 MCSs were contained in the previously found set of 2632 MCSs. To ensure that the found solutions were not only correct but also complete, we filtered the 2632 original MCSs (from Table 4.1; see also scenario 1 in Table 4.2) by their compliance with the ATP maintenance constraint and obtained an identical set of 544 MCSs. Interestingly, all these MCSs with increased ATP production capabilities use aerobic conditions, while all remaining MCSs involve the removal of the oxygen supply.

Table 4.2: Different genome-scale MCS computation setups for growth-coupled 2,3-BDO synthesis.

	Scenario	Model	Target Regions	Desired Regions	Max. MCS size/cost	Runtime [h]	Solutions
1	Standard (cf. Table 4.1)	*i*ML1515	1	1	7	3.0	2632 (Table A.5)
2	High ATP maintenance	*i*ML1515	1	2	7	3.3	544 (Table A.6)
3	Multiple substrates	*i*ML1515	2	1	6	35.3	2611 (Table A.7)
4	Multiple substrates + high ATP maintenance	*i*ML1515	2	2	6	17.3	995 (Table A.8)

To validate the operability of MCS searches also with multiple target regions and with reaction/gene additions, we included options to use alternative substrates or to use co-feeding of multiple substrates (Table 4.2, scenario 3). We extended the network with uptake reactions for acetate and glycerol and tagged them as addable, together with glucose. Accordingly, the algorithm now finds MCSs that use glucose, glycerol, or acetate or any combination of these. We adapted the previously defined yield constraint to account for all three substrates. The individual weight of the substrates (and the product) in the yield function was determined by the number of carbon atoms in the corresponding molecules. As a result, the yield constraint

expresses a minimum threshold for the carbon recovery of the combined substrates in the final product. Acetate, however, takes on a dual role in this scenario because some strain designs might require it as a co-substrate while others require its secretion as a by-product. To account for this equivocalness, the minimum yield constraint was expressed by two target regions, as described in section 4.1, one for acetate by-production (eq. (4.17)) and one for the case when acetate is used as a (co-)substrate (eq. (4.18)):

$$\textbf{T1}: \; 4\, r_{2,3\text{-}BDO,ex} + 0.3 \cdot \frac{4}{6} Y^{max}_{2,3\text{-}BDO/glc}(6\, r_{glc,up} + 3\, r_{glyc,up}) \leq 0, \qquad r_{ac,up} = 0 \quad (4.19)$$

$$\textbf{T2}: \; 4\, r_{2,3\text{-}BDO,ex} + 0.3 \cdot \frac{4}{6} Y^{max}_{2,3\text{-}BDO/glc}(6 r_{glc,up} + 2 r_{ac,up} + 3 r_{glyc,up}) \leq 0, \;\; r_{ac,ex} = 0. \quad (4.20)$$

Note that in the specification above, we follow the definition of *i*ML1515, where the exchange reactions are positive if they export a compound and negative if they import it. The carbon yield threshold was again set to 30 % of the maximum stoichiometric carbon yield, referring to glucose as the substrate. This definition of the yield threshold ensures that the subset of the found MCSs that rely solely on glucose is directly comparable to the previously found 2632 MCSs. In the latter case, the carbon-based yield formulation is equivalent to the previously used molar yield constraint. In the same manner, the desired region was now specified as biomass yield per carbon-weighted substrate:

$$\textbf{D1}: \qquad -r_{BM} - 0.005 \cdot \frac{1}{6}(6\, r_{glc,up} + 2\, r_{ac,up} + 3\, r_{glyc,up}) \leq 0. \qquad (4.21)$$

Again, an additional specification of the requested minimal 2,3-BDO yield is not necessary because all (target) regions with lower product yields will be eliminated by the MCSs while the desired Region **D1** ensures that other flux vectors, then with high product yield, still exist.

The described scenario with multiple substrates leads to a largely extended solution space of feasible flux vectors, which, together with the more complex description with two target regions, render the MCS computation significantly more complex. For this reason, we enumerated MCSs only up to the cost of 6. In this scenario, gene knockouts were attributed with a cost factor of unity, while the addition of substrates and the removal of the oxygen supply reaction remained "free" (zero costs).

The MILP for finding MCSs for growth-coupled synthesis of 2,3-BDO on multiple substrates returned 574 solutions in the compressed network. Removing the redundant (non-minimal) cut sets resulted in a solution pool of 184 MCSs which were then expanded to 2611 MCSs (see scenario 3 in Table 4.2). This pool contained 2112 MCSs with glucose as the sole substrate, 24 MCSs relying on glycerol alone and 475 MCSs that utilize glucose with co-feeding of acetate (these solutions are further discussed below). As a proof of consistency, it was verified that the 2112 MCSs with glucose as sole substrate are all contained in the pool of 2632 MCS from the initial (benchmark) setup (the additional 520 MCSs require 7 gene knockouts, whereas only 6 were allowed in the multiple substrates scenario).

In the last scenario 4 (Table 4.2), we extended the previous setup for multiple substrates (with two target regions and one desired region) by a second desired region that demands the protection of high ATP maintenance rates as previously used in scenario 2 ($r_{ATPM} \geq 18\,\mathrm{mmol\,g_{CDW}^{-1}\,h^{-1}}$). This computation returned 955 non-redundant MCSs. Among these solutions were 24 purely glucose-based MCSs, which are identical to the 24 MCSs of size 6 contained in the MCS solution pool of scenario 2 in Table 4.2, again confirming correctness and completeness. Another set of 24 MCSs, solely based on glycerol as the substrate, is identical to the set of glycerol MCSs found in scenario 3 with two target and one desired region. The other 947 of the 995 MCSs were supersets of the formerly found MCSs in scenario 2. They nevertheless represent truly distinct and minimal intervention strategies for this particular scenario: all of these MCSs add the supply of glycerol and/or acetate to satisfy the higher ATP maintenance rate. 208 of these MCSs contain further cuts because the expansion of the solution space by the additional substrates would otherwise render the target region feasible again and thus allow lower product yields. All MCSs computed in the four scenarios in Table 4.2 have been ranked according to the criteria proposed in [1] and are provided in the Tables A.5 to A.8. In the following, we discuss major principles of the found strain designs with or without substrate co-feeding in scenarios 1-3.

4.6.3 Key principles of the found intervention strategies for 2,3-BDO synthesis

The pool of the found MCSs with or without substrate co-feeding can be divided into three major classes, according to their growth-coupling mechanisms (Figure 4.6). 2088 of the 2632 MCSs with glucose as the sole substrate use anaerobic conditions (Figure 4.6, red) establishing growth-coupling by a suitable combination of redox balancing and exclusion of alternative carbon drains. Since the 2,3-BDO pathway (consuming one NADH) is imbalanced with glycolysis (yielding two NADH), these anaerobic strategies require at least one additional by-product to balance the cell's redox state (e.g. succinate). However, these MCSs must then also ensure that only a limited amount of the substrate can run to this additional side product, since otherwise only a low yield of 2,3-BDO would be achieved. The other 544 of the 2632 MCSs found in scenario 1 (with glucose as substrate) favor aerobic conditions (Figure 4.6, blue) and use oxygen as electron acceptor. In contrast to the anaerobic strategies, all pathways to alternative fermentation products as well as parts of the respiratory pathway (e.g., ATP synthase) must be blocked to prevent the complete oxidation of glucose to CO_2 in the presence of oxygen. When acetate is available as a co-feeding substrate (scenario 3), the extended MCS algorithm was able to identify completely anaerobic strain designs where redox balance could be established by reducing the provided electron acceptor acetate to ethanol (Figure 4.6, shown in green). Furthermore, it was able to identify alternative strain designs that replace glucose with glycerol as the sole carbon and energy source under aerobic conditions Table A.7).

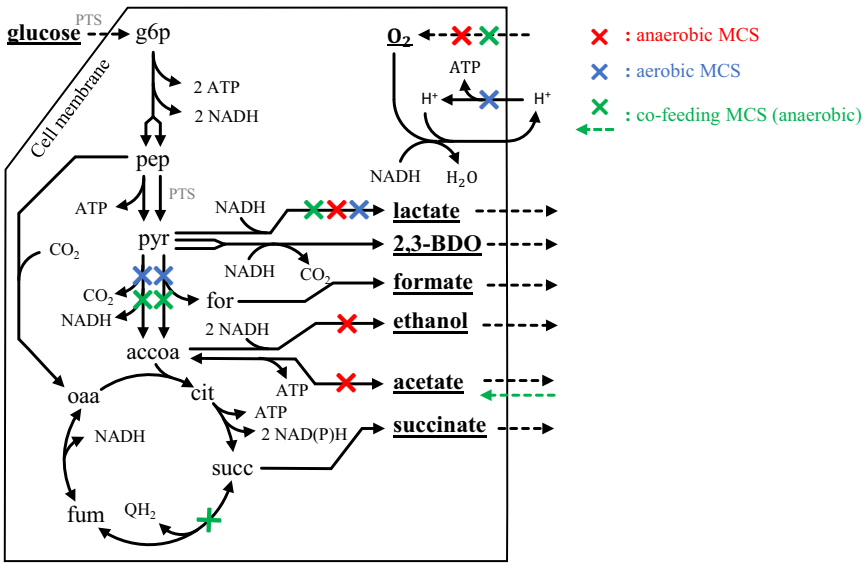


Figure 4.6: Main principles of MCS found for the growth-coupled production of 2,3-BDO in *E. coli*. Red: anaerobic production (e.g., scenario 1: geneMCS-611 (reaction MCS_043) in S1 Table: -O2, Δ*adhE*, Δ*adhP*, Δ*ldhA*, Δ*mgsA*, Δ*satP*). Blue: aerobic production (scenario 2: geneMCS-449 (reaction MCS_08) in S2 Table: +O2, Δ*aceE*, Δ*atpA*, Δ*deoC*, Δ*ldhA*, Δ*pflA*, Δ*pflC*, Δ*poxB*). Green: anaerobic production with substrate co-feeding (scenario 3: geneMCS-1039 (reaction MCS_095) in S3 Table: -O2, +acetate, Δ*aceE*, Δ*frdA*, Δ*ldhA*, Δ*pflA*, Δ*pflC*, Δ*pfo*).

4.6.4 MCSs for growth-coupled 2,3-BDO synthesis in two other genome-scale models

To finally demonstrate, that the extended MCS approach also works with other metabolic networks, including a eukaryotic one, we repeated all calculations for growth-coupled 2,3-BDO synthesis of *E. coli* (Tables 4.1 and 4.2) in the genome-scale networks of *Saccharomyces cerevisiae* (yeast-GEM [155]; 1147 genes and 3989 reactions) and *Pseudomonas putida* (*i*JN746 [156]; 746 genes and 1054 reactions; both included in Source Code A.2). We found similar trends in the benchmark computations, in particular, the beneficial effect of the GPR compression rules (see Table A.9). The results for each organism reviled the same consistency among the different scenarios that was already observed in the case of *E. coli*. Single-substrate strain designs could be found in all MCS setups, strain designs that rely on co-feeding were found for scenario 3 in *i*JN746 and yeast-GEM. The computation setup for both models and all four scenarios are provided in Source Code A.2.

4.7 Discussion

The presented features and developments generalize the framework of minimal cut sets and broaden its scope of applications, especially for metabolic network design. The first enhance-

ment allows the definition of multiple target (undesired) and multiple protected (desired) regions and enables now a precise tailoring of metabolic solution spaces of the strain to be designed. We are not aware of any strain design approach that allows such an unlimited flexibility in the description of arbitrary (especially non-convex) spaces of desired and protected metabolic behaviors. Specifying multiple target and desired regions thus represents an entirely new feature and facilitates the treatment of completely new classes of complex design problems. It is not only essential for the computation of co-feeding strategies, where a substrate may also act as potential byproduct but also allows one to demand distinct behaviors and responses to different (process) conditions. This could be used for finding growth-coupled strains designs that exploit ATP wasting, ensuring that the organism can withstand induced higher ATP demands, or, as shown in the example calculations, designing production hosts for multi-stage processes.

The second extension consists of a modified approach for integrating (compressed) gene-protein-reaction (GPR) associations in constrained-based models for the computation of intervention strategies. In the context of MCS calculations, so far, only few studies attempted to account for GPR associations. One example for such an approach is gMCS [107, 108]. In a preprocessing step, this algorithm assesses correspondences between gene knockouts and associated reaction deletions and integrates these relationships in a subsequent gene MCS computation [108]. The authors used their algorithm for computing synthetic lethalities. However, the gMCS algorithms does not support the direct computation of constrained MCSs, that maintain a desired behavior as needed, for example, for strain design applications. Moreover, compression techniques have not been proposed for reducing the dimension of the problem. Machado et al. [79] proposed a more general approach for integrating GPR rules in constraint-based metabolic network models. This approach uses pseudo-metabolites and pseudo-reactions to represent genes and their connections with enzymes and reactions in the metabolic model. Among other applications, this integrated representation was also used for the computation of gene MCSs [79]. In this chapter, this approach was adopted with a modification that ensures that the original flux space is preserved also in cases where specific flux bounds (different from zero or infinity) are given. Generally, the integration of GPR associations into the metabolic network with this approach largely increases the network size, which may limit its application in genome-scale networks. This overhead is reduced by a set of compression rules that remove redundancies in GPR associations and genes, many of which cannot be addressed by classical network compression alone. The benchmark calculations showed a performance benefit with a factor of 4-15. Importantly, these compression rules for GPR associations can as well be applied in conjunction with other strain optimization methods.

As another major extension, the MCS algorithm is now capable of combining reaction/gene deletions with additions and can also account for arbitrary cost factors for each intervention. This feature was exemplified for finding substrate co-feeding strategies for growth-coupled strain designs. However, other applications include the search for strain de-

signs where, in combination with suitable gene/reaction knockouts, heterologous reactions or pathways are added to the network to achieve optimal production behavior with minimal intervention costs. Clearly, as demonstrated in the application example, offering a larger number of addable reactions may vastly increase the solution space, especially when searching for co-feeding strategies on multiple possible substrates. Hence, the repository of optional insertions should be limited to a manageable number of (preselected) reactions or pathways. Generally, a full enumeration of all MCSs in genome-scale models is typically only feasible up to a certain maximal number of interventions (for example, an enumeration of MCSs with 9 or more interventions for 2,3-BDO synthesis in *i*ML1515 was not possible due to memory overflow). On the other hand, the calculation of single MCSs is often possible also in very large networks.

Some presented features, in particular substrate co-feeding or reaction additions, have partly been addressed also in previous studies that used bilevel optimization approaches [90–93, 115]. Yet, neither individual cost factors nor reaction additions and co-feeding strategies have so far been used in the more general framework of MCS. Moreover, co-feeding strategies, where an external metabolite may act as (addable) substrate or byproduct, requires the definition of multiple target regions and can thus only be handled by the extended MCS approach.

The new algorithmic developments have been integrated in the free open-source package *CellNetAnalyzer* (see chapter 6). The extended MCS framework is backwards compatible and can be applied, out of the box, to previous MCS setups as used, for example, by Harder et al. (2016) [54] or von Kamp and Klamt (2017) [31]. The functionality of regulatory MCSs [105] can now directly be handled within the new framework: for each reaction, whose flux is considered to be adjustable, several copies with different flux bounds are provided as addition candidates. The algorithm will then return MCSs that propose the deletion of certain reaction/genes in combination with the addition of the regulated reaction with adequate bounds. The presented comparative setup showed that the new algorithm could generate results that are consistent with those from the existing gMCS method [107, 108], confirming its suitability for the identification of synthetic lethals on the genetic level. A potential direction for future research is the combination of the extended MCS framework with nullspace-based algorithms, a recently published alternative to the Farkas-lemma-based MCS algorithm [103, 104], which holds promises to further reduce the MILP size and thus to further speed up MCS computation.

The new developments were exemplified for the MCS framework by computing strain designs for the growth-coupled production of 2,3-butanediol in *E. coli*, *S. cerevisiae* and *P. putida*. Here, benchmark calculations showed a clear benefit of the new compression rules for GPR associations. The analysis of the computation results for *E. coli* showed that the new features allow the identification of qualitatively new strain designs that could not be generated with existing methods. Previous experimental works favored the use of microaerobic conditions for 2,3-BDO production in *E. coli* [153, 154, 157]. Strain designs similar to the

ones employed could also be found among the computed strategies. Alternative fermentation pathways were blocked, e.g. by knockouts of *ldhA*, *adhE* and *pta*. In addition, the excess of reduction equivalents arising from the unbalanced net stoichiometry of the 2,3-BDO synthesis pathway must be handled by the (limited) formation of another byproduct. This includes CO_2 if some amount of oxygen is supplied. Yet, these microaerobic fermentation strategies, which balance the NADH surplus through respiration, are highly sensitive to the ratio of oxygen and substrate uptake rates. Higher oxygen uptake rates lead to an increased CO_2 and biomass production and consequently to a decreased 2,3-BDO yield. Low oxygen uptake rates, on the other hand, limit NADH recovery and result in poor cell viability, reduced 2,3-BDO productivity or increased synthesis of reduced by-products like succinate [154]. Our computed strain designs confirm that the simultaneous synthesis of reduced by-products is required to balance the reduction equivalents when operating under fully anaerobic conditions. However, the supply of a potential electron acceptor (e.g. acetate) permits the recovery of reduction equivalents under anaerobic conditions without losing the primary substrate in redox-balancing by-products. Microaerobic strategies for 2,3-BDO synthesis could also be found by the new MCS approach by offering several addable oxygen uptake reactions with different maximum uptake rates, which can then be combined with suitable gene deletions to find production strains that can operate under limited oxygen supply.

5 Characterizing and ranking computed metabolic engineering strategies

Many strain optimization methods generate large pools of alternative intervention strategies to meet their strain design goals [60, 85, 91, 124]. A necessary step between model-driven design and experimental implementation is therefore an extensive screening of the proposed strategies to identify the candidate with the best combination of low experimental effort and high production performance. While several approaches have been developed for assessing native and heterologous product synthesis pathways for metabolic engineering (for a review see Wang et al., 2017 [158]), we found only a single study [159] that addressed the problem of a systematic characterization and ranking of intervention strategies (ISs).

In this chapter, we present a catalog of ten criteria to characterize growth-coupled strain designs. Several of these criteria are new and go beyond standard metrics. The presented criteria can be used, in a first step, to preselect certain strategies if some properties are essential or imply exclusion. We then propose a scoring scheme to rank all remaining strategies, facilitating a final selection for experimental implementation (Figure 5.1). We illustrate our criteria and the ranking scheme by two different case studies, where we computed ISs in a genome-scale model of *E. coli* for the substrate-uptake-coupled (and thus also growth-coupled) production of the amino acid L-methionine and of the heterologous product 1,4-butanediol. In these case studies, the respective pools of ISs were computed as minimal cut sets, but the presented criteria and ranking scheme could as well be applied to outputs of other algorithms.

5.1 Characterization of metabolic engineering strategies by 10 different properties

We assume that a set of ISs has been computed by an appropriate strain design algorithm based on a constraint-based metabolic model of the respective wild-type production organism. Dependent on the chosen method, ISs usually contain reaction or gene knockouts, which narrow down the flux space, but they may also comprise flux up- and downregulations or the insertion of heterologous metabolic reactions and pathways. We obtain mutant models from the wild-type by introducing the respective interventions (e.g., by setting the flux bounds lb_i and ub_i to 0 if reaction r_i is knocked-out). The flux space of those models is then analyzed with

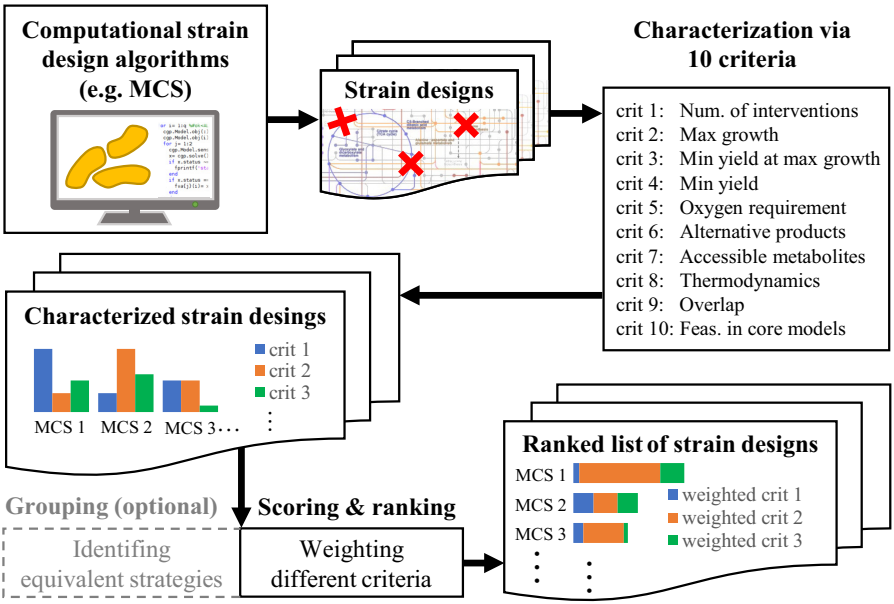


Figure 5.1: Overview of the proposed characterization criteria and ranking procedure for metabolic intervention strategies.

a variety of methods to characterize (and later rank) the given ISs with respect to their required performance, robustness and experimental effort.

