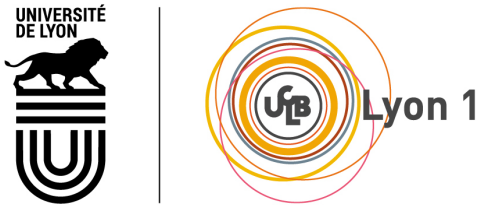
Models and algorithms for investigating and exploiting the metabolism of microorganisms

Irene Ziska



N° d'ordre NNT : xxx

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON

opérée au sein de

l'Université Claude Bernard Lyon 1

École Doctorale ED 341
E2M2

Spécialité de doctorat : Bioinformatique

Soutenue publiquement le 24/11/2020, par :

Irene Ziska

Models and algorithms for investigating and exploiting the metabolism of microorganisms

---

Devant le jury composé de :

Brochier-Armanet Céline, Professeur, Université de Lyon 1	Examinatrice
Förster Jochen, Professeur, Carlsberg Group	Rapporteur
Jourdan Fabien, DR, INRAE Toulouse	Rapporteur
Thébault Patricia, Maître de Conférences, Université de Bordeaux	Rapportrice
Cottret Ludovic, IR, INRAE Toulouse	Examinateur
Sagot Marie-France, DR, INRIA	Directrice de thèse
Vinga Susana, Associate Professor, Instituto Superior Técnico	Co-directrice de thèse
Mary Arnaud, Maître de Conférences, Université de Lyon 1	Co-encadrant

UNIVERSITE CLAUDE BERNARD-LYON 1

Administrateur provisoire de l'Université	M. Frédéric FLEURY
Président du Conseil Académique	M. Hamda BEN HADID
Vice-Président du Conseil d'Administration	M. Didier REVEL
Vice-Président du Conseil des Etudes et de la Vie Universitaire	M. Philippe CHEVALLIER
Vice-Président de la Commission de Recherche	M. Jean-François MORNEX
Directeur Général des Services	M. Pierre ROLLAND

COMPOSANTES SANTE

Département de Formation et Centre de Rechercheen Biologie Humaine	Directrice: Mme Anne-Marie SCHOTT
Faculté d'Odontologie	Doyenne: Mme Dominique SEUX
Faculté de Médecine et Maïeutique Lyon Sud -Charles Mérieux	Doyenne: Mme Carole BURILLON
Faculté de Médecine Lyon-Est	Doyen: M. Gilles RODE
Institut des Sciences et Techniques de la Réadaptation (ISTR)	Directeur: M. Xavier PERROT
Institut des Sciences Pharmaceutiques et Biologiques (ISBP)	Directrice: Mme Christine VINCIGUERRA

COMPOSANTES & DEPARTEMENTS DE SCIENCES & TECHNOLOGIE

Département Génie Electrique et des Procédés (GEP)	Directrice: Mme Rosaria FERRIGNO
Département Informatique	Directeur: M. Behzad SHARIAT
Département Mécanique	Directeur M. Marc BUFFAT
Ecole Supérieure de Chimie, Physique, Electronique (CPE Lyon)	Directeur: Gérard PIGNAULT
Institut de Science Financière et d'Assurances (ISFA)	Directeur: M. Nicolas LEBOISNE
Institut National du Professorat et de l'Education	Administrateur Provisoire: M. Pierre CHAREYRON
Institut Universitaire de Technologie de Lyon 1	Directeur: M. Christophe VITON
Observatoire de Lyon	Directrice: Mme Isabelle DANIEL
Polytechnique Lyon	Directeur: Emmanuel PERRIN
UFR Biosciences	Administratrice provisoire: Mme Kathrin GIESELER
UFR des Sciences et Techniques des Activités Physiques et Sportives (STAPS)	Directeur: M. Yannick VANPOULLE
UFR Faculté des Sciences	Directeur: M. Bruno ANDRIOLETTI

Acknowledgments

I would like to thank my advisors Marie-France Sagot and Susana Vinga and all other members of the Erable team; in particular Arnaud Mary for our discussions and my co-authors Ricardo Andrade and Mariana Ferrarini; special thanks to my colleagues Carol Moraga and Marianne Borderes who always provided support and comfort; my family for their support and for not asking too many questions.

Resumé en français

Cette thèse de doctorat porte sur l'analyse du métabolisme des micro-organismes. Le métabolisme est l'ensemble des réactions d'un organisme. Il peut être modélisé comme un réseau métabolique qui contient les métabolites présents dans un organisme et les réactions correspondantes qui les transforment. Les réseaux métaboliques peuvent être, par exemple, représentés sous forme de hypergraphes ou de matrices stœchiométriques. La modélisation du métabolisme peut être utilisée pour obtenir des informations sur l'activité métabolique d'un micro-organisme et pour prédire certains comportements.

La disponibilité croissante des données métabolomiques et leur analyse améliorent la compréhension des mécanismes cellulaires. L'intégration des données métabolomiques dans les réseaux métaboliques permet de comprendre comment les systèmes biologiques réagissent à différentes perturbations. Cela peut être, par exemple, l'adaptation du micro-organisme à un changement de milieu, au stress ou l'impact des perturbations génétiques sur l'activité métabolique du micro-organisme. La modification de l'activité métabolique d'un micro-organisme est appelée changement métabolique (en anglais "metabolic shift"). Comme il est important de comprendre les changements métaboliques dans différentes conditions pour faire progresser la recherche dans différents domaines comme la bio-ingénierie ou la santé humaine, il est nécessaire de disposer de méthodes de calcul qui facilitent la compréhension des données métabolomiques disponibles.

Un autre sujet commun dans l'analyse des réseaux métaboliques est le calcul des stratégies d'intervention optimales. Les micro-organismes sont déjà utilisés à l'échelle industrielle pour produire des substances chimiques cibles importantes. Il est essentiel de comprendre comment les différentes parties du métabolisme interagissent et sont liées entre elles pour trouver des moyens d'améliorer les taux de production. Le métabolite souhaité peut souvent n'être qu'un sous-produit du métabolisme habituel lorsque le micro-organisme est en croissance et son rendement peut être très faible. Il est donc nécessaire de modifier le métabolisme du micro-organisme afin d'augmenter le rendement du composé cible souhaité et établir ainsi une production plus efficace. L'un des domaines de la biologie synthétique est le génie génétique des micro-organismes pour améliorer la production de substances chimiques cibles importantes par un micro-organisme. L'idée est d'éliminer des réactions spécifiques en manipulant les gènes métaboliques qui codent pour les enzymes catalytiques. En fonction du rôle que les réactions d'élimination jouent dans le métabolisme, l'organisme peut utiliser différentes voies pour compenser son manque. Les réactions éliminées doivent être choisies de manière que le métabolisme modifié entraîne une augmentation de la production de la substance chimique cible.

En raison de la structure complexe du métabolisme, il est nécessaire de modéliser les perturbations *in silico* afin de proposer des stratégies d'intervention qui pourraient conduire au meilleur résultat *in vivo*. En prédisant et en analysant le comportement modifié et en identifiant les meilleures perturbations à l'aide d'approches computationnelles, la conception d'expériences pratiques peut être guidée.

De plus en plus d'approches sont mises au point pour calculer des knock-outs optimaux pour des scénarios différents. Certaines ne sont applicables qu'à des modèles de réseaux plus petits qui ne représentent qu'une version condensée ou que certaines parties du métabolisme. L'application aux réseaux métaboliques à l'échelle du génome qui modélisent le métabolisme complet d'un micro-organisme de manière aussi détaillée que possible reste particulièrement difficile. En raison de leur taille et de leur complexité, l'effort de calcul des méthodes appliquées peut augmenter considérablement.

Il y a cependant un aspect qui n'est souvent pas pris en compte lors du calcul des knock-outs

optimaux qui devraient augmenter la production cible. Dans certains cas, le produit chimique cible est un sous-produit qui est en fait toxique pour le micro-organisme. Par conséquent, l'accumulation du métabolite cible peut inhiber la croissance du micro-organisme. Elle peut également diminuer le taux de production, ce qui rendra l'application à l'échelle industrielle moins efficace.

Les knock-outs restreignent généralement le métabolisme d'un micro-organisme parce qu'ils enlèvent certaines fonctionnalités. Par conséquent, dans le cadre mentionné ci-dessus, il est crucial de s'assurer que les knock-outs effectués n'inhibent pas des processus importants que le micro-organisme peut utiliser pour établir une tolérance contre la cible toxique. Cela ne contribuera pas spécifiquement à augmenter la résistance contre le produit chimique cible, mais cela permet de s'assurer que la tolérance naturelle est préservée.

J'ai déjà travaillé sur les réseaux métaboliques pendant mon mastère, ce qui a également été l'une des raisons pour lesquelles je me suis intéressée à ce doctorat. J'ai travaillé sur deux approches pour les réseaux métaboliques qui utilisent l'optimisation multi-objectifs. Une approche était applicable aux communautés microbiennes, c'est-à-dire pour un scénario où différents micro-organismes interagissent les uns avec les autres. La seconde utilisait l'optimisation multi-objectifs pour identifier les compromis possibles entre la production de biomasse et la production cible dans les micro-organismes. Des knock-outs sont énumérées consécutivement. Cette deuxième approche a donné une première idée de base pour l'approche développée qui sera la partie principale de cette thèse. Elle sera présentée dans le chapitre 2.

Au début de cette thèse, dans le chapitre 1, je ferai un bref résumé sur certains sujets qui sont importants pour la compréhension des deux approches qui seront présentées dans les chapitres 2 et 3. Je ferai d'abord une brève introduction à la programmation linéaire, en particulier à la programmation linéaire multi-objectifs qui sera utilisée dans l'approche décrite au chapitre 2. Ensuite, je donnerai une introduction aux réseaux métaboliques, à leurs représentations et aux méthodes associées. Dans les deux dernières sections, je résumerai les approches communes pour prévoir les stratégies d'intervention et analyser les changements métaboliques.

Le chapitre 2 représente le principal travail que j'ai effectué pendant mon doctorat. Cette partie se concentre sur une approche d'*exploitation* du métabolisme des micro-organismes. La motivation principale était de développer une stratégie pour prédire les knock-outs optimaux qui augmentent la production d'une substance chimique cible dans un micro-organisme dans le cas où le métabolite cible produit est toxique pour le micro-organisme utilisé. En outre, je souhaitais une approche qui soit également applicable aux réseaux métaboliques à l'échelle du génome.

L'approche développée se compose de deux parties différentes. Dans la première partie, un problème d'optimisation multi-objectifs est formulé qui calcule les compromis entre la production de biomasse, la production cible et un score qui mesure la résistance possible à la toxicité contre le métabolite cible toxique. Dans la deuxième partie, on énumère des knock-outs qui devraient imposer des distributions de flux spécifiques. Dans un premier temps, un MILP a été proposé pour énumérer les différents knock-outs. En raison de ses limites et de ses défauts, il était nécessaire de développer une deuxième idée qui est basée sur l'identification et l'isolation de sous-réseaux.

La méthode proposée est appliquée à l'étude de cas de la production d'éthanol dans la levure. L'éthanol est déjà produit par la levure dans l'industrie et il y a un intérêt croissant pour l'éthanol en raison de son utilisation comme biocarburant. Cependant, un facteur limitant important pour la production d'éthanol dans la levure est en effet la toxicité de l'éthanol pour la levure. Lorsque l'éthanol s'accumule, il inhibe la croissance mais entraîne également un déclin de la production d'éthanol.

Il pourrait donc être important de tenir compte de cet aspect lorsque l'on essaie d'améliorer la production d'éthanol en introduisant des knock-outs qui entraîneront un changement dans l'activité métabolique de la levure. En appliquant notre approche à l'étude de cas de la production d'éthanol dans la levure, nous avons pu calculer des ensembles de désactivation avec moins de 20 réactions. Il reste à déterminer quelle est la plus petite taille possible. En outre, nous avons besoin d'une évaluation biologique plus poussée pour les ensembles de désactivation proposés afin d'établir la viabilité des réactions suggérées dans la pratique.
Les avantages de notre approche sont qu'elle est applicable sur des réseaux métaboliques à l'échelle du génome comme nous l'avons montré en utilisant le modèle de la levure 5.01. En outre, le cadre est flexible et il devrait être possible de modifier les objectifs pour différents scénarios. Nous sommes donc également intéressés par l'application de notre approche à d'autres exemples afin de confirmer son adaptabilité. Nous prévoyons de peaufiner l'approche et de la rendre disponible comme outil sur le Gitlab de l'équipe. Après avoir obtenu un avis biologique supplémentaire des collaborateurs sur nos résultats, nous voulons soumettre ce travail sous forme d'article avant la fin de l'année.
Dans le chapitre 3, je présenterai une nouvelle méthode de calcul qui s'est concentrée sur *étudier* l'activité métabolique des micro-organismes. Cette approche est appelée TOTORO, ce qui est l'abréviation de "Transient respOnse to meTabOlic pertuRbation inferred at the whole netwOrk level". Cette approche est le résultat d'une collaboration avec Ricardo Andrade et Mariana Ferrarini. Un ancien membre du groupe, Alice Julien-Laferrière, a déjà travaillé sur cette approche lors de son doctorat. Nous avons soumis un papier sur ce travail à *Bioinformatique* en Septembre 2020.
TOTORO intègre les concentrations internes de métabolites qui ont été mesurées avant et après une perturbation dans des reconstructions métaboliques à l'échelle du génome. Il prédit les réactions qui étaient actives pendant l'état transitoire qui a suivi la perturbation. Il s'agit d'une approche basée sur les contraintes qui prend en compte la stœchiométrie du réseau. Elle minimise le changement des concentrations pour les métabolites non mesurés et également le nombre de réactions actives pendant l'état transitoire pour tenir compte d'une hypothèse parcimonieuse. TOTORO est un outil disponible librement. Il a été implémenté en C++ et peut être consulté à l'adresse `https://gitlab.inria.fr/erable/totoro`.
Nous avons appliqué notre méthode à trois expériences d'impulsions dans *Escherichia coli* pour montrer qu'elle peut récupérer des voies actives connectées et prédire des directions distinctes pour des réactions réversibles qui sont conformes aux observations biologiques connues. Nous avons utilisé un modèle de base et un modèle à l'échelle du génome de *E. coli* pour montrer que notre approche est également applicable à des modèles de réseaux d'une taille plus grande.
Un autre projet sur lequel j'ai commencé à travailler pendant mon doctorat est une collaboration avec Marianne Borderes. Nous développons une approche visant à combiner les résultats de plusieurs méthodes de regroupement appliquées aux données de la métagénomique. Ce projet ne sera pas présenté dans cette thèse car il est toujours en cours et, de plus, il n'est pas directement lié à l'analyse des réseaux métaboliques.

Contents

Scope of the thesis

This PhD thesis is about the analysis of the metabolism of microorganisms. The metabolism is the set of reactions of an organism. It can be modeled as a metabolic network which contains the metabolites that are present in an organism and the corresponding reactions that transform them. Metabolic networks can be, for instance, represented as a hypergraph or a stoichiometric matrix. Modeling the metabolism can be used to gain insights on the metabolic activity of a microorganism and to predict certain behaviors,

The increasing availability of metabolomic data and their analysis are improving the understanding of cellular mechanisms. Integrating metabolomic data into metabolic networks makes it possible to understand how biological systems respond to different perturbations. This can be, for example, the adaption of the microorganism to a change in the medium or to stress, or else the impact of genetic perturbations on its metabolic activity. The change in the metabolic activity of a microorganism is called metabolic shift. Since understanding metabolic changes under different conditions is important for advancing research in different fields like bioengineering or human health, there is a need for computational methods that facilitate the comprehension of available metabolomic data.

Another common topic in the analysis of metabolic networks is the computation of optimal intervention strategies. Microorganisms are already used on an industrial scale to produce important target chemicals. Understanding how different parts of the metabolism interact and are linked together is crucial to find ways that will improve production rates. The desired compound might often just be a by-product of the usual metabolism when the microorganism is growing and its yield can be very low. Thus, it is necessary to modify the metabolism of the microorganism to increase the yield of the desired target compound and to establish a more efficient production. One of the areas of synthetic biology is the genetic engineering of microorganisms to improve the production of important target chemicals by a microorganism. The idea is to knockout specific reactions by manipulating the metabolic genes that code for the catalyzing enzymes. Depending on the role the knocked out reactions play in the metabolism, the organism might use different pathways to compensate for its lack. The reactions that are knocked out must be chosen in a way that the altered metabolism leads to an increase in the production of the target chemical.

Due to the complex structure of metabolisms, there is a need to model perturbations *in silico* in order to propose intervention strategies that might lead to the best outcome *in vivo*. By predicting and analyzing the altered behavior and identifying optimal knockouts with computational approaches, the design of practical experiments can be supported and guided.

More and more approaches are developed that aim to compute optimal knockouts for different scenarios. Some are only applicable to smaller network models which are only representing a condensed version or certain parts of the metabolism. Especially the application to genome-scale metabolic networks which model the whole metabolism of a microorganism as detailed as possible remains challenging. Due to their size and complexity, the computational effort of the applied methods can increase significantly.

When computing optimal knockouts that should increase the target production, there is however one aspect that is often not taken into account. In some cases, the desired target chemical is a by-product that is actually toxic for the microorganism and, therefore, its accumulation can severely inhibit the growth of the microorganism. It can also decrease the production rate which will make the application on a industrial scale less efficient.

Knockouts usually restrict the metabolism of a microorganism because they are taking away certain functionalities. Hence, in the above-mentioned setting, it is crucial to ensure that the inserted knockouts do not inhibit important processes that a microorganism can use to establish a tolerance against the toxic target. This will not specifically help to increase the resistance against the target chemical but it ensures that the natural tolerance is preserved.

I already worked on metabolic networks during my master's degree which was also one of the reasons why I got interested in this PhD. I worked on two approaches for metabolic networks that use multi-objective optimization. One approach was applicable to microbial communities, i.e., for a scenario where different microorganisms interact with each other. The second one was using multi-objective optimization to identify possible tradeoffs between biomass production and target production in microorganisms. Knockouts for each tradeoff were enumerated afterwards. This second approach gave a first, basic idea for the developed approach that will be the main part of this thesis. It will be presented in Chapter 2.

In the beginning of this thesis, in Chapter 1, I will give a brief summary on certain topics that are important for the understanding of the two approaches that will be presented in the Chapters 2 and 3. I will first do a short introduction to linear programming, especially multi-objective linear programming which will be used in the approach described in Chapter 2. Afterwards, I will give an introduction to metabolic networks, their representations and associated methods. In the last two sections, I will summarize common approaches to predict intervention strategies and to analyze metabolic shifts.

Chapter 2 represents the main work that I did during my PhD. This part focuses on an approach to *exploit* the metabolism of microorganisms. The main motivation was to developed a strategy to predict optimal knockouts that increase the production of a target chemical in a microorganism in the scenario where the produced target metabolite is toxic for the utilized microorganism. Furthermore, I aimed for an approach that is also applicable to genome-scale metabolic networks.

The developed approach consists of two different parts. In the first part, a multi-objective optimization problem is formulated that computes tradeoffs between biomass production, target production and a score that measures the possible toxicity resistance against the toxic target metabolite. In the second part, knockouts are enumerated that should enforce specific flux distributions. As a first idea, a mixed-integer linear program was proposed to enumerate different knockouts. Due to its limitations and flaws, there was a need to develop a second idea which is based on identifying and cutting off subnetworks.

The proposed method is applied to the case-study of ethanol production in yeast. Ethanol is already produced by yeast in industry and there is a growing interest in ethanol due to its use as bio-fuel. However, an important limiting factor for the production of ethanol in yeast is indeed the toxicity of ethanol for yeast. When ethanol accumulates, it inhibits growth but leads also to a decline of the ethanol production. It might thus be important to consider this aspect when trying to improve the ethanol output by introducing knockouts that will lead to a change in the metabolic activity of yeast.

Based on this case-study, I will describe insights that we could gain from both parts of the approach, especially on the identified subnetworks. We plan on submitting a paper on this subject this year after obtaining some more biological evaluation of the obtained results. Furthermore, the developed approach will be made available as tool on the Gitlab of our group.

In Chapter 3, I will present a novel computational method that focused on *investigating* the metabolic activity of microorganisms. The approach is called TOTORO which is short for " *Transient respOnse*

to meTabOlic pertuRbation inferred at the whole netwOrk level". This approach is the result of a collaboration with Ricardo Andrade and Mariana Ferrarini. A former member of the group, Alice Julien-Laferrière, already worked on this approach during her PhD. We submitted a paper on this work to *Bioinformatics* in September 2020.

TOTORO integrates internal metabolite concentrations that were measured before and after a perturbation into genome-scale metabolic reconstructions. It predicts reactions that were active during the transient state that occurred after the perturbation. It is a constraint-based approach that takes the stoichiometry of the network into account. It minimizes the change in concentrations for unmeasured metabolites and also the number of active reactions during the transient state to account for a parsimonious assumption. TOTORO is a freely available tool implemented in C++ and it can be accessed at `https://gitlab.inria.fr/erable/totoro`.

We applied our method to three published pulse experiments in *Escherichia coli* to show that it can retrieve connected active pathways and predict distinct directions for reversible reactions that are in accordance with known biological observations. We used a core model and a genome-scale model of *E. coli* to further demonstrate that our approach is also applicable to larger network models.

Another project that I started working on during my PhD is a collaboration with Marianne Borderes. We are developing an approach to combine the results of multiple clustering methods applied to metagenomics data. This project will not be presented in this thesis because it is still ongoing and additionally, it is not directly connected to the analysis of metabolic networks.

1 Theoretical background

Contents

In this chapter, I will first present a short introduction to linear programming, especially multi-objective linear programming which will be used in the approach described in Chapter 2. Afterwards, I will give an introduction to metabolic networks, their representations and associated methods. In the last two sections, I will summarize common approaches to predict intervention strategies and to analyze metabolic shifts.

1.1 Linear programming

As described in (Dantzig and Thapa, 2006), linear programming maximizes or minimizes a linear objective function for the variables that are subject to linear constraints. Introductions to linear programming and its applications are, for example, given in (Bertsimas and Tsitsiklis, 1997; Dantzig, 1998; Schrijver, 1998; Gass, 2003). Linear programming is part of the larger field of *mathematical programming* that includes amongst others integer programming and nonlinear programming.

1.1.1 Problem formulation

Mathematically, a linear program finds the values for the problem variables x that maximize an objective function z (Equation 1.1). The problem variables are subject to m linear constraints (Equation 1.2).

$$\max_{x} \quad z = f(x) = c^T x \tag{1.1}$$

$$\text{s.t.} \quad Ax \leq b \tag{1.2}$$

$$A \in \mathbb{R}^{m,n}, x \in \mathbb{R}^n, b \in \mathbb{R}^m, c \in \mathbb{R}^n \tag{1.3}$$

A vector $x' \in \mathbb{R}^n$ is a *feasible solution* if it does not violate any of the given constraints. The *feasible region* is the set of all feasible solutions. Geometrically, the feasible region is described by a polyhedron. A solution x^* is *optimal* if it it a feasible solution and $f(x^*) = \max\{f(x) | Ax \leq b, x \in \mathbb{R}^n\}$. Linear programs can be *infeasible* which means that no solution exists that satisfies all constraints. Furthermore, it can have several optimal solutions which means that there are multiple feasible solutions that lead to the same optimal value for the objective function. The linear program is *unbounded* if the value for the objective function can be increased (or decreased) without any limit.
If all variables are integer, the resulting problem is called *integer linear program* (ILP). If some are integer and some are continuous, it is a *mixed integer linear program* (MILP).
Solvers for linear programs are, for example, IBM ILOG CPLEX (IBM, 2019), Gurobi Optimization (Gurobi Optimization, 2020) or SCIP (Achterberg et al., 2008; Achterberg, 2009).

1.1.2 Multi-objective linear programming

In contrast to a linear program which optimizes one objective function, a *multi-objective* linear program minimizes or maximizes k objective functions at the same time (Equation 1.4). The different objective functions can be contradictory. Multi-objective programs are, for instance, described in

(Ehrgott, 2005) and (Antunes et al., 2016).

$$\begin{aligned} \max_x \quad & z_1 = f_1(x) \\ & \vdots \\ \max_x \quad & z_k = f_k(x) \end{aligned} \tag{1.4}$$

$$\text{s.t.} \quad Ax \leq b \tag{1.5}$$

$$x \in \mathbb{R}^n \tag{1.6}$$

A multi-objective linear program has, like a single-objective linear program, linear constraints that define the set of feasible solutions X in the decision space (Equation 1.7). In contrast to a linear program, here, the objective space is not one but k-dimensional where k matches with the number of objective functions. If k is one, the optimization problem is a single-objective linear program. The feasible set Z in the objective space is defined by all the points which are part of the feasible set in the decision space (Equation 1.8).

$$X = \{x \in \mathbb{R}^n | Ax \leq b, x_j \in \mathbb{Z}, j \in I\} \tag{1.7}$$

$$Z = \{z \in \mathbb{R}^k | z_i = f_i(x), x \in X, i = 1, ..., k\} \tag{1.8}$$

The concept of optimality in multi-objective optimization problems is different from single-objective linear programs because it is possible that the objective functions conflict with each other. This means that it is not necessarily possible that all objective functions can reach their optimal value at the same time. Solutions that are optimal in the sense of multi-objective programs are called efficient.

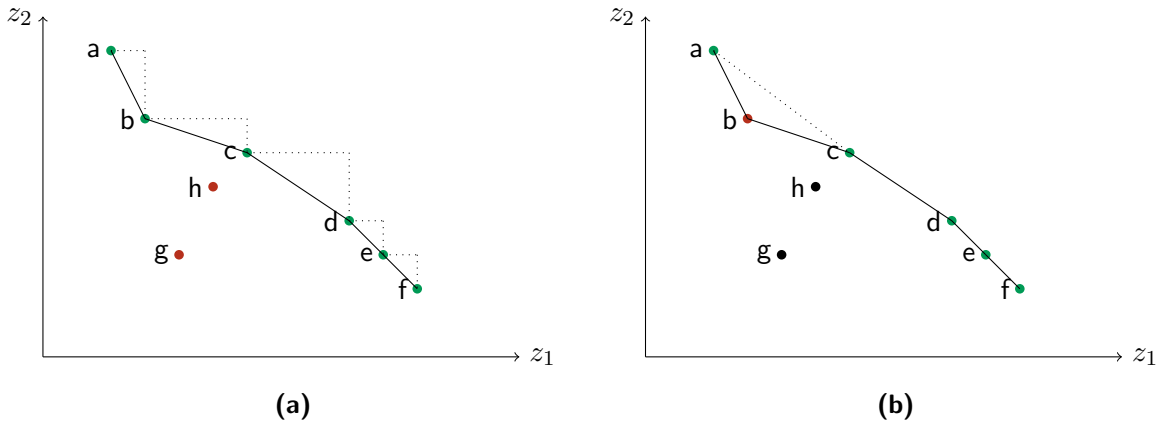


Figure 1.1: *Objective space of a multi-objective linear program for* $k = 2$. In this small example, the set of points $\{a, b, c, d, e, f, g, h\}$ should represent the full set of feasible points Z. **(a)** All nondominated points $Z_N = \{a, b, c, d, e, f\}$ are highlighted in green. The dotted lines illustrate that it is not possible to improve one of the objectives without decreasing the other objective. The points $\{g, h\}$ are dominated. For example, compared to point c, the objective values in point h are both smaller and therefore, h is dominated by c. The set of all nondominated points is also called Pareto front. **(b)** The nondominated points can be divided into supported (highlighted in green) and unsupported (highlighted in red) nondominated points. Point b is unsupported because it is dominated by an infeasible convex combination of a and c which is illustrated by the dotted line.

Hence, a new relation $\succ$ is introduced to enable the comparison of two solutions $z^1, z^2 \in \mathbb{R}^k$ of a multi-objective linear program (Equation 1.9). This relation can be used to decide which solution is better or which solution *dominates* the other. A solution z^1 dominates another solution z^2 if $z^1 \succ z^2$.

$$z^1 \succ z^2 \Longleftrightarrow z_i^1 \geq z_i^2, \quad i = 1, ..., k \quad \text{and} \quad z^1 \neq z^2 \tag{1.9}$$

A solution x^* is called *efficient* if the corresponding point in the objective space is not dominated by any other point. The set X_E includes all efficient solutions (Equation 1.10). The corresponding points in the objective space are nondominated points because there exit no other points that dominate them. The set Z_N contains all nondominated points (Equation 1.11).

$$X_E = \{x \in X | \nexists \bar{x} \in X : f(\bar{x}) \succ f(x)\} \tag{1.10}$$

$$Z_N = \{z \in Z | z = f(x), x \in X_E\} \tag{1.11}$$

In other words, a solution is treated as an efficient solution if it is not possible to increase the value of one objective function further without decreasing the values of one or more of the other objective functions. The difference between nondominated and dominated points is further illustrated in Figure 1.1a. The set of all nondominated points is called *Pareto front* which is why efficient or nondominated solutions are also referred to as Pareto optimal or Pareto efficient solutions.
Furthermore, it is possible to distinguish between supported and unsupported nondominated solutions. Unsupported nondominated solutions are dominated by a (infeasible) convex combination of other nondominated points (Figure 1.1b).