Property 1: Number of required interventions (#*int*)

The number of required interventions is an obvious measure for the future experimental effort and should be preferably small. Nevertheless, the importance of this measure compared to the other criteria may vary and depends on the time and means available for implementation.

Property 2: Maximum growth rate (r_{BM}^{max})

The maximum growth rate (r_{BM}^{max}) is a measure of a mutant's viability. In most strain design methods, the lower limit for the maximum growth rate (or biomass yield) is defined as a constraint for computing the ISs to guarantee acceptable growth rates. The importance of the maximum growth rate depends on the chosen process type and process parameters. Higher growth rates of the mutant can be a driver for good volumetric productivity, especially in one-step batch fermentation with growth-coupled product synthesis. Combined with the minimum product yield, the maximum growth rate can be used to make statements about the productivity, either through a process simulation [87, 160] or the determination of related productivity benchmarks, such as the substrate-specific-productivity (SSP) [43].

Property 3: Minimum product yield at maximum growth rate ($Y_{P/S}^{min@r_{BM}^{max}}$)

For strain design algorithms focusing on weak coupling (such as OptKnock or RobustKnock), the product synthesis rate or product yield at maximum growth rate is of major importance since this is considered as the operating point of the mutant after adaptive evolution [161]. In the following, we focus on the product yield at maximum growth rate as a performance measure, although the product synthesis rate could be used as well. The product yield at the maximum growth rate can be a unique value or lie in a certain range, where we then consider the minimum guaranteed product yield as the most relevant measure. This value can be obtained through two subsequent simple optimizations, namely a maximization of the growth rate followed by a minimization of the product yield (through linear-fractional program; see eqs. (2.9) and (2.10)) under a fixed maximum growth rate.

Property 4: Minimum product yield ($Y_{P/S}^{min}$)

Sometimes mutant strains may not attain the flux distribution with maximum growth rate in experiments, e.g., due to possible regulatory or pathway capacity constraints. We therefore consider the global minimum product yield enforced by an intervention strategy as another criterion. In fact, some strain design algorithms even enforce a minimum product yield for all feasible flux vectors in the mutant, even at non-optimal growth (substrate-uptake-coupled production). This criterion thus quantifies the strength of the coupling. The minimum product yield can again be computed through LFP.

Property 5: Requirement of aerobic conditions (O_2)

Another criterion for evaluating an IS and the resulting process concerns the necessity of oxygen supply for the mutant strain. Anaerobic growth regimes are often easier to implement in large-scale bioreactors and then preferred. In fact, some ISs even demand anaerobiosis while others require oxygen. Furthermore, the preference for an aerobic or anaerobic process may also depend on pathway capacities and regulation [162], product inhibition, process stability and other factors.

Property 6: Number of alternative products that could disrupt coupling if secreted (#*alterProd*)

Growth-coupled ISs are sometimes not successful in practice because of unexpected by-product secretion, abrogating the stoichiometric coupling of growth and product synthesis. To quantify the robustness of a growth-coupled design, we determine the number of metabolites (alternative products) that would lead to a disruption of the coupling between growth and product synthesis if the cell was able to secrete or, at least temporarily, to accumulate them. The importance of this approach arises from the fact that there are numerous promiscuous transporters with varying specificity. The citrate transporter *CitT* is such an example [163].

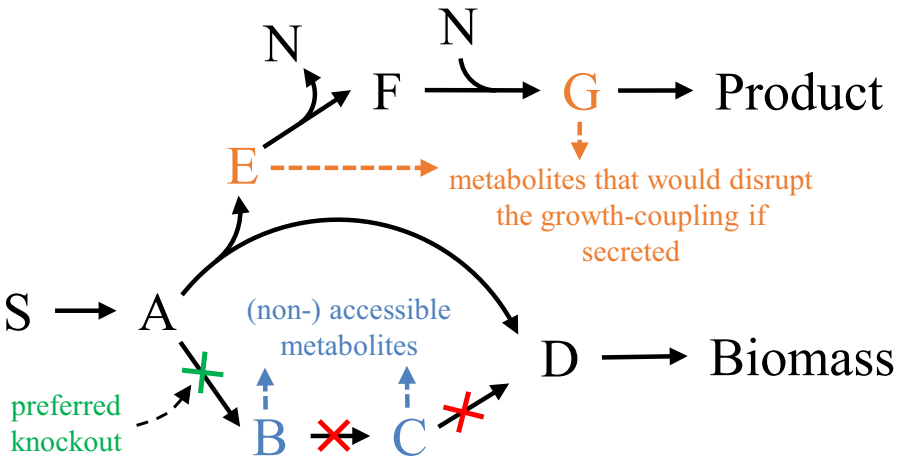


Figure 5.2: Example network to showcase the disruption of growth-coupling by product alternatives and the interception of undesired pathways at different reactions. Growth and production are coupled unless the excretion of the metabolites E or G (upper branch) is possible. In the lower branch there are three different cuts (green, red, red) that establish coupling of biomass and product synthesis. All of them cut the same pathway and lead to the same solution space but imply a different number of accessible metabolites. The green cut at the beginning of the pathway is preferred and has the lowest number of accessible metabolites. Choosing the green cut (or, less favored, one of the two red cuts), ensures GCP so that the product is synthesized when the cell grows, however, excretion of the metabolites E and G (but not F) would disrupt growth-coupling.

It mainly functions as citrate/succinate-antiporter but also shows an affinity towards other C4-dicarboxylates, such as fumarate or aspartate. However, established metabolic models often neglect secondary functionalities of transporters. For the wild type scenario and under most fermentation conditions, this assumption is realistic. Yet, in scenarios where other C4-dicarboxylates are accumulated, its export is possible [164] and there have been metabolic engineering approaches for fumarate production, that rely completely on the native *E. coli* C4-dicarboxylate transport systems [165].

In order to decrease the number of necessary interventions and because usually only few metabolites are excreted under a given condition, the set of potential product sinks can initially be reduced to the main fermentation products [31]. Searching for alternative products may then help to identify potential leaks that could abrogate coupling.

To test whether the excretion of a given metabolite could potentially disrupt growth-coupling, the model is temporarily extended by an export reaction for this metabolite and, with FBA, it is verified whether the desired growth-coupling is then still existent with desired specifications, e.g. with minimum product yield or production rate. This is done, one by one, for all metabolites that do not yet have an exchange reaction. The total number of metabolites that remove the coupled production of the actual target product gives a measure for the robustness

of the IS. An example is illustrated in Figure 5.2. Using the procedure described above reveals that the secretion of metabolites E or G (orange) would disrupt strong coupling, while F is not a possible alternative product because the cofactor N could then not be balanced by the cell.

Mutants that hold fewer alternative products tend to have a growth-coupling that is more robust, for example, because the coupling mechanisms, such as cofactor regeneration, occur at the end of the product synthesis pathway. This approach also allows one to exclude ISs with by-products that are likely to occur or have already been observed in previous experiments. Sometimes only the combined secretion of multiple alternative products leads to a disruption of growth-coupling, however, we do not consider those combinations herein because they are less likely and their detection requires higher computational effort. Furthermore, if the likelihood of secretion can somehow be quantified, the approach could also be extended by associating individual penalty scores for each metabolite. For example, metabolites that are phosphorylated or bound to Coenzyme A are already less likely to be secreted. A useful extension for the described approach is the identification of the nature of the respective coupling strategy. For this purpose, artificial reactions that recover co-factors such as ATP/ADP, NADH/NAD or NADPH/NADP "for free" from the respective unphosphorylated (ADP) / phosphorylated (ADP) or oxidized (NAD(P)) / reduced (NAD(P)H) form. If growth-coupling is, for example, disrupted by the integration of an NADH oxidizing reaction, this would indicate that the underlying coupling mechanism relies on the balance of reduction equivalents.

Equivalence classes of intervention strategies

In practice, different ISs can lead to identical solution spaces. The reason for this is the nature of the steady-state assumption. For example, a linear pathway can be interrupted through a cut of any of the sequential reactions. An example is again given in Figure 5.2 where the pathway from A to D can be interrupted by three different cuts. All three cuts (the green and both red cuts) interrupt this pathway and yet lead to the same solution space. Using this criterion, a pool of ISs can always be partitioned into (equivalence) classes, where all ISs of one class have identical solution spaces. To identify equivalence classes, a flux variability analysis is performed for each IS. All ISs with identical flux ranges belong to the same class. It is reasonable to consider only one representative strategy for each class to avoid ranking of redundant solutions. However, it remains to be specified what the best strategy of each class is. This is related to Property 7.

Property 7: Number of accessible metabolites (*#accessMet*)

Regarding linear pathways, an interruption at an intermediate step or at an endpoint of the pathway (red cuts in Figure 5.2) can lead to an undesired accumulation of intermediate metabolites. Therefore, it is advantageous to interrupt pathways at their branching point (green cut in Figure 5.2) and thus to minimize the total number of metabolites that can be produced. We therefore propose the number of accessible metabolites as a criterion to assess the risk

of accumulation (and possible secretion) of metabolites in a given strain design. We suggest identifying the (best) representative for each IS class as the IS with the lowest number of accessible metabolites. Accessible metabolites are all those metabolites that can, in principle, be synthesized by the reactions of a metabolic network. Note that this also includes metabolites for which, e.g. due to certain proposed interventions, no further metabolization pathways or sinks (excretion reaction) exist in a redesigned network. In Figure 5.2, the three different knockouts lead to the same solution space, yet, the number of accessible metabolites differs for each of them and the strategy with the smallest number (the green cut) holds the lowest risk for the accumulation of an intermediate product because the undesired pathway is interrupted at its root. We hence would take the green cut as the representative strategy for this class. In the rare case that multiple strategies of an equivalence class have the same minimal number of accessible metabolites, one may consider only one representative selected by additional criteria (see case study below) or all of them for further evaluation.

To test whether an intracellular metabolite is accessible in a mutant, the model is extended by a sink reaction for this metabolite. The steady-state constraint $\mathbf{N}\,\mathbf{r} = \mathbf{0}$ (eq. (2.26)) is then replaced with the weaker constraint $\mathbf{N}\,\mathbf{r} \geq \mathbf{0}$, hereby allowing the accumulation of metabolites. This ensures that metabolites are classified as accessible as well if their synthesis requires simultaneous accumulation or excretion of other metabolites. An optimization maximizing the flux through the new sink reaction is then performed to see whether this flux can become non-zero, indicating that the metabolite is accessible. This check is done successively for all metabolites, delivering the total number of accessible metabolites.

As suggested by [117], the undesired interruption of reactions at downstream positions of linear pathways could already be avoided during strain design computation. This would not only reduce the number of relevant strategies but also speed up the computation. However, sometimes it is not ad hoc possible to define beginning and end of the respective product synthesis pathway(s), as this may depend on the chosen intervention strategies.

Property 8: Thermodynamics: Optimal max-min driving force (*OptMDF*)

Thermodynamic pathway analysis has been used before in other studies as a feasibility constraint for pathway prediction [45] and as a criterion for pathway ranking [166, 167]. The metric of max-min driving force (MDF), introduced by Noor et al., 2014 [168], is an optimization-based technique for determining the maximum thermodynamic driving force and the thermodynamic feasibility of a given metabolic pathway under best-case conditions. The concept of MDF was already successfully applied to genome-scale net-works to rank and discriminate biosynthetic pathway candidates for expanding metabolic networks [169]. However, MDF in its original form can only be used to test thermodynamic feasibility of one given pathway. We therefore use the OptMDFpathway method, recently introduced in [170], to find the flux vector with growth-coupled product synthesis that yields the maximum MDF (optMDF) in the entire solution space of the mutant. This quantity will be used as a ranking criterion for

the respective intervention strategy. In addition, if the computed optMDF is smaller than zero, then the corresponding strategy leads to a mutant without any thermodynamically feasible flux distribution and can thus be removed from the pool.

Property 9: Overlap with other intervention strategies (*overlap*)

Regarding experimental implementation, it is advantageous to lower the initial risk of failure by choosing one with many fallback options. To have a measure that correlates with the number of fallback options, we quantify the overlap of an IS with others. We first count the number of occurrences of a certain intervention in the entire pool of ISs. The overlap measure for each strategy is then calculated as the sum of the occurrences of all its interventions divided by the number of interventions. A high overlap measure indicates more potential fallback options, which is especially useful when an iterative step-by-step approach is followed in the experimental implementation of the strain design [54]. When an IS is picked, the first interventions to be implemented would be those with the highest scores (many overlaps), moving successively towards the lower scores (few overlaps), so that an alternative strategy could be used if a strategy change becomes necessary (e.g., due to failure of growth).

Property 10: Feasibility in reduced models (*core-model*)

The underlying mechanisms of different growth-coupled ISs can sometimes be hard to identify. Some recurring patterns are cofactor regeneration, proton balancing, carbon branching and interruption of alternative pathways [116]. Nevertheless, the metabolic routes that are finally used for the synthesis of the product of interest are diverse and, in a genome-scale model, may not be limited to the central carbon metabolism and also comprise pathways with presumably smaller capacities. In principle, constraint-based models could be adjusted to consider known maximum pathway capacities to avoid solutions that rely on high metabolic fluxes through secondary pathways. However, bounds of internal fluxes are usually not known. Furthermore, one may still be interested in solutions with low capacities when there is the prospect of increasing the pathway capacity, e.g., by amplifying the corresponding genes.

We therefore suggest evaluating the reliance of an IS on pathways with small capacities. For this purpose, we test the computed ISs in a reduced core network that only comprises the major catabolic and anabolic pathways. Alternatively, when a core model is not available, one may choose other criteria, e.g. relying on GO (gene ontology) terms, to mark certain reactions to have only limited capacity. These reactions should then not be essential in a designed strain.

In our examples, we used *EColiCore2*, a model of *E. coli*'s central metabolism derived from the genome-scale model *i*JO1366 [110]. As the *EColiCore2* is a subnetwork of *i*JO1366, we can easily test whether a knockout-based IS, computed in the full model, is also applicable in the smaller model: All interventions that target elements contained in the core model are implemented in the latter and FBA is then used to check whether the desired constraints of the mutant (e.g., minimum product yield when growing) are still fulfilled. If this is not the case,

the IS is classified as reliant on secondary pathways and will thus receive a lower score in the ranking procedure.

5.2 Scoring and ranking

The ten criteria described above characterize the different ISs and can be used to compare them with each other; first, based on each single criterion. To obtain a comprehensive quantitative measure S_i for each intervention strategy i, we suggest the calculation of an overall score from the individual scores $S_{i,j}$ for each of the ten criteria $j = 1...10$. Each criterion score $S_{i,j}$ can take normalized values between 0 and 1. The intervention strategy i that takes on the most unfavorable value on the criterion j attains the score $S_{i,j} = 0$ and the strategy k with the most favorable value is scored with $S_{k,j} = 1$. Concretely, if a high value $U_{i,j}$ of a specified criterion j is desirable (e.g., minimum product yield or maximum thermodynamic driving force (optMDF)), the score for the strategy i is determined by:

$$S_{i,j} = \frac{U_{i,j} - U_{j,min}}{U_{j,max} - U_{j,min}}. \tag{5.1}$$

In the case of a preferably low value $U_{i,j}$, the term is:

$$S_{i,j} = \frac{U_{j,max} - U_{i,j}}{U_{j,max} - U_{j,min}}. \tag{5.2}$$

Criteria with a preferably low value are the number of necessary interventions, the number of metabolites that disrupt the growth-product coupling and the number of accessible metabolites. The score for the aeration requirement S_{O_2} takes the value 1 for the anaerobic and 0 for the aerobic case. The score for the feasibility in reduced networks takes the value 1 if the strategy is feasible and 0 if it is infeasible in a reduced network. In the case that an optMDF value is negative, indicating thermodynamic infeasibility of the IS with given constraints, the individual score as well as the lower reference value ($U_{j,min}$) for the normalization are then set to zero.

The total score S_i for an intervention strategy i is then a weighted sum of its scores for the ten criteria:

$$S_i = \sum_j c_j S_{i,j}\,. \tag{5.3}$$

The weights c_j can be adapted to reflect (1) the particular relevance of each criterion in a given application and (2) the spread of the values. A smaller weight should be used for a criterion if its values are distributed over a very narrow range only, indicating low variability. In the simplest and uniform approach, one may set all weights to 1. The computed overall scores S_i can finally be used to rank the ISs.

5.3 Examples for the characterization and ranking of computed metabolic engineering strategies

Model setup and computation of minimal cut sets

To illustrate our approach, we computed growth-coupled strain designs for two different products: L-methionine and 1,4-butanediol produced via a native and a heterologous pathway [51] respectively. We used the *E. coli* genome-scale model *i*JO1366 [109] with minor modifications (provided in Appendix Source Code A.3) and minimal media with glucose as the sole substrate. The ISs in these examples are knockout-based and computed as MCSs by a variant of the algorithm used by [31]. The MCS computation setup is shown in the Appendix Table A.12. Undesired phenotypes that should be eliminated through the MCSs (so-called target flux vectors) were defined as flux vectors with a product yield lower than 30 % of the theoretical maximum yield of the respective product. For the protected (desired) phenotypes, we assumed a minimum product yield above this 30 % threshold and a lower limit of the maximum growth rate of $0.05\,\mathrm{h}^{-1}$. To speed up MCS computation, the genome-scale model was first compressed by merging sets of fully coupled reactions and by removing conservation relations [31]. For the MCS search we set an upper limit of 13 reaction knockouts per strategy and the calculation was aborted if (1) the solver finished the search, (2) a time limit of 24 h was exceeded, or (3) when 200 MCSs were found in the compressed model. For all computations, we used the API functions of *CellNetAnalyzer* [74, 111] (see also chapter 6) in MATLAB 2016a with IBM CPLEX 12.6.3 as MILP solver. Standard Gibbs energies $\Delta_r G'^{\circ}$ for computing the maximum MDF in the mutants (optMDF; property 8) were available for 744 reactions and taken from [170]. For assessing property 10, the feasibility in reduced models, we used the *EColiCore2* model [110] (Source Code A.3). The full set of computed MCSs and their properties and ranking can be found in the Appendix (Tables A.10 and A.11).

Preselection and ranking

The ranking of the computed ISs involved two steps. First, to avoid occurrence of many equivalent strategies in the final ranking tableau, a preselection was performed based on equivalence classes of MCSs (see section 5.1). All MCSs of one equivalence class lead to an identical solution space of steady-state flux vectors when applying the interventions in the model. To identify equivalence classes, we performed for each IS a flux variability analysis and grouped all MCSs with identical flux ranges in one class. We then selected one (the best) representative of each class which has the minimum number of accessible metabolites. If there are, within one class, several MCSs with a minimum number of accessible species, then, from these MCSs, the one with the highest overall score is selected as the representative.

After preselection, the MCS class representatives underwent the scoring and ranking procedure as described in section 5.2. We used the weighting coefficients $c_{\#int} = 1.5$, $c_{r_{BM}^{max}} = 1$,

$c_{Y_{P/S}^{min@r_{BM}^{max}}} = 1$, $c_{Y_{P/S}^{min}} = 1$, $c_{O_2} = 0.5$, $c_{\#alterProducts} = 0.5$, $c_{\#accessMet} = 0$, $c_{optMDF} = 0.5$, $c_{overlap} = 0.5$, $c_{core\text{-}model} = 0.25$. We thus chose slightly larger weights for the first four criteria to emphasize the number of interventions, the maximal growth rate and the minimum product yield during ranking. Note that the number of accessible metabolites is only used (and only reasonable) as criterion for the preselection of ISs in the equivalence classes, and the weight of this score is therefore set to zero in the ranking of the representatives.

5.3.1 Example 1: production of L-methionine

Regarding the global market size of the different amino acids, methionine holds the third place after glutamate and lysine. Main applications of methionine lie in livestock (feed additive, especially poultry farming), pharmaceutics and nutrition [171, 172]. Despite many attempts, there are only few successful examples of metabolic engineering approaches that lead to an established industrial bioprocess. In terms of strain design, one of the main hurdles lies in the complex regulation of the L-Methionine biosynthesis in bacterial hosts [173]. A pathway for L-methionine biosynthesis was already successfully deregulated in *E. coli* [171]. In contrast to the studies that treat the regulatory restrictions, our example focuses on potential knockout strategies that reroute the metabolic flux and to enforce the overproduction of L-methionine.

We extended the *i*JO1366 model with the L-methionine proton antiporter *YjeH* [174]. In total, we found 258 intervention strategies (MCSs) with a minimum of 9 and a maximum of 13 cuts. The MCSs can be grouped in 37 equivalence classes. From each class, we picked the best representative as described above. Scoring and ranking was done as described in section 5.2 using the weighting coefficients previously given. The highest ranked candidate, MCS 41, consists of the 10 reaction deletions *TPI, ENO, CYSDS, GPDDA2, GPDDA2pp, HCYSMT, LSERDHr, MTHFC, SERD_L* and *TRPAS2* which can be established through the knockout of the genes *tpiA, eno, mmuM, folD, ydfG, sdaA, glpQ, sdaB, metC, ugpQ, tnaA* and *tdcG*.

Figure 5.4 shows the performance of five exemplary MCSs for the different criteria. As shown in Figure 5.4B, different MCSs can lead to growth-rate-product-yield space (GRPY) with similar shapes. Yet, the further assessment of the candidates reveals that a characterization solely based on these trade-off plots is limited. MCSs with similar GRPY spaces can still perform very differently on other criteria, such as the thermodynamic driving force, the number of possible by-products or the number of necessary knockouts (Figure 5.4A). For example, the highest ranked MCSs (MCS 41 - blue) is more robust and needs fewer knockouts than MCS 28 (purple), even though their GRPY spaces are almost identical. In fact, it should be noted that there is only a small variance in the minimum product yields of all MCSs nevertheless leading to different scores for this criterion. As mentioned earlier, in those cases with low variability, one could leave out the respective criterion/score by setting its weight to zero. While MCS 237 (green) is outperformed by MCS 144 (yellow) in the overlap score and the number of interventions, it is thermodynamically more favorable and offers a higher maximum growth

rate. Figure 5.4B also shows that none of the five selected MCSs could be used in the *E. coli* core model *EcoliCore2*, indicating that the computed ISs rely on pathways outside the central metabolism with possibly lower capacities. A closer analysis revealed that the crucial pathway in this case starts with the glycine C-acetyltransferase reaction *GLYAT* that finally reintroduces glycine into the pyruvate metabolism. This pathway is not contained in the *EColiCore2* model.

We further analyzed the computed MCSs to understand the coupling mechanisms. In all MCSs the coupling was established through a combined deletion of the triose-phosphate isomerase reaction and one of the two final glycolysis steps catalyzed by phosphoglycerate mutase or enolase. As a result, sugar degradation has to take place along the Entner-Doudoroff pathway (ED) which forces the carbon flux to split into pyruvate and glyceraldehyde-3-phosphate (G3P) from where the metabolic pathway to PEP and pyruvate is blocked. The flux through pyruvate can join the TCA cycle to generate reduction equivalents, mainly used for ATP synthesis via respiration. G3P cannot enter the TCA cycle directly and can only be drained via the pathways to serine and aspartate which are then further metabolized to methionine, consuming NADPH produced in the ED pathway (Figure 5.3). In addition to *tpi* and *eno* / *pgm*, essential knockouts that occur in all MCSs are the reactions of the cysteine desulfhydrase (*CYSDS*), L-serine deaminase (*SERD_L*) and tryptophanase (*TRPAS2*) by which amino acid degradation pathways are blocked. While all MCSs contain the essential knockouts described above, specific interventions in the MCS now enforce different routes through the amino acid pathways with certain flux ratios. For the flux rerouting, different combinations of reaction knockouts are possible, bearing different advantages and disadvantages. MCS 28 and 74 are less robust, as there are many more intermediates (e.g., up to 218 for MCS 28) which, if secreted, can abrogate growth-coupling.

As described in section 5.1 (property 6), it was also tested, whether coupling is still existent when ATP is available at no cost, or electron sources or sinks are added that reduce or oxidize NAD(P)/H. We found that an artificial supply with ATP leads to the disruption of the growth-coupling, suggesting that ATP synthesis (and therefore growth) is coupled to methionine synthesis. An external supply with electrons would also abolish coupling because the electron surplus can be used to increase ATP synthesis via respiration. On the contrary, giving the mutants the option to drain electrons does not affect the coupling. The thermodynamic analysis with optMDF and FBA predicts the infeasibility of 132 out of 258 computed knockout strategies (18 out of 37 equivalence classes). These strategies rely on the reversed malate oxidase reaction (*MOX*), which is thermodynamically infeasible under the considered physiological conditions. Furthermore, in all strategies, the essential knockouts of the genes *tpiA* and *eno* or *pgm* represent serious interventions in the core metabolism of the cell. While the knockout of *tpi* has been proven to be feasible in *E. coli* [175], the additional knockouts of the enolase or phosphoglycerate mutase genes may be difficult targets. It was reported that the deletion of *pgm* leads to a mutant that could not grow on minimal medium

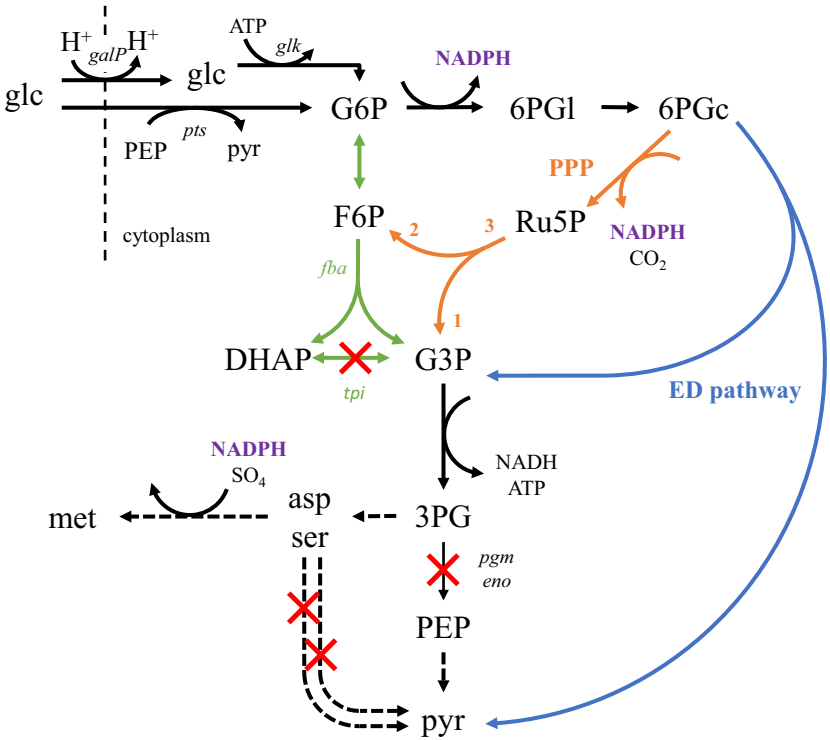


Figure 5.3: Strategies to couple growth with L-methionine synthesis in *E. coli* as revealed by the computed MCSs.

[176]. A reason may be the limited flux capacity of the ED pathway in *E. coli*, which could be overcome by additionally deregulating the ED pathway or by overexpressing its enzymes. Another potential drawback of the found intervention strategies is that the glucose uptake and activation relies on the proton symport (*galP*) and the glucokinase (*glk*) reaction because there would be not enough PEP available to use the PTS system [177]. Hence, a key requirement for implementing the found strain design strategies is to extend the capacity of the ED pathway in *E. coli*. The list of the computed and ranked MCSs is provided in Table A.10.

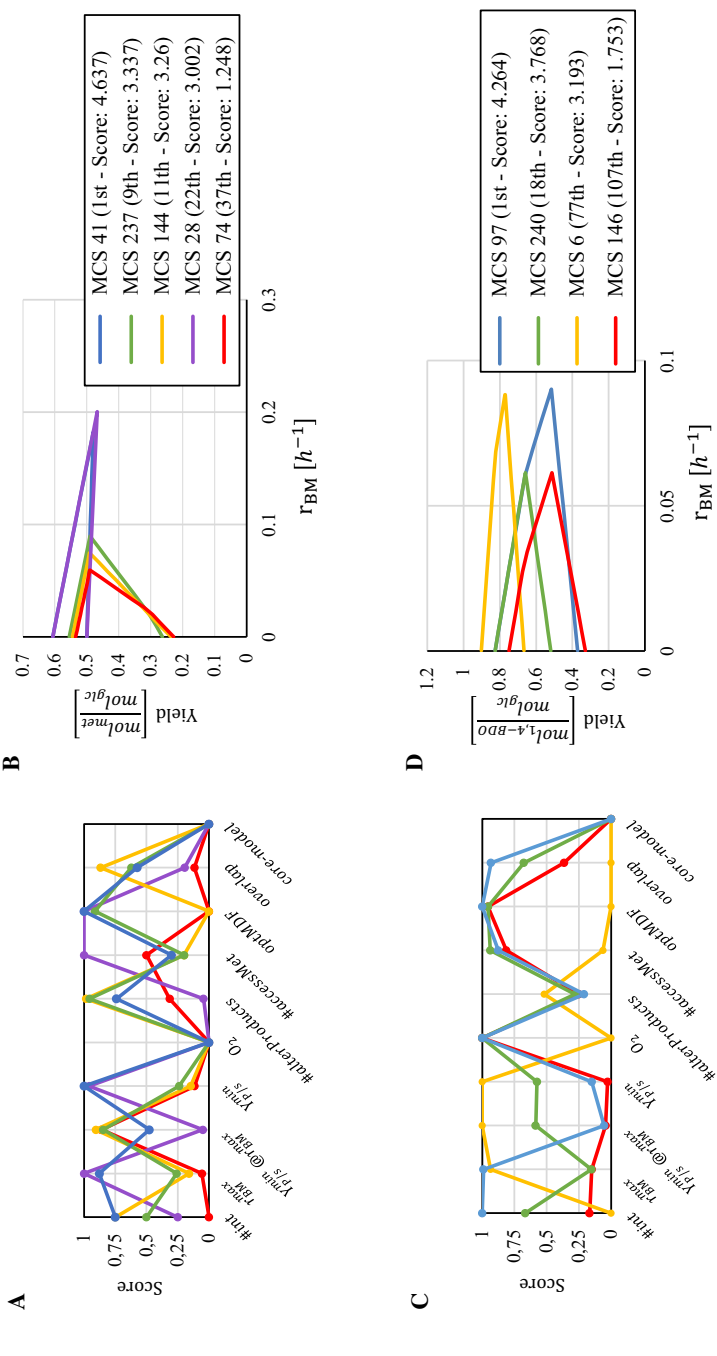


Figure 5.4: Comparison of five selected exemplary intervention strategies for the growth-coupled overproduction of L-methionine (**A, B**) and of four exemplary intervention strategies for the production of 1,4-butanediol (**C, D**) in *E. coli*. (**A,C**) Scores of the MCSs for the ten evaluation criteria. (**B,D**) Growth-rate versus product-yield plot of the selected MCSs.

5.3.2 Example 2: production of the heterologous product 1,4-butanediol

The metabolic engineering approach for the production of the non-natural bulk chemical 1,4-butanediol by *E. coli* [51] and its later commercialization has been an unprecedented success story for targeted metabolic engineering. In order to make this chemical producible by *E. coli*, Yim et al. designed a novel artificial pathway that branches from the tricarboxylic acid (TCA) cycle involving five heterologous enzymes. The strain design was supported by the computation of knockout strategies based on the OptKnock algorithm, which suggested the repression of the main fermentation pathways and of the oxidative operation of the TCA cycle to couple growth with product synthesis through the intracellular redox balance. Later studies sought to further enhance the production performance of these strain designs through enzyme engineering [178] or extended kinetic modeling [179].

We integrated the pathway from 2-oxoglutarate to 1,4-BDO, presented by Yim et al., 2011 [51], into the *i*JO1366 model (as well as in the *EColiCore2* model) and computed the minimal cut sets that establish the substrate-uptake-coupled production of 1,4-BDO with this pathway. We identified 274 MCS strategies with 6 to 12 reaction knockouts each, that could be grouped in 107 equivalence classes. Finally, we picked one representative per MCS class and ranked them as described before.

In Figure 5.4, four MCSs are compared, including the MCS with the highest and the lowest score. Figure 5.4C shows the single scores of the candidates among the different criteria, while Figure 5.4D shows their GRPY spaces. The best MCS (blue) does not have the highest minimum product yield. However, it allows anaerobic conditions and has a good balance between a high growth rate, a small number of necessary cuts (6) and an overlap with many other MCSs, which would enable switching to another IS if necessary. Nevertheless, depending on the application, MCS 6 (yellow) might also be a relevant candidate. Even though it requires 12 cuts, it has a high product yield and is more robust than the other MCSs. Sometimes, coupling can already be established with fewer interventions than the full set of a computed intervention strategy (see [54]). Generally, the determination of the best candidate may have different outcomes. However, the ranking often shows intervention strategies that can be excluded *a priori*. For example, the worst strategy (MCS 146, red) is outperformed by the best one (blue) in all criteria.

The thermodynamic analysis with optMDF and FBA predict the thermodynamic feasibility of all computed knockout strategies. In total, 104 out of 107 equivalence classes relied on anaerobic conditions all of which suggested the disruption of the tricarboxylic acid cycle (most frequently at malate dehydrogenase), the ethanol and lactate pathways, the knockout of the NAD(P) transhydrogenase (*THD2pp*) and then other (for each MCS specific) knockouts to establish full coupling. In all these strains, ATP synthesis is possible through glycolysis with the essential fermentation products formate, 1,4-BDO and acetate. While acetate is a by-product of the 1,4-BDO pathway, the ratio of formate and 1,4-BDO is predetermined by the redox-state.

Depending on the respective MCS, succinate and ethanol may occur as side products, ethanol as a product from a cycle in the glycine metabolism. The MCSs from the three remaining equivalence classes work under aerobic conditions and rely on the interruption of the TCA cycle through which 1,4-BDO becomes an essential by-product of respiration. The intervention strategy pursued by Yim et al. [51] relies on knockouts of the alcohol dehydrogenase (*adhE*), pyruvate formate lyase (*pfl*), lactate dehydrogenase (*ldh*) and the malate dehydrogenase (*mdh*) and has thus a large overlap with the computed MCSs. In addition, the aerobic respiration control protein (*arcA*) was knocked-out by Yim et al. but not considered herein as it is a regulator protein and no metabolic enzyme. As predicted, acetate excretion was observed for these strains. Differences between the strategy of Yim et al. and the computed MCSs probably arise because (1) Yim et al. used a microaerobic process by which a knockout of the pyruvate formate lyase is allowed (this mutant cannot grow under strict anaerobic conditions) such that CO_2 and acetate are the only essential by-products, and because (2) the chosen knockouts of Yim et al. do actually not yet guarantee a 1,4-BDO production in the *i*JO1366 model, even with simulated microaerobic conditions with an upper limit for oxygen uptake of 2 mmol $g_{CDW}^{-1}\,h^{-1}$.

The identified best MCS candidate (blue) consists of the reaction knockouts of the acetaldehyde dehydrogenase (*ACALD*), glucose 6-phosphate dehydrogenase (*G6PDH2r*), D-lactate dehydrogenase (*LDH_D*), malate dehydrogenase (*MDH*), phosphopentomutase 2 (*PPM2*) and the NAD(P) transhydrogenase (*THD2pp*). On the gene level this MCS can be established through the knockout of *mhpF, ldhA, pntB, zwf, mdh, adhE* and *deoB*. The list of the computed and ranked MCSs is provided in Table A.11.

5.4 Discussion

Many constraint-based strain optimization methods can generate a pool of intervention strategies from which one candidate has to be selected for strain development. This chapter introduced methods for the characterization and ranking of such a pool of strain design candidates, with a focus on growth-coupled designs. We presented a catalog of ten partially new criteria for assessing and ranking individual strain design candidates, broadly extending earlier approaches such as OptPipe [159] which assessed only four criteria (growth/production performance and an adaptability measure). Our ten criteria comprise (1) the number of interventions, (2) the maximum growth rate, (3) the minimum product yield at maximum growth rate, (4) the overall minimum product yield, (5) the required aeration strategy, (6) the number of alternative products that could disrupt growth-coupling, (7) the number of accessible (producible) metabolites, (8) the maximal thermodynamic driving force, (9) a score for the similarity to (overlap with) other ISs and (10) the feasibility of strategies in a reduced or further constrained model. Each criterion gives rise to a score which can be combined to an overall score useful to rank the IS pool. Even though the integration of ranking criteria into the initial strain design computation, e.g., as part of the strain design objective function, would be partially possible, in most cases it

would be computationally too expensive, rendering a posteriori ranking indispensable. Using real-world examples of strain design, we demonstrated the applicability and benefit of the developed characterization and ranking procedure. In our first example, we computed and ranked sets of minimal cut sets for the substrate-uptake-coupled production of L-methionine in *E. coli*, while the second example focused on the production of 1,4-butanediol via a heterologous pathway. The case studies showed that, apart from the actual ranking, the analysis of an exhaustive set of MCSs based on our ten criteria enables a thorough characterization, also supporting the elucidation of the underlying coupling mechanism. The comparison of MCSs/ISs via the different criteria shows that the selection of the best candidate is often not trivial and generally involves trade-offs. Even though the highest ranked candidate usually performs well in multiple criteria (e.g. in the expected production performance), it may be outperformed in a subset of other criteria (e.g., in the robustness of the strain designs).

Depending on the specific needs, our ranking procedure could be easily adapted (e.g., by replacing our weighted score approach with the "rank product" method used by [159]) or by analyzing additional properties. The "growth-coupling potential" [93], i.e. the difference between the maximal possible growth rate and the maximal growth rate that may be attained without mandatory product synthesis, was added a posteriori to the ranking catalog (see *CNAcharacterizeGeneMCS* in section 6.1). This measure is helpful specifically for the characterization of strain designs with wGCP. Another potential ranking criterion could be the number of undesired byproducts sometimes arising for certain ISs (e.g., acetate in the 2,3-BDO case discussed above). Furthermore, a measure for the adaptability of the resulting mutant strains could be introduced. For example, [159] used the MOMA method [180] to estimate the distance between the wild-type and the mutant flux distribution, which can then be used as a criterion to rank strain designs according to the required metabolic adjustments. However, this measure is probably not suited for a generic comparison of ISs for growth-coupled product synthesis because it is the assumption of many strain design methods that the mutant strains evolve (via adaptive laboratory evolution [161]) towards growth-optimal phenotypes. Hence, phenotypes with minimal metabolic adjustment as predicted by MOMA will then not be relevant. Furthermore, MOMA and related methods require as input valid wild-type reference flux distributions, which are often not known. This becomes even more critical if a series of intermediate strains needs to be constructed for which reference flux distributions are not available at the time of ranking the computed ISs.

As an important tool for characterizing and ranking strain designs, we also introduced the notion of equivalence classes by which designs with identical phenotypical solution spaces can be grouped. This simplifies the analysis as well as the ranking of the strategies, since then, only one representative for each class needs to be considered. Once the optimal strategy has been identified, possible alternative solutions in its equivalence class can be investigated.

Another strength of the presented approach lies in its straightforward and generic applicability as it is independent of product, substrate, host organism and the computational method used to generate the strain design candidates. The investigation of most criteria requires only an LP solver and the stoichiometric models of the wild type and the mutant. For the thermodynamic analysis, a relatively simple MILP must be solved [170].

6 Implementation in *CellNetAnalyzer*

CellNetAnalyzer (*CNA*) is a MATLAB-based toolbox, which offers methods for the analysis and design of metabolic networks. It also provides the MCS algorithm, which can be accessed through its graphical user interface (GUI) or via the application programming interface (API). So far, only the EFM-based MCS computation [57, 97] could be called from the GUI, while the original Farkas-duality-based MCS computation was covered only by the API functions *CNAMCSEnumerator* [60] and *CNAgeneMCSEnumerator* [31]. In the previous chapters, we proposed several new methods and functional extensions for the computation and analysis of MCSs. In this chapter, we describe how the presented features have been implemented in the new MCS functions for *CNA* and how these functions can be accessed through the API and the GUI.

6.1 *CellNetAnalyzer* API functions

6.1.1 Overview

Table 6.1 lists the API functions of *CNA* relevant for MCS computation and describes the extensions or improvements introduced in this work.

Table 6.1: Overview of the main *CellNetAnalyzer* API functions relevant for the computation and analysis of MCSs and the improvements introduced in this work.