Concepts to solve multi-objective linear programs

In (Antunes et al., 2016) and (Ehrgott, 2005), some basic techniques to solve multi-objective linear programs are given. They include the *weighted-sum method* and the ϵ*-constraint method* which both rely on transforming the multi-objective linear program into several single-objective linear programs that have to be solved separately.
In the case of the weighted-sum method, the corresponding single-objective linear programs maximize the weighted sum for all objective functions of the multi-objective linear program (Equation 1.12. By changing the weights λ in the sum and resolving the changed single-objective linear program, different efficient solutions for the multi-objective linear program can be computed. Usually, the weights are normalized (Equation 1.13).

$$\max_x \quad \sum_{i=1}^{k} \lambda_i f_i(x) \tag{1.12}$$

$$\sum_{i=1}^{k} \lambda_i = 1 \tag{1.13}$$

The weighted-sum method has the advantage that no additional constraints have to be added when transforming the problem. The computation complexity is therefore not changed and the transform problem requires the same computational effort as the single-objective version of the problem. On the other hand, it is only possible to compute supported nondominated points. Unsupported nondom-

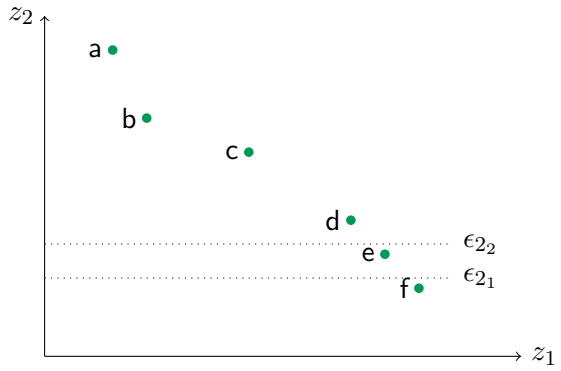


Figure 1.2: *Basic concept of the ϵ-constraint method for $k = 2$.* The idea for this figure is taken from (Antunes et al., 2016). In this example, z_1 is kept as objective function in the transformed problem. The remaining objective function z_2 is turned into the constraint $z_2 \geq \epsilon_2$. In a first step, ϵ_2 can be set to zero, that is only z_1 is considered and maximized which means that the nondominated point f can be found. Afterwards, the value of ϵ_2 needs to be increased to compute other efficient solutions.

inated points cannot be found which means that in nonconvex problems, not all efficient solutions can be found by this approach.
Similar to the weighted-sum method, the ϵ-constraint method also transforms the problem into multiple single-objective linear programs. In the transformed problem, only one of the objective functions is kept (Equation 1.14). The other objective functions are transformed into constraints (Equation 1.15).

$$\max_x \quad f_j(x) \tag{1.14}$$

$$\text{s.t.} \quad f_i(x) \geq \epsilon_i, \qquad i = 1, ..., k, \quad i \neq j \tag{1.15}$$

Changing the bounds for ϵ_i allows to compute different efficient solutions for the multi-objective linear program. An illustration of this idea can be found in Figure 1.2 for the case of two objective functions. In contrast to the weighted-sum method, the ϵ-constraint method can find all efficient solutions but as a result of the additional constraints, the transformed problem can be more difficult to solve.
Another disadvantage that both problems have in common is that it is not clear when or if all nondominated points have been found.
PolySCIP (Borndörfer et al., 2016) is, for example, a solver for multi-objective linear programs.

1.2 Metabolic networks

The *metabolism* is the set of chemical reactions that are taking place in an organism. The metabolism can be modeled through a *metabolic network*. An introduction to metabolic networks, their definitions and also associated methods are given in (Lacroix et al., 2008; Klamt et al., 2014; Cottret and Jourdan, 2010). The following definitions are taken from (Lacroix et al., 2008).
Metabolic networks consist of chemical compounds, biochemical reactions, enzymes and also genes. The chemical compounds are also called *metabolites*. These are small molecules inside an organism that can be imported and exported but also synthesized and degraded. A *biochemical reaction* pro-

duces a set of metabolites, the products, from another set of metabolites, the substrates. Reactions can import metabolites from an external source and they convert them, for example, into other building blocks needed by the organism to survive and to grow. Metabolites that are not needed by the organism and considered as waste can be excreted. Theoretically, reactions can take place in both directions which means that the set of products and substrates are interchangeable. These reactions are therefore *reversible*. However, depending on the physiological conditions, there are reactions that only take place in one specific direction. These reactions are called *irreversible*.
Reactions are catalyzed by *enzymes*. There are some reactions that happen spontaneously and that can be accelerated by enzymes but most reactions need to be catalyzed by enzymes or they cannot take place. Enzymes are proteins or protein complexes that are coded by one or by multiple *genes*. Understanding the relationships between genes, enzymes and reactions in detail is difficult because a single enzyme can catalyze different reactions and a single reaction can be catalyzed by different enzymes. Moreover, sometimes *co-factors* are necessary to enable the catalysis of a reaction by enzymes. Co-factors are small molecules that can bind enzymes thereby changing the activity of the enzyme.
A metabolic network connects metabolites with reactions that transform them. Source reactions can uptake necessary metabolites from external sources and sink reactions excrete waste products or excess metabolites. Certain sets of reactions as a whole are referred to as *metabolic pathway* if they are part of the global synthesis or degradation of a specific metabolite with intermediate steps. For example, the *glycolysis* converts glucose into pyruvate and can be seen as one metabolic pathway.

1.2.1 Metabolic network reconstruction and databases

Metabolic networks are reconstructed based on the available knowledge on relations between genes, enzymes and reactions (Lacroix et al., 2008). Usually, the reconstruction depends on comparative genomics but also on the use of metabolomic data which quantifies the metabolites that are present in the given organism. It is possible to infer from comparative genomics certain reactions that are present in an organism. In a first step, the functional annotation for genes is used to determine which enzymes are existent. Afterwards, the functional annotation is linked to the reaction by identifying which reactions are catalyzed by the present enzymes. Metabolic networks can be reconstructed automatically but the results can be erroneous. Due to their low quality, they should be used with care. To obtain accurate reconstructions, manual corrections based on the literature are necessary.
Metabolic pathways are available in databases like KEGG (Kanehisa and Goto, 2000) and BioCyc (Karp et al., 2019; Caspi et al., 2016). Some databases are specialized on a specific organism, e.g. EcoCyc (Karp et al., 2018) for *Escherichia coli* or HumanCyc (Romero et al., 2005) for the human metabolism. Depending on the organism of interest, it can be difficult to find accurate metabolic networks, especially genome-scale metabolic networks which represent the complete metabolism of an organism. Problematically, the catalogued metabolic pathways can also differ between databases (Altman et al., 2013).

1.2.2 Graph representation

Metabolic networks can be represented in different ways. One of the most common representations is the modeling of a metabolic network as a graph (Lacroix et al., 2008). In general, a graph G is formally defined as a pair (V, E). V corresponds to the set of vertices and E to the set of edges,

where an edge is a subset of 2 vertices in V. The choice of which graph model should be used depends on the application.

The *compound graph* models the metabolic network in a way where the compounds are represented by the vertices of the graph and the edges correspond to the reactions. If an edge connects two compounds, it means that there exists a reaction that transforms one compound into the other one. Therefore, one compound is a substrate and the other one is a product (see Figure 1.3a).

The problem is that there are many reactions that have multiple substrates and multiple products. In order to represent these connections, *hypergraphs* can be used to describe the metabolic network structure (Yeung et al., 2007; Pearcy et al., 2014; Klamt et al., 2009).

Hypergraphs are described in (Berge, 1973). A hypergraph allows to use edges that link more than just two compounds (see Figure 1.3b). A hypergraph H can be again denoted by a pair (V, E) where E is a set of hyperedges. While an edge is a set of two vertices, a hyperedge can be an arbitrary subset of vertices. However, this representation does not allow to distinguish between two sets of vertices that are necessary to represent the substrates and products of a reaction. Additionally, as mentioned before, metabolic networks can also contain reactions that are irreversible. To enable a more precise representation of the metabolic network, a directed hypergraph as described in (Ausiello et al., 2001) can be used. A directed hypergraph is a pair (V, A) where A is the set of directed hyperedges, that is the set of *hyperarcs*. An hyperarc e, is an ordered pair (t, h) of two sets of vertices that are called tail and head of e. They are denoted by $tail(e)$ and $head(e)$. Consequently, substrates and products of a reaction can be represented by these two sets. Here, $tail(e)$ corresponds to the substrates and $head(e)$ to the products of the reactions. Additionally, the distinct orientation of a hyperarc allows to represent the direction of the corresponding reaction (see Figure 1.3c). To

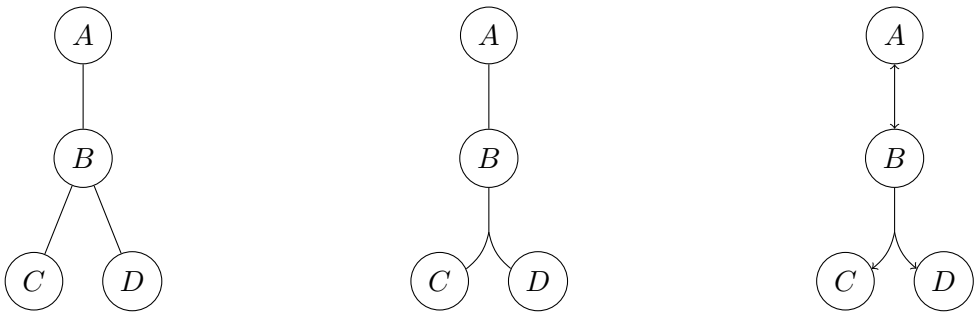
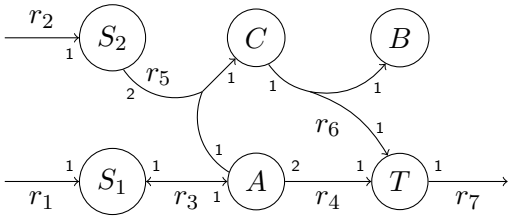


Figure 1.3: *Different compound graphs.* The figures were adapted from (Lacroix et al., 2008). **(a)** An undirected graph. The edges represent reactions that transform one compound into another one. It is difficult to represent reactions that have multiple substrates and/or products. This means that the three displayed edges $A \leftrightarrow B$, $B \leftrightarrow C$ and $B \leftrightarrow C$ could correspond to three different reactions but they could also be the result of, for example, the reaction $B \leftrightarrow A + C + D$. Since no distinct directions can be represented, it is not possible to distinguish between reversible and irreversible reactions. **(b)** An undirected hypergraph. A hyperedge has one arbitrary subset of vertices. Therefore, it is not possible to distinguish between the substrates and products of a reaction. Here, the represented reaction could be $B \leftrightarrow C + D$ but other interpretations like $C \leftrightarrow B + D$ are also possible. Additionally, it is not possible to represent that a reaction is irreversible. **(c)** A directed hypergraph. The possible directions for each reaction are displayed by arrows. This representation allows for a simple discrimination between reversible and irreversible reactions. Additionally, there is a clear separation between substrates and products. Reaction $A \leftrightarrow B$ is reversible whereas $B \rightarrow C + D$ is irreversible.

model reversible reactions, bidirectional hyperarcs can be used.

1.2.3 Stoichiometric matrix

All presented graph models represent the structure (topology) of the metabolic network but they do not take into account any information that might be available about the quantitative relationships between the substrates and the products of a reaction. Depending on the application, it might be important to include the *stoichiometry* of the reactions. A *stoichiometric coefficient* describes the quantity in which a metabolite participates in a specific reaction. For example, given the reaction $2A + 1B \rightarrow 3C$, two molecules of metabolite A and one molecule of metabolite B are needed to produce three molecules of C. It is possible to add the stoichiometric information as labels in a graph model. However, another commonly used representation is the *stoichiometric matrix*.
The stoichiometric matrix S is an $m \times n$ matrix where the m rows correspond to the metabolites in the metabolic network and the n columns to the reactions. The entry S_{ij} corresponds to the stoichiometric coefficient of metabolite i in reaction j. If the stoichiometric coefficient is positive, the metabolite is produced by the corresponding reaction. If it is negative, the metabolite is consumed. If the stoichiometric coefficient is zero, it indicates that the metabolite is not participating in this reaction or that the production compensates precisely its consumption.
The stoichiometric matrix itself does not contain any information about the reversibility of the reactions. Usually, this information is stored alongside the matrix in the form of lower bounds LB_j and upper bounds UB_j for the flux of each reaction j. For irreversible reactions, the lower bounds are greater or equal to zero. Changing the lower and upper bounds for source reactions can be used



(a)

$$S = \begin{array}{c} \begin{matrix} r_1 & r_2 & r_3 & r_4 & r_5 & r_6 & r_7 \end{matrix} \\ \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 1 & -2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & -1 \end{pmatrix} \end{array} \begin{matrix} S_1 \\ S_2 \\ A \\ C \\ B \\ T \end{matrix}$$

(b)

Figure 1.4: *Directed hypergraph and stoichiometric matrix representation.* **(a)** A metabolic network can be represented as a directed hypergraph. In a directed hypergraph, the set of reactions $R = \{r_1, r_2, r_3, r_4, r_5, r_6, r_7\}$ is represented by the hyperarcs. The set of metabolites $M = \{S_1, S_2, A, C, B, T\}$ is represented by the vertices. It is possible to display the stoichiometric values for each reactions by labeling the hyperarcs. In the given example, all reactions except reaction R_3 are irreversible. **(b)** The network can also be represented as stoichiometric matrix S. The rows of S correspond to the metabolites in M and the columns to the reactions in R. The stoichiometric values are displayed as coefficients of the matrix. If a coefficient is positive, the corresponding metabolite is produced by the reactions. If it is negative, it is consumed. Hence, in the matrix representation, a direction is assumed for each reaction. However, it is not possible to see if a reaction is reversible or not. Lower and upper bounds for reactions have to be stored separately.

to model different growth media by allowing or restricting the uptake of specific metabolites, that is the import of metabolites from an external source.

1.2.4 Constrained-based modeling for metabolic networks

The stoichiometric matrix representation is used in *constraint-based modeling* of metabolic networks (Covert and Palsson, 2003; Palsson, 2000; Lacroix et al., 2008; Lewis et al., 2012). In constraint-based modeling, the idea is to analyze or identify possible flux distributions in the metabolic network under certain constraints.
The *flux vector* describes the state of all reactions of the metabolic network and shows if a reaction has a flux, i.e. is active. The flux vector v is an n-vector in which v_j corresponds to the flux of reaction j.
A common assumption is that the metabolic network is in *steady-state* which means that every metabolite is produced in the same amount as it is consumed. Based on the stoichiometric matrix and the flux vector, these conditions can be formulated as constraints:

$$S \cdot v = 0 \tag{1.16}$$

$$LB_j \leq v_j \leq UB_j \qquad \forall j \in R_{rev} \tag{1.17}$$

$$0 \leq v_j \leq UB_j \qquad \forall j \in R_{irr}. \tag{1.18}$$

Equation 1.16 defines the mass balance for all metabolites. Additionally, lower and upper bounds are added to restrict the flux of single reactions (Equation 1.17). For irreversible reactions, the lower bound is set to zero (Equation 1.18). These constraints describe the core idea for several constraint-based approaches for metabolic networks. In the following, some of the most common among such approaches are presented that can also be used as a basis for more advanced methods.
Probably the most important approach is *Flux-balance-analysis* (FBA) which is widely used to analyze metabolic networks (Orth et al., 2010b; Mahadevan and Schilling, 2003; Edwards and Palsson, 2000; Durot et al., 2008; Covert and Palsson, 2003; Gottstein et al., 2016). FBA uses the aforementioned constraints to restrict the feasible flux space. Additionally, FBA optimizes a specific objective function (Equation 1.19), e.g. it can simulate the growth of an organism by maximizing the production of biomass compounds.

$$\max \quad c^T v \tag{1.19}$$

$$\text{s.t.} \quad S \cdot v = 0 \tag{1.20}$$

$$0 \leq v_j \leq UB_j \qquad \forall j \in R_{irr} \tag{1.21}$$

$$LB_j \leq v_j \leq UB_j \qquad \forall j \in R_{rev} \tag{1.22}$$

Common applications for FBA are, for instance, to predict the growth of an organism on different media or under different conditions (e.g. aerobic and anaerobic). It can predict the phenotype (Edwards and Palsson, 2000; Edwards et al., 2001) and hence, it is also used to analyze the altered behavior of an organism that is subject to metabolic interventions, i.e. to predict the behavior of a mutant strain. The maximization of biomass or energy (ATP) production are frequently used objectives (Feist and Palsson, 2010; Pramanik and Keasling, 1997). The core idea of FBA is also used as basis for other similar methods, such as flux variability analysis. In Figure 1.5, a general work-flow of FBA is presented.

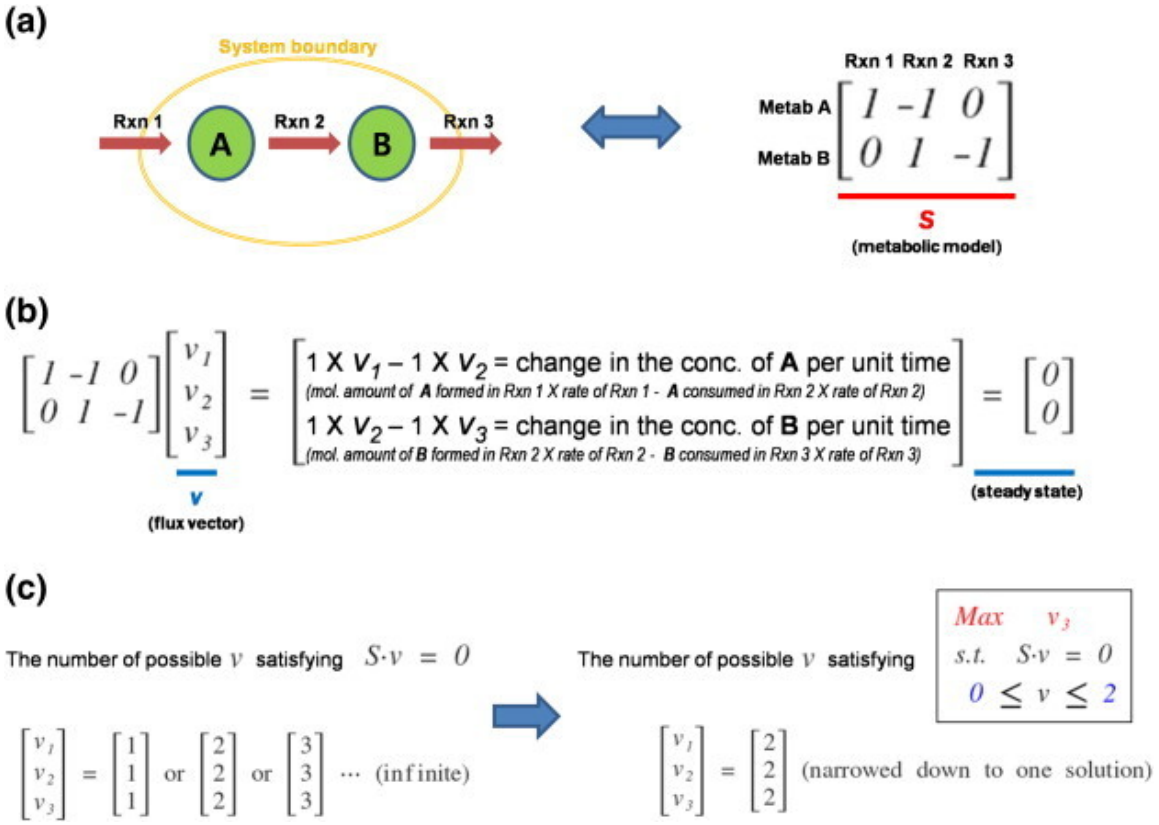


Figure 1.5: *Flux balance analysis.* The figure was taken from (Kim and Lun, 2014). **(a)** Example of a small metabolic network. Reaction 1 is a source reaction that imports metabolites into the boundaries of the system. Reaction 3 exports metabolites. The network can be represented as stoichiometric matrix. Since reaction 1 can uptake metabolites from an external source, is does not have a substrate. In the same way, reaction 3 does not have any products. **(b)** Often, steady-state is assumed which means that concentration for metabolites does not change. **(c)** The feasible flux space can be restricted by adding lower and upper bounds for reactions.

A variant of FBA, *parsimonious enzyme usage FBA* (pFBA) (Lewis et al., 2010), assumes that an optimal solution should correspond the lowest overall flux through the network which simulates the minimization of the total amount of enzyme mass that is necessary.
As argued in (Segrè et al., 2002), when analysing metabolic networks, the maximization of the biomass production can be a reasonable assumption because it reflects the hypothesis that organism tend to optimize their growth under evolutionary pressure. Problematically, this assumption might not be valid for mutant strains that are the result of metabolic engineering because they are less exposed to evolutionary pressure than the original wild type strain.
Based on these observations, the authors in (Segrè et al., 2002) present their method *Minimization of metabolic adjustment* (MOMA) which is similar to FBA but instead of maximizing the biomass to predict flux distributions for the mutant strain, they are minimizing the distance between the wild type flux and the flux in the mutant type (Equation 1.23). Here, the assumption is that the perturbed flux should remain as close as possible to the original flux because the organism might try to make as little changes as possible to its metabolic activity because using different reactions that were inactive before might also require different or additional enzymes.

$$\min \quad f(v) = \sqrt{\sum_{j=1}^{|R|}(v_j - \bar{v}_j)^2} \tag{1.23}$$

MOMA is a quadratic optimization problem because the objective function minimizes the square root of the sum of the squared distances between the wild type flux $\bar{v}$ and the perturbed flux in the mutant type v. The flux values for the wild type can, for instance, be obtained by doing a FBA that maximizes the biomass production. In many cases, there is however not only one unique flux vector that leads to the optimal biomass production (Mahadevan and Schilling, 2003). This can render the choice for the wild type flux more difficult and the outcome of MOMA may differ depending on which exact flux vector was chosen as wild type. Alternative solutions are a general problem of FBA because it only computes one flux distribution.
One possible approach to analyze alternative solutions is to identify by how much the flux for reactions can differ under the optimal condition, e.g. when the biomass is maximized. *Flux variability analysis* (FVA) allows to analyze the possible variation of the flux for all reactions under a specific condition (Mahadevan and Schilling, 2003; Burgard et al., 2001). It determines the minimum and the maximum flux values for all reactions under the optimal condition (Equation 1.24). This is achieved by introducing a constraint that fixes the biomass production to the optimal value z^* calculated by FBA (Equation 1.28).

$$\begin{aligned}
\min/\max \quad & v_j && && (1.24)\\
\text{s.t.} \quad & S \cdot v = 0 && && (1.25)\\
& 0 \leq v_j \leq UB_j && \forall j \in R_{irr} && (1.26)\\
& LB_j \leq v_j \leq UB_j && \forall j \in R && (1.27)\\
& c^T v = z^* && && (1.28)
\end{aligned}$$

In (Lee et al., 2000), a recursive MILP is proposed to find alternative solutions. Furthermore, in (Kelk et al., 2012) it is described that often, the possible flux distributions of a few subnetworks lead to the combinatorial explosion of optimal fluxes for the whole network.

A similar idea to FVA can also be used to analyze how reactions depend on each other if reactions are blocked under certain conditions (Burgard et al., 2004). *Flux coupling analysis* identifies different relations between two fluxes v_1 and v_2: They are *directionally coupled* $(v_1 \rightarrow v_2)$ if a non-zero flux for v_1 implies that v_2 must have a non-zero flux. This does not have to imply that a non-zero flux for v_2 always leads to a non-zero flux for v_1. Furthermore, they are *partially coupled* $(v_1 \leftrightarrow v_2)$ if a non-zero flux for v_1 implies a non-zero flux for v_2 and a non-zero flux for v_2 implies a non-zero flux for v_1. Finally, two fluxes are *fully coupled* $(v_1 \Leftrightarrow v_2)$ if the same conditions hold as for the partial coupling and in addition the implied fluxes are not variable but fixed which means that the ratio between the two fluxes is constant. If none of these relations holds for a pair of reactions, they are *uncoupled*. The differences are also displayed in Figure 1.6.

In (Burgard et al., 2004), the authors also identify *blocked* reactions which are reactions that can only have a zero flux under the given constraints that model, for instance, a certain medium, thereby restricting certain uptake fluxes (Equation 1.31). All reversible reactions are split into two irreversible reactions. Thus, negative fluxes are not possible (Equation 1.32) and it is sufficient to maximize the flux for a reaction to determine if the reaction is blocked or not. If reversible reactions are not split, both the maximum and minimum have to be computed which shows the direct relation to the FVA.

$$\max \quad v_j \tag{1.29}$$

$$\text{s.t.} \quad S \cdot v = 0 \tag{1.30}$$

$$v_j^{uptake} \leq v_j^{uptake\_max} \qquad \forall j \in R_{transport} \tag{1.31}$$

$$v_j \geq 0 \qquad \forall j \in R \tag{1.32}$$

Identifying blocked reactions is of interest because it can help to simplify the underlying network structure for specific cases. If reactions are blocked under the conditions that are modeled, it is possible to remove these reactions for other analyses under the same conditions. However, it is important to note that whether a reaction is blocked or not is very depending on the specific

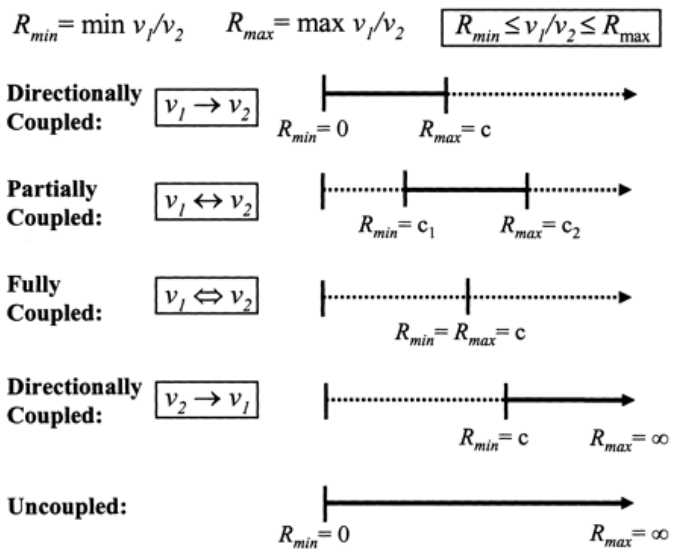


Figure 1.6: *Different types of flux coupling*. The figure was taken from (Burgard et al., 2004). Different scenarios for flux coupling are shown based on the flux ratio limits of $R_{min} = \min v_1/v_2$ and $R_{max} = \max v_1/v_2$ which have to be computed.

conditions (e.g. steady-state or uptake rates).
Another way to analyze the steady-state flux distributions in metabolic networks are *elementary flux modes* (EFM) (Schuster and Hilgetag, 1994). A more recent review on the calculation and application of EFMs is given in (Zanghellini et al., 2013). The support of a flux vector v is defined by $supp(v) = \{i|v_i \neq 0\}$ that is the set of indices of non-zero elements. A flux vector v is called mode if it is non-trivial (i.e. $v \neq 0$) and is feasible under steady-state conditions. An EFM is a mode whose support cannot be written as proper subset of any other feasible mode. Therefore, EFMs are non-decomposable and represent minimal functional building blocks. A closely related idea are *extreme pathways* which are a subset of elementary modes (Schilling et al., 2000).

1.3 Predicting optimal intervention strategies

One of the areas of synthetic biology is the genetic engineering of microorganisms to improve the production of important target chemicals by a microorganism. Microorganisms are able to produce chemical compounds that can, for instance, be used in industry. The desired compound might often just be a by-product of the usual metabolism when the microorganism is growing and the yield can be very low. Thus, it is necessary to modify the metabolism of the organism to increase the yield of the desired target compound and to establish a more efficient production. The idea is to deactivate (knockout) specific reactions by manipulating the metabolic genes that are coding for the catalyzing enzymes. Depending on the role the knocked out reactions play in the metabolism, the organism might use different pathways to compensate for its lack.
Given the complexity of the problem, there is a need to identify the most promising intervention strategies *in silico* and it is necessary to predict and analyze the altered behavior after a knockout is introduced. A review on constraint-based methods for optimal strain design *in silico* is given in (Maia et al., 2016). In the following, I will summarize some computational approaches to predict optimal intervention strategies.
In (Burgard et al., 2003), the authors present OptKnock which is one of the most well known methods for identifying knockouts. Their method is based on a bi-level optimization problem that aims to find a flux distribution that maximizes the production of the compound of interest while also maximizing the production of the biomass given some knocked out reactions.

$$\max_{y_j} \quad v_{target} \tag{1.33}$$

s.t.

$$\left\{ \begin{array}{lll} \max\limits_{v_j} & v_{biomass} & \\ \text{s.t.} & S \cdot v = 0 & \\ & LB_j(1-y_j) \leq v_j \leq UB_j(1-y_j) & \forall j \in R \end{array} \right\} \tag{1.34}$$

$$\sum_{j \in R} y_j = K \tag{1.35}$$

$$y_j \in \{0,1\}, v_j \in \mathbb{R} \qquad \forall j \in R \tag{1.36}$$

The outer problem of the bi-level framework maximizes the bioengineering objective, i.e. the production of the target (Equation 1.33). The inner problem maximizes a cellular objective like the biomass production (Equation 1.34). The knockouts are modeled using a binary variable y_j for each reaction

j. If the binary is set to one, the lower and upper bounds of the corresponding reaction are forced to zero which imitates a knockout (Equation 1.34). The approach identifies the optimal knockout set for a maximum number of knockouts (Equation 1.35). OptKnock is integrated in the COBRA Toolbox (Heirendt et al., 2019) available in MATLAB.

OptKnock can be overoptimistic because it assumes that both biomass and target production are maximized by the organism. For a given biomass production, even though a specific maximum target yield is possible, it does not mean that it will be how the organism behaves in reality. As explained in Section 1.2.4, there can be lots of alternative flux distributions that lead to the same optimal value. To account for competing pathways, in (Tepper and Shlomi, 2009), the authors present another bi-level program which is called RobustKnock. It is similar to OptKnock. However, the crucial difference is that in the outer problem the minimum of the bioengineering objective is maximized. In this way, they ensure that the target metabolite has to be produced at least in the determined minimum amount in all flux distributions that are still achievable after knockouts have been introduced.

The approaches considered so far only model knockouts. To consider also up- an downregulation of fluxes, OptForce is presented in (Ranganathan et al., 2010). Like OptKnock, it is a bi-level optimization problem. In a first step, for each reaction, the possible lower and upper bounds are computed by FVA to characterize the wild-type. This information is used to identify reactions sets that must change to achieve a desired overproduction specified by the user. These sets are called *MUST* sets. Based on the them, *FORCE* sets are computed which are minimal sets of reactions that must be genetically manipulated to force a change and to achieve the desired production yield.