Function name	Function description and improvements introduced in this work	Function dependence
CNAMCSEnumerator2 (new)	Extends the original *CNAMCSEnumerator* function for the computation of MCSs with the support of multiple target and desired regions, reaction additions and individual intervention costs (see chapter 4 and section 6.1.3).	*CNAoptimizeFlux*, *CNAfluxVariability*, *CNAcompressMFNetwork*

Function name	Function description and improvements introduced in this work	Function dependence
CNAgeneMCSEnumerator2 (new)	Computes gene-based MCSs, sped up by GPR-rule compression; supports all features from *CNAMCSEnumerator2* (see chapter 4 and section 6.1.3).	*CNAMCSEnumerator2*, *CNAoptimizeFlux*, *CNAfluxVariability*, *CNAintegrateGPRrules*
CNAMCSEnumerator3 (new)	Computes MCSs with all features from *CNAMCSEnumerator2*, additionally supporting optimality constraints to describe undesired and desired flux states for the computation of pGCP and wGCP strain designs (see chapter 3 and section 6.1.4).	*CNAoptimizeFlux*, *CNAfluxVariability*, *CNAcompressMFNetwork*
CNAgeneMCSEnumerator3 (new)	Computes gene-based MCSs with GPR-rule compression and supports all features from *CNAMCSEnumerator3* (see chapter 3 and section 6.1.4).	*CNAMCSEnumerator3*, *CNAoptimizeFlux*, *CNAfluxVariability*, *CNAintegrateGPRrules*
CNAintegrateGPRrules (new)	Integrates GPR rules (given in disjunctive normal form) as pseudo reactions and metabolites in a metabolic model (see chapter 4) used for the computation of gene-based MCS.	none
CNAcharacterizeIS (new)	Characterizes and ranks computed reaction-based metabolic engineering strategies (see chapter 5 and section 6.1.5).	*CNAoptimizeFlux*, *CNAfluxVariability*,
CNAcharacterizeGeneMCS (new)	Characterizes and ranks computed gene-based metabolic engineering strategies with growth-coupled product synthesis at different strengths (see section 6.1.6).	*CNAoptimizeFlux*, *CNAfluxVariability*,
CNAgenerateMap (new)	Automatically generates default flux maps for the GUI-based analysis of metabolic networks. Especially helpful for the analysis of GPR-extended networks.	none
CNAoptimizeFlux	FBA. Support of the Gurobi-solver was added.	none
CNAfluxVariability	FVA. Support of the Gurobi-solver was added, as well as support for parallel computing.	none
CNAcompressMFNetwork	Network compression. Used to speed up the computation of MCSs (see chapter 4). No changes were introduced to this function.	none

6.1.2 *CNAMCSEnumerator2*

The *CNAMCSEnumerator2* function extends *CNAMCSEnumerator* [31] and supports the additional MCS features from chapter 4, that is, multiple undesired and desired flux regions, reaction additions and intervention cost factors. While offering more flexibility, the function is still fully compatible with the previous *CNAMCSEnumerator*.

To improve the MILP runtime, *CNAMCSEnumerator2* performs a number of preprocessing steps before running the actual MCS MILP:

(1) Verification that all given sets of target and desired constraints are feasible in the original model (via *CNAoptimizeFlux*).

(2) Identification of blocked and irreversible reactions via FVA (*CNAfluxVariability*).

(3) Network compression via *CNAcompressMFNetwork* (removal of blocked reactions and conservation relations, lumping reactions).

(4) Determination of effective flux bounds for the target and desired regions in the compressed model (via *CNAfluxVariability*, see section below).

(5) Construction of the MCS MILP as shown in eqs. (2.47) and (4.13) either with indicator or with big-M constraints.

(6) Enumeration of all solutions to the MCS MILP with increasing cardinality (*enumerateMCS*: all MCS of size n are found in the n-th iteration).

or (Repeated) solution of the MCS MILP to find smallest MCSs (*findSmallestMCS*: n-th smallest MCS is found in the n-th iteration).

or (Repeated) solution of the MCS MILP to find arbitrary MCSs (*findAnyMCS*: an arbitrary MCS is found in the n-th iteration).

(7) Expansion of the computed MCSs to the uncompressed network.

(8) MCS verification.

In addition to the added MCS features, the function also received a number of improvements not mentioned in chapter 4. One is the more thorough network compression. Previous MCS algorithms protected non-targetable (not-knockable) reactions from being lumped with knockable ones, to prevent prohibited knockouts. *CNAMCSEnumerator2* allows the lumping of both, leading to smaller overall MILP sizes. Invalid MCSs with non-targetable reactions are, instead, discarded in a post-processing step, when the compressed MCSs are expanded and mapped back to the original network.

The construction of the MCS MILP is supported by multiple FVAs that identify the effective flux bounds, as well as blocked and essential reactions during preprocessing. In many models, (dummy) flux bounds are provided for all reactions, most of which cannot be reached due to restricting bounds on only very few reactions (e.g. substrate uptake). In these cases, the unnecessary (dummy) bounds are omitted.

Furthermore, the algorithm for computing MCSs of random size was adjusted. In the first step, the minimization objective is omitted to compute a solution to the general cut set problem, which usually does not return a *minimal* cut set. Previous approaches then generated an irreducible solution, i.e. an MCS, by successively removing interventions and verifying the resulting subset until a true MCS was found. *CNAMCSEnumerator2*, instead, minimizes the number of interventions in a confined search space, allowing only interventions present in the preliminary solution. This minimization usually completes within seconds, even in genome-scale setups and may generate smaller MCSs than the approach that was previously used.

Although not part of this thesis, several other improvements of the *CNAMCSEnumerator2* deserve mentioning. Steffen Klamt and Axel von Kamp added a nullspace-based approach for the computation of MCSs as an alternative to the Farkas-dual approach [104], implemented several alternative ways to build the MILP (e.g. that split equality constraints or combine z variables of different subsystems) and improved the solver support for CPLEX by using the CPLEX-MATLAB-API and intlinprog, available through the MATLAB Optimization Toolbox.

6.1.3 *CNAgeneMCSEnumerator2* and *CNAintegrateGPRrules*

The computation of gene-based MCS makes use of the gene-protein-reaction associations that are available for many metabolic models. After performing the gene-protein-reaction rule compression steps introduced in section 4.5, *CNAgeneMCSEnumerator2* integrates the compressed GPR rules into the metabolic reaction network as pseudoreactions and pseudometabolites through the function *CNAintegrateGPRrules*, which is part of the *CNA* API (and can also be called separately). Finally, *CNAMCSEnumerator2* is called, which then proceeds with a network compression and the actual MCS computation.

In addition to the GPR and network compression, *CNAgeneMCSEnumerator2* also offers an early reaction network compression *prior* to the GPR-integration. The GPR-rules are then lumped alongside their reactions, for instance when reactions r_1 and r_2 with the genes g_1 and g_2 are lumped, the lumped reaction $r_{1,2}$ receives the gene rule $g_1 \wedge g_2$. However, this option is not used by default because combining GPR-rules from multiple reactions may entail a higher combinatorial complexity that sometimes outweighs the benefits from the network compression. This is the case when lumping gene rules leads to many permutations, e.g., when the reaction r_1 with the rule $g_{1a} \vee g_{1b}$ and r_2 with the rule $g_{2a} \vee g_{2b}$ are lumped, the combined GPR rule of $r_{1,2}$ reads $(g_{1a} \wedge g_{2a}) \vee (g_{1a} \wedge g_{2b}) \vee (g_{1b} \wedge g_{2a}) \vee (g_{1b} \wedge g_{2b})$.

The gene MCSs returned by *CNAgeneMCSEnumerator2* refer to the full metabolic network with integrated GPR associations, so that gene knockouts affect the gene synthesis pseudoreactions. The fully extended network is also generated during gene MCS computation and can be used to analyze the computed gene MCSs. Additionally, the function also returns equivalent reaction-based MCSs that refer to the original network, which allows a quicker analysis of intervention strategies with *CNAcharacterizeGeneMCS*. Finally, also the compressed MCSs are returned together with the fully compressed network. This feature is helpful when decompressing MCSs involves many possible permutations and results in an excessive number of full solutions.

6.1.4 *CNAMCSEnumerator3* and *CNAgeneMCSEnumerator3*

CNAMCSEnumerator3 and *CNAgeneMCSEnumerator3* are extensions of *CNAMCSEnumerator2* and *CNAgeneMCSEnumerator2* and support the new MCS features introduced in chapters 3 and 4, most notably the computation of strain designs with weaker degrees of growth-coupled production. Sections 3.2.2 and 3.2.3 have shown that, for this purpose, an inequality-based description of undesired or desired flux regions, as used in *CNAMCSEnumerator2* (e.g., $\mathbf{T}\,\mathbf{r} \leq \mathbf{t}$), must be supplemented with an inner objective function that demands growth optimality (generally described by: maximize $\mathbf{c}^\intercal\mathbf{r}$). The integration of this feature into the MCS algorithm required fundamental adaptions, giving rise to the new MCS algorithm *CNAMCSEnumerator3*.

To standardize the user-defined strain design specifications, i.e. undesired or desired flux states with or without optimality constraints, *CNA(gene)MCSEnumerator3* now uses strain design "modules" (see Figure 6.1, top right). Every module specifies (a) desired or (b) undesired flux states described by either (1) a set of inequality constraints (for traditional MCS strain design), (2) an optimality constraint together with one or more inequality constraints (for pGCP or wGCP strain design), or (3) a flux-ratio (i.e. yield) constraint that can also be complemented with additional flux constraints (which may be used as an alternative to the traditional approach for dGCP or SUCP). Analogous to *CNAMCSEnumerator2*, described in section 4.1, arbitrary strain design modules for target and desired regions can be combined to specify an MCS setup. An MCS MILP with two undesired and two desired modules, for instance, was shown in eq. (4.13).

Solver-independent construction of the MCS MILP

The core MCS algorithms of *CNAMCSEnumerator2* and *3* are substantially different. With *CNAMCSEnumerator3*, the entire MILP processing pipeline was reimplemented using a modular and object-oriented approach that decouples the construction of the MCS MILP from its solution. The main reasons for this separation are the increasing number of supported solvers and the introduction of advanced MCS features. An implementation of optimality constraints in the core algorithm of *CNAMCSEnumerator2* would have necessitated an adaption of all solver-specific MILP construction procedures, increasing the risk of errors and future

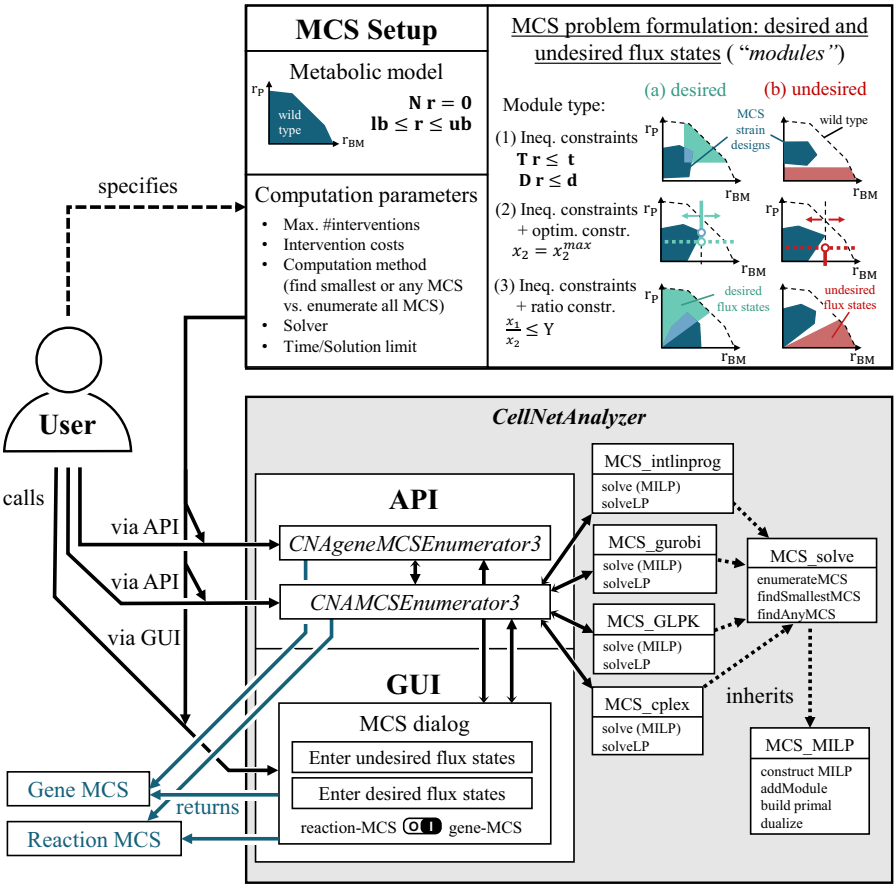


Figure 6.1: Overview of the different components used in the API- and GUI-based computation of MCSs with *CNAMCSEnumerator3*. The upper box shows the main constituents of an MCS strain design setup. These include the base model, desired and undesired flux states defined by *"modules"* and several computation parameters which must be specified by the user before starting the computation. The plots (production envelopes) shown for each module show typical desired or undesired flux regions for production strain design and flux spaces of exemplary strain designs that fulfill the module's demands. The lower box shows the main functions and classes that are involved in the MCS computation. After specifying an MCS setup, there are two ways to start the (gene or reaction-based) MCS computation in *CellNetAnalyzer*, either through the API or through the GUI via a dialog box. In both cases, the MCS MILP is constructed by functions of the MCS_MILP class. The MILP solution and MCS verification steps are implemented in MCS_solve that, in turn, uses the LP and MILP solver interfaces provided by MCS_cplex, MCS_intlinprog, MCS_gurobi or MCS_GLPK. Dashed arrows indicate the inheritance relationships between classes, whereas the arrow head points towards the parent class (e.g., MCS_solve is inherited from MCS_MILP and contains all its properties and functions).

maintenance effort. The new and unified MILP construction process supports all features introduced in this work. Once constructed, the MCS MILP can be solved with any of the linked solvers; CPLEX, MATLAB intlinprog, Gurobi or GLPK. This architecture additionally ensures that all future extensions of the MCS approach will be supported immediately with all solvers, while, vice-versa, new solvers may be employed for the MCS computation, directly supporting the full feature set of the MCS framework.

The structure of *CNA(gene)MCSEnumerator3*, including all involved classes, is shown in the bottom of Figure 6.1. Functions that translate a user-defined MCS setup into the standardized MILP form are provided by the MCS_MILP class. The strain design modules are first translated into *sub-MILPs*, which are then assembled to the final MCS MILP. Recurring steps for the construction of *sub-MILPs*, for instance, the dualization of LPs or the application of Farkas' Lemma, are gathered in different functions of MCS_MILP. Methods for the MCS computation, i.e. the iterative solution of the MCS MILP, are located in MCS_solve which is a class derived from MCS_MILP. It provides the three MCS computation approaches from *CNAMCSEnumerator2*, all making use of the previously built MILP: an iterative search for the smallest MCSs (*findSmallestMCS*), an iterative search for MCSs of random size (*findAnyMCS*) and a full enumeration of MCSs with increasing cardinality (up to a predefined limit) that uses the populate function available in CPLEX and Gurobi (*enumerateMCS*). Further auxiliary functions are also integrated in MCS_solve, e.g., to verify found MCS or to introduce constraints that exclude previously found MCSs as MILP solutions (see eq. (2.25)). The classes MCS_cplex, MCS_intlinprog, MCS_gurobi and MCS_GLPK mainly redirect the function calls for the solution of LP and MILP to the individual solver interfaces and return the optimization results in a standardized manner. As all four classes are inherited from MCS_solve and hence also from MCS_MILP, they contain all necessary functions to construct and solve an MCS problem. *CNAMCSEnumerator3*, constructs one of these four classes, depending on the chosen solver, to prepare and start the MCS search.

The solver interfaces of CPLEX, Gurobi, MATLAB intlinprog and GLPK accept MILP problems (cf. eq. (2.11)) in the standardized form:

$$\begin{aligned} &\text{minimize} && \mathbf{c}^{\mathsf{T}}\mathbf{x} \\ &\text{subject to} && \begin{bmatrix} \mathbf{A_{eq}} \\ \mathbf{A_{ineq}} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{b_{eq}} \\ \leq \mathbf{b_{ineq}} \end{bmatrix} \\ & && \mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub} \\ & && x_i \in \mathbb{Z}, \quad \forall i \in Z, \end{aligned} \tag{6.1}$$

where a set of indices Z designates the integer variables. CPLEX and Gurobi furthermore support indicator constraints of the form:

$$x_i = 0 \rightarrow \mathbf{A}_k\mathbf{x} \leq b_k, \qquad x_i \in \{0,1\}, \quad \forall k \in K, \quad i \in Z, \tag{6.2}$$

that is to say a value $x_i = 0$ *implies* that a set of constraints $\mathbf{A}_k\mathbf{x} \leq b_k, \forall k \in K$ holds (K is the index set of all inequalities, for which this indicator constraint holds). As indicated, a variable x_i may be linked to multiple indicator constraints. The details of this relationship were discussed in section 2.1.2 and are shown in Table 2.5 (here, binary variables are named z_i). Throughout the assembly process, the association between the original targetable reactions and genes with the indices $i \in I$ and the associated variables and constraints ($k \in K$) of the preliminary MCS MILP is tracked and updated. In the last step of MILP construction, indicator constraints are used to link the binary variables z_i that represent reaction (or gene) knockouts (or additions) to the associated variables and constraints. The resulting MILP can be used right away with solvers that support indicator constraints (CPLEX, Gurobi). As was explained in section 2.1.2, indicator constraints can generally be replaced with big-M constraints. This is necessary for the solvers that lack the support of indicator constraints (intlinprog, GLPK), making the substitution of indicator constraints the only solver-specific step of MILP construction.

There are some pitfalls when picking values for M_k. Too small values may only shift the bound on a constraint but not relax it sufficiently to simulate its removal, while too large values entail numerical issues. As intlinprog and GLPK have tolerance ranges for integer variables, a small relaxation of a constraint

$$\mathbf{A}_k\mathbf{x} - z_i \cdot M_k \leq b_k, \tag{6.3}$$

may then take place, also when $z_i = 0$, as with a tolerance of 10^{-6} on z_i and $M_k = 10^6$ the constraint is relaxed by a value of $z_i \cdot M_k = 1$ (hence $\mathbf{A}_k\mathbf{x} \leq b_k + 1$) by default. It is sometimes possible to determine a suitable value of M_k for a big-M constraint (eq. (6.3)) prior to the MCS computation. For this purpose, one uses the set of non-targetable constraints (J) in the MCS MILP to determine individual values M_k via linear programming:

$$\begin{aligned} &\text{maximize} && \mathbf{A}_k\,\mathbf{x} - b_k (= M_k) \\ &\text{subject to} && \mathbf{A}_j\mathbf{x} \leq b_j, \forall j \in J, \end{aligned} \tag{6.4}$$

and insert the computed values for M_k in the MCS MILP.

As eq. (6.4) is in many cases unbounded, but values for M_k are still required when using big-M-based solvers like intlinprog or GLPK, the M_ks can only be chosen arbitrarily. However, when bounds can be determined, the big-M method can be combined with indicator constraints to decrease MILP runtimes. This is especially effective when the M_ks are tightly bounded [181].

Target and desired modules with flux-ratio constraints

Modules of type (3) with flux-ratio constraints have not been applied in any of the preceding chapters and present an alternative approach for demanding, for example, a minimum product yield. They comprise the equations

$$\frac{\mathbf{e}^\intercal \mathbf{r}}{\mathbf{f}^\intercal \mathbf{r}} \leq Y, \quad \mathbf{T}\,\mathbf{r} \leq \mathbf{t}, \tag{6.5}$$

with $\mathbf{e}$ and $\mathbf{f}$ as coefficient vectors for the numerator and denominator of the flux-ratio constraint and Y as the upper limit of the flux ratio. As in the other two module types, $\mathbf{T}$ and $\mathbf{t}$ further bound the flux space with a set of inequality constraints. During the construction of the MCS MILP, the module is translated into a system of linear inequalities via an approach similar to linear-fractional program (LFP) (cf. eq. 2.9). Here, one must ensure first that the denominator term has a purely positive (or negative range). For a positive denominator, the linear inequality system reads:

$$\begin{bmatrix} \mathbf{G} & -\mathbf{g} \\ \mathbf{T} & -\mathbf{t} \\ \mathbf{e} - Y \cdot \mathbf{f} & 0 \\ \mathbf{f} & 0 \end{bmatrix} \begin{bmatrix} \leq \\ \leq \\ \leq \\ = \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{r}} \\ y \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ 0 \\ 1 \end{bmatrix}, \qquad y \geq 0. \tag{6.6}$$

The original reaction rate vector $\mathbf{r}$ can be retrieved from $\tilde{\mathbf{r}}$ through $\mathbf{r} = y^{-1} \cdot \tilde{\mathbf{r}}$. $\mathbf{G}$ and $\mathbf{g}$ describe the base model through $\mathbf{G}\,\mathbf{r} \leq \mathbf{g}$ (see eq. (2.30)). The third row contains the linearized yield constraint shown in eq. (2.36). Although this approach is similar to the traditional one with a linearized yield constraint shown in eqs. (2.36), (3.10) and (4.4), there are two main differences. First, the inequality system describing the flux region is homogenized, second, the denominator of the yield function is fixed to one. As a result, the denominator term of the yield constraint no longer needs an explicit minimum threshold. This gives the LFP-type formulation a numerical advantage, especially when otherwise the user-defined denominator minimum is set to a very small value. In almost all cases, traditional and LFP-type modules will, however, behave identically, for instance, when the module is used to enforce a product yield in a network that has a non-zero NGAM demand that enforces substrate uptake. The denominator of the yield function is then bound to be strictly positive (and maybe even greater than one) leading to good numerical stability of the traditional approach.