Another different concept that can be used to identify intervention strategies in metabolic networks are *minimal cut sets* (MCS) which are described in (Klamt and Gilles, 2004; Klamt, 2006; Ballerstein et al., 2012). In general, MCSs are suitable to determine reactions that can be used to block metabolic functionalities. With regard to a specific target reaction that should be blocked, a set of reactions is a cut set if after all these reactions have been removed (e.g. knocked out), there is no feasible flux distribution possible that contains the target reaction. A cut set C is minimal if no proper subset of C is also a cut set.

The idea of MCS led to the introduction of *constraint minimal cut sets* (cMCS) in (Hädicke and Klamt, 2011). Whereas MCSs are used to block undesired phenotypes, cMCSs allow to specify desired behaviors that should be kept. It prevents that MCSs are computed that block important functions. This concept was further generalized to *regulatory* MCSs that are able to not only model the knockout of reactions but also up/downregulations of reaction rates (Mahadevan et al., 2015). Regulatory MCSs and cMCSs were already applied successfully to identify intervention strategies in genome-scale metabolic networks (von Kamp and Klamt, 2014; Mahadevan et al., 2015).

1.4 Metabolic shifts

Metabolomics is a field that concerns itself with the measurement of metabolites (Roessner and Bowne, 2009). It studies the metabolome which refers to the total number of metabolites that are found in the cells of an organism. Experiments to measure metabolite levels can be targeted, which means that a specific set of metabolites is quantified that are suspected to be relevant for the given experiment but there are also global analyses techniques available that take into account all measurable metabolites (Fiehn, 2002).

Like transcriptomics and proteomics, metabolomics plays an important part in understanding how an organism reacts to changes in the environment. This can be, for example, the adaption of the organism to a change in the medium (e.g. nutritional stress). Furthermore, also the impact of genetic perturbations (e.g. mutations) on the organism can be analysed. Metabolomics makes it possible to explain how the metabolic activity of an organism changes after a perturbation and to identify which parts of the organism are affected. The change in metabolic activity is called *metabolic shift*. Understanding metabolic changes under different conditions is important for advancing research in different fields like bioengineering or human health (Sevin et al., 2015; Goel et al., 2012).
In enrichment analyses of metabolomic data based on the metabolites and their relative abundances, pathways are identified that are more impacted by the perturbation than others (Booth et al., 2013). Several tools are already published that facilitate the analysis of metabolomic data. For example, MetaboAnalyst (Xia et al., 2015; Chong et al., 2018) is a web-based tool that includes different modules for the integration of metabolomic data. Amongst others, it provides modules for metabolite set enrichment analysis (Xia and Wishart, 2010b), metabolic pathway analysis (Xia and Wishart, 2010a), as well as two-factor and time-series analyses (Xia et al., 2011). MetExplore (Cottret et al., 2018), another web-based application, is a versatile tool for the analysis of metabolic networks. It also implements MetaboRank (Frainay et al., 2019) which can be used to interpret and enrich metabolomic data. MBRole (Chagoyen and Pazos, 2011), which is another web-server, can also be used to perform enrichment analysis of metabolomic data. Different tools and methods for the enrichment analysis of metabolomics are compared and evaluated in (Marco-Ramell et al., 2018; Booth et al., 2013; Alonso et al., 2015).
As also mentioned in (Frainay and Jourdan, 2017), one of the main issues in the analysis of metabolomic data is that there is no single metabolomics technology that allows to measure all metabolites. This means that only a partial view of the metabolome can be obtained which makes the subsequent analysis more complicated. Furthermore, as summarized in (Booth et al., 2013) there is still a large number of metabolites that is unidentified and therefore also not characterized or annotated in known databases. A lack of refined networks and databases for organisms that are less commonly studied makes the analysis of metabolomic data more challenging.

2 Identifying knockouts when the target chemical is toxic for the organism

Contents

2.1 Introduction

Microorganisms are already used on an industrial scale to produce important target chemicals. Metabolic engineering of strains has been widely used in order to optimize the bioconversion pathways aiming at higher product yield. However, accumulation of the target chemical can often negatively impact the cultivation of the microorganism (Mukhopadhyay, 2015). Problematically, there are even more extreme cases where the target chemical is toxic for the used microorganism which might therefore restrict the efficiency of the production. Strain engineering also focuses on alleviating these bottlenecks, reducing cellular burdens that might limit product yield in order to achieve optimal production.

Among such chemicals, there is a growing interest in ethanol due to its use as bio-fuel (Mussatto et al., 2010). Yeasts can convert sugars to ethanol during a process called fermentation (Lin and Tanaka, 2006; Bai et al., 2008; Boulton et al., 1999) which is already used to produce ethanol industrially. The production of ethanol in yeast can be improved by metabolic engineering (Borodina and Nielsen, 2014; Nielsen et al., 2013; Van Vleet and Jeffries, 2009). However, an important limiting factor for the production of ethanol in yeast on an industrial scale is indeed the toxicity of ethanol for the yeast. When ethanol accumulates, it inhibits growth but leads also to a decline of the ethanol production (Van Uden, 1985; Dombek and Ingram, 1986; D'Amore and Stewart, 1987; Casey and Ingledew, 1986; Lam et al., 2014; D'amore et al., 1989). Therefore, different approaches are investigated to increase the tolerance of yeast to ethanol (Dombek and Ingram, 1986; Alper et al., 2006; Lam et al., 2014). Such tolerance can differ between yeast strains (Casey and Ingledew, 1986).

As mentioned in Section 1.3, different constraint-based approaches were already developed that can be used to identify optimal intervention strategies in metabolic networks. The methods that were presented (e.g. OptKnock, RobustKnock) are some of the most common ones for metabolic engineering but they are mostly based on single-objective linear programs or bi-level approaches. However, in general, multi-objective optimization is already commonly applied in bioinformatics and computational biology (Handl et al., 2007). Multi-objective approaches have the advantage that they can investigate conflicting objectives at the same time.

In the context of metabolic engineering, this can be, for example, the maximization of a cellular objective (e.g. biomass production) and the maximization of an engineering objective (e.g. the production of a target chemical). For example, in (Patané et al., 2019), the authors propose a multi-objective metabolic engineering algorithm that can model gene knockouts but also up- and downregulation of enzymes. Their method solves a multi-objective optimization problem for biomass production and ethanol production. Other examples of the application of multi-objective frameworks for ethanol production are presented in (Vera et al., 2003; Sendín et al., 2006). In (Andrade et al., 2020), a multi-objective approach called MOMO is introduced that identifies points in the Pareto front that represent different tradeoffs between biomass and target production. Afterwards, MOMO enumerates and analyzes possible single knockouts for each point in the Pareto front. These methods however do not account for the fact that ethanol is toxic for yeast.

In this chapter, we present a new constraint-based approach inspired by MOMO (Andrade et al., 2020) that proposes knockout sets to improve the production of any target chemical that is also toxic for the microorganism and whose accumulation might inhibit growth and slow down the production. We therefore also ensure that the introduced knockouts do not restrict the metabolic activity of the organism that is crucial for a better resistance against the toxic target chemical. The approach is separated in two parts. In the first part, a multi-objective optimization problem is used to identify

different efficient flux distributions that maximize the biomass production, the target production and the resistance against the toxic target. In the second part, different approaches are presented to enforce a flux distribution of interest that was identified in the first part. For the second part, different ideas were explored. As a first idea, a MILP is proposed to enumerate different knockouts. Due to its limitations, there was a need to develop a second idea which is based on identifying and cutting off subnetworks. The present method is applied to the case-study of ethanol production in yeast.

2.2 Computation of tradeoffs

In the first part of this approach, a multi-objective optimization problem is used to identify tradeoffs between biomass production, target production and the resistance against the toxic target. To calculate a score that describes the resistance of the organism against the toxic target, it is necessary to have information on certain critical reactions that must have been identified beforehand. A critical reaction is a reaction that increases the resistance of the organism against the toxic target when it is active. Computing these tradeoffs allows to gain an overview of optimal outcomes for which feasible flux distributions exist.

2.2.1 Toxicity resistance score

To evaluate the resulting resistance against the toxic target for a specific flux distribution, a score r is defined that depends on the flux values of the critical reactions:

$$r = \sum_{j \in R_{critical}} (\omega_j t_j + \omega_j \frac{v_j}{UB_j}). \tag{2.1}$$

The score is based on three different assumptions concerning the reactions that are critical for the resistance against the target metabolite:

1. *A non-zero flux is more important than the amount of flux for a critical reaction.* This assumption is represented by the first part of the sum. For each reaction j that was identified to be important for the toxicity resistance, a binary variable t_j is introduced. If the corresponding reaction has a flux, the binary variable is set to one. Thus, if many important reactions have a flux, the score will be high. The exact quantity of the flux has no impact on this part of the score.

2. *A higher flux is better than a lower one.* It is represented by the second part of the sum. The flux values v_j are normalized by their upper bound UB_j to be between 0 and 1 before they are added to the score to ensure that they are less significant for the total score than the binary variables.

3. *Some reactions have a higher impact on the toxicity resistance than other reactions.* These differences can be modeled by setting individual weights w_j for each critical reaction.

2.2.2 Multi-objective optimization problem

The toxicity resistance score is used to formulate a multi-objective optimization problem that maximizes three different objectives: The biomass production $v_{biomass}$, the target production v_{target} and

the toxicity resistance score r.

$$\max \quad v_{target}, v_{biomass}, \sum_{j \in R_{critical}} (\omega_j t_j + \omega_j \frac{v_j}{UB_j}) \tag{2.2}$$

$$\text{s.t.} \quad S \cdot v = 0 \qquad \forall i \in M \tag{2.3}$$

$$LB_j \leq v_j \leq UB_j \qquad \forall j \in R \tag{2.4}$$

$$v_j = 0 \Longrightarrow t_j = 0 \qquad \forall j \in R_{critical} \tag{2.5}$$

$$v_j, \omega_j \in \mathbb{R}; t_j \in \{0, 1\} \tag{2.6}$$

The system assumes a steady-state (Equation 2.3) and for each reaction upper and lower bounds are specified (Equation 2.4). Furthermore, constraints are introduced to connect the flux values v_j to the binary variables t_j for all critical reactions $R_{critical}$ (Equation 2.5). If the corresponding flux value is zero, the binary variable must be set to zero too. If the reaction has a non-zero flux, the binary variable can be set to one. Here, it is not necessary to explicitly model $v_j \neq 0 \Longrightarrow t_j = 1$ that would ensure that a binary variable has to be set to one if the corresponding flux value is different from zero because the optimization problem is maximizing the toxicity resistance score. This means that if a t_j can be set to one, the solver will set it to one to increase the score.
In practice, a small threshold m is used to identify reaction fluxes different from zero and the implication constraint is remodeled:

$$-\infty \leq t_j - v_j \leq 1 - m. \tag{2.7}$$

There are two possibilities. If the reaction has a flux greater or equal to m, the difference $d_j = v_j - m$ is greater to equal to zero. Therefore, $t_j \leq 1 + d_j$ holds and t_j can be set to one. If the reaction flux is smaller than m (and hence it is assumed to be zero), d_j is negative and t_j can only be set to a value smaller than one, and since t_j is a binary, it can only be set to zero.
However, this is only viable if the reaction cannot have a negative flux. For reversible reactions, the constraint needs to be adjusted:

$$-\infty \leq t_j - |v_j| \leq 1 - m. \tag{2.8}$$

To model the absolute value $|v_j|$, this part can be split:

$$-\infty \leq t_j - v_j^+ - v_j^- \leq 1 - m. \tag{2.9}$$

In this case, v^+ and v^- are two new non negative variables with the corresponding bounds:

$$0 \leq v_j^+ \leq UB_j \cdot y_j \tag{2.10}$$

$$0 \leq v_j^- \leq |LB_j| \cdot (1 - y_j). \tag{2.11}$$

The idea is the same as splitting a reversible reaction in two irreversible reactions. The forward direction is represented by v_j^+ and the backward direction by v_j^-. Since in the case of a reversible reaction, the lower bound is negative, the absolute value of it is used as upper bound for v_j^-. The new binary variable y_j prevents that v_j^+ and v_j^- are non zero at the same time. The variables v_j^+ and v_j^- are only needed if the reaction is reversible (i.e. the lower bound is negative). Otherwise,

Equation 2.7 is sufficient.
In the resulting Pareto front of this multi-objective optimization problem, each point describes a different possible tradeoff between the three objectives. Computing only the extreme points is enough to gain a broad overview of the whole Pareto front. The idea is to choose a point in the Pareto front that has a high target yield and a sufficient biomass production and to find a way to enforce the values for target and biomass yield and the toxicity score. One possibility is to introduce knockouts that restrict the phenotypic space around the desired values for target and biomass production as mentioned in Section 1.3.
The multi-objective optimization problem was implemented in C++ and PolySCIP (Borndörfer et al., 2016) was used as solver which is part of the SCIP Optimization Suite 5.0 (Gleixner et al., 2017a,b). The metabolic network was modeled using the library metnetlib which is available on `https://gitlab.inria.fr/erable/kirikomics/metnetlib`.

2.3 First approach - Identifying knockouts that enforce the tradeoff flux

2.3.1 MILP formulation

After a promising point p of the Pareto front has been picked, a modified version of the first MILP is used to search for knockouts that might restrict the phenotypic space enough to enforce the desired values for the toxicity score and the target and biomass yield. The values are fixed to the corresponding values from the Pareto front v^p_{target}, $v^p_{biomass}$ and r^p (Equations 2.15, 2.16, 2.17). It is not necessary to minimize or maximize an objective function because we are looking for different feasible solutions that correspond to these fixed values.
Additionally, new binary variables x_j are introduced to model reaction knockouts. It is possible that not all reactions are potential candidates for a knockout. If it is known before that it is not viable to knockout a certain reaction *in vivo*, it is not included in $R_{knockout}$. Each reaction j in $R_{knockout}$ has an associated binary variable x_j. If the binary is set to one, the reaction flux is forced to zero (Equation 2.18). Furthermore, the number of total knockouts that are introduced in the metabolic network are fixed to a specified number K (Equation 2.19).

$$
\begin{aligned}
\text{s.t.} \quad & \sum_{j \in R} s_{ij} v_j = 0 && \forall i \in M && (2.12)\\
& LB_j \leq v_j \leq UB_j && \forall j \in R && (2.13)\\
& v_j = 0 \Longrightarrow t_j = 0 && \forall j \in R_{critical} && (2.14)\\
& v_{target} = v^p_{target} && && (2.15)\\
& v_{biomass} = v^p_{biomass} && && (2.16)\\
& r = r^p && && (2.17)\\
& x_j = 1 \Longrightarrow v_j = 0 && \forall j \in R_{knockout} && (2.18)\\
& \sum x_j = K && \forall j \in R_{knockout} && (2.19)\\
& v_j, \omega_j \in \mathbb{R}; t_j, x_j \in \{0,1\}; K \in \mathbb{N}^+ && && (2.20)
\end{aligned}
$$

To model $x_j = 1 \Longrightarrow v_j = 0$, in CPLEX, indicator constraints can be used. They are also called

IfThen constraints. In SCIP, the included indicator constraint is modeled as follows:

$$z = 1 \Longrightarrow ax \leq 0. \tag{2.21}$$

Here, z is a binary variable. To model the equality in Scip if the flux can be negative, the constraint in Equation 2.18 has to be split into two new indicator constraints:

$$x_j = 1 \Longrightarrow v_j \leq 0 \tag{2.22}$$
$$x_j = 1 \Longrightarrow -v_j \leq 0. \tag{2.23}$$

The proposed MILP can be used to propose reaction knockouts that still allow the desired values for the three objectives. To enumerate different combination of knockouts, it has to be solved multiple times and after each iteration, the computed knockout set has to be excluded as solution by adding a corresponding constraint to the optimization problem. A knockout set P contains all variables x_j that were set to one by the solver. By adding Equation 2.24 as constraint, it can be prevented that the exact same combination of x_j is chosen again:

$$\sum_{x_j \in P} x_j \leq K - 1. \tag{2.24}$$

The problem can be solved repeatedly until it becomes infeasible to enumerate all knockout sets that are possible for the fixed values for biomass and target production and the toxicity score. For a small K, i.e. $K = 1$ or $K = 2$, it is possible to enumerate all knockout sets even for larger networks. Depending on the size of the network and on the size of $R_{knockout}$, for larger K, this task becomes more difficult. Therefore, reducing the size of $R_{knockout}$ can help simplify the problem.

2.3.2 Reducing the number of knockout candidates

Several steps can be taken to reduce the size of $R_{knockout}$. First of all, a pre-selection should be done. For example, it can be difficult to knockout some transport reactions, given that a substrate can sometimes be transported by more than one system and also because transporters can be nonspecific. Exchange reactions are artificial reactions added to the networks to model boundaries of the organism. Furthermore, all genome-wide reconstructed models have reactions that do not have any associated genes. These reactions should be removed from $R_{knockout}$ because their knockout is not applicable *in vivo*.

As a next step, a certain aspect of Flux coupling analysis is used to determine groups of reactions that are knocked out simultaneously. We are interested in identifying such groups of reactions. For each two reactions i, j in a group, it holds that:

$$v_j == 0 \Longleftrightarrow v_i == 0. \tag{2.25}$$

To achieve this, single knockouts for all reactions currently in $R_{knockout}$ are done and afterwards, FVA is used to analyze the possible flux values for the other reactions in $R_{knockout}$. If a reaction can only have a zero flux, the other direction of the equivalence of Equation 2.25 is verified. As a result, reactions will be grouped together and only one representative for each group is present in $R_{knockout}$. All others are removed. If the representative is chosen as knockout, afterwards, it is

possible to replace it by other reactions from the same group without changing the result. This step also eliminates the need to simplify linear pathways. For instance, if there is a linear pathway like $A \rightarrow B \rightarrow C \rightarrow D$, it is possible to reduce it to $A \rightarrow D$ thereby lowering the number of metabolites and reactions without changing the overall behavior of the metabolic network. By determining the coupled groups of reactions and only using one representative that can be knocked out, it is also possible to remove parts of linear pathways from $R_{knockout}$ without changing the network.

To further reduce the size of $R_{knockout}$, it might be important to analyze the influence of single knockouts. All remaining reactions in $R_{knockout}$ are knocked out separately (Equation 2.29) and afterwards, FBA is used to maximize the biomass production. If it is zero, the candidate k can be removed from $R_{knockout}$. Otherwise, the target production is maximized subsequently. Likewise, if the maximum target production is zero, the candidate k is removed.

$$\max \quad v_{biomass} \quad (\text{or} \quad v_{target}) \tag{2.26}$$

$$\text{s.t.} \quad \sum_{j \in R} s_{ij} v_j = 0 \qquad \forall i \in M \tag{2.27}$$

$$LB_j \leq v_j \leq UB_j \qquad \forall j \in R \tag{2.28}$$

$$v_k = 0 \tag{2.29}$$

$$v_j \in \mathbb{R} \tag{2.30}$$

The reactions that are removed in this way from $R_{knockout}$ cannot be chosen by the MILP presented in Section 2.3.1 because they do not allow for a biomass or target production. However, removing them beforehand leads to less binary variables in the MILP because only for reactions in $R_{knockout}$, an associated binary variable x_j is introduced.

All the steps to reduce the size of $R_{knockout}$ that are presented until now are independent of the point in the Pareto front that was chosen. The next steps however will depend on the exact point in the Pareto front and have to be repeated if different points are investigated.

For a specific tradeoff p, it is possible to verify for each candidate k if there exists a flux distribution where reaction k can have a zero flux (i.e. it can be knocked out). The corresponding MILPs are shown below (Equation 2.31 to 2.38).

$$\min/\max \quad v_k \tag{2.31}$$

$$\text{s.t.} \quad \sum_{j \in R} s_{ij} v_j = 0 \qquad \forall i \in M \tag{2.32}$$

$$LB_j \leq v_j \leq UB_j \qquad \forall j \in R \tag{2.33}$$

$$v_j = 0 \Longrightarrow t_j = 0 \qquad \forall j \in R_{critical} \tag{2.34}$$

$$v_{target} = v^p_{target} \tag{2.35}$$

$$v_{biomass} = v^p_{biomass} \tag{2.36}$$

$$r = r^p \tag{2.37}$$

$$v_j \in \mathbb{R}; t_j \in \{0, 1\} \tag{2.38}$$

To model the specific tradeoff, the values for the biomass and target productions and the toxicity score are fixed to the values of the tradeoff. For irreversible reactions, it is sufficient to minimize v_k. If the minimum is greater than zero, it has to be active in all flux distributions that are possible

for the fixed values of the tradeoff and it cannot be knocked out. Thus, it can be removed from $R_{knockout}$. For reversible reactions, it needs to be verified that the minimum is smaller or equal to zero and that the maximum is greater or equal to zero. Otherwise, it will be removed from $R_{knockout}$. Again, the removed candidates cannot be chosen by the MILP in Section 2.3.1 but their early removal will reduce the number of binary variables in the MILP.

In the next step, the idea is to compare the FVA values between the wild type flux and the tradeoff flux for all remaining candidates in $R_{knockout}$. Therefore, first, FVA is done for all reactions in $R_{knockout}$ for the wild type. Afterwards, the values for biomass production, target production and the toxicity resistance score are fixed to the values of the tradeoff and the FVA is repeated. Based on the comparison of the possible flux values between wild type and tradeoff flux, the reactions are sorted into groups. The idea is that some reactions might be more likely to have an impact on the resulting flux than others. It might be advantageous to try candidates that are more likely to change the flux first or, when looking for double (or larger) knockouts, to ensure that every knockout set contains at least one of the higher priority candidates.

The first group contains all reactions that must have a non zero flux in the wild type and that must have a zero flux in the tradeoff flux. The flux of these reactions must change to establish the tradeoff flux and therefore, these are potentially very strong candidates and they should have a high priority. The assumption for the next group is similar but less strict. It includes all reactions that must have a non zero flux in the wild type and that can have a zero flux in the tradeoff. All remaining candidates are put into the last group.

2.3.3 Evaluation of knockout sets

After reducing the size of $R_{knockout}$ as much as possible, the MILP in Section 2.3.1 is used to compute different knockout sets that are possible for the chosen point in the Pareto front. The disadvantage of the given formulation is that the proposed knockout sets do not necessarily enforce the desired values for biomass production, target production and the toxicity resistance score but all knockout sets are computed that still allow for these values but that can also result in less optimal ones. Hence, it is necessary to evaluate all computed knockout sets afterwards to identify the ones that lead to the best results.

To evaluate a knockout set P, first, the biomass production is maximized after all reactions in P are knocked out.

$$\max \quad v_{biomass} \tag{2.39}$$

$$\text{s.t.} \quad \sum_{j \in R} s_{ij} v_j = 0 \qquad \forall i \in M \tag{2.40}$$

$$LB_j \leq v_j \leq UB_j \qquad \forall j \in R \tag{2.41}$$

$$v_j = 0 \Longrightarrow t_j = 0 \qquad \forall j \in R_{critical} \tag{2.42}$$

$$x_j = 1 \Longrightarrow v_j = 0 \qquad \forall j \in R_{knockout} \tag{2.43}$$

$$x_j = 1 \qquad \forall j \in P \tag{2.44}$$

$$v_j \in \mathbb{R}; t_j, x_j \in \{0, 1\} \tag{2.45}$$

Then, the computed maximum biomass $v^*_{biomass}$ is fixed and the target production is minimized and

the toxicity resistance score is maximized.

$$\begin{aligned} \min \quad & v_{target} \quad (\text{or} \quad \max \quad r) && (2.46)\\ \text{s.t.} \quad & (2.40) - (2.44) && (2.47)\\ & v_{biomass} = v^*_{biomass} && (2.48)\\ & v_j \in \mathbb{R}; t_j, x_j \in \{0,1\} && (2.49) \end{aligned}$$

Here, the target production is minimized to account for the worst case in which the organism does not prioritize the target production at all. As described in Section 1.3, due to alternative pathways, it is not enough to maximize the target production because this might not be in accordance with the behavior of the organism and no real improvement for the target yield might be achieved which also happened in some cases for OptKnock. The evaluation is based on the assumption that the biomass production is still maximized after introducing knockouts. As described in Section 1.2.4, this is not necessarily the case and it is also possible to use an approach like MOMA that minimizes the distance to the wild type flux. The toxicity resistance score however is maximized because if the organism is under the stress of a toxic environment, it should have a biological incentive to increase its own resistances.

2.3.4 Main drawbacks and other approaches

In the results that are presented in Section 2.5.3, it will become clear that this approach is not viable for larger networks. The required number of knockouts to enforce the desired tradeoff values might be too large and enumerating all possible combinations would take too much time and is not reasonable. After evaluating the computed knockout sets, it becomes clear that almost all of the enumerated knockout sets do not lead to a change in the target production compared to the wild type. The MILP is not restrictive enough. It is possible to enumerate single or double knockouts but already knockout sets of size three are problematic because there are too many combinations that are possible. Hence, it was necessary to change the approach.
A first idea was to address the main problem of the used MILP and render it more restrictive. This could be done by using a max-min problem (similar to RobustKnock) while computing the knockout sets. The problem should maximize the minimum ethanol production. In this way, it should be ensured that the introduced knockouts lead to a change in the target production.
Although the idea itself is simple, the main drawback is that the modified problem is no longer a linear optimization problem and solving it becomes considerably more difficult.

2.4 Second approach - Isolating the active subnetwork

2.4.1 Hyperpaths

Different flux distributions can result in the same values for biomass production, target production and toxicity resistance score. To analyze different feasible flux distributions for one point in the Pareto front, we introduce the notion of a hyperpath for a solution which allows to compare solutions on a topological level. The hyperpath of a flux distribution consists of all reactions that have a non-zero

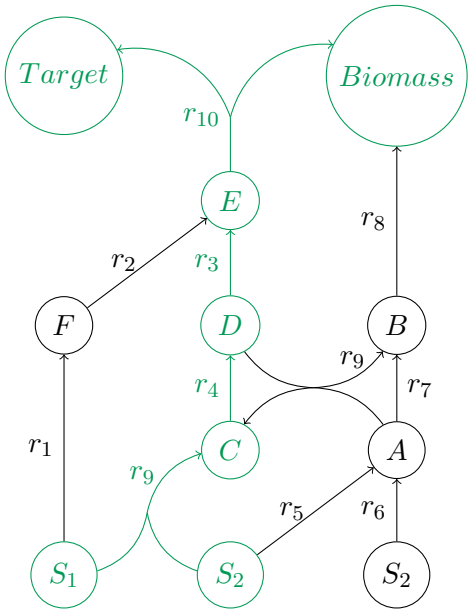


Figure 2.1: *Example for a hyperpath.* All reactions that have a non-zero flux in the solution and participating metabolites are highlighted in green. The active reactions are r_9, r_4, r_3, r_{10}. The only internal metabolite is $Target$. The metabolites $S_1, S_2, C, D, E, Biomass$ are on the border of the hyperpath. Reactions r_2, r_8, r_9 are incoming and reactions r_1, r_5, r_9 are outgoing. Given that reaction r_3 is a critical reaction, using reactions r_9, r_4, r_3 instead of r_1, r_2 is increasing the resistance against the toxic target.

flux value and of all metabolites that are either produced or consumed by these reactions. Reactions having a non-zero flux are called *active reactions*. All other reactions are *inactive reactions*.

Metabolites that are exclusively connected to active reactions are referred to as *internal metabolites*. Metabolites that are connected to at least one active reaction and at least one reaction with a zero flux are metabolites lying on the *border* of the hyperpath.

Furthermore, inactive reactions that are connected to at least one metabolite lying on the border of the hyperpath are called *incoming* or *outgoing* reactions. An inactive reaction is an incoming (outgoing) reaction if it is producing (consuming) a metabolite on the border. It is possible for a reaction to be incoming and outgoing at the same time.

Enumerating different hyperpaths

To enumerate topologically different solutions for one point p of the Pareto front, a MILP is solved. Again, we assume that the system is in steady-state (Equation 2.50), each reaction has lower and upper bounds (Equation 2.51) and the critical reactions are associated with binary variables (Equation 2.52). Additionally, the values for target production, biomass production and the toxicity score are fixed at their optimal values in p (Equation 2.53, 2.54, 2.55). Since we are interested in topologically different solutions, each reaction j is associated with a binary variable a_j that indicates if the corresponding reaction has a non-zero flux and therefore participates in the solution (Equation 2.56). Additionally, smaller hyperpaths (solutions that have less active reactions) are preferable because fewer reactions of the whole network are needed. However, minimizing the number of active reactions means that $\sum_{j \in R} a_j$ has to be minimized which renders the problem computationally more expensive. Hence, only the size of the resulting hyperpath is limited to K (Equation 2.57) which does not

compute the smallest hyperpaths but avoids that large hyperpaths appear as a solution.

$$\text{s.t.} \quad \sum_{j \in R} s_{ij} v_j = 0 \qquad \forall i \in M \tag{2.50}$$

$$LB_j \leq v_j \leq UB_j \qquad \forall j \in R \tag{2.51}$$

$$v_j = 0 \Longrightarrow t_j = 0 \qquad \forall j \in R_{critical} \tag{2.52}$$

$$v_{target} = v^p_{target} \tag{2.53}$$

$$v_{biomass} = v^p_{biomass} \tag{2.54}$$

$$r = r^p \tag{2.55}$$

$$a_j = 0 \Longleftrightarrow v_j = 0 \qquad \forall j \in R \tag{2.56}$$

$$\sum_{j \in R} a_j = K \tag{2.57}$$

$$v_j, \omega_j \in \mathbb{R}; t_j, a_j \in \{0, 1\} \tag{2.58}$$

After one hyperpath H is computed, it has to be excluded as solution by adding another constraint before the problem is solved again:

$$\sum_{j \in H} a_j \leq |H|. \tag{2.59}$$

This process can be repeated until the problem becomes infeasible and therefore no more new topologically different solutions exist or until the desired number of hyperpaths have been enumerated.