Equation (6.6) shows the primal form of the flux-ratio-constraint module. To define desired behavior with flux-ratio constraints, the variables in $\tilde{\mathbf{r}}$ of eq. (6.6) are linked to the binary knockout variables $\mathbf{z}$ to simulate potential metabolic interventions. For undesired

behaviors, knockout constraints $\mathbf{I_{KO}}\tilde{\mathbf{r}} = \mathbf{0}$ are added and Farkas' lemma is applied as shown in many previous examples (e.g., eqs. (2.41) and (3.20)):

$$\begin{bmatrix} \mathbf{G}^\mathsf{T} & \mathbf{T}^\mathsf{T} & (\mathbf{e}^\mathsf{T} - Y \cdot \mathbf{f}^\mathsf{T}) & \mathbf{f}^\mathsf{T} & \mathbf{I_{KO}} \\ -\mathbf{g}^\mathsf{T} & -\mathbf{t}^\mathsf{T} & 0 & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0 & 1 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \\ h \\ q \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \geq \\ \leq \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ 0 \\ -1 \end{bmatrix} \tag{6.7}$$

$$\mathbf{u} \geq \mathbf{0}, \quad \mathbf{w} \geq \mathbf{0}, \quad h \geq 0.$$

Finally, the slack variables $\mathbf{v}$ are replaced by indicator constraints that are linked to the binary variables $\mathbf{z}$ to simulate the knockouts of reactions, i.e., their corresponding constraints. Due to the quasi-homogeneity of the inequality system, $q \leq -1$ may be replaced with $q = -1$ to further simplify eq. (6.7). The MILP for an MCS setup with one LFP-type target module reads

$$\begin{aligned} &\text{minimize} && \sum z_i \\ &\text{subject to} && \begin{bmatrix} \mathbf{g}^\mathsf{T} & \mathbf{t}^\mathsf{T} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \end{bmatrix} \leq 0 \\ & && \forall i : z_i = 0 \rightarrow \begin{bmatrix} \mathbf{G}_i^\mathsf{T} & \mathbf{T}_i^\mathsf{T} & (e_i - Y \cdot f_i) \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \\ h \end{bmatrix} = f_i \\ & && \mathbf{u} \geq \mathbf{0}, \quad \mathbf{w} \geq \mathbf{0}, \quad h \geq 0, \quad z_i \in \{0,1\}. \end{aligned} \tag{6.8}$$

6.1.5 *CNAcharacterizeIS*

CNAcharacterizeIS automates all necessary computational steps for the characterization and ranking of metabolic engineering strategies with the ten different criteria presented in chapter 5. The function can be called through the *CNA* application programming interface (API). As parameters, this function takes a (usually wild type) reference model, defined by its stoichiometric matrix and flux boundaries, and, the actual intervention strategies defined by sets of upper and lower flux boundaries distinct from the original model but also referring to the same stoichiometric matrix. Reactions that are knocked out are marked by upper and lower boundaries of zero. Other types of interventions, such as up- or downregulations are simulated through tighter or more relaxed flux boundaries with respect to the reference model. Furthermore, the user may select the catalog of characterization criteria that should be assessed and set individual weighting coefficients for each criterion to rank the ISs.

The assessment of some criteria requires additional parameters in the function call. For properties 2-5 (chapter 5), the user needs to indicate the reactions of growth, product synthesis and substrate and oxygen uptake, furthermore the user may indicate the *internal* metabolites (needed for the properties 6 and 7), inequality constraints that describe undesired behavior

(property 7), thermodynamic data (property 8) and the subset of reactions representing the core metabolism within the given full model.

In a first step, FVA is used to group the ISs in equivalence classes, second, all ISs are characterized with the methods presented in chapter 5 and finally, the non-equivalent ISs are ranked. For the LP-dependent characterizations the user may choose between the CPLEX, gurobi, intlinprog or the GLPK solver. The MILP used for the thermodynamic criterion (Optimal max-min driving force) depends on the CPLEX-Java-API. When the MATLAB Parallel Computing Toolbox is installed, *CNAcharacterizeIS* will process the IS in multiple parallel threads, speeding up the characterization process significantly. The results of the characterization and ranking are presented in a MATLAB structure and, alternatively, as a 2-D-cell-array that can be stored in a spreadsheet file.

6.1.6 *CNAcharacterizeGeneMCS*

For the characterization and ranking of gene-based MCSs (as, for instance, computed in chapter 4), *CNAcharacterizeIS*, defined for reaction knockouts, has several shortcomings. Although it is possible to integrate GPR rules into the constraint-based network structure as pseudoreactions and pseudometabolites and use *CNAcharacterizeIS* for the characterization of gene-based IS, the inflated network size (3-fold in the case of *i*ML1515) results in massive runtime increases. Furthermore, different gene intervention strategies may generate identical metabolic phenotypes which, in practice, results in many IS duplicates and equivalent ISs to be assessed.

CNAcharacterizeGeneMCS, therefore, uses an adapted approach. As function parameters, it receives all gene MCSs, a corresponding set of reaction-based ISs and a mapping between gene and reaction MCSs. The reaction-based ISs are, as before, represented by sets of upper and lower bounds and analyzed with the methods of *CNAcharacterizeIS*, resulting in shorter runtimes. As an exception, the number of interventions (property 1) and the overlap score (property 9) are based on gene MCS instead. The final ranking table then lists the reaction-based ISs and provides to each reaction IS the sets of gene MCSs that generate it, whereas the number of interventions and the overlap score are taken from the best performing associated gene IS.

Originally, the *CNAcharacterizeGeneMCS* function was created for the characterization and ranking of the (SUCP) MCSs computed in chapter 4. To account for the developments in chapter 3, two properties were added to the characterization catalog for the analysis of ISs with pGCP and wGCP, the maximum product yield at maximum growth and the *growth-coupling potential* [93]. The growth-coupling potential describes the spread between the maximal growth rate and the maximal growth rate that is possible without production. ISs with high values in these properties receive a higher score. Like its predecessor, the function is part of the *CNA* API.

6.2 GUI-based MCS computation

The duality-based MCS computation can now also be accessed through the *CellNetAnalyzer* GUI, which fully supports the features added in chapters 3 and 4. Figure 6.2 shows the new MCS dialog box with exemplary parameters for the computation of strain designs for weakly growth-coupled production. This GUI-based MCS problem specification supports almost all options from the API function and allows also beginners or less trained users to conveniently set up and solve various MCS problems.

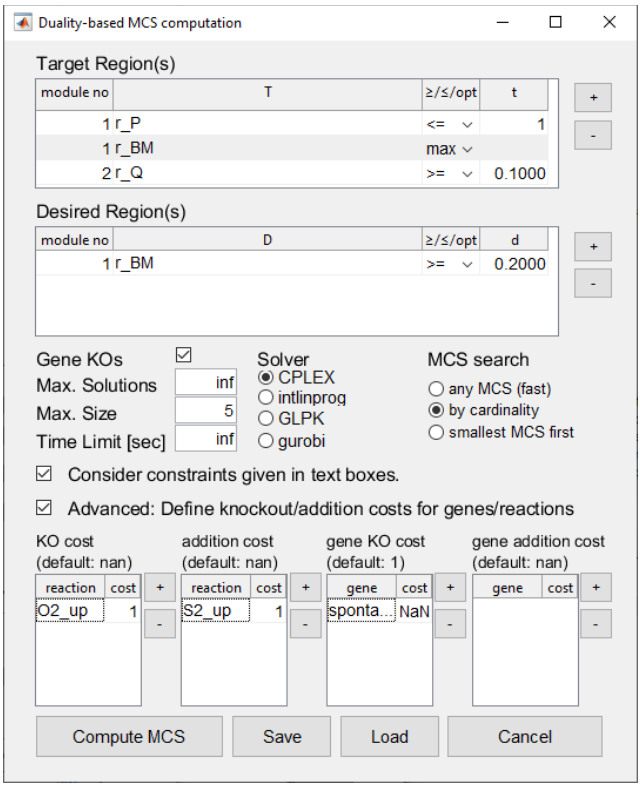


Figure 6.2: Screenshot of the MCS computation dialog box, which can be accessed through the *CNA* GUI. The shown parameters define a setup for the computation of gene-based MCSs for weakly growth-coupled product synthesis. Two target regions are defined, one to eliminate flux states where the product synthesis rate (r_P) is less or equal to 1 under maximal growth (r_BM), the other one to block the by-production of a metabolite Q (r_Q). The desired region ensures that the growth rate is able to reach at least a value of 0.2. Manually entered values in the visualized flux maps of *CNA* can be taken into account. The knockout of the oxygen supply (O2_up) by the MCS algorithm is explicitly allowed (at the cost of one intervention) and the additional supply of a secondary substrate (r_S2) is added if necessary. The knockout of reactions marked as spontaneous is prohibited. The algorithm will perform a full enumeration of MCSs with increasing cardinality, up to the intervention costs of 5, using the CPLEX solver without any time or solution limit.

6.3 Code availability and compatibility

The described new API functions and features have been integrated in *CellNetAnalyzer* (version 2021.1, see Appendix Source Code A.4) and have been tested and used under MATLAB® 2020b with IBM ILOG® CPLEX® 12.10. All scripts and network files needed to reproduce the example calculations from chapters 3-5 are compatible with this setup and can also be found in the Appendix (Source Codes A.1 to A.3). As was described above, the MCS algorithm is now also compatible with Gurobi™ (tested with version 9.1.1), with the intlinprog function from MATLAB's Optimization Toolbox (version 9.0 / MATLAB 2020b) and with the open-source solver GLPK (version 4.62). The latter two solvers, which are limited to the big-M based MCS computation, should be used only for small and medium-sized networks. Especially, the stability of GLPK strongly depends on the used version, the operating system and the model size.

7 Discussion and outlook

The reconstruction of genome-scale metabolic networks for several thousand organisms [34–36] has fostered the development of new metabolic modeling approaches and tools. Constraint-based methods allow the analysis of large metabolic networks with MILP and LP techniques [85, 86] and computational strain design has become an indispensable step in many present-day metabolic engineering studies [29, 49–56]. The minimal cut set approach is one of the most powerful frameworks for strain design and has been applied successfully in several computational and experimental studies [29, 31, 54, 82, 110]. In this work, a number of developments has been introduced that extend and generalize the concept of MCS and broaden its scope of applications.

An essential first step of any strain design approach is the formalization of a design objective. A common goal of many existing methods, including MCS, is to enforce growth-coupled production, i.e., to render a product of interest a mandatory by-product of growth [41, 58, 59]. Since different interpretations of this principle exist in the literature, we systematized in chapter 3 the predominant notions of GCP by introducing four distinct coupling types with gradually increasing strengths: pGCP, ensuring that product synthesis is possible under maximum growth, wGCP, enforcing product synthesis at all growth-maximal flux states, dGCP, demanding product synthesis is present in all growing phenotypes, and SUCP, enforcing production in all flux states where substrate is taken up, that is, in all realistic flux states. Most strain design approaches consider one or sometimes two of these coupling notions. OptKnock, for instance, generates strain designs for pGCP, ModCell2 allows for wGCP and SUCP [117] and MCS approaches have been used to compute SUCP strain designs [54, 74] and their use for dGCP has been proposed but not been applied so far [136]. To allow the computation of strain designs for *all four* types through the MCS framework, a feature that allows the specification of growth-optimal desired and undesired flux states was developed in this work. Such *optimality constraints* ensure that desired or undesired flux states are optimal regarding a given objective function, such as growth maximization, also under potential network interventions that shift this optimum. This feature can therefore now be used to compute MCSs for pGCP and wGCP.

Computing MCS for all four coupling degrees in application examples, it could be illustrated that the computed MCSs reflect the hierarchy of the four coupling types. For

instance, the computation of MCSs with dGCP demands that all remaining flux states with growth carry product synthesis. This, of course, includes the growth-maximal flux states, so that the requirements for wGCP and pGCP are automatically fulfilled as well. Hence, any MCS that enforces dGCP will automatically also enforce wGCP and pGCP. In turn, the MCSs found for dGCP and wGCP must also be retrievable for wGCP or pGCP, either identically or as a subset with fewer interventions. This hierarchical relationship holds for any pair of coupling degrees.

In two genome-scale computations, strain designs for the production of 12 relevant products were determined using the *E. coli* model *i*ML1515, first identifying MCSs with an arbitrary and then with a minimum number of knockouts. Growth-coupled production could be established for all products and for almost all coupling strengths. As expected, a stronger coupling correlates with a higher number of knockouts. On the contrary, stronger coupling, on average, did not increase the computation effort, nor lower the chances of a successful computation. In fact, the highest success rates and lowest computation runtimes were observed for the weakest (pGCP) and the strongest (SUCP) coupling degree. As SUCP guarantees a minimum product yield, it should generally be favored for computations.

The concept of coupled product synthesis was finally generalized to the biological function of ATP regeneration, so that the product of interest becomes a byproduct of ATP synthesis. All MCSs for directionally ATP-coupled product synthesis were enumerated in the *E. coli* core model up to the size of 4 knockouts. The results confirm that (directionally) ATP-coupled production is possible and implies growth-coupled production, i.e., it takes an intermediate role between dGCP and SUCP, such that MCSs for SUCP imply dACP and MCSs for dACP imply dGCP.

In chapter 4, the MCS approach was extended by a number of functional enhancements that provide more flexibility for tailoring the metabolic flux space. Allowing also gene and reaction additions in combination with knockouts, the extended MCS algorithm is able to generate new types of strain design that could not be found with the traditional approach, i.e. using knockouts only. Reaction additions can, for example, be used for a flexible substrate choice, so that the most suitable substrate (or substrate combination) is selected on demand instead of a fixed selection of substrate(s). By supporting multiple “regions” of desired and undesired flux states, more complex strain design setups become possible because combinations of multiple regions may be used to properly describe non-convex sets of flux states. In combination with reaction additions, intervention strategies can be computed that rely on substrate co-feeding, in particular for the case that a co-substrate used for one strain design is also a potential byproduct of other designs. To give the user the option to prioritize certain network interventions, all potential interventions can be associated with individual cost factors. As the MCS algorithm minimizes the total cost, cheap interventions are then favored. Finally, to improve the performance of

gene-based MCS computations, compression techniques for GPR associations were developed and added to the algorithm.

The new features were first tested in small example networks, then applied in a medium-sized and finally in a genome-scale MCS setup to compute strain designs for the substrate-uptake-coupled production of 2,3-BDO in *E. coli*. A full enumeration of gene-based MCSs was possible for the production of 2,3-BDO from glucose, with up to 9 interventions in medium-scale and 7 interventions in genome-scale respectively. The additional compression of GPR rules reduces the computation runtime by factors of 15 and 4, respectively, compared to traditional network compression. Using multiple target and desired regions together with a selection of three different (optional) substrates led to intervention strategies that effectively exploit co-feeding for improved bioproduction. While an additional desired region had only a slight effect on the MCS-enumeration runtime, an additional target region rendered the MCS enumeration problem significantly more complex. To show that the MCS extensions and improvements are independent of the used model, the computations for all scenarios were repeated using the yeast-GEM model for *Saccharomyces cerevisiae* [155] and the *Pseudomonas putida* model *i*JN756 [156].

MCS and other computational strain design approaches, can often generate a pool of hundreds of intervention strategies to a given design problem. In practice, having a pool of strategies to choose from is favorable, since every strain design has its particular advantages and disadvantages. A high product yield might only be achieved at the expense of a low growth rate, or good thermodynamic production properties may require many interventions. For assessing ISs with respect to their production performance, their robustness and their experimental implementation effort, chapter 5 presented a catalog of 10 criteria. These include obvious measures such as the maximal growth rate or the number of interventions but also new and advanced criteria such as the thermodynamic driving force of the product pathway or the number of accessible metabolites. Furthermore, as a new concept we introduced *equivalence classes*, which groups ISs with identical flux spaces. Equivalence classes can be used to greatly reduce the number of ISs that need to be assessed.

In two example MCS computations, large pools of IS were generated for the substrate-uptake-coupled production of L-methionine and the heterologous product 1,4-BDO in *E. coli* using the *i*JO1366 model. For both cases, the solutions were characterized along the 10 criteria and then ranked by the weighted sum of their performance in each criterion compared to their competitors. While the ranking order of the "best" strategies often varies with the weighting factors used for each criterion, weak candidates can be exposed more effectively, as they usually perform poorly in multiple or even all criteria. Together with the grouping of ISs in equivalence classes, the ranking approach is able to narrow down the pool of ISs effectively. The individual characterization results present a good information basis to select a strain design for experimental implementation.

The algorithmic developments from the chapters 3 to 5 have been implemented and added to the MATLAB-based *CellNetAnalyzer* toolbox. In addition to the CPLEX solver, the full-featured MCS algorithm can now also be used with the solvers Gurobi, intlinprog and GLPK. For this purpose, the MCS algorithm has been reimplemented from scratch, offering a modular, object-oriented and solver-independent construction of the MCS MILP. With this functional separation, new features can be introduced to the MCS framework without the need for solver-specific adaptions, and, vice-versa, additional solvers may be deployed without adapting the MILP construction. The source code of all computations in this work is available in the Appendix (Source Codes A.1 to A.3) and part of the latest *CellNetAnalyzer* release.

While the concept of MCS, as such, has already been applied in experimental studies [29, 54], experimental applications of the new algorithmic developments are still pending but in planning or even underway. For example, gene-based MCSs have been computed and characterized for the growth-coupled production of octyl acetate in *E. coli* (with all four coupling strengths) and a promising strain design, suggested by an MCS, is currently being implemented and tested in our lab. For another experimental study that aims to produce terpenes with *E. coli*, MCSs have been computed and ranked and await implementation. Finally, the metabolic drafting of microbial communities, which presents an entirely new application for the MCS algorithm, is planned as a separate study. Finding suitable genetic interventions for that purpose will involve most new MCS features introduced in this work, such as genetic interventions, optimality constraints, the specification of multiple desired or undesired flux regions. For example, one could compute MCS demanding that any member of the co-culture may only reach its maximal growth rate when the other is also able to grow.

A major challenge for the computation of MCSs is the long runtime, especially in advanced genome-scale setups, and many of the extensions further increase the complexity of the MILPs. A full enumeration of MCSs is often impossible, even for relatively small cardinalities, and the runtime for the computation of single MCSs may still lie in the range of hours (cf. Table 3.2). It is therefore vital, to further improve network compression (and possibly reduction) and apply more resourceful problem formulations. An existing example for such an algorithmic improvement, is the formalization of target flux vectors via an alternative dual system based on the nullspace instead of Farkas' Lemma [103, 104]. In benchmarks, this approach sped up the MCS computation by a factor of approximately 2.5. Another way to improve the MILP performance, could be a replacement of indicator constraints with tightly bounded big-M constraints [181]. This would additionally ensure the compatibility with free MILP solvers, that do not support integer constraints. To bound big-M constraints (e.g., of the form $\mathbf{A}_i\mathbf{x} \leq b_i + z_i \cdot M_i$, see eqs. (2.14) and (6.3)), it is sometimes possible to determine ideal (small) values for M_i that maintain the full space of integral solutions, by maximizing the term $\mathbf{A}_i\mathbf{x}$. Since, in the context of MCS, many of these terms are actually unbounded, other algebraic or heuristic approaches must be used to set M_i. For instance, to compute suitable values of M_i,

one could linearize all big-M constraints by replacing $z_i \cdot M_i$ with a continuous variable s_i and then determine the maximal value that s_i takes in *a vertex* of the otherwise unbounded problem space (also known as the "optimal vertex problem" [182]). Those values could then be used to fix M_i, as it is guaranteed that the full combinatoric complexity of the original MILP solution space is maintained. In problems with single target or desired regions, this approach is not expected to improve performance because the complexities of determining M_is and computing MCSs are similar. With multiple regions, however, the problem complexity of finding M_is is lower than the MCS problem, as the variables z_i that present the link between the different subsystems and thereby render the problem space significantly more complex, are absent.

Beyond the experimental validation and performance improvements, the MCS algorithm may also be augmented by additional features. One direction is to include additional metabolic information in the MCS computation, e.g., for proteome allocation [82] or thermodynamics [183]. Also, further conceptual extensions of the computational framework might be possible. For example, the approach for wGCP strain design could be adapted, with little effort, to demand a minimum "growth-coupling potential" [93], that is, a minimum difference between the maximal growth rate and the maximal growth rate without production. Other advanced features could be added as well. The computation of minimal sets of *continuous* interventions, i.e. overexpressions and downregulations (in addition to binary knockouts or additions) would present a very potent tool for strain design. An advantage of MCS with continuous interventions is, that production strain design would also be possible in cases where a rigid and stoichiometric coupling is not feasible, e.g., due to high energy demand and stoichiometric limitations. A framework has already been developed to compute regulatory MCSs with different predefined expression levels [105], but these levels are set arbitrarily and lead to much higher MILP dimensions. Ideally, continuous interventions would provide more flexibility and not require a definition of different fixed expression levels by the user. While an integration of continuous interventions is straightforward in desired MILP subsystems, it is still unresolved if and how they can be embedded in a Farkas-lemma-based linear inequality systems and how they can be linked to all desired or undesired subsystems.