Rating a hyperpath

After having enumerated different hyperpaths for one point in the Pareto front, it must be investigated which of these hyperpaths are preferable compared to the others. The idea is to compare what kind of production values a specific hyerpath can achieve. One hyperpath is representing a certain subnetwork which must be cut off from the rest of the metabolic network.

The hyperpath can be isolated by knocking out all incoming and outgoing reactions. Alternatively, it is possible to remove all sources that are not part of the hyperpath and knock out only the outgoing reactions. If all sources of the network are part of the hyperpath, it is enough to knock out all outgoing reactions. Indeed, doing this prevents that the flux can deviate from the hyperpath. Removing all external sources is equal to knocking out all incoming reactions because if all outgoing reactions are removed, incoming reactions can only have a flux if they are fed from an external source that is not part of the subnetwork. An illustration is shown in Figure 2.2.

With the hyperpath isolated from the rest of the network, it is now possible to compute the biomass production, target production and the toxicity score just for the remaining subnetwork. We assume that the organism is still maximizing the biomass production after knocking out all outgoing reactions. This assumption is not accurate in all cases but it is possible to change it if a better estimation is on hand and to modify the MILP accordingly.

The following MILP can be used to compute the maximum biomass production $v^*_{biomass}$ in the hyperpath.

$$\max \quad v_{biomass} \tag{2.60}$$

$$\text{s.t.} \quad \sum_{j \in R} s_{ij} v_j = 0 \qquad \forall i \in M \tag{2.61}$$

$$LB_j \leq v_j \leq UB_j \qquad \forall j \in R \tag{2.62}$$

$$v_j = 0 \Longrightarrow t_j = 0 \qquad \forall j \in R_{critical} \tag{2.63}$$

$$v_j = 0 \qquad \forall j \in R_{incoming} \tag{2.64}$$

$$v_j = 0 \qquad \forall j \in R_{outgoing} \tag{2.65}$$

$$v_j, \omega_j \in \mathbb{R}; t_j \in \{0, 1\} \tag{2.66}$$

As already mentioned, it is possible that $R_{incoming} = \emptyset$.

Afterwards, minimum and maximum target productions and minimum and maximum toxicity resistance scores are calculated when the biomass production is fixed to its optimum $v^*_{biomass}$.

$$\min/\max \quad v_{target} \tag{2.67}$$

$$\text{s.t.} \quad (2.61) - (2.65) \tag{2.68}$$

$$v_{biomass} = v^*_{biomass} \tag{2.69}$$

$$v_j, \omega_j \in \mathbb{R}; t_j \in \{0, 1\} \tag{2.70}$$

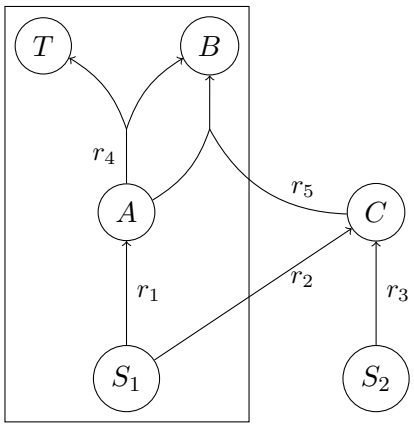


Figure 2.2: *Illustration how incoming reactions can be blocked by removing external sources.* In this small example, the hyperpath is highlighted by the rectangle which means that r_1 and r_4 are the active reactions of the hyperpath. r_5 is an incoming reaction and r_2 is an outgoing reactions. The metabolites S_1 and S_2 are sources of the network. T is an internal metabolite. A, B and S_1 are metabolites on the border of the hyperpath. We are interested in producing the target metabolite T while also maintaining some biomass production (metabolite B). To make sure that the flux cannot deviate from the desired hyperpath flux distribution, all incoming and outgoing reactions have to be knocked out. An alternative way is to only knock out all outgoing reactions and to remove all external sources that remain outside of the hyperpath. After knocking out all outgoing reactions, the only way that the substrates of incoming reactions are present is through external sources of the network. So in this case, by knocking out reaction r_2 and removing the metabolite S_2 from the medium, C cannot be produced and the incoming reaction r_5 cannot take place.

$$\min/\max \quad r \tag{2.71}$$
$$\text{s.t.} \quad (2.61) - (2.65) \tag{2.72}$$
$$v_{biomass} = v^*_{biomass} \tag{2.73}$$
$$v_j, \omega_j \in \mathbb{R}; t_j \in \{0, 1\} \tag{2.74}$$

The result can be used to compare different hyperpaths. A hyperpath is rated high if it has a good target production. A high maximum target production is not sufficient. To account for the worst case, it is more important to have a high minimum target production. Preferably, minimum and maximum target productions are very close to each other. Furthermore, it should be possible to achieve high values for the toxicity resistance score.
Contrary to the target production, for the toxicity resistance score, it is more important to have a high maximum value. Since we assume that the critical reactions might only be active if the need arises, that is when the concentration of the toxic product is high, a minimum score close to zero should not be a criterion to dismiss the corresponding hyperpath. In this case, it is more important to be able to achieve a high maximum toxicity resistance score because then, it is possible that a significant number of critical reactions can be activated if the organism is exposed to the toxic product.
Based on these assumptions, hyperpaths with high minimum target production and high maximum resistance score should be considered for further investigation. Hyperpaths that do not fit these criteria are less efficient.

2.4.2 Identifying smaller knockout sets

For simplification, the next steps assume that all sources of the network are included in the hyperpath. If this is not the case, it is necessary to remove the remaining sources (e.g. by removing them from the media). If there are sources outside of the hyperpath and it is not possible to remove these sources, not only the outgoing but also the incoming reactions have to be considered in the subsequent approaches.
So far, to isolate the hyperpath from the rest of the network, all outgoing reactions are knocked out. However, the number of such reactions can be high and it might not be feasible to knockout such a high amount of reactions *in vivo*. Consequently, it is necessary to reduce the number of reactions that have to be knocked out. Our first ideas were based on minimal cut sets and topological precursors. In the final approach, we decided to use a random exploration approach to find smaller subsets that are still leading to good values for minimum target production and maximum resistance score.

Topological precursor cut sets

The first approach is based on the identification of minimal topological sources for the incoming and outgoing reactions. The concept for topological precursors is introduced in (Cottret et al., 2008; Acuña et al., 2012b).
In (Acuña et al., 2012b), the notion of *forward propagation* is used to define what a precursor is. If M is a set of metabolites, then the forward propagation of M, denoted by $Fwd(M)$, is the set of metabolites that can be produced from M. $Subs(r)$ are all metabolites that are substrates of reaction r. Likewise, $Prod(r)$ contains all metabolites that are products of reaction r. $Reac(M)$ is the set of reactions that can take place if all the metabolites in M are present in the network because

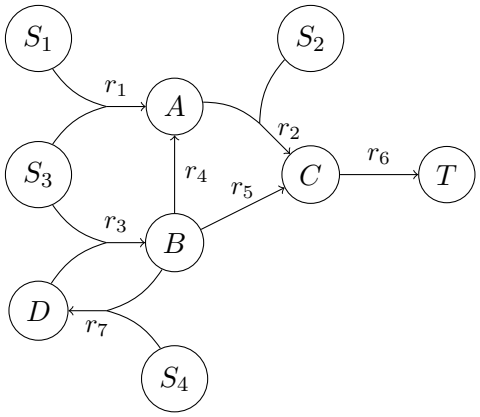


Figure 2.3: *Topological precursors*. The figure was adapted from (Acuña et al., 2012b). The metabolites S_1, S_2, S_3 and S_4 are the sources of the metabolic network. Metabolite T is the target node whose topological sources should be identified. For example, if $M_0 = \{S_1, S_2, S_3\}$, it is possible to produce A because both substrates of reaction r_1 are present. Therefore, $M_1 = \{S_1, S_2, S_3, A\}$. Now, both substrates of reaction r_2 are available and C can be produced which means that subsequently also T can be produced. Afterwards, no more changes can be made and a fixed point is reached. Hence, $Fwd(\{S_1, S_2, S_3\}) = \{S_1, S_2, S_3, A, C, T\}$.

all substrates of these reactions are available which means that $Reac(M) = \{r \in R | Subs(r) \subseteq M\}$. Furthermore, if R is a set of reactions, the authors define two more sets: $Subs(R) = \cup_{r \in R} Subs(r)$ and $Prod(R) = \cup_{r \in R} Prod(r)$. Based on these notations, $Fwd(M)$ can be computed by the recursion $M_{i+1} = M \cup Prod(Reac(M_i))$. The recursion starts from $M_0 = M$ and finishes when a fixed point is reached which means that no more new metabolites can be added. An example is shown in Figure 2.3. Moreover, as proposed in (Cottret et al., 2008), the authors include other metabolites in their model which are called *internal supply* and are always available. $Fwd_Z(M)$ is the forward propagation of M given that the set of metabolites Z is an internal supply and the recursion can be reformulated as $M_{i+1} = M \cup Prod(Reac(M_i \cup Z))$.

The set of metabolites S denotes the set of all source metabolites of the metabolic network which means that they can be available as an external supply. Based on the idea of forward propagation and internal supply, a set of source metabolites $X \subseteq S$ is a *precursor set* of the set of target metabolites T if $Fwd_Z(X)$ contains both T and also Z. It is necessary that Z is included to make sure that it can be reproduced.

Since we are not interested in producing a set of target metabolites but our goal is to cut off all incoming and outgoing reactions, the most important idea is the *precursor cut sets*. The authors of (Cottret et al., 2008) define that a set of sources $X \subseteq S$ is a cut set of the set of target metabolites T if and only if the set $S \setminus X$ is not a precursor set of T. Hence, the production of T is prevented by removing the the elements in X as source metabolites. To ensure that the production of all metabolites in T is cut off, it is necessary to introduce a special target metabolite t^*. For each metabolite $t_i \in T$, a new reaction r_i is added with $Subs(r_i) = \{t_i\}$ and $Prod(r_i) = \{t^*\}$. Afterwards, the new set of target metabolites is $T' = \{t^*\}$.

Since we are interested in cutting off reactions, we have to modify the network because the idea of precursor cut sets is used to prevent the production of metabolites. One possibility is to add a new metabolite for each incoming and outgoing reaction that splits the reactions in two. For each incoming or outgoing reaction r_i, a new metabolite m_i is created and r_i is split into r_i' and r_i'' whereas $Subs(r_i') = Subs(r_i)$, $Prod(r_i') = \{m_i\}$ and $Subs(r_i'') = \{m_i\}$, $Prod(r_i') = Prod(r_i)$.

Furthermore, as explained before, a special target metabolite t^* has to be created and one additional reaction for each m_i must be added that consumes m_i and produces t^*.
To identify precursor cut sets for the outgoing reactions, the directions for all reactions of the network have to be reversed. This means that all metabolites that were sources of the network become sinks of the network and likewise, all metabolites that were previously sinks are now sources of the metabolic network. However, this implies that the precursor cut sets for incoming and outgoing reactions have to be identified separately. Furthermore, only metabolites that are not part of the hyperpath can be considered as sources because otherwise they are needed to obtain the biomass and target production. It might be preferable to cut out the identified hyperpath and to only take into account the remaining part of the network when looking for precursor cut sets.
As presented in (Acuña et al., 2012b), a minimal precursor cut set can be found quite easily. Starting from $X' = \emptyset$, sources are added if the target cannot be produced. As a result, the set $X = S \setminus X'$ is a minimal precursor cut set. Identifying minimum precursor cut sets is however considerably more difficult but also more important because we are interested in the smallest changes that are necessary to achieve our goal. The authors in (Cottret et al., 2008) present an approach to enumerate all precursor sets that uses a kind of depth-first search on the hyperpath that traverses the reactions in their opposite direction to explore paths from the target to the sources. To avoid fake solutions, some cycles have to be removed from the metabolic network in a preprocessing step because they can lead to the production of certain metabolites without any external sources. During the preprocessing, these undesired cycles are removed by deleting some reactions. It is a heuristic and the preprocessed network is not always the same which also influences the minimal precursor sets that are computed afterwards.
In practice, we abandoned the idea to use topological precursors cut sets as a solution to our problem due to difficulties that were linked to the removal of the aforementioned cycles. Additionally, it can only identify sources as cut points and we are also interested in proposing reaction knockouts as intervention strategy. Furthermore, since this approach is based on the topology of the network, it might have been preferable to also remove small metabolites and co-factors from the network to simplify the network structure. It can however be difficult to automatically identify and remove co-factors in a way that the stoichiometry of the network is kept intact. The stoichiometry is not needed to identify minimal precursor sets but it might be important for subsequent analyses since in general, our method should include the stoichiometry of the network. It might be possible to circumvent some of the difficulties by using the idea of *stoichiometric precursor sets* (Andrade et al., 2016) but so far, we did not explore this concept further because we assumed that the computation might be more difficult and there might be other approaches that are more accessible for our problem.

Minimal cut sets

Next, we were interested in applying the concept of minimal cut sets to our problem. In this case, we want to find the smallest MCSs that block all outgoing reactions for a hyperpath. Therefore, all outgoing reactions are set as the target reactions. Additionally, it is not possible to knockout the active reactions of the hyperpath.
The MSCEnumerator presented in (von Kamp and Klamt, 2014) enumerates MCSs and cMCSs. An implementation is available in CellNetAnalyzer (Klamt and von Kamp, 2011; Klamt et al., 2007; von Kamp et al., 2017). CellNetAnalyzer uses MATLAB.
In (von Kamp and Klamt, 2014), MSCEnumerator is used to compute the smallest MCSs in

a genome-scale metabolic network. However, as shown in their results, it can be computationally expensive for larger networks to compute MCSs. In our case, this approach was not suitable in practice. We assume that one of the main problems is that a larger number of reactions has to be knocked out to block all of the outgoing reactions.

Random exploration

In this last approach, we propose a random exploration that identifies smaller subsets of the outgoing reactions that are still leading to good values for minimum target production and maximum resistance score. The starting point for this approach are all outgoing reactions for a hyperpath that was chosen based on its promising values. The outgoing reactions are forming the reaction pool P_0. The maximum biomass for this hyperpath is used as reference value for the biomass production $v^{ref}_{biomass}$.

Step 1: The reaction pool is split into two different subsets. For each reaction r in the reaction pool P_0, all reactions in P_0 are knocked out except reaction r. Afterwards, the maximum biomass is computed. If the maximum biomass remains the same as $v^{ref}_{biomass}$, not knocking out only reaction r does not change the result and reaction r will be part of subset 2. If the maximum biomass is not the same as $v^{ref}_{biomass}$, not knocking out reaction r does lead to a change and reaction r will be part of the subset 1. This procedure is repeated for all reactions in P_0. Consequently, subset 1 contains all reactions that have to be knocked out at all time because otherwise it is not possible to obtain the same production values as in 2.4.1. However, subset 2 contains reactions that do not have to be necessarily knocked out and it might be possible to remove several of them from the knockout set without changing the production values.

Step 2: A new knockout P_1 set is generated. All reactions of subset 1 must be in P_1. One by one, reactions from subset 2 are randomly drawn and added to P_1. Every time a new reaction is added, the maximum biomass production is computed given that all reactions in P_1 are knocked out. If the maximum biomass production is not equal to $v^{ref}_{biomass}$, the next reaction is drawn and added to P_1. If the maximum biomass production is equal to $v^{ref}_{biomass}$, no more reactions are added to P_1.

Step 3: Step 2 is repeated n times to compute $P_{11}, P_{12}, ..., P_{1n}$. Then, the smallest (with regard to the number of reactions it contains) valid knockout set P_{1x} is chosen. If P_{1x} and P_0 are identical, no smaller knockout set can be chosen and the procedure is stopped. Otherwise, P_{1x} is set to be the new P_0 and Steps 2 and 3 are repeated.

It is not necessary to recompute the subsets 1 and 2. Subset 1 always remains the same and the new subset 2 will be a subset of the previous subset 2 containing only the reactions that are present in P_{1x}.
Furthermore, it is important to observe that this approach is based on using the biomass production as main indicator because in our case study, the target production depends heavily on the obtainable biomass production. Therefore, it is not necessary to also compare the target production after each added reaction which saves computation time. In different cases, it might be advantageous to chose another main indicator or to compare both values after each newly drawn reaction
The exploration is random and greedy and it is not guaranteed to obtain the smallest possible knockout set. The probability to achieve a small knockout set can be increased by choosing a high n

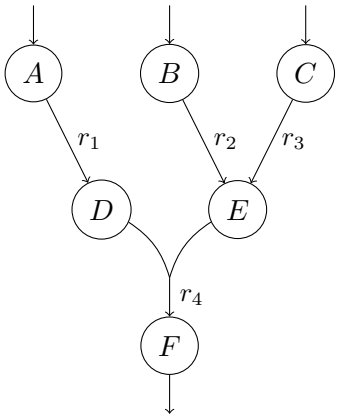


Figure 2.4: *Disadvantage of the first version of the random exploration.* In this example, reactions r_1, r_2 and r_3 are outgoing reactions and the metabolites A, B and C are metabolites that lie on the border of the hyperpath. The metabolites D, E and F and reaction r_4 are outside of the hyperpath. When drawing reactions for the next knockout sets, if r_2 and r_3 are getting picked before r_1, the limit for the biomass production might already be reached before r_1 is added because by knocking out r_2 and r_3, the flux cannot deviate from the hyperpaths via r_4. If this knockout set gets chosen for the next iteration, it is not possible anymore that r_1 gets picked anymore instead of r_2 and r_3 which would reduce the size of the knockout set.

and by repeating the whole exploration several times and comparing the results. Since the approach is greedy, always the smallest knockout set is chosen for the next iteration. It is likely that there are cases where choosing a different knockout set will lead to a better final result because it contains a better combination of reactions. An example is shown in Figure 2.4.

After applying this random exploration to the case study that will be presented in the results, it becomes clear that this described random exploration is too straightforward and it would be better to modify it in order to make it less greedy. The main idea is to not always take the smallest knockout set as next candidate but to introduce probabilities that guide the choice of the next candidate and aid in better exploring possible solutions. Still, smaller solutions should be prioritized.

Steps 1 and 2 of the modified version remain unchanged. Step 3 is modified in the following way: Instead of repeating Step 2 n times, it is only done once to compute one new knockout set P_1. All already computed knockout sets can be potential candidates for the next iteration. All computed knockout sets are unique. This means that if the newly computed knockout set was already computed previously, it is not added to the pool of all knockout sets. All computed knockout sets are sorted by size and split into buckets. In the following explanation, s is the size of the currently smallest knockout set. Furthermore, B is the size that is chosen for the buckets. This means that the first bucket b_0 contains all knockout sets with a size in between s and $s + B - 1$ and the bucket b_n contains all knockout sets with a size in between $s + n \cdot B$ and $s + (n + 1) \cdot B - 1$. However, there is a further adjustment for the first bucket. It contains all knockout sets of size s and additionally, the same amount of larger solutions. For example, if there are 50 knockout sets of size s, it also contains the 50 next bigger knockout sets. Furthermore, it should have a minimum size F. If there are less than $F/2$ knockout sets of size s, it will be filled with the next bigger knockout sets until it contains a total of F knockout sets. If in the current iteration, less than F knockout sets have been computed in total, the first bucket contains all knockout sets. Consequently, the limits for the subsequent buckets have to be adjusted. Given that the largest size that is still contained in bucket b_0 is s^*, the bucket b_n contains all knockout sets with a size in between $(s^* + 1) + n \cdot B$ and

$(s^* + 1) + (n + 1) \cdot B - 1$.

To choose the next candidate, first a bucket is chosen and afterwards a knockout set in the chosen bucket is picked. Starting with b_0, each bucket has a probability p_1 to be accepted. If the bucket is rejected, the next bucket is tried. If a bucket is accepted, a knockout set from this bucket is drawn. In this case, a uniform distribution is used which means that all knockout sets inside one bucket have the same probability.

After a knockout set is chosen, steps 2 and 3 are repeated. After each iteration, it is verified if a smaller knockout set was found. If after a specified amount of iterations, no smaller knockout set was computed, the random exploration stops.

The main idea behind the buckets is to make it easier that larger knockout sets get chosen as candidates but at the same time knockout sets of same or similar size have the same probability of getting picked. Additionally, since all computed knockout sets can be potentially drawn in the next iteration, this approach is less greedy than the first version where it was not possible to pick a previous candidate in a later iteration.

Implementation

Since in the second part, only single-objective optimization problems are solved, PolySCIP is no longer needed and all linear programs are solved with CPLEX 12.71 (IBM, 2016). They are implemented in C++ and the metabolic network was modeled using the library metnetlib which is available on `https://gitlab.inria.fr/erable/kirikomics/metnetlib`.

2.5 Results

To apply our approach, we used the production of ethanol in yeast since ethanol is an interesting target chemical due to its use as biofuel but it is also toxic to yeast.

All computations are done using the yeast 5.01 model (Heavner et al., 2012). The network model contains a total of 2109 reactions and 2759 metabolites. After the removal of blocked reactions which are all reactions that can only have a zero flux for the given lower and upper bounds (see Section 1.2.4) and the subsequent removal of isolated metabolites, the model contains 1165 reactions and 914 metabolites.

We will first present the observations on the computed Pareto front. Afterwards, the results for the first approach are shortly summarized before showing the results for the second approach that is based on the idea of the hyperpaths.

2.5.1 Critical reactions

To identify reactions that might be involved in increasing the resistance of yeast against ethanol, the growth of single gene knockout yeast strains under ethanol stress was compared to the wild type growth under the same conditions. If the biomass production was decreased in the modified strain, it is assumed that the knocked out gene is implicated in developing a resistance against ethanol. Genes that are known to participate in growth mechanisms were not tested since knocking these genes out would likely lead to a reduced growth but not because they are influencing the resistance against ethanol.

Depending on the amount that the growth decreased compared to the wild type, the genes were separated into two groups. Group 1 contains genes whose knock out resulted in a biomass production between 50% and 70% of the wild type biomass production. Knocking out the genes in group 2 led to less than 50% of the wild type biomass production. Hence, genes in group 2 might be playing a more significant role in building a resistance against ethanol than genes in group 1.

Afterwards, the gene identifiers were used to link these genes to reactions in the network file. Not all genes can be associated with reactions. It is also possible to have several genes linked with the same reaction and one gene connected with several reactions in the network file.

The reactions that can be connected to the genes in groups 1 and 2 are the critical reactions that are used to compute the toxicity resistance score. A total of 61 critical reactions could be identified. After removing blocked reactions, the list could be reduced to 41 remaining critical reactions. The reduced list can be found in the Supplementary Table A.3.

2.5.2 Computation of the Pareto front

The critical reactions were used to compute different tradeoffs between biomass production, ethanol production and the score measuring the potential resistance against ethanol. The multi-objective optimization problem was solved using PolySCIP version 2.0 (Borndörfer et al., 2016). The critical reactions in group 1 have weight 1.0 when calculating the resistance score, reactions in group 2 have a weight of 2.0. Only the extreme points of the Pareto front were computed which allows to gain a broad overview of the whole Pareto front (see Figure 2.5). Some extreme points have very similar ethanol and biomass production values but differ considerably in their ethanol resistance score. In this case, a point with a higher ethanol resistance is preferable.

A list of computed extreme points and their identification number can be found in the Supplementary Table A.2. A reduced version that contains only the extreme points that will be referred to in the text is shown in 2.1.

Id	Biomass	Ethanol	Toxicity
1	0.131931	17.6169	37.0072
2	0.138936	17.6120	36.0071
3	0.145084	17.5654	35.0066
4	0.157094	17.6384	26.0035
6	0.165196	17.5216	26.0035
9	0.180441	16.6031	37.0120
11	0.187788	16.6035	36.0115
13	0.185612	15.0541	43.0187
19	0.103374	13.4770	52.0291
47	0.347631	10.2221	43.0447

Table 2.1: *Computed tradeoffs between biomass production, ethanol production and toxicity score - short version.* Only the extreme points in the Pareto front were computed. Lower bounds for biomass production was set to 0.1, lower bounds for ethanol production to 10. The list is sorted by descending ethanol production. The list presented here contains only those tradeoffs that are explicitly mentioned in the text. For a full list, we refer the reader to the Supplementary Table A.2.

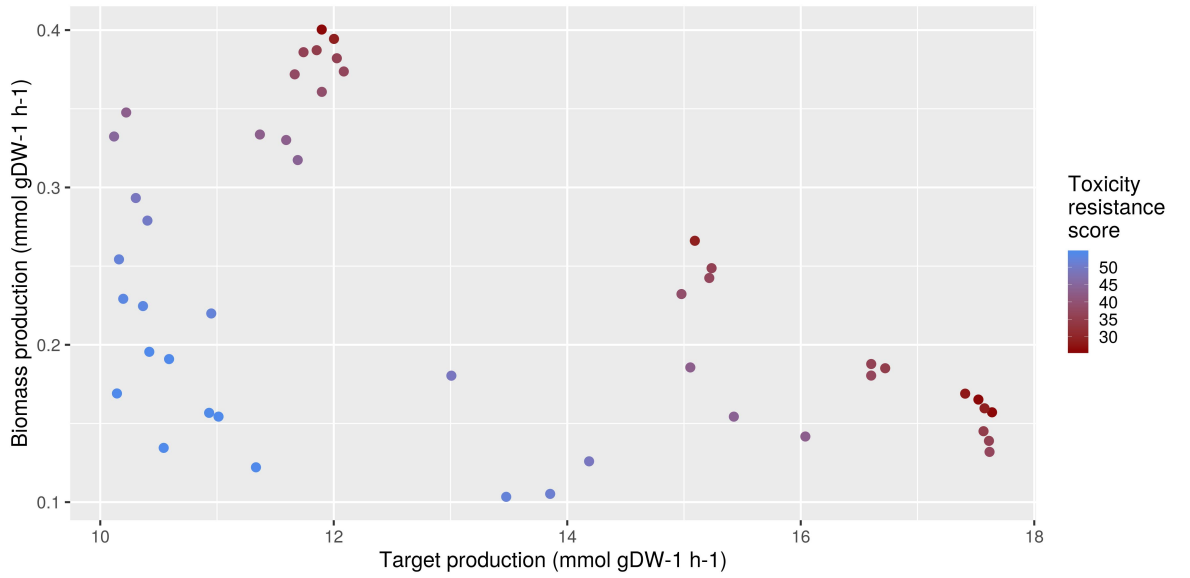


Figure 2.5: *Computed tradeoffs between biomass production, target production and toxicity resistance score.* Only the extreme points were computed. The lower bound for the biomass production was set to 0.1 and the lower bound for ethanol production was set to 10. For a clearer representation, the third dimension, the toxicity resistance score, is displayed by a color scale. Red points have a low toxicity resistance score, blue points have a higher score. It is interesting to see that there are some points that have very similar ethanol and biomass production values but differ in the resistance score. It shows that it might be possible to gain a considerably better resistance against ethanol by choosing slightly worse production values for ethanol and biomass.

2.5.3 First approach

An important remark for the results for the first approach is that they were obtained using the original network model. The preprocessing step was introduced at a later point in time when the idea based on the identification of the hyperpaths was developed.

As described in Section 2.3.2, the number of candidates for knockouts was reduced. After eliminating all reactions without a gene association and all transport and exchange reactions, 660 reactions remained in the candidate pool. Afterwards, for the remaining candidates, the coupled groups were identified and only one representative for each group was kept in the pool which reduced the candidate pool to 492 reactions. Subsequently, biomass and target production in the wild type were verified for single knockout of the remaining candidates. After discarding all candidates whose knockout resulted in a zero biomass or target production, 435 reactions were left in the candidate pool. These steps are the same for all tradeoffs and do not have to be repeated.

The last step is shown on the example of the tradeoff 1 (as presented in Table 2.1). Having verified if a candidate reaction can have a zero flux in at least one tradeoff flux distribution, 65 more reactions could be removed from the pool. Finally, the remaining 370 candidates were sorted into the three groups. The first group was empty (must have non zero flux in the wild type and zero flux in the tradeoff). The second group contained 28 reactions (must have non zero flux in the wild type and can have zero flux in the tradeoff). Therefore, the third group included the last 342 reactions.

Single knockouts were enumerated for all extreme points. These are however never sufficient to enforce the desired tradeoff flux. When we computed the double knockouts, we realized that enumerating all solutions is not practicable because there are only a few double knockouts that actually

make the optimization problem infeasible. Consequently, almost all pairs of candidates were enumerated as double knockouts. After each iteration, for the newly found knockout set, a new constraint was added which meant that the time it took to enumerate the next solution slowly increased.
To reduce the number of enumerated pairs, we used the groups to prioritize certain candidates. We enumerated pairs that needed to include at least one of the 28 reactions from group 2 (i.e. they could appear together in a pair). This led to the computation of around 10000 double knockouts. Hence, each of the 28 reactions was paired with almost all of the other remaining 369 candidates (groups 2 and 3). This showed again that the main problem is that almost all pairs of candidates are feasible and do not disrupt the tradeoff flux. The underlying MILP is not restrictive enough. It might be faster just to try all pairs and verify the resulting biomass and target productions because it would avoid to solve the MILP for each new pair which renders this approach impracticable.

2.5.4 Second approach

In the following, the results for the second approach that is based on the hyperpaths are shown. Before applying the random exploration to identify knockout sets, the similarities and differences between the hyperpaths and tradeoffs are investigated.