Finally, the usability of the MCS approach may be further improved, especially to lower the bar for beginners. Although the MCS algorithm is easily accessible through the *CellNetAnalyzer* GUI and API, there are many hurdles on the way of applying MCS to a strain design problem that discourage potential users. For the OptKnock and Robustknock approaches it suffices to indicate the (pseudo)reactions of growth and product synthesis, while no further knowledge about the model or other network analysis tools is required. The MCS approach, on the other hand, relies on tailoring the metabolic flux space through thresholds for growth rate or productivity. The often arbitrary thresholds usually necessitate a prior analysis of the model with YS, PE or FBA. Additionally, the concept of defining "undesired" or "desired" flux spaces carries many pitfalls and can be confusing. For example, to design production strains with MCS,

one is tempted to specify only a set of desired flux states with high product yields. Experienced users know that such an MCS computation will fail because the demanded flux states with high product yields are already feasible in the wild type model, and no further knockouts are required. Instead, competing flux states with a low product yield (that possibly allow for faster cell growth) must be defined as *undesired* flux states, to be later eliminated through knockouts. As for many other constraint-based strain design methods, there are also a number of limitations to the MCS approach that a user might not be aware of. While the growth-coupled synthesis of different target metabolites can often be enforced stoichiometrically through MCS, this coupling is harder and sometimes impossible for more complex products like proteins. For the user, it is then hard to understand whether the computation failed because (1) there exists no solution to the strain design problem, (2) the MCS setup is too complex to complete the computation within the runtime limit or (3) a mistake was made in setting up the model or MCS setup. To prevent most pitfalls and support users in applying MCS, a better guidance is necessary, e.g., by automatically plotting target and desired regions, or offering a beginner-friendly MCS computation wizard from within the *CellNetAnalyzer* toolbox. Furthermore, accessibility of the MCS framework may be enhanced by improving the compatibility with the COBRA and COBRApy frameworks and implementing a Python-based MCS algorithm. The latter is already in progress as part of the Python-based *CNA* toolbox *CNApy* [112].

Appendix

Table A.1: MCS setups for the computation of pGCP, wGCP, dGCP, dACP and SUCP strain designs shown in Figure 3.5A and B, Tables 3.2 and 3.3 and Figure 3.6. In addition to the main MCS setup, such as the used model and the target and desired flux states, also the technical parameters for the MCS computation are shown. Details on each computation can be found in the readme and comments of the computation scripts (Source Code A.1). All genes were marked "knockable" and additionally oxygen uptake could be switched off by MCS if needed. Uncommon metabolite exchanges were removed prior to the computation. Allowed product exchanges were: acetate, CO_2, ethanol, formate, H_2, lactate, succinate, methanol

	Figure 3.5A	Figure 3.5B	Table 3.2	Table 3.3	Figure 3.6
Model	*i*ML1515 core (Model A.1)	*i*ML1515 core (Model A.1)	*i*ML1515	*i*ML1515	*i*ML1515 core (Model A.1)
(1) Desired region pGCP	$r_{BM} = r_{BM}^{max}$, $r_{BM} \geq 0.05$, $r_P \geq 0.01$	$r_{BM} = r_{BM}^{max}$, $r_{BM} \geq 0.05$, $r_P \geq 0.01$	$r_{BM} = r_{BM}^{max}$, $r_{BM} \geq 0.05$, $r_P \geq 0.2\ r_P^{max}$	$r_{BM} = r_{BM}^{max}$, $r_{BM} \geq 0.05$, $r_P \geq 0.2\ r_P^{max}$	$r_{BM} = r_{BM}^{max}$, $r_{BM} \geq 0.05$, $r_P \geq 0.01$
(2-5) Desired region wGCP, dGCP, SUCP, dACP	$r_{BM} \geq 0.05$	$r_{BM} \geq 0.05$	$r_{BM} \geq 0.05$	$r_{BM} \geq 0.05$	$r_{BM} \geq 0.05$
(2) Target wGCP	$r_{BM} = r_{BM}^{max}$, $r_P = 0$	$r_{BM} = r_{BM}^{max}$, $r_P = 0$	$r_{BM} = r_{BM}^{max}$, $r_P \leq 0.2\ r_P^{max}$	$r_{BM} = r_{BM}^{max}$, $r_P \leq 0.2\ r_P^{max}$	
(3) Target dGCP	$r_{BM} \geq 0.001$, $r_P = 0$	$r_{BM} \geq 0.001$, $r_P = 0$	$\frac{r_P}{r_{BM}} \leq \frac{0.2\ r_P^{max}}{0.2\ r_{BM}^{max}}$	$\frac{r_P}{r_{BM}} \leq \frac{0.2\ r_P^{max}}{0.2\ r_{BM}^{max}}$	$r_{BM} \geq 0.001$, $r_P = 0$
(4) Target SUCP	$r_P = 0$	$r_P = 0$	$\frac{r_P}{r_S} \leq \frac{0.2\ r_P^{max}}{r_S^{max}}$	$\frac{r_P}{r_S} \leq \frac{0.2\ r_P^{max}}{r_S^{max}}$	$r_P = 0$
(5) Target dACP					$r_{ATPM} \geq 0.01$, $r_P = 0$
Min ATPM	$r_{ATPM} \geq 6.86$	$r_{ATPM} \geq 0$	$r_{ATPM} \geq 6.86$	$r_{ATPM} \geq 6.86$	$r_{ATPM} \geq 0$
Enumeration method	exhaustive enumeration (*enumerateMCS*)	exhaustive enumeration (*enumerateMCS*)	find smallest MCS (*findSmallestMCS*)	find any MCS (*findAnyMCS*)	exhaustive enumeration (*enumerateMCS*)
MCS size limit	3	3	60	60	4
MCS solution limit	none	none	1	1	none
Time limit	none	none	6 runs, each 4 h per coupling type per product	12 runs, each 2 h per coupling type per product	none
Product	ethanol	ethanol	various	various	ethanol
Script file (included in Source Code A.1)	Source Code A.1 MCS_1_coupling_degrees.m	Source Code A.1 MCS_1_coupling_degrees.m	Source Code A.1 MCS_2_smallest.m	Source Code A.1 MCS_3_any.m	Source Code A.1 MCS_4_ACP.m
Identifier: r_{BM}	BIOMASS_Ec_iML1515_core_75p37M				
Identifier: r_S	EX_glc__D_e				
Identifier: r_P	EX_etoh_e	EX_etoh_e	see Table A.2	see Table A.2	EX_etoh_e

The following appendices are provided in digital form and are available at www.shaker.de/shop/978-3-8440-8411-5:

Table A.2: Reactions and species of the heterologous pathways that were added to the *i*ML1515 model for the exemplary MCS computations from Tables 3.2 and 3.3. All pathways were adapted from literature. Mass and charge balances were verified. The table can be found in: Table_A02_GCP_heterologous_pathways.xlsx.

Table A.3: Detailed results and benchmark for the computation of any MCSs with different degrees of GCP in a genome-scale setup (see Table 3.2). The table can be found in: Table_A03_GCP_benchmarks_any_mcs.xls.

Table A.4: Detailed results and benchmark for the computation of smallest MCSs with different degrees of GCP in a genome-scale setup (see Table 3.3). The table can be found in: Table_A04_GCP_benchmarks_smallest_mcs.xls.

Table A.5: Ranked list of MCSs computed in scenario 1 (see chapter 4, Tables 4.1 and 4.2) for substrate-uptake-coupled 2,3-BDO synthesis in the *E. coli* genome-scale model *i*ML1515. The table can be found in: Table_A05_MCS_extensions_scenario_1.xlsx.

Table A.6: Ranked list of MCSs computed in scenario 2 (see chapter 4, Table 4.2) for substrate-uptake-coupled 2,3-BDO synthesis in the *E. coli* genome-scale model *i*ML1515. The table can be found in: Table_A06_MCS_extensions_scenario_2.xlsx.

Table A.7: Ranked list of MCSs computed in scenario 3 (see chapter 4, Table 4.2) for substrate-uptake-coupled 2,3-BDO synthesis in the *E. coli* genome-scale model *i*ML1515. The table can be found in: Table_A07_MCS_extensions_scenario_3.xlsx.

Table A.8: Ranked list of MCSs computed in scenario 4 (see chapter 4, Table 4.2) for substrate-uptake-coupled 2,3-BDO synthesis in the *E. coli* genome-scale model *i*ML1515. The table can be found in: Table_A08_MCS_extensions_scenario_4.xlsx.

Table A.9: Results of MCS computations in chapter 4 for substrate-uptake-coupled 2,3-BDO synthesis in the genome-scale models of *S. cerevisiae* and *P. putida*. The table can be found in: Table_A09_MCS_extensions_S_cerevisiae.xlsx.

Table A.10: Characterization and ranking of MCS for the substrate-uptake-coupled production of L-methionine in *E. coli*. The table can be found in: Table_A10_Ranking_L_methionine_mcs.xlsx.

Table A.11: Characterization and ranking of MCS for the substrate-uptake-coupled production of 1,4-BDO in *E. coli*. The table can be found in: Table_A11_Ranking_1-4-BDO_mcs.xlsx.

Table A.12: MCS setups for the computation of strain designs with substrate-uptake-coupled production of L-methionine and 1,4-BDO using an adapted version of the genome-scale *E. coli* model *i*JO1366. The table can be found in: Table_A12_Ranking_mcs_setup.xlsx.

Model A.1: *i*ML1515core - A metabolic core model of *E. coli* derived from *i*ML1515. The *i*ML1515core model is provided in the SBML file Model_A1_iML1515core.xml.

Source Code A.1: The MATLAB source code and the models for all computations in chapter 3 can be found in the directory Source_Code_S1_GCP.

Source Code A.2: The MATLAB source code and the models for all computations in chapter 4 can be found in the directory Source_Code_S2_MCS_extensions.

Source Code A.3: The MATLAB source code for all computations in chapter 5, including the *i*JO1366 model, can be found in the directory Source_Code_S3_Ranking_MCS.

Source Code A.4: The MATLAB toolbox *CellNetAnalyzer* (version 2021.1) can be found in the directory Source_Code_S4_CellNetAnalyzer_2021.1.

Bibliography

[1] Schneider P., Klamt S. (2019). Characterizing and ranking computed metabolic engineering strategies. *Bioinformatics*, **35**, 17, 3063–3072, doi: 10.1093/bioinformatics/bty1065.

[2] Schneider P., Kamp A. v., Klamt S. (2020). An extended and generalized framework for the calculation of metabolic intervention strategies based on minimal cut sets. *PLOS Computational Biology*, **16**, 7, e1008110, doi: 10.1371/journal.pcbi.1008110.

[3] Schneider P., Mahadevan R., Klamt S. (2021). Systematizing the different notions of growth-coupled product synthesis and a single framework for computing corresponding strain designs. *Biotechnology Journal*, **16**, 12, 2100236, doi: 10.1002/biot.202100236.

[4] United Nations F. C. o. C. C. (2016). *Report of the Conference of the Parties on its twenty-first session, held in Paris from 30 November to 13 December 2015*, https://unfccc.int/resource/docs/2015/cop21/eng/10a01.pdf (**accessed:** 2021/03/09).

[5] Van Dien S. (2013). From the first drop to the first truckload: commercialization of microbial processes for renewable chemicals. *Current Opinion in Biotechnology*, **24**, 6, 1061–1068, doi: 10.1016/j.copbio.2013.03.002.

[6] Liu J.-M., Solem C., Jensen P. R. (2020). Harnessing biocompatible chemistry for developing improved and novel microbial cell factories. *Microbial Biotechnology*, **13**, 1, 54–66, doi: 10.1111/1751-7915.13472.

[7] Pickens L. B., Tang Y. (2009). Decoding and engineering tetracycline biosynthesis. *Metabolic Engineering*, **11**, 2, 69–75, doi: 10.1016/j.ymben.2008.10.001.

[8] Ledesma-Amaro R., Jiménez A., Santos M. A., Revuelta J. L. (2013). Biotechnological production of feed nucleotides by microbial strain improvement. *Process Biochemistry*, **48**, 9, 1263–1270, doi: 10.1016/j.procbio.2013.06.025.

[9] Spadiut O., Capone S., Krainer F., Glieder A., Herwig C. (2014). Microbials for the production of monoclonal antibodies and antibody fragments. *Trends in Biotechnology*, **32**, 1, 54–60, doi: 10.1016/j.tibtech.2013.10.002.

[10] Yuan S.-F., Alper H. S. (2019). Metabolic engineering of microbial cell factories for production of nutraceuticals. *Microbial Cell Factories*, **18**, 1, 46, doi: 10.1186/s12934-019-1096-y.

[11] Tripathi N. K., Shrivastava A. (2019). Recent Developments in Bioprocessing of Recombinant Proteins: Expression Hosts and Process Development. *Frontiers in Bioengineering and Biotechnology*, **7**, doi: 10.3389/fbioe.2019.00420.

[12] Zhang J., Petersen S. D., Radivojevic T., Ramirez A., Pérez-Manríquez A., Abeliuk E., Sánchez B. J., Costello Z., Chen Y., Fero M. J., Martin H. G., Nielsen J., Keasling J. D., Jensen M. K. (2020). Combining mechanistic and machine learning models for predictive engineering and optimization of tryptophan metabolism. *Nature Communications*, **11**, 1, 4880, doi: 10.1038/s41467-020-17910-1.

[13] Hanga M. P., Ali J., Moutsatsou P., Raga F. A. d. l., Hewitt C. J., Nienow A., Wall I. (2020). Bioprocess development for scalable production of cultivated meat. *Biotechnology and Bioengineering*, **117**, 10, 3029–3039, doi: https://doi.org/10.1002/bit.27469.

[14] Choi S., Song C. W., Shin J. H., Lee S. Y. (2015). Biorefineries for the production of top building block chemicals and their derivatives. *Metabolic Engineering*, **28**, 223–239, doi: 10.1016/j.ymben.2014.12.007.

[15] Biz A., Proulx S., Xu Z., Siddartha K., Mulet Indrayanti A., Mahadevan R. (2019). Systems biology based metabolic engineering for non-natural chemicals. *Biotechnology Advances*, **37**, 6, 107379, doi: 10.1016/j.biotechadv.2019.04.001.

[16] Lee S. Y., Kim H. U., Chae T. U., Cho J. S., Kim J. W., Shin J. H., Kim D. I., Ko Y.-S., Jang W. D., Jang Y.-S. (2019). A comprehensive metabolic map for production of bio-based chemicals. *Nature Catalysis*, **2**, 1, 18–33, doi: 10.1038/s41929-018-0212-4.

[17] Casini A., Chang F.-Y., Eluere R., King A. M., Young E. M., Dudley Q. M., Karim A., Pratt K., Bristol C., Forget A., Ghodasara A., Warden-Rothman R., Gan R., Cristofaro A., Borujeni A. E., Ryu M.-H., Li J., Kwon Y.-C., Wang H., Tatsis E., Rodriguez-Lopez C., O'Connor S., Medema M. H., Fischbach M. A., Jewett M. C., Voigt C., Gordon D. B. (2018). A Pressure Test to Make 10 Molecules in 90 Days: External Evaluation of Methods to Engineer Biology. *Journal of the American Chemical Society*, **140**, 12, 4302–4316, doi: 10.1021/jacs.7b13292.

[18] Yu D., Ellis H. M., Lee E.-C., Jenkins N. A., Copeland N. G., Court D. L. (2000). An efficient recombination system for chromosome engineering in *Escherichia coli*. *Proceedings of the National Academy of Sciences*, **97**, 11, 5978–5983, doi: 10.1073/pnas.100127597.

[19] Jensen S. I., Lennen R. M., Herrgård M. J., Nielsen A. T. (2015). Seven gene deletions in seven days: Fast generation of *Escherichia coli*strains tolerant to acetate and osmotic stress. *Scientific Reports*, **5**, 1, 17874, doi: 10.1038/srep17874.

[20] Doudna J. A., Charpentier E. (2014). The new frontier of genome engineering with CRISPR-Cas9. *Science*, **346**, 6213, 1258096, doi: 10.1126/science.1258096.

[21] Chen G.-Q., Jiang X.-R. (2018). Next generation industrial biotechnology based on extremophilic bacteria. *Current Opinion in Biotechnology*, **50**, 94–100, doi: 10.1016/j.copbio.2017.11.016.

[22] Burg J. M., Cooper C. B., Ye Z., Reed B. R., Moreb E. A., Lynch M. D. (2016). Large-scale bioprocess competitiveness: the potential of dynamic metabolic control in two-stage fermentations. *Current Opinion in Chemical Engineering*, Biotechnology and bioprocess engineering / Process systems engineering **14**, 121–136, doi: 10.1016/j.coche.2016.09.008.

[23] Woodley J. M. (2020). Towards the sustainable production of bulk-chemicals using biotechnology. *New Biotechnology*, **59**, 59–64, doi: 10.1016/j.nbt.2020.07.002.

[24] Lane J. (2018). Biofuels Digest: Rennovia's demise, the Triple Rule, and the pursuit of sustainable nylon in a world of low oil prices, https://www.biofuelsdigest.com/bdigest/2018/03/09/rennovias-demise-the-rule-of-thirds-and-the-pursuit-of-sustainable-nylon-in-a-world-of-low-oil-prices/ (**accessed:** 2021/03/11).

[25] Keasling J. D. (2010). Manufacturing Molecules Through Metabolic Engineering. *Science*, **330**, 6009, 1355–1358, doi: 10.1126/science.1193990.

[26] Becker J., Wittmann C. (2015). Advanced Biotechnology: Metabolically Engineered Cells for the Bio-Based Production of Chemicals and Fuels, Materials, and Health-Care Products. *Angewandte Chemie International Edition*, **54**, 11, 3328–3350, doi: 10.1002/anie.201409033.

[27] Lee S. Y., Kim H. U. (2015). Systems strategies for developing industrial microbial strains. *Nature Biotechnology*, **33**, 10, 1061–1072, doi: 10.1038/nbt.3365.

[28] Tsuge Y., Kawaguchi H., Sasaki K., Kondo A. (2016). Engineering cell factories for producing building block chemicals for bio-polymer synthesis. *Microbial Cell Factories*, **15**, 1, doi: 10.1186/s12934-016-0411-0.

[29] Banerjee D., Eng T., Lau A. K., Sasaki Y., Wang B., Chen Y., Prahl J.-P., Singan V. R., Herbert R. A., Liu Y., Tanjore D., Petzold C. J., Keasling J. D., Mukhopadhyay A. (2020). Genome-scale metabolic rewiring improves titers rates and yields of the non-native product indigoidine at scale. *Nature Communications*, **11**, 1, 5385, doi: 10.1038/s41467-020-19171-4.

[30] Heirendt L., Arreckx S., Pfau T., Mendoza S. N., Richelle A., Heinken A., Haraldsdóttir H. S., Wachowiak J., Keating S. M., Vlasov V., Magnusdóttir S., Ng C. Y., Preciat G., Žagare A., Chan S. H. J., Aurich M. K., Clancy C. M., Modamio J., Sauls J. T., Noronha A., Bordbar A., Cousins B., El Assal D. C., Valcarcel L. V., Apaolaza I., Ghaderi S., Ahookhosh M., Ben Guebila M., Kostromins A., Sompairac N., Le H. M., Ma D., Sun Y., Wang L., Yurkovich J. T., Oliveira M. A. P., Vuong P. T., El Assal L. P., Kuperstein I., Zinovyev A., Hinton H. S., Bryant W. A., Aragón Artacho F. J., Planes F. J., Stalidzans E., Maass A., Vempala S., Hucka M., Saunders M. A., Maranas C. D., Lewis N. E., Sauter T., Palsson B. Ø., Thiele I., Fleming R. M. T. (2019). Creation and analysis of biochemical constraint-based models using the COBRA Toolbox v.3.0. *Nature Protocols*, **14**, 3, 639–702, doi: 10.1038/s41596-018-0098-2.

[31] Kamp A. von, Klamt S. (2017). Growth-coupled overproduction is feasible for almost all metabolites in five major production organisms. *Nature Communications*, **8**, 15956, doi: 10.1038/ncomms15956.

[32] Yasemi M., Jolicoeur M. (2021). Modelling Cell Metabolism: A Review on Constraint-Based Steady-State and Kinetic Approaches. *Processes*, **9**, 2, 322, doi: 10.3390/pr9020322.

[33] Lawson C. E., Martí J. M., Radivojevic T., Jonnalagadda S. V. R., Gentz R., Hillson N. J., Peisert S., Kim J., Simmons B. A., Petzold C. J., Singer S. W., Mukhopadhyay A., Tanjore D., Dunn J. G., Garcia Martin H. (2021). Machine learning for metabolic engineering: A review. *Metabolic Engineering*, Tools and Strategies of Metabolic Engineering **63**, 34–60, doi: 10.1016/j.ymben.2020.10.005.