Active reactions

In the subsequent computations of the hyperpaths, the number of active reactions was always limited to 500 reactions to prevent that too large hyperpaths can be enumerated.
To gain an overview of different flux distributions that are possible in the network, for 47 different tradeoffs, one hyperpath was computed and the active reactions were compared (see Figure 2.6). More than 300 active reactions were present in all hyperpaths. Since ethanol and biomass are produced in all of these 47 tradeoffs, reactions that are specific to these pathways have to participate in the solution. Although their values for biomass and ethanol production can differ significantly, topologically these solutions have many reactions in common. Furthermore, there are more than 100 reactions that appear in five or less hyperpaths. It is already an indicator that there might be several reactions that can be easily added or removed from a hyperpath without changing the resulting flux significantly.
This impression is confirmed when comparing different hyperpaths for one tradeoff. The results for 1000 different hyperpaths for tradeoff 1 are shown in Figure 2.7. There are more than 200 reactions that are participating only in less than 10 hyperpaths. Hence, they do not seem to have a big impact on the solution and are probably only part of the hyperpath because it has to differ from previously computed hyperpaths that are then excluded by constraints. Besides, more than 350 reactions appear in all solutions.

Comparison of 10 hyperpaths for each of the 47 tradeoffs

Comparing the computed tradeoffs, some seem to be preferable because they have a higher ethanol production value and a better resistance score than others. However, it is necessary to compare many hyperpaths for each tradeoff because knocking out all outgoing reactions of a hyperpath does usually not enforce the exact values of the tradeoff. Since many reactions are essential for biomass production and ethanol production, the remaining subnetwork is still quite large and thereby also contains a lot of variability.

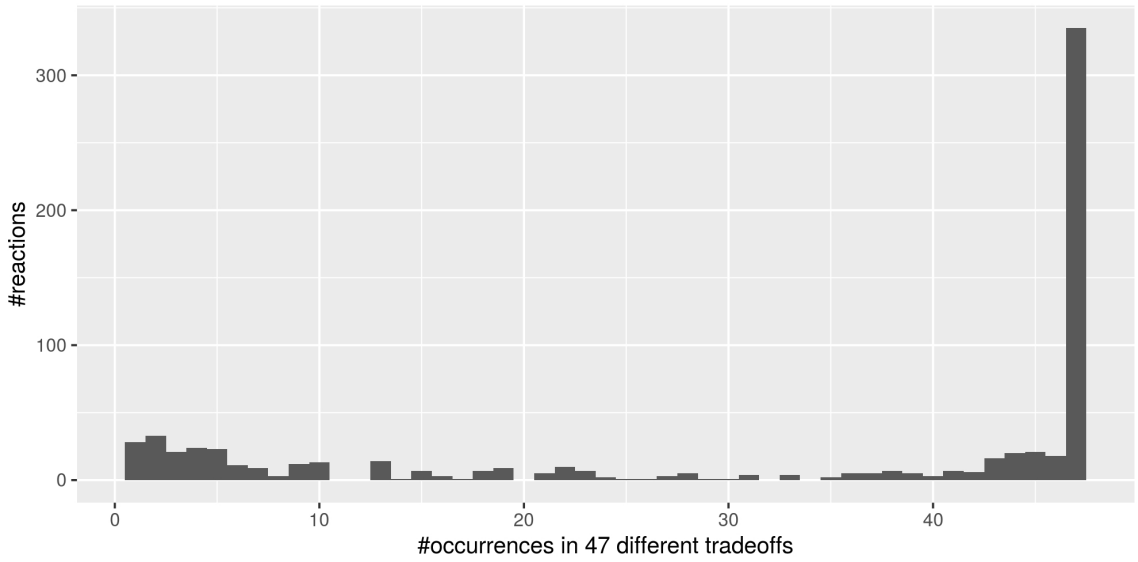


Figure 2.6: *Histogram of active reactions in 47 different tradeoffs.* For 47 extreme points, one hyperpath was computed for each. The figure shows in how many hyperpaths a reaction was active among these 47. A large part of the active reactions was present in all of them.

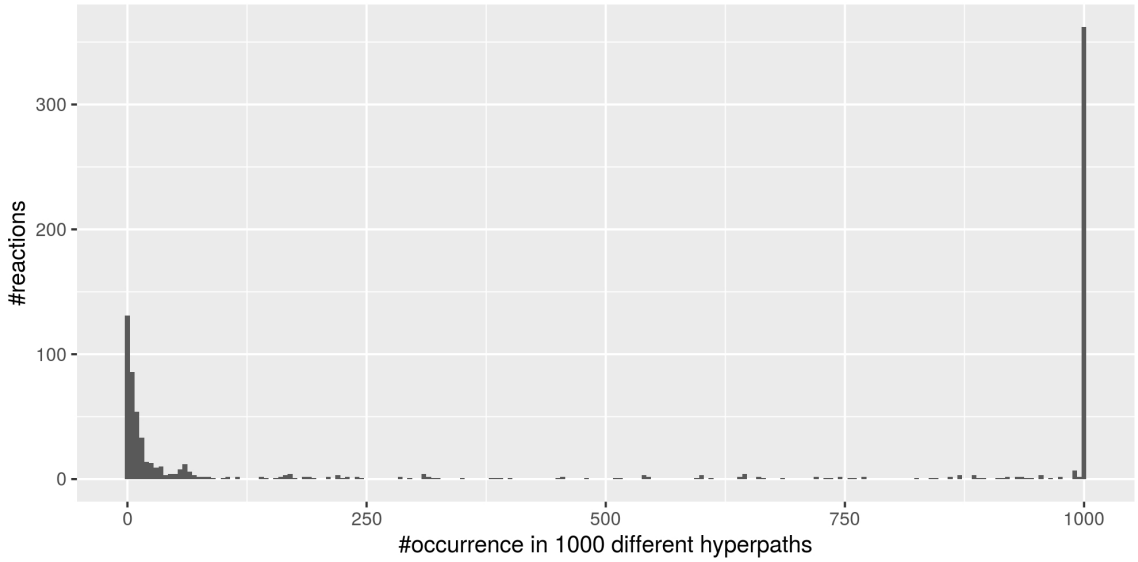


Figure 2.7: *Histogram of active reactions in 1000 different hyperpaths for tradeoff 1.* In total, 884 reactions were active in at least one of these 1000 hyperpaths. Most of them participate either in all of the 1000 hyperpaths or they appear in just a very few cases. The binwidth for this histogram was set to five which means that, for example, the first bar counts all reactions that appeared exactly zero to four times in all 1000 hyperpaths. However, none of the represented reactions appeared zero times because reactions that were not part of any hyperpaths were not considered for this plot.

Before applying the random exploration approach to compute knockout sets, an adequate hyperpath has to be chosen. Thus, in a first step, for each tradeoff, 10 different hyperpaths were computed. All hyperpaths are evaluated by their ethanol production and resistance score when the biomass production is maximized as explained in 2.4.1.

First of all, it can be observed that ethanol production is dependent on the maximum biomass production. A high ethanol production can only be achieved if the maximum biomass production is low. If a high maximum biomass production is possible, it prevents a profitable ethanol production (see Figure 2.8). This happens because both biomass and product are directly dependent on the same carbon sources, which means that if all carbon molecules uptaken from the substrate are used for biomass production, there is nothing left to produce ethanol and vice versa. This fact can be exploited during the random exploration.

Moreover, different hyperpaths can lead to the same production values (see Table 2.2 for some selected results). For some tradeoffs, all ten hyperpaths have the same results (e.g. tradeoffs 9 and 11). In most of the other cases, the results for all ten hyperpaths are at least very similar (e.g. tradeoffs 2 and 4). Problematically, there are also instances where the results between the different hyperpaths can differ considerably. For example, the first nine hyperpaths for tradeoff 1 have very low maximum toxicity resistance scores and the achievable ethanol production is only mediocre. However, the tenth computed hyperpath differs in an important way and is superior in all values. Even the biomass production is slightly higher. This shows how important the choice of a specific hyperpath is. Enumerating ten hyperpaths for a tradeoff is not sufficient to ensure that favorable hyperpaths are available.

To show how the result is influenced if the toxicity resistance score is not taken into account, for each tradeoff, 10 hyperpaths were computed where only biomass production and target production were fixed in the optimization. The constraint (Equation 2.55) that is fixing the toxicity resistance to its optimal value from the tradeoff is removed from the optimization problem. Ideally, when including the toxicity resistance score in the optimization problem, the obtained hyperpaths should have a better toxicity resistance score than when it is omitted. Not fixing the toxicity score will not prevent the solver to choose a flux distribution that results in a high toxicity resistance score. However, it also does not encourage it and therefore, the probability for choosing a hyperpath with a lower toxicity should be higher.

The maximum toxicity resistance scores for both cases are displayed in Figures 2.9 and 2.10. Comparing the results, globally, higher maximum resistance can be achieved when the score is included in the optimization problem. Omitting it, only two of the resulting hyperpaths (tradeoff 19) have a maximum toxicity score over 20 as opposed to all ten when it is included. By including the score when computing the hyperpath, this can be achieved more consistently.

However, there are a few cases (e.g. tradeoff 4) where the ten hyperpaths computed without taking the toxicity resistance score into account have better maximum scores. A higher number of hyperpaths might be necessary to investigate if for this tradeoff a better toxicity resistance score can be reached when omitting the score from the optimization problem. It is possible that by enumerating more hyperpaths, other solutions are found where the score is at least as good as the highest one that can be obtained when the toxicity score is omitted from the optimization problem. Indeed, when enumerating 1000 hyperpaths for tradeoff 4, it is possible to obtain some hyperpaths that have a better toxicity score than any of the hyperpaths that are computed when omitting the score.

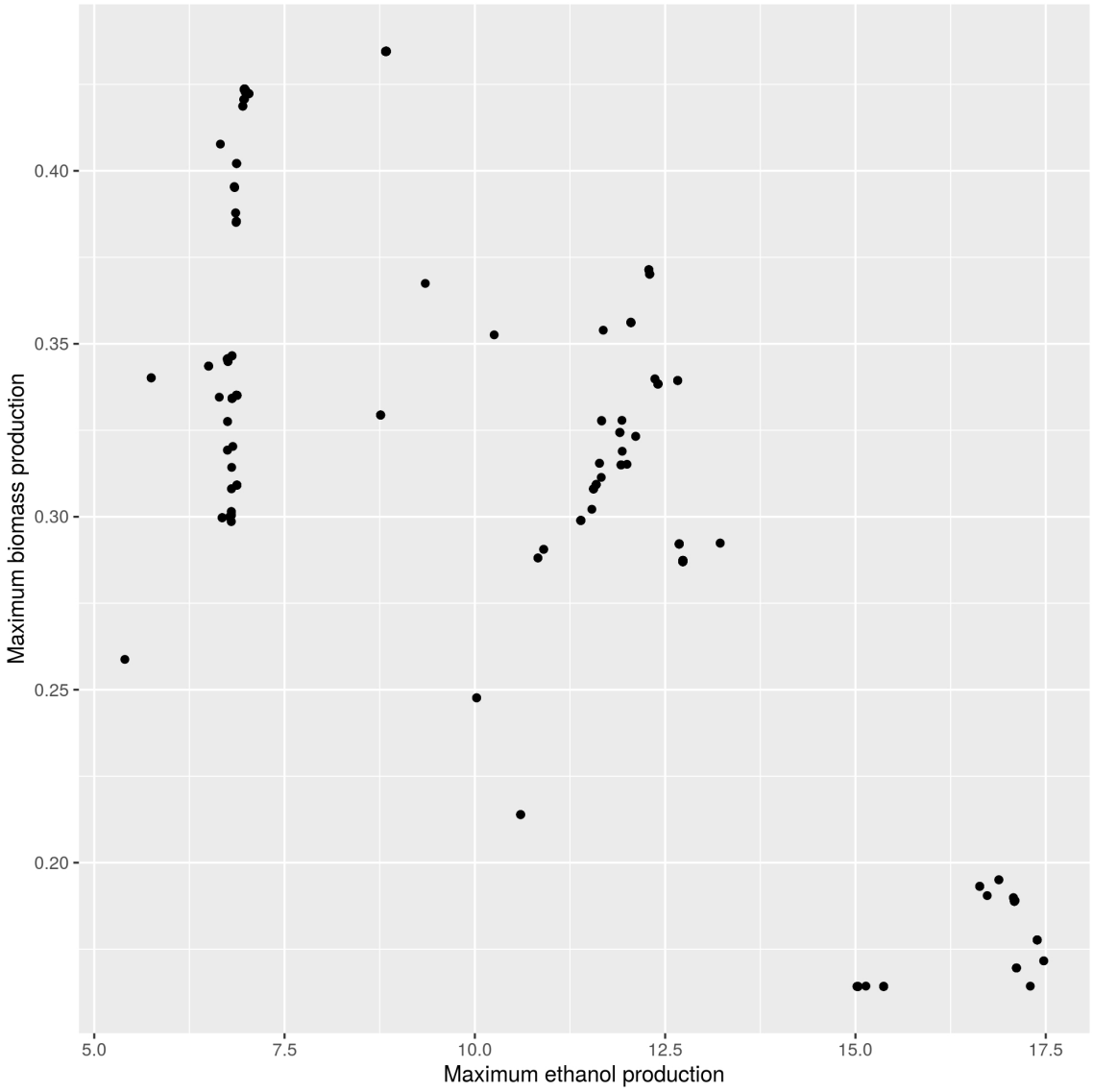


Figure 2.8: *Dependency of the maximum ethanol production on the maximum biomass production.* For 47 tradeoffs with 10 different hyperpaths each, the resulting maximum biomass production is plotted against the maximum ethanol production that is possible. An ethanol production of more than 14 can only be achieved if the maximum biomass production is lower than 0.2.

Id	Max. biomass	Min. ethanol	Max. ethanol	Max. toxicity score
1	0.1642	14.677	15.370	13.003
	0.1642	14.677	15.370	13.003
	0.1642	14.677	15.370	13.003
	0.1642	14.677	15.370	13.003
	0.1642	14.677	15.370	13.003
	0.1642	14.677	15.370	13.003
	0.1642	14.677	15.370	13.003
	0.1642	14.677	15.021	13.003
	0.1642	14.677	15.021	13.003
	0.1643	17.101	17.297	21.008
2	0.1642	14.689	15.033	13.003
	0.1642	14.689	15.033	13.003
	0.1642	14.689	15.033	13.003
	0.1642	14.689	15.033	13.003
	0.1642	14.689	15.033	13.003
	0.1642	14.689	15.033	13.003
	0.1642	14.689	15.033	13.003
	0.1642	14.689	15.033	13.003
	0.1643	14.788	15.137	13.003
	0.1642	14.676	15.020	13.003
4	0.1716	17.433	17.473	11.002
	0.1716	17.433	17.473	11.002
	0.1777	17.348	17.388	10.002
	0.1777	17.348	17.388	10.002
	0.1777	17.348	17.388	10.002
	0.1777	17.348	17.388	10.002
	0.1777	17.348	17.388	10.002
	0.1777	17.348	17.388	10.002
	0.1777	17.348	17.388	10.002
	0.1777	17.348	17.388	10.002
9	0.2870	12.716	12.732	20.041
	0.2870	12.716	12.732	20.041
	0.2870	12.716	12.732	20.041
	0.2870	12.716	12.732	20.041
	0.2870	12.716	12.732	20.041
	0.2870	12.716	12.732	20.041
	0.2870	12.716	12.732	20.041
	0.2870	12.716	12.732	20.041
	0.2870	12.716	12.732	20.041
	0.2870	12.716	12.732	20.041
11	0.2874	12.717	12.733	22.041
	0.2874	12.717	12.733	22.041
	0.2874	12.717	12.733	22.041
	0.2874	12.717	12.733	22.041
	0.2874	12.717	12.733	22.041
	0.2874	12.717	12.733	22.041
	0.2874	12.717	12.733	22.041
	0.2874	12.717	12.733	22.041
	0.2874	12.717	12.733	22.041
	0.2874	12.717	12.733	22.041

Table 2.2: *Results for ten different hyperpaths for certain tradeoffs.* For tradeoffs 1, 2, 4, 9 and 11 the production values are shown for ten different hyperpaths each.

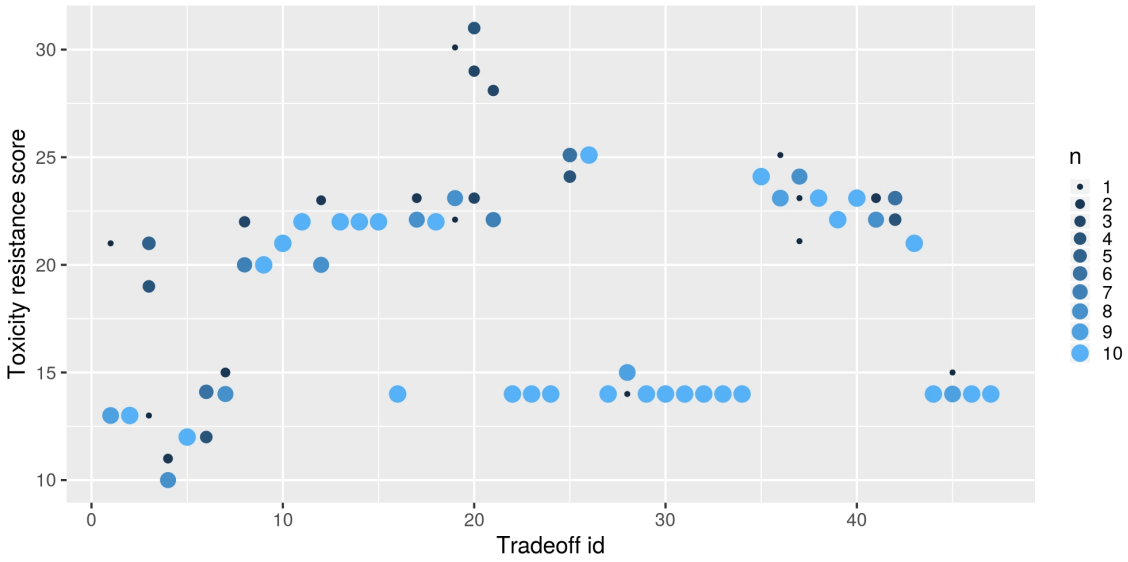


Figure 2.9: *Maximum toxicity resistance score for hyperpaths where the score was included in the optimization problem.* For each of the 47 tradeoffs, the maximum possible toxicity resistance score for ten different hyperpaths is plotted. To group similar results together, the score was rounded to the first digit. For some tradeoffs, all ten hyperpaths lead to the same or very similar toxicity resistance scores. For others (e.g. tradeoff 1 or 20), the results differ significantly.

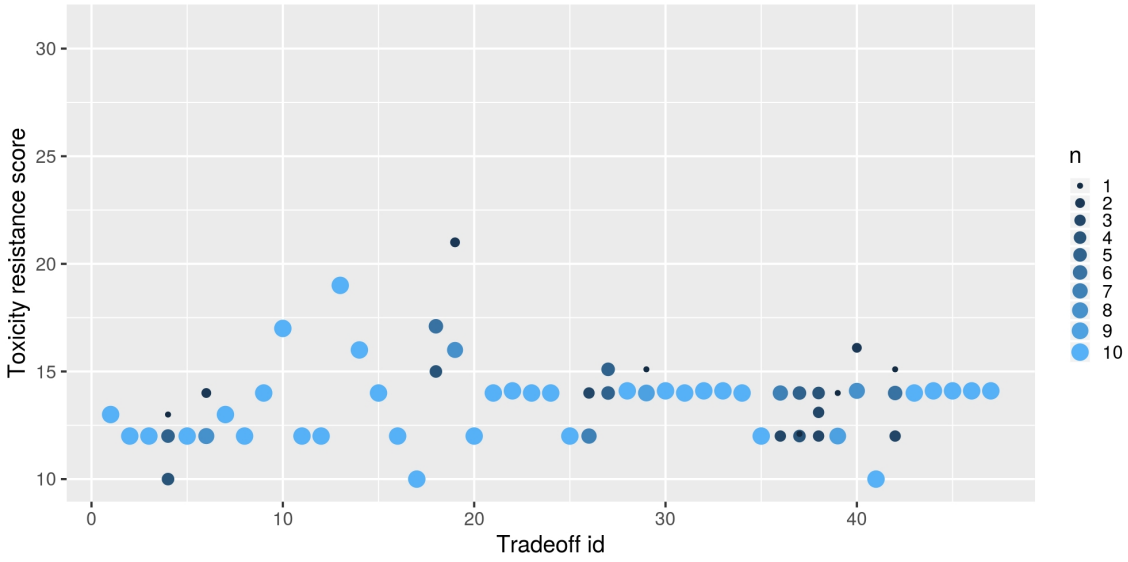


Figure 2.10: *Maximum toxicity resistance score for hyperpaths where the score was omitted in the optimization problem.* For each of the 47 tradeoffs, the maximum possible toxicity resistance score for ten different hyperpaths is plotted. These hyperpaths were computed without fixing the toxicity resistance score in the optimization problem. To group similar results together, the score was rounded to the first digit. Comparing the results to Figure 2.9, globally the scores are lower. Only two hyperpaths (tradeoff 19) have maximum scores that are over 20.

Comparison of 1000 hyperpaths for chosen tradeoffs

Therefore, to get a more detailed view, 1000 different hyperpaths are enumerated for specific tradeoffs. Tradeoff 1 was chosen to investigate the fact that its tenth hyperpath differed significantly from the other nine and because it showed promising values for ethanol production. Additionally, the tenth hyperpath also had a high maximum toxicity score. Tradeoff 3 also had high values for ethanol production and toxicity score. One of the main objectives is to find a hyperpath with a very high ethanol production value and a high toxicity resistance score. Hence, both tradeoff 1 and tradeoff 3 needed to be explored further. Tradeoff 6 was chosen because the ten hyperpaths were inferior to the ones from tradeoffs 1 and 3 but we wished to explore whether enumerating a higher number might lead to better results. Tradeoff 13 was chosen for the same reason. Tradeoff 47 was selected because it is the tradeoff with the highest biomass production.

The results for tradeoff 1 are shown in Figure 2.11. First of all, by enumerating 1000 hyperpaths, it was possible to find more hyperpaths that have high ethanol production and high toxicity resistance score. Before, the tenth hyperpath was definitely the best result for tradeoff 1. It has a minimum target production of 17.10 and a maximum score of 21.01. The same result can be obtained for one of the newly enumerated hyperpaths. Additionally, several other hyperpaths were computed that have a slightly lower minimum ethanol production (in between 16.5 and 16.9) but also a slightly higher toxicity resistance score of around 22. Therefore, they are also interesting candidates for the random exploration.

Compared to tradeoff 1, the results for tradeoff 3 were inferior (see Figure 2.12). The highest minimum ethanol production is at 17.09. However, the corresponding maximum toxicity resistance score is only 13.01. The highest found maximum toxicity score was around 21.01 and the matching minimum ethanol production at 16.97. These values are slightly inferior to the best results of tradeoff 1.

For tradeoff 6, the 1000 enumerated hyperpaths did not contain any with a high maximum toxicity resistance score (see Figure 2.13). This might be also due to the fact that the optimal toxicity resistance score for tradeoff 6 was clearly lower than for tradeoff 1 or 3. Similarly, for tradeoff 13 which has a higher optimal toxicity resistance score than tradeoffs 1 and 3, it was possible to obtain hyperpaths with maximum toxicity resistance scores that were higher than any of the scores for tradeoffs 1 and 3. However, the corresponding minimum ethanol productions are lower (see Figure 2.14) because the optimal ethanol production of tradeoff 6 is also significantly lower than for tradeoff 1 or 3.

As expected, the computed hyperpaths for tradeoff 47 do not have very high values (see Figure 2.15). Especially, the ethanol productions are very limited. Interestingly, the minimum ethanol productions and the maximum toxicity resistance scores are not very diverse. It might show again that the variability of the flux distributions that remain possible is limited if the biomass production is high.

All in all, the results of these five different tradeoffs confirm that choosing a tradeoff with a high optimal toxicity resistance score can lead to hyperpaths with higher minimum toxicity resistance scores. Likewise, choosing a tradeoff with a lower optimal toxicity resistance score limits the minimum toxicity resistance scores for its hyperpaths. Furthermore, selecting a tradeoff with a high optimal ethanol production allows for the computation of hyperpaths with a high minimum ethanol production. Consequently, even though the computed hyperpaths have values for ethanol production and toxicity resistance score that differ from the optimal values of the corresponding tradeoff, the choice of the tradeoff does influence the result. It is important to choose a tradeoff whose optimal

values are very close to the desired ones for target and biomass production and the toxicity score.
To justify the importance of the toxicity resistance score, the minimum ethanol production and maximum toxicity resistance score for hyperpaths including and omitting the toxicity resistance score in the optimization problem are compared. In tradeoffs 1 and 3 (see Figures 2.11 and 2.12), by including the toxicity resistance score, higher minimum toxicity resistance scores can be reached. Discarding the toxicity resistance score, a slightly higher ethanol production can be achieved.
In the case of tradeoff 47, the toxicity resistance scores are better when ommitting the score (see Figure 2.15). However, as already mentioned before, fixing the toxicity score when enumerating the hyperpaths is restricting the model. Hence, all solutions that are feasible in this case are also feasible when omitting the toxicity score. It is therefore possible to obtain hyperpaths with a high toxicity score when the score is omitted.
The results for tradeoff 13 are similar to those for tradeoffs 1 and 3 but the difference between the highest minimum ethanol productions is larger. In contrast to tradeoffs 1 and 3, the results for tradeoff 6 are different because similar minimum toxicity resistance scores are obtained for both cases. This can be explained by the fact that tradeoff 3 has a lower optimal toxicity resistance score. Accordingly, it influences the resulting hyperpaths less and they are more similar to the hyperpaths generated without taking the toxicity resistance score into account.
Overall, analyzing the different hyperpaths, it is preferable to include the toxicity resistance score in the optimization problem to obtain hyperpaths with a higher maximum toxicity resistance score and similar minimum ethanol production.

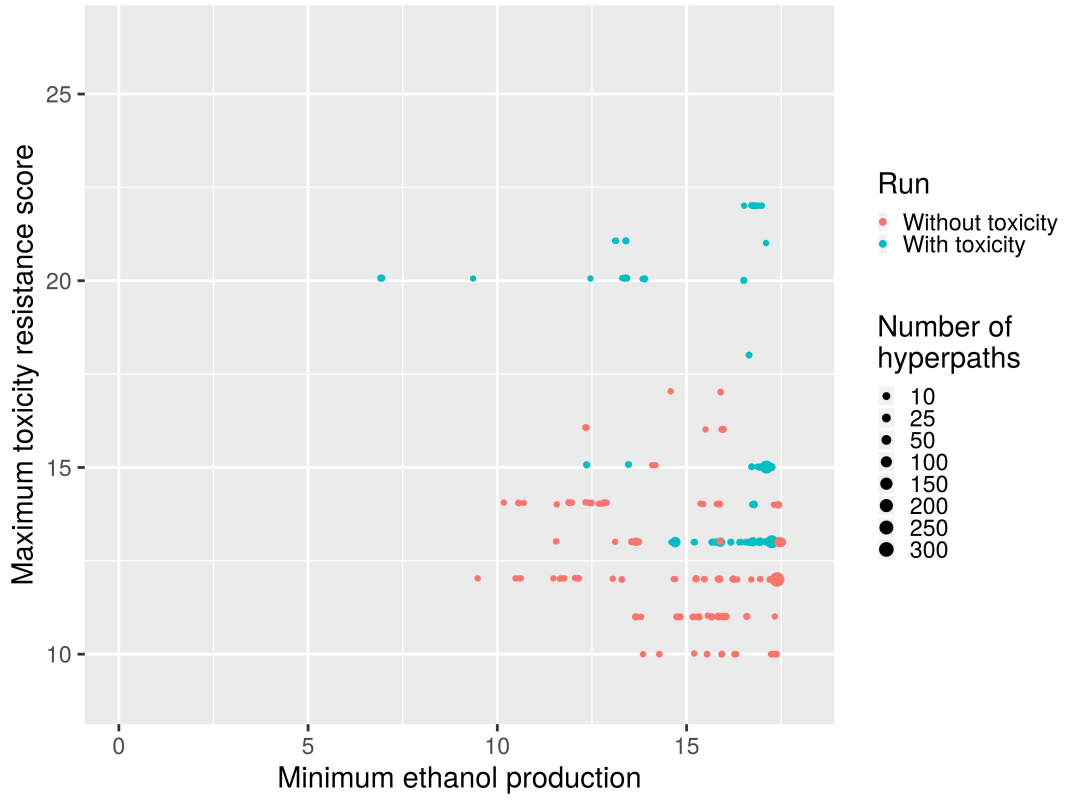


Figure 2.11: *Results of different hyperpaths for tradeoff 1.* One thousand different hyperpaths were computed taking the toxicity resistance score into account (red points) and 1000 different hyperpaths were computed omitting it (blue points).