[34] Gu C., Kim G. B., Kim W. J., Kim H. U., Lee S. Y. (2019). Current status and applications of genome-scale metabolic models. *Genome Biology*, **20**, 1, 121, doi: 10.1186/s13059-019-1730-3.

[35] Fang X., Lloyd C. J., Palsson B. O. (2020). Reconstructing organisms in silico: genome-scale models and their emerging applications. *Nature Reviews Microbiology*, **18**, 12, 731–743, doi: 10.1038/s41579-020-00440-4.

[36] King Z. A., Lu J., Dräger A., Miller P., Federowicz S., Lerman J. A., Ebrahim A., Palsson B. O., Lewis N. E. (2016). BiGG Models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Research*, **44** D1, D515–D522, doi: 10.1093/nar/gkv1049.

[37] Bordbar A., Monk J. M., King Z. A., Palsson B. O. (2014). Constraint-based models predict metabolic and associated cellular functions. *Nature Reviews Genetics*, **15**, 2, 107–120, doi: 10.1038/nrg3643.

[38] Feist A. M., Herrgård M. J., Thiele I., Reed J. L., Palsson B. Ø. (2009). Reconstruction of biochemical networks in microorganisms. *Nature Reviews. Microbiology*, **7**, 2, 129–143, doi: 10.1038/nrmicro1949.

[39] Klamt S., Hädicke O., Kamp A. von (2014). Stoichiometric and Constraint-Based Analysis of Biochemical Reaction Networks, in: Modeling and Simulation in Science, Engineering and Technology, 263–316, ISBN: 978-3-319-08437-4, doi: 10.1007/978-3-319-08437-4_5.

[40] Lewis N. E., Nagarajan H., Palsson B. O. (2012). Constraining the metabolic genotype-phenotype relationship using a phylogeny of in silico methods. *Nature Reviews. Microbiology*, **10**, 4, 291–305, doi: 10.1038/nrmicro2737.

[41] Burgard A. P., Pharkya P., Maranas C. D. (2003). Optknock: A bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnology and Bioengineering*, **84**, 6, 647–657, doi: 10.1002/bit.10803.

[42] Sandberg T. E., Salazar M. J., Weng L. L., Palsson B. O., Feist A. M. (2019). The emergence of adaptive laboratory evolution as an efficient tool for biological discovery and industrial biotechnology. *Metabolic Engineering*, **56**, 1–16, doi: 10.1016/j.ymben.2019.08.004.

[43] Feist A. M., Zielinski D. C., Orth J. D., Schellenberger J., Herrgard M. J., Palsson B. Ø. (2010). Model-driven evaluation of the production potential for growth-coupled products of *Escherichia coli*. *Metabolic Engineering*, **12**, 3, 173–186, doi: 10.1016/j.ymben.2009.10.003.

[44] Tepper N., Shlomi T. (2010). Predicting metabolic engineering knockout strategies for chemical production: accounting for competing pathways. *Bioinformatics*, **26**, 4, 536–543, doi: 10.1093/bioinformatics/btp704.

[45] Campodonico M. A., Andrews B. A., Asenjo J. A., Palsson B. O., Feist A. M. (2014). Generation of an atlas for commodity chemical production in *Escherichia coli* and a novel pathway prediction algorithm, GEM-Path. *Metabolic Engineering*, **25**, 140–158, doi: 10.1016/j.ymben.2014.07.009.

[46] Tervo C. J., Reed J. L. (2014). Expanding metabolic engineering algorithms using feasible space and shadow price constraint modules. *Metabolic Engineering Communications*, **1**, 1–11, doi: 10.1016/j.meteno.2014.06.001.

[47] Klamt S., Mahadevan R. (2015). On the feasibility of growth-coupled product synthesis in microbial strains. *Metabolic Engineering*, **30**, 166–178, doi: 10.1016/j.ymben.2015.05.006.

[48] Garcia S., Trinh C. T. (2019). Multiobjective strain design: A framework for modular cell engineering. *Metabolic Engineering*, **51**, 110–120, doi: 10.1016/j.ymben.2018.09.003.

[49] Fong S. S., Burgard A. P., Herring C. D., Knight E. M., Blattner F. R., Maranas C. D., Palsson B. O. (2005). In silico design and adaptive evolution of *Escherichia coli* for production of lactic acid. *Biotechnology and Bioengineering*, **91**, 5, 643–648, doi: 10.1002/bit.20542.

[50] Trinh C. T., Unrean P., Srienc F. (2008). Minimal *Escherichia coli* Cell for the Most Efficient Production of Ethanol from Hexoses and Pentoses. *Applied and Environmental Microbiology*, **74**, 12, 3634–3643, doi: 10.1128/AEM.02708-07.

[51] Yim H., Haselbeck R., Niu W., Pujol-Baxley C., Burgard A., Boldt J., Khandurina J., Trawick J. D., Osterhout R. E., Stephen R., Estadilla J., Teisan S., Schreyer H. B., Andrae S., Yang T. H., Lee S. Y., Burk M. J., Van Dien S. (2011). Metabolic engineering of *Escherichia coli* for direct production of 1,4-butanediol. *Nature Chemical Biology*, **7**, 7, 445–452, doi: 10.1038/nchembio.580.

[52] Ng C., Jung M.-y., Lee J., Oh M.-K. (2012). Production of 2,3-butanediol in *Saccharomyces cerevisiae* by in silico aided metabolic engineering. *Microbial Cell Factories*, **11**, 1, 68, doi: 10.1186/1475-2859-11-68.

[53] Otero J. M., Cimini D., Patil K. R., Poulsen S. G., Olsson L., Nielsen J. (2013). Industrial Systems Biology of *Saccharomyces cerevisiae*Enables Novel Succinic Acid Cell Factory. *PLOS ONE*, **8**, 1, e54144, doi: 10.1371/journal.pone.0054144.

[54] Harder B.-J., Bettenbrock K., Klamt S. (2016). Model-based metabolic engineering enables high yield itaconic acid production by *Escherichia coli*. *Metabolic Engineering*, **38**, 29–37, doi: 10.1016/j.ymben.2016.05.008.

[55] Wilbanks B., Layton D. S., Garcia S., Trinh C. T. (2018). A Prototype for Modular Cell Engineering. *ACS Synthetic Biology*, **7**, 1, 187–199, doi: 10.1021/acssynbio.7b00269.

[56] Garcia S., Trinh C. T. (2020). Harnessing Natural Modularity of Metabolism with Goal Attainment Optimization to Design a Modular Chassis Cell for Production of Diverse Chemicals. *ACS Synthetic Biology*, **9**, 7, 1665–1681, doi: 10.1021/acssynbio.9b00518.

[57] Hädicke O., Klamt S. (2011). Computing complex metabolic intervention strategies using constrained minimal cut sets. *Metabolic Engineering*, **13**, 2, 204–213, doi: 10.1016/j.ymben.2010.12.004.

[58] Machado D., Herrgård M. J. (2015). Co-evolution of strain design methods based on flux balance and elementary mode analysis. *Metabolic Engineering Communications*, **2**, 85–92, doi: 10.1016/j.meteno.2015.04.001.

[59] Maia P., Rocha M., Rocha I. (2016). In Silico Constraint-Based Strain Optimization Methods: the Quest for Optimal Cell Factories. *Microbiology and Molecular Biology Reviews*, **80**, 1, 45–67, doi: 10.1128/MMBR.00014-15.

[60] Kamp A. von, Klamt S. (2014). Enumeration of Smallest Intervention Strategies in Genome-Scale Metabolic Networks. *PLOS Computational Biology*, **10**, 1, e1003378, doi: 10.1371/journal.pcbi.1003378.

[61] Song C., Park J., Chung S., Lee S., Song H. (2019). Microbial production of 2,3-butanediol for industrial applications. *Journal of Industrial Microbiology and Biotechnology*, **46**, 11, 1583–1601, doi: 10.1007/s10295-019-02231-0.

[62] Boecker S., Harder B.-J., Kutscha R., Pflügl S., Klamt S. (2021). Increasing ATP turnover boosts productivity of 2,3-butanediol synthesis in *Escherichia coli*. *Microbial Cell Factories*, **20**, 1, 63, doi: 10.1186/s12934-021-01554-x.

[63] Matousek J., Gärtner B. (2007). Understanding and Using Linear Programming., ISBN: 978-3-540-30697-9, doi: 10.1007/978-3-540-30717-4.

[64] Schrijver A. (2011). Theory of linear and integer programming., ISBN: 978-0-471-98232-6.

[65] Karmarkar N. (1984). A new polynomial-time algorithm for linear programming. *Combinatorica*, **4**, 4, 373–395, doi: 10.1007/BF02579150.

[66] Vanderbei R. J. (2008). Linear programming: foundations and extensions. 114, ISBN: 978-0-387-74388-2 978-0-387-74387-5.

[67] Charnes A., Cooper W. W. (1962). Programming with linear fractional functionals. *Naval Research Logistics Quarterly*, **9**, 3, 181–186, doi: 10.1002/nav.3800090303.

[68] Garey M. R., Johnson D. S. (2009). Computers and intractability: a guide to the theory of NP-completeness., ISBN: 978-0-7167-1045-5 978-0-7167-1044-8.

[69] Bohlinger J. D., Miller R. E. (1972). Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York., ISBN: 978-1-4684-2001-2.

[70] Vlassis N., Pacheco M. P., Sauter T. (2014). Fast Reconstruction of Compact Context-Specific Metabolic Network Models. *PLOS Computational Biology*, **10**, 1, e1003424, doi: 10.1371/journal.pcbi.1003424.

[71] Lieven C., Beber M. E., Olivier B. G., Bergmann F. T., Ataman M., Babaei P., Bartell J. A., Blank L. M., Chauhan S., Correia K., Diener C., Dräger A., Ebert B. E., Edirisinghe J. N., Faria J. P., Feist A. M., Fengos G., Fleming R. M. T., García-Jiménez B., Hatzimanikatis V., Helvoirt W. van, Henry C. S., Hermjakob H., Herrgård M. J., Kaafarani A., Kim H. U., King Z., Klamt S., Klipp E., Koehorst J. J., König M., Lakshmanan M., Lee D.-Y., Lee S. Y., Lee S., Lewis N. E., Liu F., Ma H., Machado D., Mahadevan R., Maia P., Mardinoglu A., Medlock G. L., Monk J. M., Nielsen J., Nielsen L. K., Nogales J., Nookaew I., Palsson B. O., Papin J. A., Patil K. R., Poolman M., Price N. D., Resendis-Antonio O., Richelle A., Rocha I., Sánchez B. J., Schaap P. J., Malik Sheriff R. S., Shoaie S., Sonnenschein N., Teusink B., Vilaça P., Vik J. O., Wodke J. A. H., Xavier J. C., Yuan Q., Zakhartsev M., Zhang C. (2020). MEMOTE for standardized genome-scale metabolic model testing. *Nature Biotechnology*, **38**, 3, 272–276, doi: 10.1038/s41587-020-0446-y.

[72] Ebrahim A., Lerman J. A., Palsson B. O., Hyduke D. R. (2013). COBRApy: COnstraints-Based Reconstruction and Analysis for Python. *BMC Systems Biology*, **7**, 1, 74, doi: 10.1186/1752-0509-7-74.

[73] Rocha I., Maia P., Evangelista P., Vilaça P., Soares S., Pinto J. P., Nielsen J., Patil K. R., Ferreira E. C., Rocha M. (2010). OptFlux: an open-source software platform for in silico metabolic engineering. *BMC Systems Biology*, **4**, 1, 45, doi: 10.1186/1752-0509-4-45.

[74] Kamp A. von, Thiele S., Hädicke O., Klamt S. (2017). Use of *CellNetAnalyzer* in biotechnology and metabolic engineering. *Journal of Biotechnology*, **261**, 221–228, doi: 10.1016/j.jbiotec.2017.05.001.

[75] Orth J. D., Thiele I., Palsson B. Ø. (2010). What is flux balance analysis? *Nature biotechnology*, **28**, 3, 245–248, doi: 10.1038/nbt.1614.

[76] Rawls K. D., Dougherty B. V., Blais E. M., Stancliffe E., Kolling G. L., Vinnakota K., Pannala V. R., Wallqvist A., Papin J. A. (2019). A simplified metabolic network reconstruction to promote understanding and development of flux balance analysis tools. *Computers in Biology and Medicine*, **105**, 64–71, doi: 10.1016/j.compbiomed.2018.12.010.

[77] Klamt S., Regensburger G., Gerstl M. P., Jungreuthmayer C., Schuster S., Mahadevan R., Zanghellini J., Mueller S. (2017). From elementary flux modes to elementary flux vectors: Metabolic pathway analysis with arbitrary linear flux constraints. *PLOS Computational Biology*, **13**, 4, e1005409, doi: 10.1371/journal.pcbi.1005409.

[78] Schuster S., Fell D. A., Dandekar T. (2000). A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature Biotechnology*, **18**, 3, 326–332, doi: 10.1038/73786.

[79] Machado D., Herrgård M. J., Rocha I. (2016). Stoichiometric Representation of Gene-Protein-Reaction Associations Leverages Constraint-Based Analysis from Reaction to Gene-Level Phenotype Prediction. *PLOS Computational Biology*, **12**, 10, e1005140, doi: 10.1371/journal.pcbi.1005140.

[80] Covert M. W., Schilling C. H., Palsson B. (2001). Regulation of Gene Expression in Flux Balance Models of Metabolism. *Journal of Theoretical Biology*, **213**, 1, 73–88, doi: 10.1006/jtbi.2001.2405.

[81] Jensen P. A., Papin J. A. (2011). Functional integration of a metabolic network model and expression data without arbitrary thresholding. *Bioinformatics*, **27**, 4, 541–547, doi: 10.1093/bioinformatics/btq702.

[82] Bekiaris P. S., Klamt S. (2020). Automatic construction of metabolic models with enzyme constraints. *BMC Bioinformatics*, **21**, doi: 10.1186/s12859-019-3329-9.

[83] Sánchez B. J., Zhang C., Nilsson A., Lahtvee P.-J., Kerkhoven E. J., Nielsen J. (2017). Improving the phenotype predictions of a yeast genome-scale metabolic model by incorporating enzymatic constraints. *Molecular Systems Biology*, **13**, 8, doi: 10.15252/msb.20167411.

[84] Soh K. C., Hatzimanikatis V. (2014). Constraining the Flux Space Using Thermodynamics and Integration of Metabolomics Data, 49–63, ISBN: 978-1-4939-1170-7, doi: 10.1007/978-1-4939-1170-7_3.

[85] Maranas C. D., Zomorrodi A. R. (2016). Optimization Methods in Metabolic Networks., ISBN: 978-1-119-02849-9.

[86] Palsson B. O. (2015). Systems Biology: Constraint-Based Reconstruction and Analysis., ISBN: 978-1-139-85461-0, doi: 10.1017/CBO9781139854610.

[87] Klamt S., Müller S., Regensburger G., Zanghellini J. (2018). A mathematical framework for yield (vs. rate) optimization in constraint-based modeling and applications in metabolic engineering. *Metabolic Engineering*, **47**, 153–169, doi: 10.1016/j.ymben.2018.02.001.

[88] Monk J. M., Lloyd C. J., Brunk E., Mih N., Sastry A., King Z., Takeuchi R., Nomura W., Zhang Z., Mori H., Feist A. M., Palsson B. O. (2017). *i*ML1515, a knowledgebase that computes *Escherichia coli* traits. *Nature Biotechnology*, **35**, 10, 904–908, doi: 10.1038/nbt.3956.

[89] Long M. R., Ong W. K., Reed J. L. (2015). Computational methods in metabolic engineering for strain design. *Current Opinion in Biotechnology*, **34**, 135–141, doi: 10.1016/j.copbio.2014.12.019.

[90] Pharkya P., Burgard A. P., Maranas C. D. (2004). OptStrain: a computational framework for redesign of microbial production systems. *Genome Research*, **14**, 11, 2367–2376, doi: 10.1101/gr.2872004.

[91] Kim J., Reed J. L. (2010). OptORF: Optimal metabolic and regulatory perturbations for metabolic engineering of microbial strains. *BMC Systems Biology*, **4**, 1, 53, doi: 10.1186/1752-0509-4-53.

[92] Tervo C. J., Reed J. L. (2012). FOCAL: an experimental design tool for systematizing metabolic discoveries and model development. *Genome Biology*, **13**, 12, R116, doi: 10.1186/gb-2012-13-12-r116.

[93] Jensen K., Broeken V., Hansen A. S. L., Sonnenschein N., Herrgård M. J. (2019). OptCouple: Joint simulation of gene knockouts, insertions and medium modifications for prediction of growth-coupled strain designs. *Metabolic Engineering Communications*, **8**, e00087, doi: 10.1016/j.mec.2019.e00087.

[94] Klamt S., Gilles E. D. (2004). Minimal cut sets in biochemical reaction networks. *Bioinformatics*, **20**, 2, 226–234, doi: 10.1093/bioinformatics/btg395.

[95] Klamt S. (2006). Generalized concept of minimal cut sets in biochemical networks. *Biosystems*, 5th International Conference on Systems Biology **83**, 2, 233–247, doi: 10.1016/j.biosystems.2005.04.009.

[96] Haus U.-U., Klamt S., Stephen T. (2008). Computing knock-out strategies in metabolic networks. *Journal of Computational Biology: A Journal of Computational Molecular Cell Biology*, **15**, 3, 259–268, doi: 10.1089/cmb.2007.0229.

[97] Jungreuthmayer C., Beurton-Aimar M., Zanghellini J. (2013). Fast computation of minimal cut sets in metabolic networks with a Berge algorithm that utilizes binary bit pattern trees. *IEEE/ACM transactions on computational biology and bioinformatics*, **10**, 5, 1329–1333, doi: 10.1109/tcbb.2013.116.

[98] Jungreuthmayer C., Nair G., Klamt S., Zanghellini J. (2013). Comparison and improvement of algorithms for computing minimal cut sets. *BMC Bioinformatics*, **14**, 1, 318, doi: 10.1186/1471-2105-14-318.

[99] Klamt S., Stelling J. (2002). Combinatorial complexity of pathway analysis in metabolic networks. *Molecular Biology Reports*, **29**, 1, 233–236, doi: 10.1023/a:1020390132244.

[100] Ballerstein K., Kamp A. von, Klamt S., Haus U.-U. (2012). Minimal cut sets in a metabolic network are elementary modes in a dual network. *Bioinformatics*, **28**, 3, 381–387, doi: 10.1093/bioinformatics/btr674.

[101] Tobalina L., Pey J., Planes F. J. (2016). Direct calculation of minimal cut sets involving a specific reaction knock-out. *Bioinformatics*, **32**, 13, 2001–2007, doi: 10.1093/bioinformatics/btw072.

[102] Röhl A., Riou T., Bockmayr A. (2019). Computing irreversible minimal cut sets in genome-scale metabolic networks via flux cone projection. *Bioinformatics*, **35**, 15, 2618–2625, doi: 10.1093/bioinformatics/bty1027.

[103] Miraskarshahi R., Zabeti H., Stephen T., Chindelevitch L. (2019). MCS2: minimal coordinated supports for fast enumeration of minimal cut sets in metabolic networks. *Bioinformatics*, **35**, 14, i615–i623, doi: 10.1093/bioinformatics/btz393.

[104] Klamt S., Mahadevan R., Kamp A. von (2020). Speeding up the core algorithm for the dual calculation of minimal cut sets in large metabolic networks. *BMC Bioinformatics*, **21**, 1, 510, doi: 10.1186/s12859-020-03837-3.

[105] Mahadevan R., Kamp A. von, Klamt S. (2015). Genome-scale strain designs based on regulatory minimal cut sets. *Bioinformatics*, **31**, 17, 2844–2851, doi: 10.1093/bioinformatics/btv217.

[106] Venayak N., Kamp A. von, Klamt S., Mahadevan R. (2018). MoVE identifies metabolic valves to switch between phenotypic states. *Nature Communications*, **9**, 5332, doi: 10.1038/s41467-018-07719-4.

[107] Apaolaza I., José-Eneriz E. S., Tobalina L., Miranda E., Garate L., Agirre X., Prósper F., Planes F. J. (2017). An in-silico approach to predict and exploit synthetic lethality in cancer metabolism. *Nature Communications*, **8**, 1, 1–9, doi: 10.1038/s41467-017-00555-y.

[108] Apaolaza I., Valcarcel L. V., Planes F. J. (2019). gMCS: fast computation of genetic minimal cut sets in large networks. *Bioinformatics*, **35**, 3, 535–537, doi: 10.1093/bioinformatics/bty656.

[109] Orth J. D., Conrad T. M., Na J., Lerman J. A., Nam H., Feist A. M., Palsson B. Ø. (2011). A comprehensive genome-scale reconstruction of *Escherichia coli* metabolism. *Molecular Systems Biology*, **7**, 1, 535, doi: 10.1038/msb.2011.65.

[110] Hädicke O., Klamt S. (2017). *EColiCore2*: a reference network model of the central metabolism of *Escherichia coli* and relationships to its genome-scale parent model. *Scientific Reports*, **7**, 39647, doi: 10.1038/srep39647.