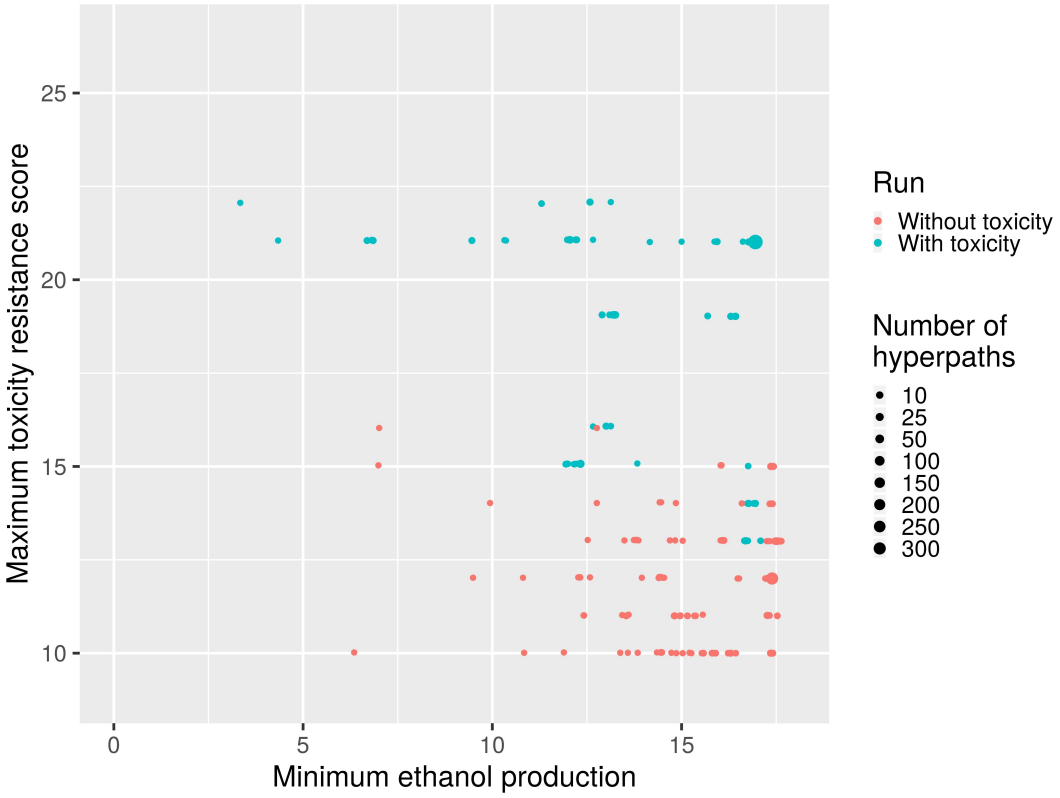


Figure 2.12: *Results of different hyperpaths for tradeoff 3.*

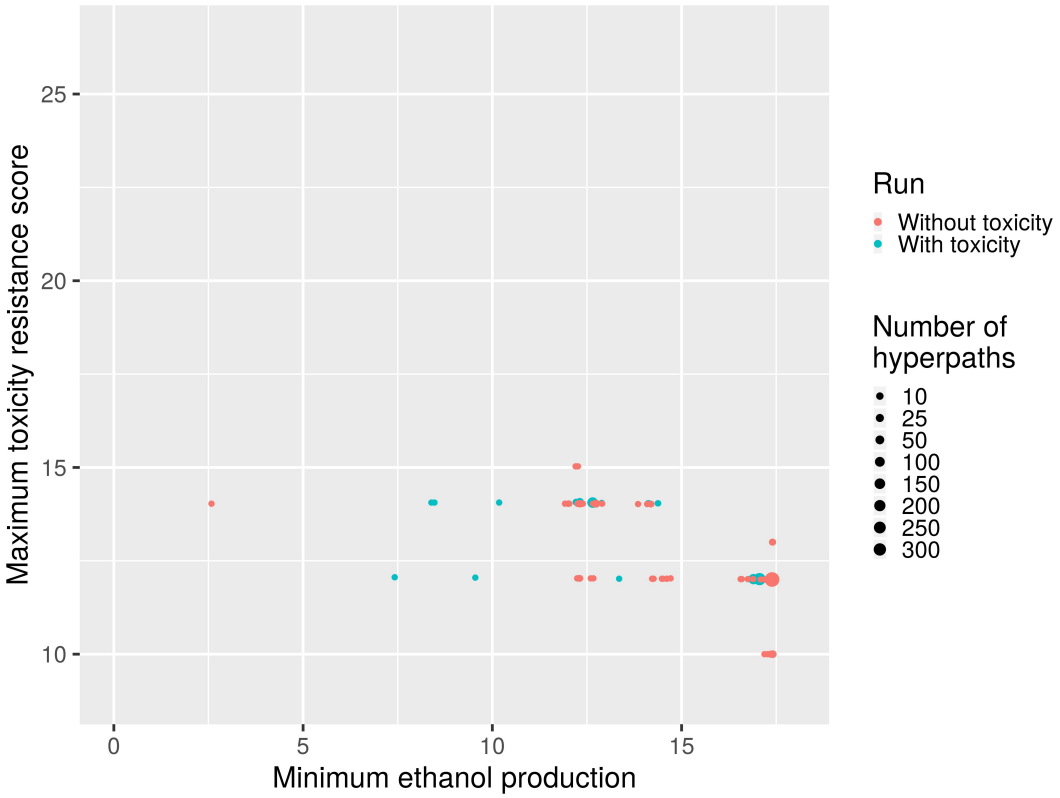


Figure 2.13: *Results of different hyperpaths for tradeoff 6.*

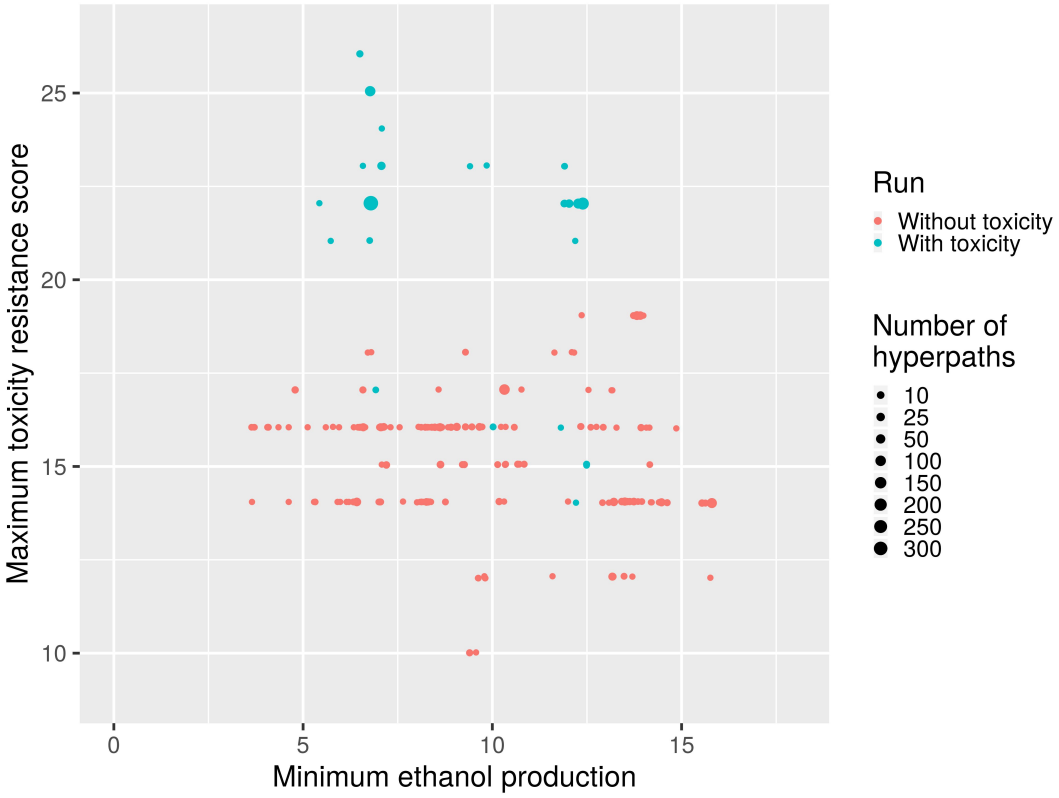


Figure 2.14: *Results of different hyperpaths for tradeoff 13.*

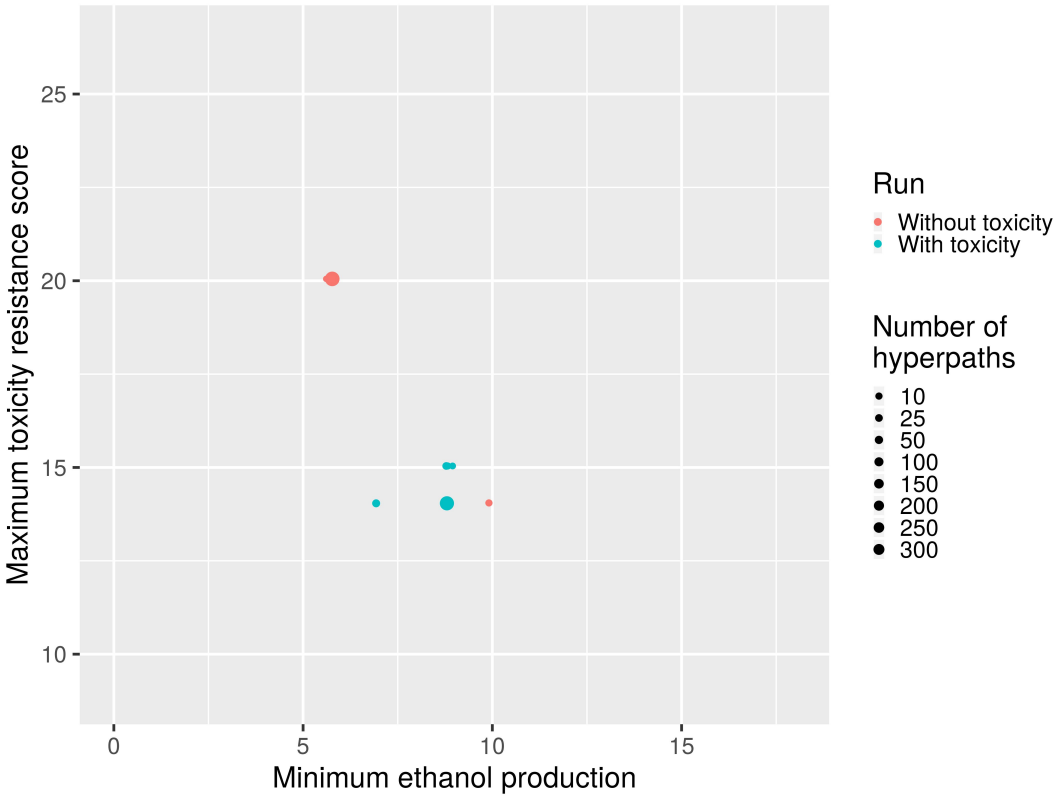


Figure 2.15: *Results of different hyperpaths for tradeoff 47.*

Random exploration results

Afterwards, we used the random exploration presented in Section 2.4.2 to find a subset of the outgoing reactions that can be used as knockout set. To compare the values for the biomass productions that are computed during the random exploration to $v^{ref}_{biomass}$, a small tolerance of $\epsilon = 0.005$ was used. This means that no new reactions are drawn for a knockout set if the maximum possible biomass production is smaller than or equal to $v^{ref}_{biomass} + \epsilon$. The bin size was set to 10 and the minimum size for the first bin was set to 50. In each iteration, one new knockout set was computed. The enumeration was stopped if no new smaller knockout set was found after 1000 iterations. When choosing the candidate for the next iteration, the probability to accept a bin was set to 0.5 to increase the chance that not only the smallest current knockout sets are picked as candidates and that more diverse solutions are explored.
Based on the results of the analyses of the different hyperpaths, knockout sets were enumerated for different hyperpaths of tradeoff 1 that were chosen based on their minimum ethanol productions and maximum toxicity scores (see Table 2.3). A total of 13 different hyperpaths were chosen and for each of these hyperpaths, the random exploration was repeated twice. After all runs were finished, interesting knockout sets were selected as follows. Only knockout sets that contained less than 25 reactions were selected. Furthermore, the minimum ethanol production needed to be above 14 and the maximum toxicity score above 20. For each run, only the smallest knockout sets were selected. However, if there was a slightly larger knockout set that led to an increase in the minimum target production, it was picked in addition. For example, if for one run the two smallest knockout sets have size 19 and both fulfill the desired threshold for ethanol production and toxicity score, they are selected. If there are knockout sets of size 20, they are not selected because the smaller knockout sets that are available are likely subsets of them. Therefore, the only reason to choose them is in the case where they lead to an increase in ethanol production and/or toxicity resistance score. In total, 18 knockout sets were selected in this way. The results are shown in Table 2.4.
When selecting the smallest computed knockout sets, a first observation was that not all runs led to knockout sets that fulfilled the above mentioned criteria. In Table 2.4, we can see that in total, knockout sets got selected only from 12 out of the 26 runs that were done (two runs for 13 hyperpaths). Additionally, for several hyperpaths (ids 799, 800, 657, 31), neither of the two runs led to very small knockout sets. There are several possibilities to explain that. Since the proposed approach to select the knockout sets is randomized, it might be necessary to do more repetitions which might lead to smaller knockout sets also for these hyperpaths. Another possibility is that indeed, the outgoing reactions for these hyperpaths contain less favorable combinations of reactions and it is actually necessary to knock out more reactions to cut off the desired subnetwork. Lastly, the proposed parameter settings for the random exploration are the results of multiple test runs and seemed to offer a good tradeoff between time, exploring different (larger) solutions and driving the exploration towards smaller knockout sets. However, the possibility remains that further fine tuning of the parameters might improve the results.
The 18 knockout sets contain 73 different reactions. A full version can be found in the Supplementary Table A.5. In Table 2.5, only the reactions are shown that occurred at least in five out of the 18 knockout sets. The three reactions that occurred the most (r_1110, r_0766 and r_1112) were always the reactions of subset 1 (reactions that have to be knocked out). The only exception was for hyperpath #169 for which both r_0766 and r_1112 were not part of subset 1. A possible explanation for this difference in relation to the other hyperpaths is that hyperpath #169 also had a slightly lower

H-Id	Max. biomass	Min. ethanol	Max. toxicity score
799	0.1690	16.713	22.016
800	0.1686	16.747	22.015
656	0.1706	16.753	22.014
657	0.1706	16.753	22.014
599	0.1661	16.815	22.011
31	0.1659	16.831	22.011
537	0.1659	16.831	22.011
536	0.1659	16.831	22.011
638	0.1653	16.899	22.011
169	0.1671	16.520	22.011
879	0.1657	16.961	22.010
638	0.1652	16.992	22.009
10	0.1643	17.101	21.008

Table 2.3: *List of hyperpaths that were chosen for the random exploration.* The first column indicates the id of the hyperpath. A total of 13 hyperpaths of tradeoff 1 were chosen based on their minimum ethanol productions and their maximum toxicity scores. The table is sorted by decreasing toxicity scores. The last hyperpaths (#10) has a slightly lower maximum toxicity score.

H-Id	Run	Size	Max. biomass	Min. ethanol	Max. ethanol	Max. toxicity score
10	1	17	0.1691	14.829	17.327	21.024
10	1	18	0.1690	16.119	17.331	21.016
10	2	21	0.1692	15.912	16.687	21.012
169	1	24	0.1721	14.773	16.819	22.024
536	1	19	0.1709	15.487	16.698	21.019
536	2	20	0.1709	14.627	15.864	21.012
536	2	20	0.1707	14.836	15.978	21.011
537	1	18	0.1709	14.220	15.864	21.012
537	1	18	0.1709	14.220	15.864	21.012
537	1	19	0.1709	15.427	15.864	21.012
537	1	19	0.1709	15.427	15.864	21.012
599	1	17	0.1720	15.177	15.689	21.011
638	1	19	0.1702	16.943	17.109	21.010
638	2	18	0.1739	14.637	16.524	22.028
638	2	19	0.1739	15.394	16.563	22.024
656	1	17	0.1709	15.801	16.702	21.014
658	2	20	0.1729	16.211	16.817	22.021
879	1	22	0.1706	15.177	16.421	21.018

Table 2.4: *Size and production values for selected knockout sets.* In total, 18 interesting knockout sets got selected based on their size and their values for ethanol productions and toxicity scores. The smallest knockout sets that were enumerated contained 17 reactions.

Sbml Id	Reaction name	Gene associations	#
r_1110	ADP/ATP transporter	(YBL030C or YBR085W or YMR056C)	18
r_0766	NAD kinase	YPL188W	17
r_1112	AKG transporter mitochonrial	YMR241W	17
r_0489	glycerol-3-phosphatase	(YER062C or YIL053W)	14
r_0324	D-sorbitol reductase	YHR104W	13
r_0149	adenylate kinase	YER170W	12
r_1623	5-formethyltetrahydrofolate cyclo-ligase		12
r_1183	L-alanine transport	(YBR068C or YCL025C or YDR046C or YKR039W or YOL020W or YOR348C or YPL265W)	11
r_2037	reduced thioredoxin transport		11
r_0174	aldehyde dehydrogenase	YOR374W	10
r_0658	isocitrate dehydrogenase	(YNL037C and YOR136W)	10
r_1239	oxaloacetate transport	YKL120W	10
r_0713	malate dehydrogenase	YKL085W	9
r_1118	aspartate-glutamate transporter	YPR021C	9
r_0111	acetyl-CoA hydrolase	YBL015W	8
r_0416	fatty-acyl-ACP hydrolase	(YKL182W and YPL231W)	8
r_0940	proline oxidase (NAD)	YLR142W	8
r_1096	(R)-mevalonate transport		8
r_0415	fatty-acyl-ACP hydrolase	(YKL182W and YPL231W)	7
r_0552	hydrogen peroxide reductase	((YGR209C and YLR109W) or (YLR043C and YLR109W))	7
r_1117	aspartate transport	YPR021C	7
r_1777	fatty acid transport		6
r_0113	acetyl-CoA synthetase	YAL054C	5
r_0300	citrate synthase	(YNR001C or YPR001W)	5
r_1030	tetrahydrofolate aminomethyltransferase	(YAL044C and YBR221C and YDR019C and YER178W and YMR189W)	5
r_1780	fatty-acyl-ACP transport		5

Table 2.5: *Number of reaction occurrences in 19 different knockout sets - short version.* The table shows how often a reaction occurred in the 19 selected knockout sets. The number of occurrences is shown in the last column. In this version, only reactions that appeared at least five times are shown. For the full version, see Supplementary Table A.5. In total, 74 reactions appeared in the 19 knockout sets. Problematically, quite a few of the proposed reactions are transports that might be difficult to knock out *in vivo*.

minimum ethanol production compared to all the other chosen hyperpaths (see Table 2.3).
The smallest knockout set that could be found and that satisfied the thresholds included 17 reactions. Problematically, many out of the 73 reactions represent transports in the organism (see Table 2.5) and it might be difficult to knock them out *in vivo* which means that the proposed knockout sets might not be practicable. As can be seen in Table 2.5, not all of the reactions do have associated genes which also makes it infeasible to knock them out *in vivo*. Furthermore, for example, reaction r_1110, which is an ADP/ATP transporter and was always part of subset 1 for the chosen hyperpaths and therefore played an important role in attaining the desired values for ethanol production and toxicity scores, has three associated genes. This means that it might require a greater effort to actually knockout this reaction in practice compared to other reactions that have less associated genes. The selected knockout sets were chosen based on the number of reactions that they contained. It might be necessary to select them based on the number of associated genes to identify knockout sets that need the smallest amount of gene knockouts. It is however not always necessary to knock out all associated genes of a reaction. If multiple associated genes for one reaction correspond to an enzymatic complex, knocking out one single gene might be enough to already disrupt its activity. For example, reaction r_1030 has five associated genes that are connected by 'and' in the model. Thus, knocking out one of the five genes should already interrupt the activity of this reaction.
Although certain reactions appeared in most of the selected knockout sets, a certain variability remained in the reactions that occurred less often which might make it possible to exchange at least some of the reactions that are more problematic or more costly (i.e. in terms of gene associations) to knock out. This also implies that by doing more runs or by broadening the selection of knockout sets, this choice could be further increased which could also help to identify knockout sets with fewer gene associations.

2.6 Discussion

We showed on the example of ethanol production in yeast that our developed approach can compute knockout sets that increase the target production and that ensure that reactions that are critical for a tolerance against the target can be active. To apply this approach, prior knowledge or experiments are essential to identify the reactions that are critical to improve the resistance of the microorganism against the toxic target. Without this information, the formulated multi-objective optimization problem is not applicable. If detailed knowledge for different reactions is on hand, their weights can be adjusted accordingly to improve the model. It is however also possible to modify the score that was used to capture the resistance against the toxic target if in specific cases a different model is preferable.
First of all, we computed the Pareto front to gain valuable insights about different efficient flux distributions in the network. As an advantage of this approach, we could see that there are some extreme points that have very similar ethanol and biomass production values but differ considerably in their toxicity resistance score. This already demonstrates that it is beneficial to include the toxicity resistance score in the optimization problem because desirable values for biomass and ethanol production can be reached without having reactions active that are critical for ethanol resistance. Consequently, when focusing exclusively on biomass production and target production, it is possible that knockouts are proposed that will reduce the resistance against the toxic target. The Pareto front that displays tradeoffs between all three objectives can be used to choose more favorable outcomes

for which knockouts should be enumerated.

Problematically, the MILP that was initially formulated to enumerate different knockout sets was not viable. It is possible to enumerate single knockouts but they are not sufficient to enforce the aspired flux values. For larger knockouts, the approach is unsuitable because the MILP is not restrictive enough. Moreover, even for single knockouts, the value of this approach is questionable because in retrospect, it seems to be easier to simply check all single knockouts instead of enumerating them. As already proposed before, it might be possible to narrow down the enumerated knockouts to the ones that actually lead to a change by using a min-max formulation. The resulting increase in the complexity of the problem however might also make this approach less viable in practice, especially for larger networks.

In the second approach, we enumerated and analyzed hyperpaths for different points in the Pareto front. Interestingly, there was a huge overlap between all hyperpaths which confirmed that certain parts of the metabolic network are indispensable for the production of biomass and ethanol. Moreover, we could show the advantage of including the toxicity resistance score in our model. It must be said that by fixing the toxicity score when enumerating the hyperpaths, the model is more restricted. This means that all solutions that are feasible in this case are also feasible when omitting the constraint that fixes the toxicity score to a specific value. However, when omitting the score, the feasible space is larger and therefore, the solver is not guided to include the critical reactions in the hyperpath. Therefore, the probability that the computed hyperpaths lead to a good toxicity resistance score (based on the evaluation that was used) is lower. Analysing the different results revealed that multiple reactions that are critical for the resistance against ethanol are not essential for biomass and ethanol production.

One other benefit of the presented approach is that it is applicable to genome-scale networks. Although metabolic networks are commonly subject to analyses, many approaches are only applicable to smaller networks that represent just a condensed version of the metabolism and they struggle with larger networks due to the increased complexity. It is however important to analyse the complete metabolism to understand more complex relations and to model the organisms in a way that is as detailed as possible. Our method manages to extract knockout sets for very specific conditions from genome-scale metabolic networks.

The smallest computed knockout set contained 17 reactions. Realizing knockouts of this size *in vivo* is still challenging. However, we did not prove that 17 is the smallest knockout set that is possible. The results that we obtained with the proposed parameter settings for the random exploration were promising but, of course, tuning the parameters further might actually also improve the results further, e.g. lead to the computation of smaller knockout sets. Additionally, we chose hyperpaths for the random exploration that aimed for a very high ethanol production. Giving more freedom and reducing these expectations might also allow for the computation of smaller knockout sets.

Up to now, the presented results are purely theoretical and their practical value remains to be evaluated biologically. Although the utilized metabolic network of yeast is very detailed and well curated, the simulated flux distributions for the introduced knockouts do not necessarily match reality. Furthermore, it is important to remark that it still needs to be investigated how applicable the proposed reactions are for knockouts. As mentioned during the presentation of the results, many of the reactions that were part of the smallest computed knockout sets were modeling different transports in the organism. It is known that these kinds of reactions are not always usable as candidates for knockouts *in vivo*. Some reactions also did not have associated genes. To avoid that

reactions get included in the computed knockout sets that are unsuitable in practice, a list could be provided that contains only reactions that can be knocked out *in vivo*. This list can, for example, also be used to exclude reactions that have no known gene associations. By providing this type of list, the evaluation of the different hyperpaths also needs to be adapted because it will lead to outgoing reactions that cannot be knocked out. This means that the remaining variability of the hyperpath can be higher because it will be less restricted. Hence, different hyperpaths might actually get chosen as good candidates for the random exploration. One disadvantage of excluding certain reactions is that it might also bias the outcome and it is possible to miss important insights. So far, the approach models reactions knockouts. The smallest reaction knockouts do not necessarily correspond to the smallest gene knockouts. Hence, it might be preferable to adapt the approach to directly model gene knockouts.

It must also be noted that it is likely that certain genes have other functions that are not modeled in the metabolic network. Knocking them out might impact the results in unexpected ways. On the other hand, it could also mean that less knockouts are needed to achieve the theoretical results *in vivo*. It might therefore be advantageous to perform the knockouts in a sequential way. For example, knocking out one single gene from a reaction with multiple genes that correspond to an enzymatic complex might be enough to already disrupt the activity.

So far, we only considered knockouts in our approach. Other methods like OptForce (Ranganathan et al., 2010) and regulatory MCSs (Mahadevan et al., 2015) show that it is also interesting to identify up- and downregulation that will force a specific overproduction. In our case, the values that we wanted to enforce, which correspond to the different points in the Pareto front, were not obtainable in the enumerated hyperpaths. Indeed, when we maximized the biomass production and fixed it to its maximum for the consecutive minimization of ethanol production or maximization of the toxicity resistance score, the resulting values differed from the ones of the points in the Pareto front.

The subnetworks that remain after removing all of the outgoing reactions of a hyperpath include more than 300 reactions. This size also leads to high variability for the possible flux distributions. Especially the biomass production was often not restricted significantly which is problematic because we could also show that higher values for the ethanol production can only be obtained if the value for the biomass production does not exceed a certain limit. On the other hand, we also know that we cannot knockout any of the reactions on the hyperpath to further restrict the biomass production because the active reactions of the hyperpath were identified as vital to reach the values of the point in the Pareto front (i.e. for biomass production, ethanol production and toxicity resistance score). Since we cannot knockout any of the active reactions, one possibility could be to up- or downregulate some reactions on the hyperpath to reduce the undesired variability.

Another interesting aspect of this approach is that the overall framework is very flexible. We developed our approach to address situations where the desired target metabolite is toxic for the producing microorganism and might therefore result in a less efficient production. Hence, we computed tradeoffs between biomass production, ethanol production and a toxicity resistance score. It is however quite straightforward to modify this multi-objective optimization problem to account for different circumstances. For example, if the desired target metabolite is not toxic for the microorganism, the toxicity resistance score can be omitted from the problem. It would also be possible to compute tradeoffs between biomass production and the productions of two different target metabolites. Another example would be to consider a case where the production of the target metabolite also leads to the production of a toxic by-product. Here, the idea would be to identify tradeoffs that

maximize the target production and minimize the production of the by-product. Independently of the adapted multi-objective optimization problem, hyperpaths can be enumerated accordingly, i.e. the newly formulated objective functions have to be fixed to their optimal values. Apart from that, the enumeration remains unchanged and also the concept for the random exploration does not change. Since our approach is suitable for genome-scale metabolic networks, this flexibility might make it interesting for different types of analyses.

2.7 Conclusion and perspectives

In this chapter, we presented a constraint-based approach that uses multi-objective optimization to identify tradeoffs between biomass production, target production and a score that measures the potential resistance of the microorganism against the toxic target. After tradeoffs have been computed, our method determines subnetworks that are needed to uphold the values of the three objectives. The different subnetworks are evaluated based on the assumption that the biomass production is prioritized. The target production and the score are therefore always computed for the maximum possible biomass production. This assumption can be changed if a better model is on hand. Afterwards, our approach uses a random exploration to identify smaller knockout sets for promising subnetworks based on the outgoing reactions.

Applying our approach to the case-study of ethanol production in yeast, we were able to compute knockout sets with less than 20 reactions. It remains an open question to determine what the smallest possible size is. Additionally, we need more biological evaluation for the proposed knockout sets to establish how viable the suggested reactions are in practice.

The advantages of our approach are that it is applicable on genome-scale metabolic networks as we showed by using the yeast 5.01 model. Moreover, the framework is flexible and it should be possible to alter the objectives for different scenarios. Hence, we are also interested in applying our approach to other examples to further confirm its adaptability.

We plan on polishing the approach and making it available as a tool on the Gitlab of the team. After obtaining some more biological opinion from collaborators on our results, we want to submit this work as a paper later this year.

3 Identifying active reactions during the transient state

Contents

3.1 Introduction

This chapter is based on the publication (Ziska et al., 2020) that was submitted to *Bioinformatics*. The work is the result of a collaboration with Ricardo Andrade and Mariana Ferrarini. It is the continuation of an approach that Alice Julien-Laferrière developed during her PhD (Julien-Laferrière, 2016). This work was performed using the computing facilities of the CC LBBE/PRABI.

The increasing availability of metabolomic data and their analysis are improving the understanding of cellular mechanisms and elucidating how biological systems respond to different perturbations (Sevin et al., 2015). Metabolomics can identify the metabolic capacities of an organism and this fact can be used to obtain a metabolic profile that characterizes the physiological response of a cell, tissue or organism to stress or to a general perturbation (Roessner and Bowne, 2009). Different network-based strategies for metabolomic data analysis have been recently reviewed in (Perez de Souza et al., 2020) and amongst others, such strategies can be used to establish associations between metabolites or to integrate them into metabolic pathways.

Metabolic profiles are often analyzed and interpreted with the help of bioinformatic software such as MetExplore (Cottret et al., 2018; Frainay et al., 2019), MetaboAnalyst (Xia et al., 2015; Chong et al., 2018) or 3Omics (Kuo et al., 2013) that can identify the set of metabolites with a significant change in their concentration. The metabolomic data are projected on the annotated metabolic pathways in order to highlight the processes that may be linked to the observed changes. The aforementioned software also try to integrate different kinds of omic data (such as transcriptomic, metabolomic or proteomic data) in order to give a deeper understanding of the studied mechanisms (Cambiaghi et al., 2017). Different approaches were reviewed in (Rosato et al., 2018; Ivanisevic and Want, 2019; Stanstrup et al., 2019) and software for the enrichment analysis of metabolomic data were evaluated and their results compared in (Marco-Ramell et al., 2018). However, metabolic pathways have subjective definitions and can differ between databases (Ginsburg, 2009). Additionally, this kind of analysis can make it hard to identify the connections between metabolites since they can be part of many pathways and it is thus possible to miss paths which traverse several pathways.

Another approach is to use graph-based methods that allow us to consider the whole metabolism as an integrated system focusing on the parts that are connecting the metabolites of interest. Usually, these methods rely mainly on the network structure, chemical information and on an input list of metabolites (Frainay and Jourdan, 2017).