[111] Klamt S., Saez-Rodriguez J., Gilles E. D. (2007). Structural and functional analysis of cellular networks with *CellNetAnalyzer*. *BMC Systems Biology*, **1**, 2, doi: 10.1186/1752-0509-1-2.

[112] Thiele S., Kamp A. von, Bekiaris P. S., Schneider P., Klamt S. (2021). CNApy: a CellNetAnalyzer GUI in Python for Analyzing and Designing Metabolic Networks. *Bioinformatics*, btab828, doi: 10.1093/bioinformatics/btab828.

[113] Vilaça P., Maia P., Giesteira H., Rocha I., Rocha M. (2018). Analyzing and Designing Cell Factories with OptFlux. *Methods in Molecular Biology (Clifton, N.J.)*, **1716**, 37–76, doi: 10.1007/978-1-4939-7528-0_2.

[114] Klamt S., Stelling J., Ginkel M., Gilles E. D. (2003). FluxAnalyzer: exploring structure, pathways, and flux distributions in metabolic networks on interactive flux maps. *Bioinformatics*, **19**, 2, 261–269, doi: 10.1093/bioinformatics/19.2.261.

[115] Kim J., Reed J. L., Maravelias C. T. (2011). Large-Scale Bi-Level Strain Design Approaches and Mixed-Integer Programming Solution Techniques. *PLOS ONE*, **6**, 9, e24162, doi: 10.1371/journal.pone.0024162.

[116] Alter T. B., Blank L. M., Ebert B. E. (2018). Genetic Optimization Algorithm for Metabolic Engineering Revisited. *Metabolites*, **8**, 2, 33, doi: 10.3390/metabo8020033.

[117] Trinh C. T., Liu Y., Conner D. J. (2015). Rational design of efficient modular cells. *Metabolic Engineering*, **32**, 220–231, doi: 10.1016/j.ymben.2015.10.005.

[118] Trinh C. T., Srienc F. (2009). Metabolic engineering of *Escherichia coli* for efficient conversion of glycerol to ethanol. *Applied and Environmental Microbiology*, **75**, 21, 6696–6705, doi: 10.1128/AEM.00670-09.

[119] Alter T. B., Ebert B. E. (2019). Determination of growth-coupling strategies and their underlying principles. *BMC Bioinformatics*, **20**, 1, 447, doi: 10.1186/s12859-019-2946-7.

[120] Groot D. H. de, Lischke J., Muolo R., Planqué R., Bruggeman F. J., Teusink B. (2020). The common message of constraint-based optimization approaches: overflow metabolism is caused by two growth-limiting constraints. *Cellular and Molecular Life Sciences*, **77**, 3, 441–453, doi: 10.1007/s00018-019-03380-2.

[121] Burgard A. P., Nikolaev E. V., Schilling C. H., Maranas C. D. (2004). Flux Coupling Analysis of Genome-Scale Metabolic Network Reconstructions. *Genome Research*, **14**, 2, 301–312, doi: 10.1101/gr.1926504.

[122] Larhlimi A., David L., Selbig J., Bockmayr A. (2012). F2C2: a fast tool for the computation of flux coupling in genome-scale metabolic networks. *BMC Bioinformatics*, **13**, 57, doi: 10.1186/1471-2105-13-57.

[123] Reimers A. C., Goldstein Y., Bockmayr A. (2015). Generic flux coupling analysis. *Mathematical Biosciences*, **262**, 28–35, doi: 10.1016/j.mbs.2015.01.003.

[124] Patil K. R., Rocha I., Förster J., Nielsen J. (2005). Evolutionary programming as a platform for in silico metabolic engineering. *BMC Bioinformatics*, **6**, 308, doi: 10.1186/1471-2105-6-308.

[125] Vieira V., Maia P., Rocha M., Rocha I. (2019). Comparison of pathway analysis and constraint-based methods for cell factory design. *BMC Bioinformatics*, **20**, 1, 350, doi: 10.1186/s12859-019-2934-y.

[126] Hädicke O., Klamt S. (2010). CASOP: A Computational Approach for Strain Optimization aiming at high Productivity. *Journal of Biotechnology*, **147**, 2, 88–101, doi: 10.1016/j.jbiotec.2010.03.006.

[127] Shabestary K., Hudson E. P. (2016). Computational metabolic engineering strategies for growth-coupled biofuel production by *Synechocystis*. *Metabolic Engineering Communications*, **3**, 216–226, doi: 10.1016/j.meteno.2016.07.003.

[128] Navas M. A., Cerdán S., Gancedo J. M. (1993). Futile cycles in *Saccharomyces cerevisiae* strains expressing the gluconeogenic enzymes during growth on glucose. *Proceedings of the National Academy of Sciences of the United States of America*, **90**, 4, 1290–1294, doi: 10.1073/pnas.90.4.1290.

[129] Hädicke O., Bettenbrock K., Klamt S. (2015). Enforced ATP futile cycling increases specific productivity and yield of anaerobic lactate production in *Escherichia coli*. *Biotechnology and Bioengineering*, **112**, 10, 2195–2199, doi: 10.1002/bit.25623.

[130] Hädicke O., Klamt S. (2015). Manipulation of the ATP pool as a tool for metabolic engineering. *Biochemical Society Transactions*, **43**, 6, 1140–1145, doi: 10.1042/BST20150141.

[131] Liu J., Kandasamy V., Würtz A., Jensen P. R., Solem C. (2016). Stimulation of acetoin production in metabolically engineered *Lactococcus lactis* by increasing ATP demand. *Applied Microbiology and Biotechnology*, **100**, 22, 9509–9517, doi: 10.1007/s00253-016-7687-1.

[132] Semkiv M. V., Dmytruk K. V., Abbas C. A., Sibirny A. A. (2016). Activation of futile cycles as an approach to increase ethanol yield during glucose fermentation in *Saccharomyces cerevisiae*. *Bioengineered*, **7**, 2, 106–111, doi: 10.1080/21655979.2016.1148223.

[133] Boecker S., Zahoor A., Schramm T., Link H., Klamt S. (2019). Broadening the Scope of Enforced ATP Wasting as a Tool for Metabolic Engineering in *Escherichia coli*. *Biotechnology Journal*, **14**, 9, 1800438, doi: 10.1002/biot.201800438.

[134] Zahoor A., Messerschmidt K., Boecker S., Klamt S. (2020). ATPase-based implementation of enforced ATP wasting in *Saccharomyces cerevisiae* for improved ethanol production. *Biotechnology for Biofuels*, **13**, 1, 185, doi: 10.1186/s13068-020-01822-9.

[135] Koebmann B. J., Westerhoff H. V., Snoep J. L., Nilsson D., Jensen P. R. (2002). The glycolytic flux in *Escherichia coli* is controlled by the demand for ATP. *Journal of Bacteriology*, **184**, 14, 3909–3916, doi: 10.1128/jb.184.14.3909-3916.2002.

[136] Klamt S., Mahadevan R., Hädicke O. (2018). When Do Two-Stage Processes Outperform One-Stage Processes? *Biotechnology Journal*, **13**, 2, 1700539, doi: 10.1002/biot.201700539.

[137] Jouhten P., Huerta-Cepas J., Bork P., Patil K. R. (2017). Metabolic anchor reactions for robust biorefining. *Metabolic Engineering*, **40**, 1–4, doi: 10.1016/j.ymben.2017.02.010.

[138] Abbad-Andaloussi S., Amine J., Gerard P., Petitdemange H. (1998). Effect of glucose on glycerol metabolism by *Clostridium butyricum* DSM 5431. *Journal of Applied Microbiology*, **84**, 4, 515–522, doi: 10.1046/j.1365-2672.1998.00374.x.

[139] Simeonidis E., Price N. D. (2015). Genome-scale modeling for metabolic engineering. *Journal of Industrial Microbiology & Biotechnology*, **42**, 3, 327–338, doi: 10.1007/s10295-014-1576-3.

[140] Yao R., Liu Q., Hu H., Wood T. K., Zhang X. (2015). Metabolic engineering of *Escherichia coli* to enhance acetol production from glycerol. *Applied Microbiology and Biotechnology*, **99**, 19, 7945–7952, doi: 10.1007/s00253-015-6732-9.

[141] Lloyd C. J., King Z. A., Sandberg T. E., Hefner Y., Olson C. A., Phaneuf P. V., O'Brien E. J., Sanders J. G., Salido R. A., Sanders K., Brennan C., Humphrey G., Knight R., Feist A. M. (2019). The genetic basis for adaptation of model-designed syntrophic co-cultures. *PLOS Computational Biology*, **15**, 3, e1006213, doi: 10.1371/journal.pcbi.1006213.

[142] Aslan S., Noor E., Benito Vaquerizo S., Lindner S. N., Bar-Even A. (2020). Design and engineering of *E. coli* metabolic sensor strains with a wide sensitivity range for glycerate. *Metabolic Engineering*, **57**, 96–109, doi: 10.1016/j.ymben.2019.09.002.

[143] Babel W. (2009). The Auxiliary Substrate Concept: From simple considerations to heuristically valuable knowledge. *Engineering in Life Sciences*, **9**, 4, 285–290, doi: 10.1002/elsc.200900027.

[144] Park J. O., Liu N., Holinski K. M., Emerson D. F., Qiao K., Woolston B. M., Xu J., Lazar Z., Islam M. A., Vidoudez C., Girguis P. R., Stephanopoulos G. (2019). Synergistic substrate cofeeding stimulates reductive metabolism. *Nature Metabolism*, **1**, 6, 643–651, doi: 10.1038/s42255-019-0077-0.

[145] Liu N., Santala S., Stephanopoulos G. (2019). Mixed carbon substrates: a necessary nuisance or a missed opportunity? *Current Opinion in Biotechnology*, **62**, 15–21, doi: 10.1016/j.copbio.2019.07.003.

[146] Lun D. S., Rockwell G., Guido N. J., Baym M., Kelner J. A., Berger B., Galagan J. E., Church G. M. (2009). Large-scale identification of genetic design strategies using local search. *Molecular Systems Biology*, **5**, 1, 296, doi: 10.1038/msb.2009.57.

[147] Costanza J., Carapezza G., Angione C., Lió P., Nicosia G. (2012). Robust design of microbial strains. *Bioinformatics*, **28**, 23, 3097–3104, doi: 10.1093/bioinformatics/bts590.

[148] Gagneur J., Klamt S. (2004). Computation of elementary modes: a unifying framework and the new binary approach. *BMC Bioinformatics*, **5**, 1, 175, doi: 10.1186/1471-2105-5-175.

[149] Chindelevitch L., Trigg J., Regev A., Berger B. (2014). An exact arithmetic toolbox for a consistent and reproducible structural analysis of metabolic network models. *Nature Communications*, **5**, 1, 1–9, doi: 10.1038/ncomms5893.

[150] Ji X.-J., Huang H., Ouyang P.-K. (2011). Microbial 2,3-butanediol production: A state-of-the-art review. *Biotechnology Advances*, **29**, 3, 351–364, doi: 10.1016/j.biotechadv.2011.01.007.

[151] Yang T., Rao Z., Zhang X., Xu M., Xu Z., Yang S.-T. (2017). Metabolic engineering strategies for acetoin and 2,3-butanediol production: advances and prospects. *Critical Reviews in Biotechnology*, **37**, 8, 990–1005, doi: 10.1080/07388551.2017.1299680.

[152] Yang Z., Zhang Z. (2019). Recent advances on production of 2,3-butanediol using engineered microbes. *Biotechnology Advances*, **37**, 4, 569–578, doi: 10.1016/j.biotechadv.2018.03.019.

[153] Erian A. M., Gibisch M., Pflügl S. (2018). Engineered *E. coli* W enables efficient 2,3-butanediol production from glucose and sugar beet molasses using defined minimal medium as economic basis. *Microbial Cell Factories*, **17**, 1, 190, doi: 10.1186/s12934-018-1038-0.

[154] Nielsen D. R., Yoon S.-H., Yuan C. J., Prather K. L. J. (2010). Metabolic engineering of acetoin and meso-2,3-butanediol biosynthesis in *E. coli*. *Biotechnology Journal*, **5**, 3, 274–284, doi: 10.1002/biot.200900279.

[155] Lu H., Li F., Sánchez B. J., Zhu Z., Li G., Domenzain I., Marcišauskas S., Anton P. M., Lappa D., Lieven C., Beber M. E., Sonnenschein N., Kerkhoven E. J., Nielsen J. (2019). A consensus S. cerevisiae metabolic model Yeast8 and its ecosystem for comprehensively probing cellular metabolism. *Nature Communications*, **10**, 1, 3586, doi: 10.1038/s41467-019-11581-3.

[156] Nogales J., Palsson B. Ø., Thiele I. (2008). A genome-scale metabolic reconstruction of *Pseudomonas putida* KT2440: iJN746 as a cell factory. *BMC Systems Biology*, **2**, 1, 79, doi: 10.1186/1752-0509-2-79.

[157] Nakashima N., Akita H., Hoshino T. (2014). Establishment of a novel gene expression method, BICES (biomass-inducible chromosome-based expression system), and its application to the production of 2,3-butanediol and acetoin. *Metabolic Engineering*, **25**, 204–214, doi: 10.1016/j.ymben.2014.07.011.

[158] Wang L., Dash S., Ng C. Y., Maranas C. D. (2017). A review of computational tools for design and reconstruction of metabolic pathways. *Synthetic and Systems Biotechnology*, **2**, 4, 243–252, doi: 10.1016/j.synbio.2017.11.002.

[159] Hartmann A., Vila-Santa A., Kallscheuer N., Vogt M., Julien-Laferrière A., Sagot M.-F., Marienhagen J., Vinga S. (2017). OptPipe - a pipeline for optimizing metabolic engineering targets. *BMC Systems Biology*, **11**, 143, doi: 10.1186/s12918-017-0515-0.

[160] Zhuang K., Yang L., Cluett W. R., Mahadevan R. (2013). Dynamic strain scanning optimization: an efficient strain design strategy for balanced yield, titer, and productivity. DySScO strategy for strain design. *BMC Biotechnology*, **13**, 1, 8, doi: 10.1186/1472-6750-13-8.

[161] Conrad T. M., Lewis N. E., Palsson B. Ø. (2011). Microbial laboratory evolution in the era of genome-scale science. *Molecular Systems Biology*, **7**, 509, doi: 10.1038/msb.2011.42.

[162] Heerden C. D. van, Nicol W. (2013). Continuous and batch cultures of *Escherichia coli* KJ134 for succinic acid fermentation: metabolic flux distributions and production characteristics. *Microbial Cell Factories*, **12**, 80, doi: 10.1186/1475-2859-12-80.

[163] Pos K. M., Dimroth P., Bott M. (1998). The *Escherichia coli* citrate carrier *CitT*: a member of a novel eubacterial transporter family related to the 2-oxoglutarate/malate translocator from spinach chloroplasts. *Journal of Bacteriology*, **180**, 16, 4160–4165.

[164] Engel P., Krämer R., Unden G. (1992). Anaerobic fumarate transport in *Escherichia coli* by an *fnr*-dependent dicarboxylate uptake system which is different from the aerobic dicarboxylate uptake system. *Journal of Bacteriology*, **174**, 17, 5533–5539.

[165] Song C. W., Kim D. I., Choi S., Jang J. W., Lee S. Y. (2013). Metabolic engineering of *Escherichia coli* for the production of fumaric acid. *Biotechnology and Bioengineering*, **110**, 7, 2025–2034, doi: 10.1002/bit.24868.

[166] Carbonell P., Parutto P., Herisson J., Pandit S. B., Faulon J.-L. (2014). XTMS: pathway design in an eXTended metabolic space. *Nucleic Acids Research*, **42**, W389–W394, doi: 10.1093/nar/gku362.

[167] Kuwahara H., Alazmi M., Cui X., Gao X. (2016). MRE: a web tool to suggest foreign enzymes for the biosynthesis pathway design with competing endogenous reactions in mind. *Nucleic Acids Research*, **44**, W217–W225, doi: 10.1093/nar/gkw342.

[168] Noor E., Bar-Even A., Flamholz A., Reznik E., Liebermeister W., Milo R. (2014). Pathway Thermodynamics Highlights Kinetic Obstacles in Central Metabolism. *PLOS Computational Biology*, **10**, 2, e1003483, doi: 10.1371/journal.pcbi.1003483.

[169] Asplund-Samuelsson J., Janasch M., Hudson E. P. (2018). Thermodynamic analysis of computed pathways integrated into the metabolic networks of *E. coli* and *Synechocystis* reveals contrasting expansion potential. *Metabolic Engineering*, **45**, 223–236, doi: 10.1016/j.ymben.2017.12.011.

[170] Hädicke O., Kamp A. von, Aydogan T., Klamt S. (2018). OptMDFpathway: Identification of metabolic pathways with maximal thermodynamic driving force and its application for analyzing the endogenous CO2 fixation potential of *Escherichia coli*. *PLOS Computational Biology*, **14**, 9, e1006492, doi: 10.1371/journal.pcbi.1006492.

[171] Huang J.-F., Liu Z.-Q., Jin L.-Q., Tang X.-L., Shen Z.-Y., Yin H.-H., Zheng Y.-G. (2017). Metabolic engineering of *Escherichia coli* for microbial production of L-methionine. *Biotechnology and Bioengineering*, **114**, 4, 843–851, doi: 10.1002/bit.26198.

[172] Willke T. (2014). Methionine production—a critical review. *Applied Microbiology and Biotechnology*, **98**, 24, 9893–9914, doi: 10.1007/s00253-014-6156-y.

[173] Figge R. M. (2006). Methionine Biosynthesis in *Escherichia coli* and *Corynebacterium glutamicum*, 163–193, ISBN: 978-3-540-48595-7 978-3-540-48596-4, doi: 10.1007/7171_2006_059.

[174] Liu Q., Liang Y., Zhang Y., Shang X., Liu S., Wen J., Wen T. (2015). *YjeH* Is a Novel Exporter of l-Methionine and Branched-Chain Amino Acids in *Escherichia coli*. *Applied and Environmental Microbiology*, **81**, 22, 7753–7766, doi: 10.1128/AEM.02242-15.

[175] Facchetti G. (2016). A simple strategy guides the complex metabolic regulation in *Escherichia coli*. *Scientific Reports*, **6**, 27660, doi: 10.1038/srep27660.

[176] Foster J. M., Davis P. J., Raverdy S., Sibley M. H., Raleigh E. A., Kumar S., Carlow C. K. S. (2010). Evolution of Bacterial Phosphoglycerate Mutases: Non-Homologous Isofunctional Enzymes Undergoing Gene Losses, Gains and Lateral Transfers. *PLOS ONE*, **5**, 10, e13576, doi: 10.1371/journal.pone.0013576.

[177] Hernández-Montalvo V., Martínez A., Hernández-Chavez G., Bolivar F., Valle F., Gosset G. (2003). Expression of *galP* and *glk* in a *Escherichia coli* PTS mutant restores glucose transport and increases glycolytic flux to fermentation products. *Biotechnology and Bioengineering*, **83**, 6, 687–694, doi: 10.1002/bit.10702.

[178] Hwang H. J., Park J. H., Kim J. H., Kong M. K., Kim J. W., Park J. W., Cho K. M., Lee P. C. (2014). Engineering of a butyraldehyde dehydrogenase of *Clostridium saccharoperbutylacetonicum* to fit an engineered 1,4-butanediol pathway in *Escherichia coli*: Engineering of BLD for enhancement of 1,4-BDO. *Biotechnology and Bioengineering*, **111**, 7, 1374–1384, doi: 10.1002/bit.25196.

[179] Andreozzi S., Chakrabarti A., Soh K. C., Burgard A., Yang T. H., Van Dien S., Miskovic L., Hatzimanikatis V. (2016). Identification of metabolic engineering targets for the enhancement of 1,4-butanediol production in recombinant *E. coli* using large-scale kinetic models. *Metabolic Engineering*, **35**, 148–159, doi: 10.1016/j.ymben.2016.01.009.

[180] Segrè D., Vitkup D., Church G. M. (2002). Analysis of optimality in natural and perturbed metabolic networks. *Proceedings of the National Academy of Sciences of the United States of America*, **99**, 23, 15112–15117, doi: 10.1073/pnas.232349399.

[181] Belotti P., Bonami P., Fischetti M., Lodi A., Monaci M., Nogales-Gómez A., Salvagnin D. (2016). On handling indicator constraints in mixed integer programming. *Computational Optimization and Applications*, **65**, 3, 545–566, doi: 10.1007/s10589-016-9847-8.

[182] Kaibel V., Pfetsch M. E. (2003). Some Algorithmic Problems in Polytope Theory, 23–47, ISBN: 978-3-642-05539-3 978-3-662-05148-1, doi: 10.1007/978-3-662-05148-1_2.

[183] Bekiaris P. S., Klamt S. (2021). Designing microbial communities to maximize the thermodynamic driving force for the production of chemicals. *PLOS Computational Biology*, **17**, 6, e1009093, doi: 10.1371/journal.pcbi.1009093.