In (Acuña et al., 2012a; Milreu et al., 2014), a method is proposed in this direction that is based on the enumeration of metabolic stories. A metabolic story is defined there as the set of reactions that summarize the flow of matter from a set of source metabolites to a set of target metabolites and is characterized as a maximum directed acyclic sub-graph connecting the metabolites of interest. One of the drawbacks of this approach is that a metabolic story is acyclic and thus, it is not possible to obtain sets of reactions that contain cycles. However, cycles are common in metabolic networks and this assumption might thus not reflect reality. Additionally, it does not take the stoichiometry of the reactions into account. This can in turn lead to a set of reactions that is not feasible in practice.

Metabolite concentrations have been used to assess the responses to small perturbations in the context of constraint-based models (Palsson, 2000; Covert and Palsson, 2003; Klamt et al., 2014). In (Reznik et al., 2013), the authors used a method that is derived from the classical flux balance analysis (FBA) framework. They showed that the variables of the dual problem, the so-called shadow prices, which correspond to the sensitivity of FBA to imbalances in the flux, can indicate if a metabolite is a growth-limiting metabolite in FBA. In (Rohwer and Hofmeyr, 2008; Christensen et al., 2015),

methods are presented to identify regulatory metabolites and paths by varying *in silico* their known concentrations in a measured steady-state using supply-demand analysis. Therefore, these methods are based on the response of an organism to a relatively small perturbation and on the influence of the metabolite concentrations on the reaction rates of the system to return to the original equilibrium.
In this work, we focus not on the metabolite pools in one condition but on the difference of the obtained measurements between two conditions. We suppose that the difference of metabolite pools between two metabolic states can provide information on the transient state, that is, on the transition between the two measured conditions. In (Sajitz-Hermstein et al., 2016), the authors provide a method to integrate relative metabolomic measurements in order to make predictions about differential fluxes. They used a constraint-based approach which minimizes the distance between the two flux vectors of the two different states based on the ratio between the measured metabolite concentrations in both conditions. For both states, steady-state is assumed for the flux vectors. The authors identify differential fluxes between the two conditions whereas our approach will aim to find reactions that are potentially active during the transient state.
In (Case et al., 2016), a similar problem was studied. The authors investigated reachability problems in chemical reaction networks. Given two different states of the network, the goal is to identify a path that leads the network from the first state to the second one. They prove that this problem can be solved in polynomial time. However, they also discuss that a variant of this problem in which the maximum size of the path is fixed is more difficult to solve. Our approach overcomes this limitation and is able to minimize the number of active reactions which is important since we are interested in identifying only the parts of the network that are potentially active during the transient state.
The method we propose uses a constraint-based modeling to enumerate sets of reactions that explain the changes in concentrations for some measured metabolites, *i.e.* how the system moved from a state to another. We implemented our approach in a software we called TOTORO (for "Transient respOnse to meTabOlic pertuRbation inferred at the whole netwOrk level"), that is publicly available at https://gitlab.inria.fr/erable/totoro. It is implemented in C++ and depends on IBM CPLEX which is freely available for academic purposes. We also tested our method with data from pulse experiments with different carbon sources (glucose, pyruvate and succinate) in *Escherichia coli*.

3.2 Methods

A metabolic network can be represented as a directed hypergraph $H(\mathcal{V}, \mathcal{R}, \mathcal{S})$ where $\mathcal{V}$ is the set of vertices, $\mathcal{R}$ the set of hyperarcs and $\mathcal{S}$ the stoichiometric matrix. Each $c \in \mathcal{V}$ represents a metabolite of the network and each hyperarc $r \in \mathcal{R}$ a reaction that connects two sets of disjoint metabolites $Subs_r, Prod_r$ with $Subs_r, Prod_r \subseteq \mathcal{V}$. The stoichiometric matrix $\mathcal{S}$ is a $m \times n$ matrix where each column represents a reaction and each row a different metabolite. It contains the stoichiometric coefficients which are positive if a metabolite is produced by a reaction and negative if it is consumed. The set $X \subseteq \mathcal{V}$ contains all measured metabolites. The metabolomic data is given as a list which, for each measured metabolite in X, contains an interval. This interval describes by how much the internal metabolite concentration changed between two different states. Usually, small deviations for the measurements are available which can be used to calculate the minimum and the maximum possible difference between the internal metabolite concentrations. Furthermore, all reversible reactions of the network are split into forward and backward reactions.
We are interested in solving the following problem: Given a network H and a list containing the

changes for some metabolite concentrations before and after a perturbation, we want to identify sets of reactions that were involved in diverting the system from the initial state before the perturbation to the state after the perturbation. We will present a constraint-based approach to solve this problem where the change of concentrations between two states is represented as an interval.

3.2.1 Core model

The variation of the concentrations in time of the metabolites in X can be written as:

$$\frac{dX}{dt} = (\mathcal{S} \cdot v)_X. \tag{3.1}$$

In this equation, v is a flux vector and the $(\cdot)_X$ operator means that only the entries of the vector corresponding to the metabolites in X are taken into account. We use $[X]_t$ to denote the concentration for the metabolites in X at time point t. Considering two points t_0 and t_1 in time and $\Delta_X = [X]_{t_1} - [X]_{t_0}$, one can write:

$$\Delta_X = \mathcal{S} \cdot \varphi. \tag{3.2}$$

In this case, each entry of the vector φ can be interpreted as the overall number of moles that passed through the reaction j during the time interval $[t_0, t_f]$ which corresponds to the area under the reaction rate curve in this time interval:

$$\varphi_j = \int_{t_0}^{t_1} v_j(t) \cdot dt. \tag{3.3}$$

Due to biological and technical variability that can arise from different replicates of the same experiment, we assume that the measured variations in concentrations of the metabolites in X are represented by an interval rather than using a fixed number:

$$\Delta_X = [\Delta_X^{\min},\ \Delta_X^{\max}]. \tag{3.4}$$

Furthermore, for the non-measured metabolites, we do not know if their concentration changed or not. Therefore, we will assume that a *small* variation is possible for all non-measured metabolites $\overline{X} = \mathcal{V} \setminus X$:

$$\Delta_{\overline{X}} = [\epsilon^{\min},\ \epsilon^{\max}]. \tag{3.5}$$

Based on these assumptions, we can model the production or consumption of metabolites between two states by the following constraints:

$$\begin{aligned} &\Delta^{\min} \leq \mathcal{S} \cdot \varphi \leq \Delta^{\max} \\ &0 \leq \varphi_j \leq u_j \qquad\qquad \forall j \in \mathcal{R}. \end{aligned} \tag{3.6}$$

All φ_j are positive and have an upper bound u_j. $\Delta^{\min}$ is a vector composed of $\Delta_X^{\min}$ and $\epsilon^{\min}$ while $\Delta^{\max}$ is composed of $\Delta_X^{\max}$ and $\epsilon^{\max}$.

The numerical values of the φ vector are difficult to interpret. The variable φ can only be zero or have a positive value. This means that we do not know if the activity of the corresponding reaction

was increased or decreased during the shift compared to the baseline. We only know that if φ_j is zero in the solution, reaction j was not active during the shift while if φ_j has a non-zero value, reaction j was active during the shift. Hence, we are only interested in the reactions that have a non-zero φ because we want to identify the part of the metabolic network that was active during the metabolic shift. These reactions are represented by the support of the vector φ.

3.2.2 Minimizing the number of reactions and the variation of the concentrations for the non-measured metabolites

Since the number of possible paths that can explain the measured metabolic shifts can be very large, we will focus on finding the smallest solutions with regard to the number of active reactions that still explain the metabolic shift. This corresponds to the parsimonious assumption that the fewest possible resources are used or the smallest changes are made. Thus, we are interested in identifying minimum sets of reactions that play a major role in the metabolic shift. For each reaction j, a binary variable y_j is then introduced that is set to zero if and only if the corresponding φ_j is zero and therefore, the reaction is not part of the solution. In this way, these variables will correspond to the support vector of φ and it will be sufficient to minimize their sum:

$$\begin{aligned} &y_j = 0 \leftrightarrow \varphi_j = 0 \qquad \forall j \in \mathcal{R} \\ &y_j \in \{0,1\}. \end{aligned} \tag{3.7}$$

Additionally, to prevent that both a reaction j and its reversible $\bar{j}$ can be picked at the same time for one solution, the following constraint is used:

$$y_j + y_{\bar{j}} \leq 1 \qquad \forall (j,\bar{j}) \in \mathcal{R}. \tag{3.8}$$

To minimize the number of reactions that are part of the solution, the objective function is written as:

$$\min \sum_{j=1}^{m} y_j. \tag{3.9}$$

However, we are not only interested in minimizing the number of reactions in the solution but also in minimizing the variation in concentration for the non-measured metabolites $\overline{X}$. Since the measured compounds are usually the more important ones for analyzing the biological experiment, it is reasonable to aim for solutions where other compounds do not accumulate or deplete a lot. This leads to the following minimization:

$$\min \sum |\mathcal{S} \cdot \varphi|_{\overline{X}}\,. \tag{3.10}$$

On the other hand, we are trying to explain as much change in the concentration as possible for the measured metabolites:

$$\max \sum |\mathcal{S} \cdot \varphi|_{X}\,. \tag{3.11}$$

To combine both ideas in one objective function, a weight λ is used for both objectives:

$$\min \lambda \sum_{j=1}^{m} y_j + (1-\lambda) \sum |\mathcal{S} \cdot \varphi|_{\overline{X}} - (1-\lambda) \sum |\mathcal{S} \cdot \varphi|_X \,. \tag{3.12}$$

The value for λ should lie between 0 and 1. Finding a good balance between these two objectives can be challenging but necessary to identify meaningful biological solutions. This will be further discussed in Section 3.3.
Summing up, the mixed-integer linear program (MILP) that is implemented in our software TOTORO is the following:

$$\begin{aligned}
\min_{\varphi, y} \quad & \lambda \sum_{j=1}^{m} y_j + (1-\lambda) \sum |\mathcal{S} \cdot \varphi|_{\overline{X}} - (1-\lambda) \sum |\mathcal{S} \cdot \varphi|_X \\
s.t \quad & \Delta^{\min} \leq \mathcal{S} \cdot \varphi \leq \Delta^{\max} \\
& 0 \leq \varphi_j \leq u_j && \forall j \in \mathcal{R} \\
& y_j = 0 \leftrightarrow \varphi_j = 0 && \forall j \in \mathcal{R} \\
& y_j + y_{\bar{j}} \leq 1 && \forall (j, \bar{j}) \in \mathcal{R} \\
& y_j \in \{0,1\}; \lambda \in (0,1); u_j, \varphi_j \in \mathbb{R}.
\end{aligned} \tag{3.13}$$

3.2.3 Enumerating different solutions

To enumerate different solutions, once a solution is found, it must be excluded for the next iteration. Two solutions are different if they do not contain the same reactions. We are using the following constraint where y^* is a previously found solution vector:

$$\sum_{j \in \mathcal{R}: y_j^* = 1} y_j \leq \sum_{j=1}^{m} y_j^* - 1. \tag{3.14}$$

This prevents that the exact same combination of reactions gets chosen again. Afterwards, we can solve the updated MILP again to compute a different solution. We repeat this process until no more new solutions can be found or until a desired number of solutions has been computed.

3.2.4 Dealing with source/sink reactions and co-factors

In graph-based methods, it is known that looking for shortest paths without taking into consideration co-factors (for example ADP) can lead to irrelevant paths because such metabolites can introduce shortcuts through the network (Frainay and Jourdan, 2017). In our case, although we are using a constraint-based approach and taking stoichiometry in account, similar problems can arise. When considering only shortest paths, depending on the presence of source or sink reactions and/or the value chosen for $\epsilon^{\min}$ and $\epsilon^{\max}$, the active reactions in the solution can be highly disconnected. This makes them biologically less meaningful because it is not possible to identify possible pathways that played a role during the metabolic shift.
For example, if only the size of the solution is minimized, it is possible that changes in the concentration are just transferred to a close source or sink without actually selecting a pathway. To avoid this effect, it is important to block transport reactions in the network. Blocking transport reactions

means that they cannot be part of a solution. However, if the substrates of a sink reaction are accumulated or the products of a source reaction are depleted in a solution, this indicates that the corresponding transport reaction can be part of the solution. Their use is limited by the chosen ϵ but it can be set to a very low or large value to imitate an infinite source or sink. Specific sources or sinks can be added to the problem by specifying a large negative $\Delta^{\min}$ or a large positive $\Delta^{\max}$ for certain metabolites. Therefore, transport reactions should always be blocked by setting their lower and upper bounds to zero.

A similar effect can happen if the value of epsilon for the non-measured metabolites is chosen too big. In this case, the changes in concentration of the measured metabolites can simply be distributed on (accumulated on or taken from) the nearby non-measured metabolites. This prevents that longer pathways are chosen which would actually connect several measured metabolites and could explain how the depletion of one measured metabolite leads to the accumulation of another measured metabolite (and vice-versa). However, this issue can be addressed by decreasing the value of λ in the objective function and thereby giving more weight to the function that minimizes the accumulation in non-measured metabolites. This should result in solutions that are larger but that connect the measured metabolites better than when only the number of reactions is minimized. Furthermore, it might be preferable to choose smaller epsilons to further restrict the accumulation/depletion of non-measured metabolites.

Before we minimized the accumulation and depletion of non-measured metabolites, we tried to prevent that shortcuts through the network are taken by limiting the amount of connected reactions that appear in the solution as active for each metabolite. Co-factors can usually have a very high degree which means that they are involved in many reactions. To prevent a high degree in the solution, the following constraint was used:

$$\sum_{j \in \mathcal{R}: i \in Prod_j \cup Subs_j} y_j \leq D \quad \forall i \in \mathcal{V}. \tag{3.15}$$

As a consequence, a metabolite cannot have more than D producing and consuming reactions that are part of the solution.

However, after including the minimization of the accumulation and depletion of non-measured metabolites in the objective function, we omitted this constraint. It was not necessary to treat co-factors differently since we can still obtain connected pathways as will be shown in Section 3.3.

3.2.5 Calculating the input deltas

To apply TOTORO, it is necessary to provide the input deltas for measured metabolites. In most cases, they will not be available but they have to be calculated from the available metabolite concentrations of the different two conditions.

In the data that we used to apply TOTORO, internal metabolite concentrations for the first condition (glucose baseline experiments) were given. They also included deviations that resulted from different replicates of the same experiment. The concentrations for the second condition (a pseudo-steady state after a pulse) had to be calculated from the metabolite concentrations of the first conditions and a fold change.

Therefore, for each measured metabolite $x \in X$, we computed the minimum (maximum) internal concentration for the pseudo-steady state $[x]_{pss-min}$ ($[x]_{pss-max}$) using the baseline glucose

concentration $[x]_{baseline}$ with the given deviations d_x and the fold changes f_x:

$$\begin{aligned} [x]_{pss-min} &= ([x]_{baseline} - d_x) \cdot f_x \\ [x]_{pss-max} &= ([x]_{baseline} + d_x) \cdot f_x. \end{aligned} \tag{3.16}$$

Afterwards, we computed the minimum difference $\Delta_x^{\min}$ and maximum difference $\Delta_x^{\max}$ between the internal concentrations of the glucose baseline and the pseudo-steady state:

$$\begin{aligned} \Delta_x^{\min} &= [x]_{pss-min} - ([x]_{baseline} + d_x) \\ \Delta_x^{\max} &= [x]_{pss-max} - ([x]_{baseline} - d_x). \end{aligned} \tag{3.17}$$

When the differences are negative, $\Delta_x^{\min}$ and $\Delta_x^{\max}$ are swapped.
The interval defined by $\Delta_x^{\min}$ and $\Delta_x^{\max}$ must have a distinct direction meaning that it is not possible that $\Delta_x^{\min}$ is negative and $\Delta_x^{\max}$ is positive. Therefore, measured metabolites with a fold change of 1.0 are considered as non-measured metabolites and are assigned the chosen generic ϵ. In some cases, for example if the fold change is close to 1.0 and the baseline deviations are large enough, it is possible that $\Delta_x^{\min}$ is negative and $\Delta_x^{\max}$ is positive. If one of them is clearly closer to zero, it will be set to zero. If no clear direction is determinable, the corresponding metabolite will be treated as a non-measured metabolite.

3.3 Results

To evaluate our approach, we used data from different pulse experiments with different carbon sources in *E.coli* as presented in (Taymaz-Nikerel et al., 2013). The authors measured the internal concentrations for several metabolites for a glucose baseline and for glucose, pyruvate and succinate pulse experiments. These data were used to apply the method on the *E.coli* core model (Orth et al., 2010a) and the *E. coli* iJO1366 model (Orth et al., 2011) available from the BiGG database (King et al., 2015b). The *E. coli* core model consists of 72 metabolites and 95 reactions, the *E. coli* iJO1366 model of 1805 metabolites and 2583 reactions.
We were interested in the difference between the glucose baseline and the pseudo-steady state shortly after the pulse experiment. In (Taymaz-Nikerel et al., 2013), the authors provided the internal concentrations for the baseline, including the deviations for their measurements and the fold changes for the three different pseudo-steady states which we used to calculate the internal concentrations for each pseudo-steady state. In (Taymaz-Nikerel et al., 2013), deviations for the measured concentration of the glucose baseline are given that were derived from several replicates of the same experiment. We used them to be able to calculate the minimum difference $\Delta_X^{\min}$ and maximum difference $\Delta_X^{\max}$ in the concentrations between the glucose baseline and each pseudo-steady state. A detailed explanation can be found in the Supplementary Material Section 3.2.5. The calculated $\Delta_X^{\min}$ and $\Delta_X^{\max}$ for all three pulse experiments can be found in the Supplementary Tables A.6, A.7 and A.8.
We used all measured metabolites that are present in the network and that had a significant change in their concentration as input. It should be noted that a change for each given metabolite must be either positive or negative. For further details, see the Supplementary Material Section 3.2.5.
Furthermore, transport reactions cannot be chosen as part of the solution and therefore glucose, pyruvate and succinate were added as sources for the corresponding pulse experiment. Oxygen was added as another source because in (Taymaz-Nikerel et al., 2013), the authors identified increased

oxygen uptake rates during the pulse experiment. To allow unlimited growth, the biomass was added as sink.
The expected active reactions in the core metabolism of *E. coli* are displayed in Figure 3.1 for each pulse experiment.

3.3.1 E. coli core model

At first, the method was applied using the *E. coli* core model. To better understand how the different parts of our model impacted the solutions, we did several runs with different values for λ (0.1, 0.5 and 0.9) and ϵ (5 and 10) for each pulse experiment. Although a single solution should be enough to identify some pathways responsible for the shift, we wanted to see if we could also obtain alternative pathways. Furthermore, we wanted to investigate how the solutions evolve when they are slightly suboptimal. For each different parameter setting, 100 different solutions were therefore enumerated. The results are displayed using *Escher* (King et al., 2015a) in the Supplementary Figures A.1 to A.15.
In general, we could observe that solutions with $\lambda = 0.1$ are preferable since usually the goal is to have a final solution which is overall more connected. In this way, we were able to extract complete pathways that played a role during the metabolic shifts. This was the case for all three pulse experiments. A higher λ leads to solutions that are less connected since the optimizer prioritizes solutions with fewer active reactions in this case. However, this means that it is difficult to obtain complete pathways as solutions and it might be hard to interpret these solutions biologically. Nevertheless, the user is able to fine-tune the number of reactions in the final solution and the degree of connectivity (for instance, if the goal is to highlight only parts of the complete metabolic network instead of finding a connected pathway). We show this fine-tuning for the case of the glucose pulse, in which decreasing the parameter ϵ was used to obtain more connected pathways (see Figure 3.3).
By adjusting the parameters λ and ϵ, TOTORO could propose complete pathways for all three pulse experiments. The predicted solutions did not use co-factors as shortcuts through the network. We therefore did not modify our method further to treat co-factors separately.

Pyruvate pulse

For the pyruvate pulse, we expected that the activity of the TCA cycle would go up and that reactions for gluconeogenesis would be active (see Figure 3.1). Both observations could be reproduced for $\lambda = 0.1$ for $\epsilon = 5$ or 10. The TCA cycle was proposed to be active by TOTORO in all the 100 enumerated solutions. The four measured metabolites citrate, isocitrate, L-malate and fumarate had positive input deltas and could thus be used as sinks. The results showed how the TCA cycle was fed from pyruvate either by the Phosphoenolpyruvate carboxylase (PPC) or by the combination of Pyruvate dehydrogenase (PDH) and Citrate synthase (CS). Furthermore, the pathway from pyruvate to D-fructose 6-phosphate was active, thereby also producing the biomass precursor glyceraldehyde 3-phosphate (G3P) in all of the 100 solutions. The pathway from pyruvate to G3P contains five reactions including the reversible reactions Enolase (ENO), Phosphoglycerate mutase (PGM), Phosphoglycerate kinase (PGK) and Glyceraldehyde-3-phosphate dehydrogenase (GAPD). Especially here, it is important to state that all these reversible reactions were predicted in the correct direction going from pyruvate towards G3P. One important difference between the results for $\epsilon = 5$ and $\epsilon = 10$ was that for $\epsilon = 5$, the biomass reaction was chosen in all solutions which makes it slightly

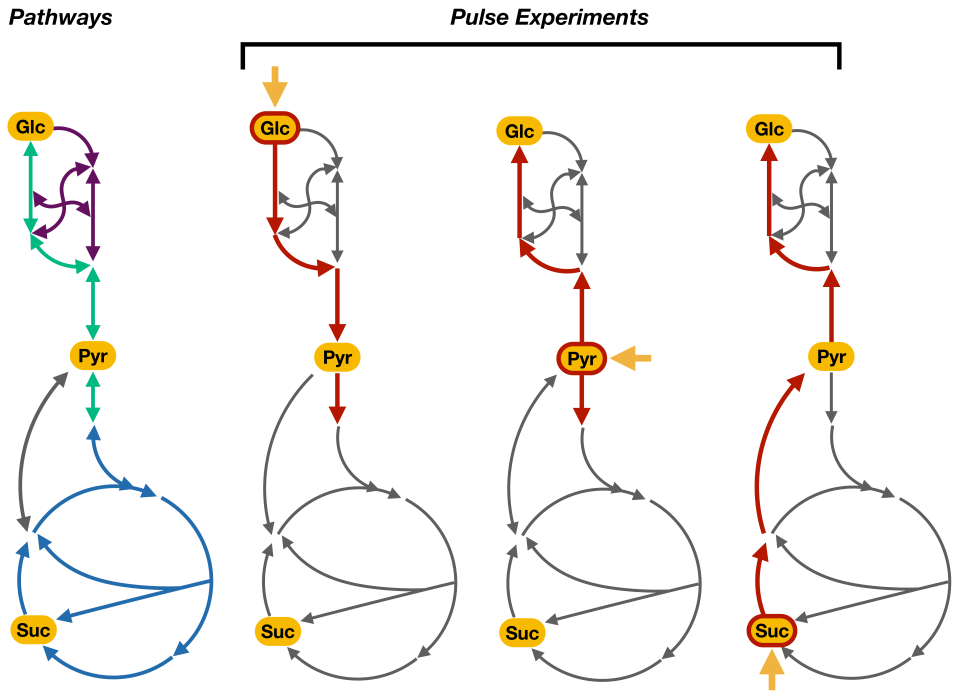


Figure 3.1: *Expected active reactions for different pulse experiments.* These essential reactions along with their expected directions are highlighted in red whereas other non-essential reactions (but which nonetheless could be chosen) are depicted in grey. Each pulse is indicated by the short yellow arrow (Glc: glucose; Pyr: pyruvate and Suc: Succinate). During the glucose pulse, the glycolysis reactions (depicted in green) should be active in order to generate ATP from the hydrolysis of glucose. On the other hand, the pyruvate and succinate pulse experiments should show gluconeogenesis activation (also depicted in green but in the opposite sense), generating glucose-6-phosphate from these two carbon sources. Furthermore, the TCA cycle (depicted in blue) can be fed from pyruvate during the pyruvate and glucose pulses. During the succinate pulse, the overflow in the TCA cycle should lead to the production of pyruvate with a subsequent activation of gluconeogenesis to produce biomass precursors. The pentose phosphate pathway (depicted in purple) is most likely active in all pulses in order to generate biomass precursors; however, since this pathway is a mere interconversion of carbohydrates, there is no particular expectation as to the actual direction of these reactions.

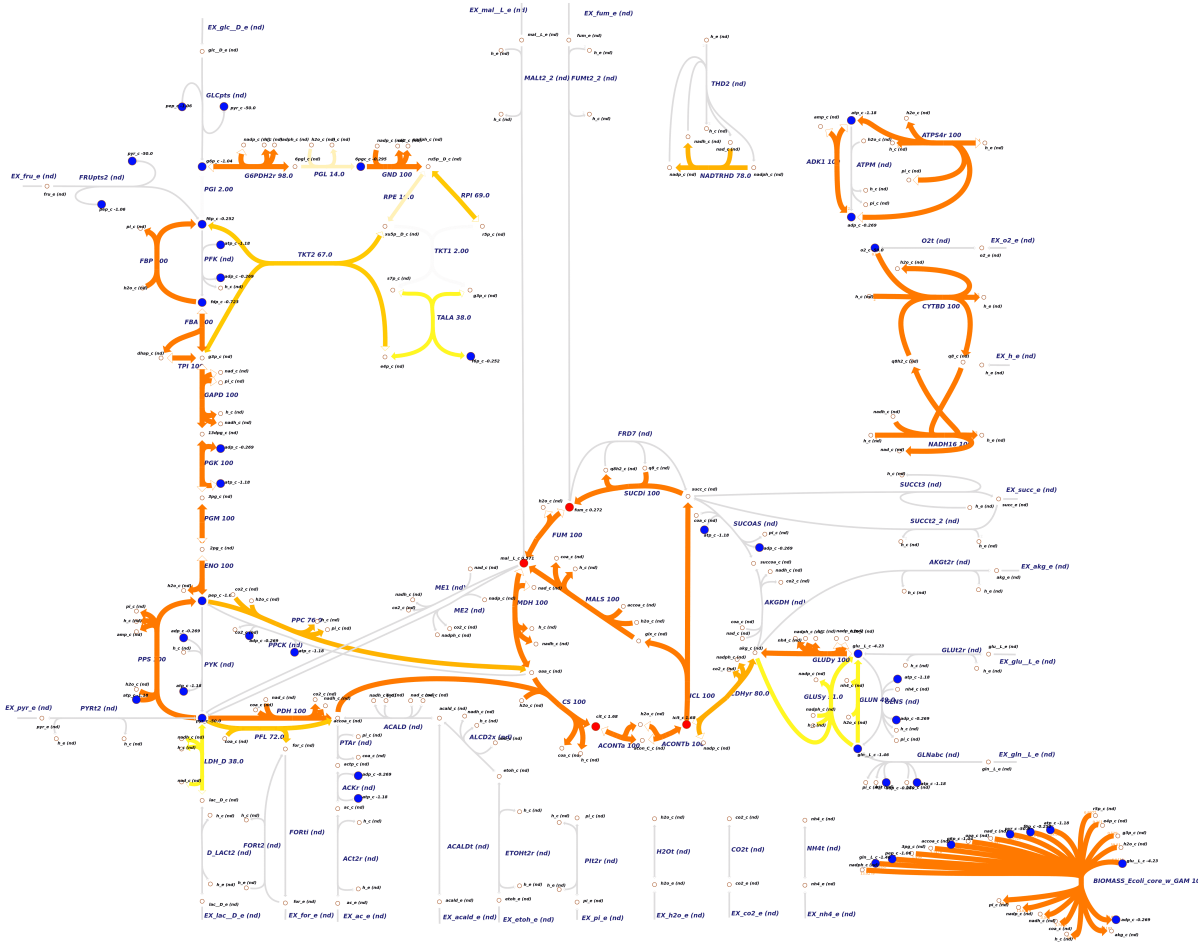


Figure 3.2: *E. coli core model - Results for pyruvate pulse ($\lambda = 0.1$, $\epsilon = 5$).* The metabolites that were given as input are highlighted in blue if the corresponding input deltas were below zero and red if they were above zero. Reactions that are highlighted in orange were chosen in almost all of the enumerated solutions. Reactions that are yellow were chosen only in around half of the solutions. White reactions were not chosen in any solution. The expected reactions of the gluconeogenesis and part of the TCA cycle are active in all 100 solutions. The reversible reactions of the gluconeogenesis are chosen in the correct direction. The figure was created using *Escher* (King et al., 2015a).

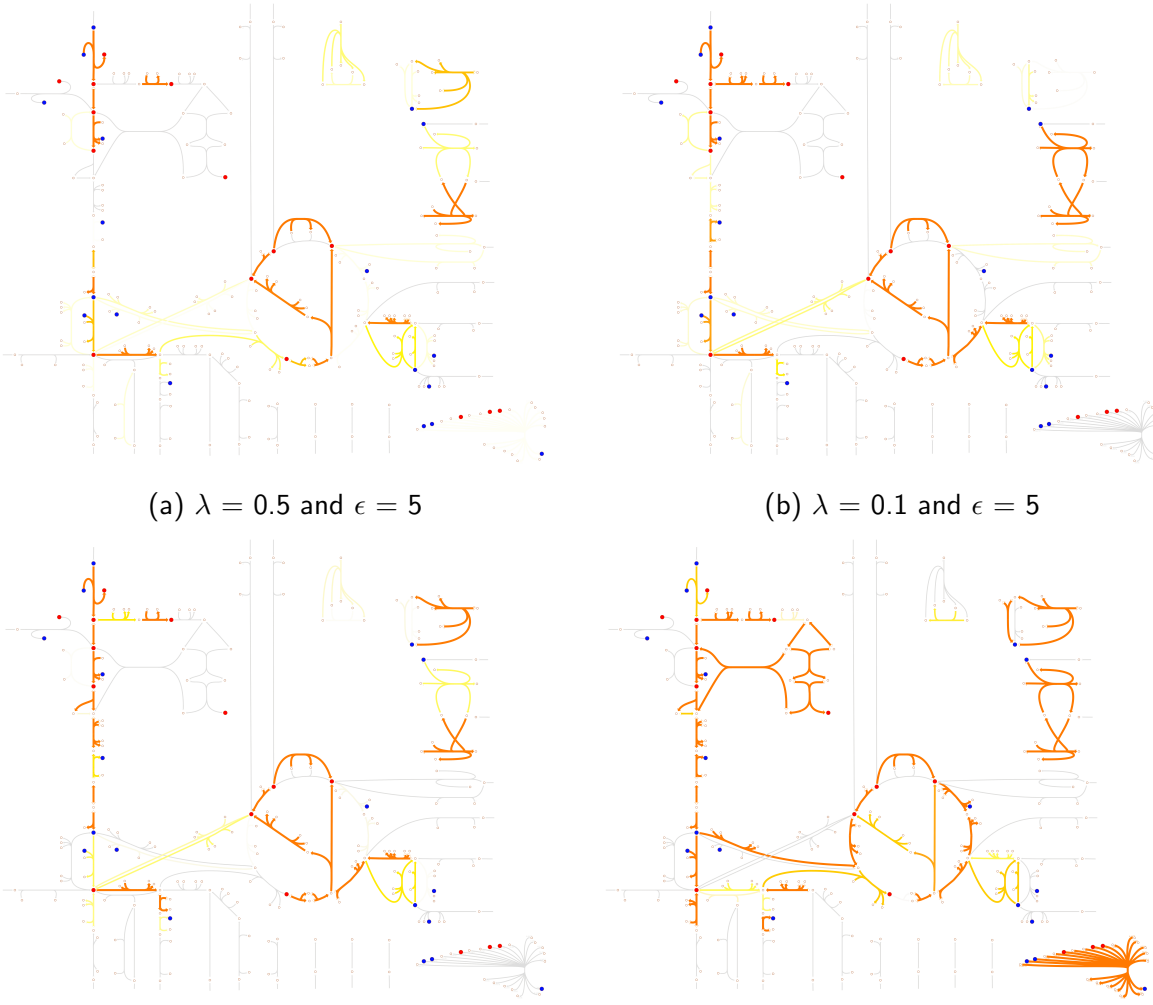
Pulse experiment	1st sol.	100th sol.	Abs. diff.	%
Pyruvate ($\lambda = 0.1$, $\epsilon = 5$)	-32.1394	-30.6635	1.48	5.5%
Glucose ($\lambda = 0.1$, $\epsilon = 1.2$)	5.3830	6.5582	1.18	21.8%
Succinate ($\lambda = 0.1$, $\epsilon = 5$)	-158.1770	-157.5760	0.60	0.4%

Table 3.1: *Comparison of different objective values for the best runs for each experiment.* Since we are not fixing the objective value of the first solution in our optimization problem, the objective values for the subsequent solutions can be worse. In this table, we are comparing the difference in the objective values between the first solution and the 100th solution. In addition to the absolute differences, also the percentage of how much the objective value worsened compared to the first solution is displayed. The underlying optimization problem is a minimization problem. Therefore, smaller objective values are better.

preferable. Besides the biomass precursor G3P, TOTORO proposed the production of alpha-D-ribose 5-phosphate (R5P) via ribose-5-phosphate isomerase (RPI) and the production of D-erythrose 4-phosphate (E4P) via Transketolase (TKT2). The results for $\lambda = 0.1$ and $\epsilon = 5$ are shown in Figure 3.2 (see Supplementary Figure A.5 for $\epsilon = 10$).
For $\lambda = 0.9$ (see Supplementary Figures A.1 and A.2), neither the TCA nor the gluconeogenesis pathway were proposed to be active. Setting λ to 0.5 already improved the results: the TCA cycle was proposed as active but the gluconeogenesis pathway was only recovered in less than 50% of the solutions (see Supplementary Figures A.3 and A.4).
We do not fix the objective value in our optimization problem after obtaining the first solution but in every iteration, the minimization problem is solved again after excluding the newly found solution. This means that the next solution can have the same objective value but it is also possible that the objective value is worse than in the previous iteration. In this particular case, the 100th solution had an objective value that is only 5.5% worse than the objective value of the first solution (see Table 3.1) which shows that, as concerns optimality, all 100 solutions were very similar. They also had very similar active reactions. Comparing the 100 enumerated solutions for $\lambda = 0.1$ and $\epsilon = 5$, a total of 43 reactions with a specific direction were chosen in all solutions. Out of these 43 reactions, 24 were chosen in every solution (including reactions in the TCA cycle and the gluconeogenesis pathway). This means that certain core pathways were consistently picked also in slightly suboptimal solutions. Looking at only the ten best solutions, already 38 out of the 43 reactions were identified. The missing reactions were mostly part of the pentose phosphate pathway which also contains reactions that were part of the solution only in a few cases. Even with only ten solutions, we were able to obtain the alternative pathways feeding the TCA cycle (PPC/PDH). This indicates that it is not necessary to enumerate a large amount of solutions to get significant results and to identify alternative pathways.

Glucose pulse

For the glucose pulse, we expected that reactions that are part of the glycolysis pathway would be active as they convert glucose into pyruvate generating energy. Consequently, the TCA cycle should also be fed (see Figure 3.1). For $\lambda = 0.9$ and 0.5, the active reactions proposed by TOTORO were disconnected and it was not possible to identify active pathways. However, even for $\lambda = 0.1$ and $\epsilon = 5$, only disconnected parts of the network were active (see Figure 3.3). Since we were interested in connected pathways, we decided to fine-tune the solutions by lowering the value of ϵ as much as possible. The result for $\epsilon = 1.2$ can be found in Figure 3.3. Lowering the value of ϵ to 1.1 rendered



(a) $\lambda = 0.5$ and $\epsilon = 5$

(b) $\lambda = 0.1$ and $\epsilon = 5$

(c) $\lambda = 0.1$ and $\epsilon = 2$

(d) $\lambda = 0.1$ and $\epsilon = 1.2$

Figure 3.3: *E. coli core model - Results for glucose pulse.* The labels for reaction and metabolite names were omitted. The metabolites that were given as input are highlighted in blue if the corresponding input deltas were below zero and red if they were above zero. Reactions that are highlighted in orange were chosen in almost all of the enumerated solutions. Reactions that are yellow were chosen only in around half of the solutions. White reactions were not chosen in any solution. In **(a)** and **(b)**, even for $\lambda = 0.1$ the active reactions remain disconnected. In **(c)** and **(d)**, lowering ϵ allowed for a fine-tuning of the solution which made it possible to obtain complete pathways. All figures were created using *Escher* (King et al., 2015a).

the underlying optimization problem infeasible. As previously mentioned, decreasing the value of ϵ allowed us to obtain more connected solutions. For $\epsilon = 1.2$, we got solutions that linked intermediate metabolites of the glycolysis pathway to the TCA cycle through the PPC reaction. Followed by a reversed Malate dehydrogenase (MDH), reversed Fumarase (FUM) and Fumarate reductase (FRD7), the solutions allowed for an accumulation of the input metabolites L-malate, fumarate and succinate. However, in some solutions, the TCA cycle was additionally fed by PDH and Citrate synthase (CS) to account for the accumulation of citrate. This means that when the solutions are disconnected and this is unwanted, better results might be obtained by lowering the value of parameter ϵ.
Again, the 100 solutions were very similar ($\lambda = 0.1$, $\epsilon = 1.2$). They accounted for a total of 47 reactions (with distinct directions) and 30 of these appeared in all solutions. Similarly to the pyruvate pulse, the difference in these solutions were mostly based on a few reactions that are not part of the main pathways (glycolysis/TCA cycle). One critical observation is that the D-glucose transport reaction (GLCpts) was not part of every solution although glucose should be used as important source. However, the bacteria were already grown in glucose as the baseline, which in turn might be a reason why glucose was already internalized prior to the initial pulse. When comparing the objective values for these 100 solutions, the absolute difference between the first solution and the 100th one was similar to the one observed for the pyruvate pulse (see Table 3.1). However, proportionally this value was 21.8% worse than for the first solution. When we repeated the run for $\lambda = 0.1$ and $\epsilon = 1.2$ with 50 iterations, the D-glucose transport reaction was part of 42 solutions. For ten iterations, this reaction was picked in all ten solutions. Hence, the glucose transport reaction was active in solutions with the best objective values. This showed that although the solutions remained very similar, there was a decline in their quality. For the pyruvate pulse, we saw that it is not necessary to enumerate a large amount of solutions.

Succinate pulse

After the succinate pulse, part of the TCA cycle should always be active. Furthermore, the gluconeogenesis pathway should be active to produce G3P and glucose-6-phosphate from succinate. Again, the results for $\lambda = 0.5$ and 0.9 led to smaller solutions that were more disconnected (see Supplementary Figures A.12 - A.15). Therefore, we focused on the analysis of the results for $\lambda = 0.1$ (see Supplementary Figure A.16 and Figure 3.4). For both $\epsilon = 5$ and 10, succinate entered the TCA cycle and turned into oxaloacetate. TOTORO proposed two possibilities to output the excess of the TCA cycle: Either phosphoenolpyruvate (PEP) was produced by PEP carboxykinase (PPCK) or by PEP synthase (PPS) using pyruvate as intermediate substrate. Subsequently, PEP was, as expected, transformed to G3P. The lower right part of the TCA cycle predicted as active can be explained by the fact that the concentration of L-glutamate decreased and the concentration of citrate increased. The active reaction in this part connected these two metabolites. Furthermore, reactions of the pentose phosphate pathway were proposed as active and the biomass precursors R5P, E4P and G3P were produced.
The results for $\epsilon = 5$ and 10 were very similar. For example, one difference was that for $\epsilon = 10$, the reverse D-lactate dehydrogenase (LDH) was predicted to be active in 56 solutions which led to a small accumulation of D-lactate. It does make sense biologically because in general, D-lactate is one of the main products of the fermentation but we do not have the measurements for the concentration of D-lactate for this pulse experiment to actually verify this observation. However, in total, the differences were negligible and in contrast to the glucose pulse, the parameter ϵ had a

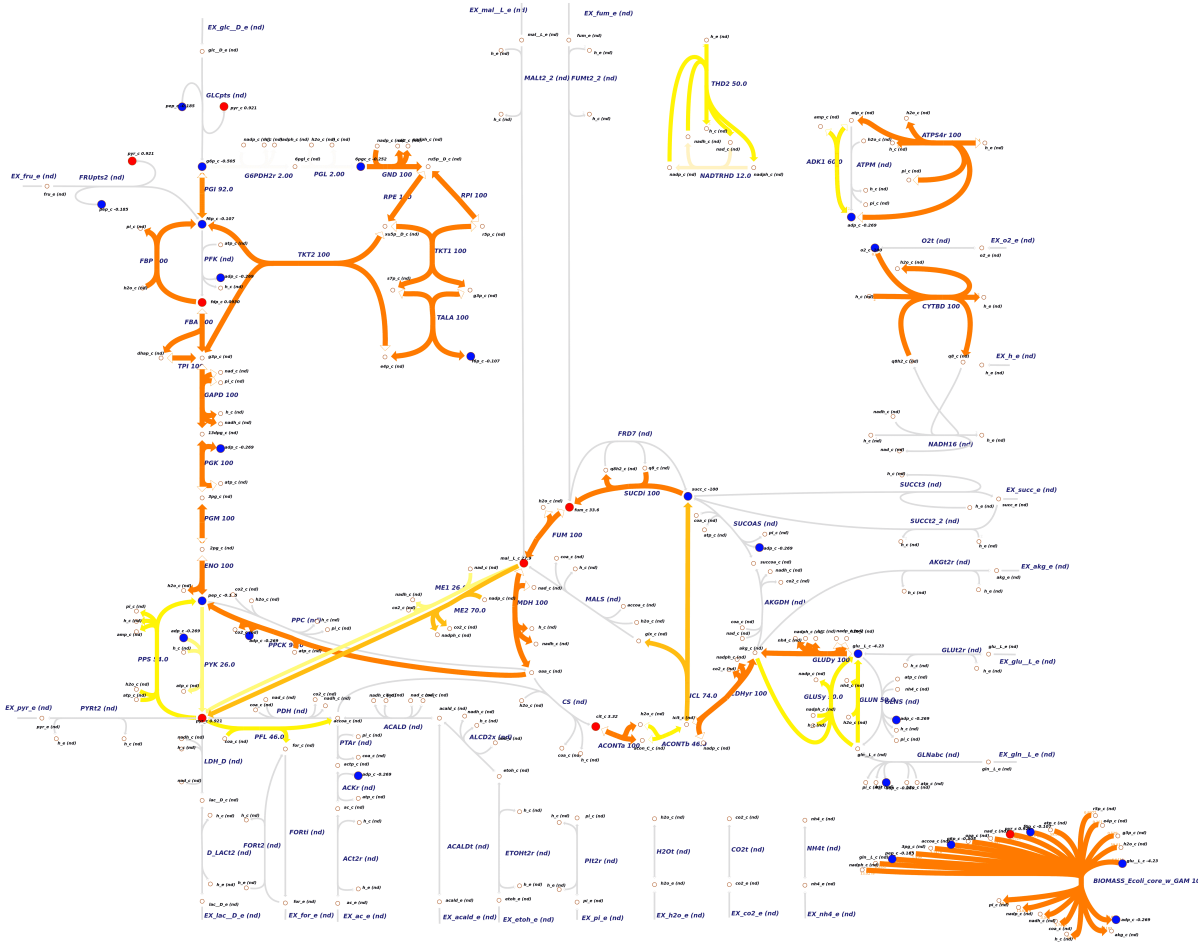


Figure 3.4: *E. coli core model - Results for succinate pulse ($\lambda = 0.1$, $\epsilon = 5$).* The metabolites that were given as input are highlighted in blue if the corresponding input deltas were below zero and red if they were above zero. Reactions that are highlighted in orange were chosen in almost all of the enumerated solutions. Reactions that are yellow were chosen only in around half of the solutions. White reactions were not chosen in any solution. The reactions of the gluconeogenesis pathway and the reactions that transform succinate in the TCA cycle and subsequently into pyruvate are active in all 100 solutions. The figure was created using *Escher* (King et al., 2015a).

lower impact on the outcome.
Again, the core reactions of all 100 solutions were very similar. In total, 41 reactions (with distinct directions) appeared in all 100 solutions (for $\lambda = 0.1$, $\epsilon = 5$). We observed that 22 of these were always active (mostly in the gluconeogenesis pathway and part of the TCA cycle). The objective values for all 100 solutions were extremely close (see Table 3.1).

3.3.2 E. coli iJO1366 model

Based on the results for the *E. coli* core model, we only did runs with $\lambda = 0.1$ for the *E. coli* iJO1366 model. The inputs were updated because this network contains more metabolites and therefore, more measured metabolites could be added. The amount of iterations was decreased to ten because the runtime in the larger network is significantly higher and we had already established in the core model that it is not necessary to enumerate a larger amount of solutions. To decrease the runtime for each solution, CPLEX was configured differently. The relative MIP gap tolerance was set to 0.05 which means that the solver will stop an iteration if a solution is found that is within 5% of the optimal. This allows for a faster result and we could see in the core model that the first 100 solutions tend to be very similar. This means that even if we are enumerating slightly suboptimal solutions, we should be able to compute solutions that are very similar to the actual optimal solution. If the 5% limit is not reached after 48h, the iteration is stopped. The memory usage of CPLEX was limited to 10 GB. The runtime for the different pulse experiments differed a lot. The results for the pyruvate and glucose pulse were computed on a cluster. For the pyruvate pulse, the 5% limit was reached only in three iterations (see Supplementary Table 3.2). All other iterations were stopped after 48h. However, all solutions obtained were within 7% of the optimum. Thus, we still took them into account when analyzing the predicted active reactions. In none of the iterations for the glucose pulse, the 5% limit was reached. The obtained solutions were within 8.5% of the optimal value (see Supplementary Table 3.3).
In contrast to the pyruvate and the glucose pulse, the 5% limit was reached in all iterations for the succinate pulse and computing all ten solutions took less than 5 minutes on a personal machine (2.90GHz Intel i7-7820HQ CPU, 16GB RAM). This shows that the constraints describing the input deltas in the MILP have a large influence on the difficulty of the optimization problem, and thus also on the runtime.
However, although the obtained solutions were suboptimal, the active reactions predicted by Totoro for the core metabolism are very similar to the best results of the *E. coli* core model for all three pulse experiments.
The additional measurements that were added as input deltas for the large network were mostly amino acids (see Supplementary Tables A.6 to A.8). In (Waschina et al., 2016), the authors show for the example of amino acid production in *E. coli* how the production cost for individual amino acids can depend on the available carbon source, and reactions close to the entry point of the carbon source might have considerably higher fluxes (see Figure 3.5). Indeed, from the experimental data, valine only accumulated during the pyruvate pulse, and was depleted with the other two carbon sources. Pyruvate is a direct precursor for valine production. Therefore, we expected that reactions of the valine metabolism should play a greater role in the predicted results for pyruvate compared to the other two pulses. Totoro predicted a higher turnover from pyruvate to valine, which resulted in the accumulation of this amino acid. Even though the pathway was also predicted as active for the glucose pulse, it was consumed more in this case (see Supplementary Table 6). In accordance

Iteration	Objective value	Optimality (%)	Runtime (h)
1	-135.112	5.66	48
2	-134.915	5.48	48
3	-134.879	5.00	29
4	-134.902	6.19	48
5	-134.976	5.00	9
6	-135.309	5.00	23
7	-135.303	6.62	48
8	-135.142	6.31	48
9	-135.117	5.65	48
10	-135.146	6.45	48

Table 3.2: *Results for pyruvate ($\lambda = 0.1$, $\epsilon = 1.0$).* In total, 10 solutions were computed. The table shows the objective value for each solution and how close this value is to the optimum (in %). The solver stopped either if a solution within 5% of the optimal value was found or after 48 hours. Only in three iterations, the 5% limit was reached. In all other iterations, the solver stopped after 48 hours. However, the obtained solutions had objective values with 7% of the optimum. Thus, we still took them into account when analyzing the predicted active reactions.

Iteration	Objective value	Optimality (%)	Runtime (h)
1	-147.134	8.39	48
2	-147.153	8.35	48
3	-147.744	6.58	48
4	-147.75	8.09	48
5	-147.744	7.72	48
6	-147.772	7.47	48
7	-147.809	8.32	48
8	-147.998	7.30	48
9	-148.259	7.47	48
10	-148.331	8.08	48

Table 3.3: *Results for glucose ($\lambda = 0.1$, $\epsilon = 1.0$).* In total, 10 solutions were computed. The table shows the objective value for each solution and how close this value is to the optimum (in %). The solver stopped either if a solution within 5% of the optimal value was found or after 48 hours. The 5% limit was never reached and the solver always stopped after 48 hours. All computed solutions were within 8.50% of the optimal value.

with the predictions in (Waschina et al., 2016), another example is the accumulation of threonine during the succinate pulse. As shown in Figure 3.5, threonine and succinate are closely connected, and TOTORO predicted active reactions leading to its accumulation during the succinate pulse. Compared to the results for succinate, TOTORO predicted more active reactions consuming threonine during the glucose pulse and less active reactions producing threonine during the pyruvate pulse, resulting in the depletion of this amino acid in these two cases (see Supplementary Table 6). Moreover, only during the glucose pulse, phenylalanine was accumulated. As a result, TOTORO proposed more reactions of the phenylalanine metabolism as active compared to the pyruvate and succinate pulses, in accordance with the predictions in (Waschina et al., 2016) of lower cost to produce this amino acid with glucose as carbon source (see Figure 3.5).

Another interesting pathway we noticed as active only in the glucose results was the murein recycling pathway. Murein (or peptidoglycan) is a polymer consisting of sugars and aminoacids and is a major component of cell wall in bacteria. As bacteria grow and the cells divide, this layer of peptidoglycans breaks and the fragments are transported back for reutilization (Goodell, 1985; van Heijenoort, 2011). Indeed, around 60% of the murein sacculus is thought to be cleaved and reused at each generation in *E. coli* (Park and Uehara, 2008). Although merely speculative at this point, as glucose is the carbon source with the highest growth rate for *E. coli*, it is plausible to assume that the higher growth rate when compared to pyruvate or succinate results in a higher amount of murein to be recycled at each generation.

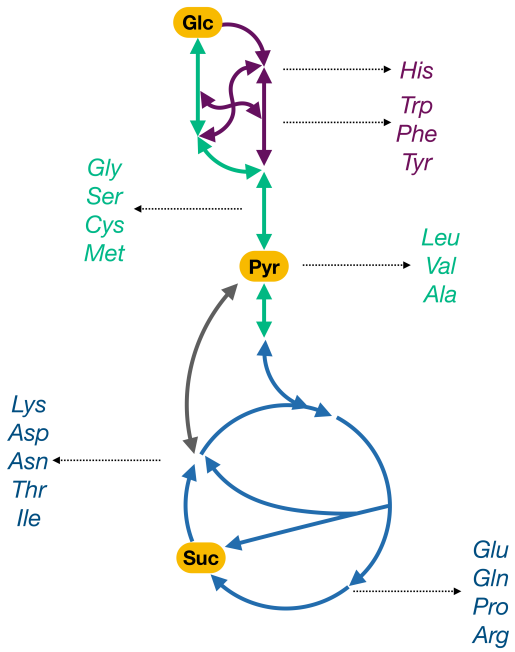


Figure 3.5: *Carbon sources and closely connected amino acids.* The production cost for individual amino acids can depend on the available carbon source, and reactions close to the entry point of the carbon source might have considerably higher fluxes. This figure shows the entry points for different carbon sources and closely connected amino acids. For example, Pyruvate is a direct precursor for valine production.

3.4 Discussion

TOTORO was able to predict expected pathways as active based on the differences in the measured concentrations for some internal metabolites. We could show that in general, it is preferable to use $\lambda = 0.1$ though the method is not critically sensible to this setup, being robust to small perturbations. However, it is worth noting that a higher λ can lead to smaller solutions but they are biologically irrelevant. We were interested in extracting complete pathways that explain the changes in concentration between two different conditions. We could see that a reduction of ϵ could be used to obtain a more connected solution and therefore, in our case, a smaller ϵ led to better solutions. However, there might be situations where we are more interested in only local changes around the measurements. In this context, it might be advantageous to choose a higher ϵ. Furthermore, since we are doing more than just minimizing the number of active reactions, something which would be closer to looking for shortest paths, we did not encounter problems specific to co-factors. By splitting reversible reactions, TOTORO is able to predict distinct directions for them.
Both in the core network and in the larger network, we were able to recover pathways that make sense biologically. Additionally, although the larger network contains more reactions and we added more input deltas, the predictions for the core metabolism of *E. coli* were similar to the results for the core network. It must be however noted that the predictions do depend on the measured metabolites. If for large parts of the network, no metabolite concentrations are measured, TOTORO will likely not be able to find active pathways for these parts of the network.
Moreover, we could also see that it is not necessary to enumerate a high number of solutions which is especially important when larger networks are used and the runtime of TORORO increases. We enumerated 100 different solutions for the core network. However, in our case, the enumerated solutions are very similar and a large amount of reactions appeared in all 100 solutions. Therefore, already one solution would have been sufficient to infer the most important reactions that were proposed to be active.

3.5 Conclusion

In this work, we presented TOTORO, a method that identifies active reactions during the transient state based on the differences in the concentrations for some measured metabolites in two different states and we showed its prediction power on the example of different pulse experiments in *E. coli*. Our method TOTORO only uses metabolomic data as basis for the prediction. It is able to handle full networks which take into account in the model stoichiometry, cycles, reversible reactions as well as co-factors.
With the current developments, it gets more common to have different kinds of data available which creates a need for methods that combine, for instance, metabolomic, transcriptomic and proteomic data. In (Pandey et al., 2019), the authors proposed a framework that predicts differential fluxes. It consists of three different methods that use either thermodynamic and transcriptomic data or thermodynamic and metabolomic data or thermodynamic, transcriptomic and metabolomic data at the same time. It might then be interesting in the future to adapt our approach to be able to integrate different kinds of data.

Conclusions and perspectives

In this thesis, I presented two different methods for the analyses of metabolisms of microorganisms. One approach focused on the identification of knockout sets, the other one on the analysis of metabolic shifts.

The first method that I described was the main work of my PhD. It can be used to identify knockout sets that increase the production of a valuable target metabolite in a microorganism in the scenario where the target metabolite is toxic for the microorganism and its accumulation can therefore inhibit growth or lead to a decline in its production. In the given approach, the resistance of the microorganism against the toxic target was measured based on the activity of some critical reactions that were identified experimentally beforehand.

In the studied example of ethanol production in yeast, already in the first part of the approach where tradeoffs between biomass and ethanol productions and the toxicity resistance score are calculated, we could indeed see that there are flux distributions that have very similar biomass and ethanol productions but that differed significantly in the calculated toxicity resistance score. This showed that not all of the critical reactions are necessary for growth and ethanol production. Hence, accounting specifically for the toxicity is important to ensure that the microorganism can keep its natural resistance. We could further demonstrate the advantage of the toxicity resistance score by comparing different hyperpaths in the second part of our approach.

Applying the random exploration, we were able to identify smaller subsets of the outgoing reactions for some hyperpaths that can be used as knockout sets. However, we still need to obtain a more biological examination of the computed knockout sets to establish how viable they would be in practice.

One of the main advantages of our approach is that it is applicable to genome-scale metabolic networks. Moreover, the described framework is flexible and it should be possible to adapt it to identify knockout sets for different examples, not just cases where the target is toxic for the microorganism. We plan to submit our method as paper after receiving a biological view of our results from our collaborator. Furthermore, the implementation in C++ that uses PolyScip as solver for the multi-objective optimization problems and CPLEX for all other MILPs will be made available on the Gitlab of our group.

The second method that I presented in this thesis is called Totoro and it can be used to analyze metabolic shifts. Totoro predicts reactions that are active during the transient state that occurs after a perturbation. It minimizes the change in concentrations for unmeasured metabolites and also the number of active reactions during the transient state to account for a parsimonious assumption. It predicts distinct directions for reversible reactions. An implementation of Totoro in C++ that uses CPLEX as solver for the underlying MILPs is available at `https://gitlab.inria.fr/erable/totoro`.

On the example of three different pulse experiments in *E. coli*, we could show that this constraint-based approach is also applicable to larger network models. We could reproduce the main observations obtained in the *E. coli* core model in the *E. coli* iJO1366 model. Furthermore, we showed that by limiting the accumulation/depletion of unmeasured metabolites in the MILP and also prioritizing their minimization in the objective function, Totoro can predict connected pathways. We did not encounter problems specific to co-factors.

With the current developments, it gets more common to have different kinds of data available.

Hence, it might be interesting to see if our method is adaptable to also integrate, for instance, transcriptomic and/or proteomic data.

List of Figures

List of Tables

Bibliography

Achterberg, T. (2009). Scip: solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41.

Achterberg, T., Berthold, T., Koch, T., and Wolter, K. (2008). Constraint integer programming: A new approach to integrate cp and mip. In Perron, L. and Trick, M. A., editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 6–20, Berlin, Heidelberg. Springer Berlin Heidelberg.

Acuña, V., Birmelé, E., Cottret, L., Crescenzi, P., Jourdan, F., Lacroix, V., Marchetti-Spaccamela, A., Marino, A., Milreu, P. V., Sagot, M.-F., and Stougie, L. (2012a). Telling stories: Enumerating maximal directed acyclic graphs with a constrained set of sources and targets. *Theoretical Computer Science*, 457(457):1–9.

Acuña, V., Milreu, P. V., Cottret, L., Marchetti-Spaccamela, A., Stougie, L., and Sagot, M.-F. (2012b). Algorithms and complexity of enumerating minimal precursor sets in genome-wide metabolic networks. *Bioinformatics*, 28(19):2474–2483.

Alonso, A., Marsal, S., and Julià, A. (2015). Analytical methods in untargeted metabolomics: state of the art in 2015. *Frontiers in bioengineering and biotechnology*, 3:23.

Alper, H., Moxley, J., Nevoigt, E., Fink, G. R., and Stephanopoulos, G. (2006). Engineering yeast transcription machinery for improved ethanol tolerance and production. *Science*, 314(5805):1565–1568.

Altman, T., Travers, M., Kothari, A., Caspi, R., and Karp, P. D. (2013). A systematic comparison of the MetaCyc and KEGG pathway databases. *BMC bioinformatics*, 14(1):112.

Andrade, R., Doostmohammadi, M., Santos, J. L., Sagot, M.-F., Mira, N. P., and Vinga, S. (2020). Momo-multi-objective metabolic mixed integer optimization: application to yeast strain engineering. *BMC bioinformatics*, 21(1):1–13.

Andrade, R., Wannagat, M., Klein, C. C., Acuña, V., Marchetti-Spaccamela, A., Milreu, P. V., Stougie, L., and Sagot, M.-F. (2016). Enumeration of minimal stoichiometric precursor sets in metabolic networks. *Algorithms for Molecular Biology*, 11(1):25.

Antunes, C. H., Alves, M. J., and Clímaco, J. (2016). *Multiobjective linear and integer programming*. Springer.

Ausiello, G., Franciosa, P. G., and Frigioni, D. (2001). Directed hypergraphs: Problems, algorithmic results, and a novel decremental approach. In *Italian conference on theoretical computer science*, pages 312–328. Springer.

Bai, F., Anderson, W., and Moo-Young, M. (2008). Ethanol fermentation technologies from sugar and starch feedstocks. *Biotechnology advances*, 26(1):89–105.

Ballerstein, K., von Kamp, A., Klamt, S., and Haus, U.-U. (2012). Minimal cut sets in a metabolic network are elementary modes in a dual network. *Bioinformatics*, 28(3):381–387.

Berge, C. (1973). *Graphs and Hypergraphs (North-Holland Mathematical Library; V. 6)*. Elsevier.

Bertsimas, D. and Tsitsiklis, J. N. (1997). *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA.

Booth, S. C., Weljie, A. M., and Turner, R. J. (2013). Computational tools for the secondary analysis of metabolomics experiments. *Computational and structural biotechnology journal*, 4(5):e201301003.

Borndörfer, R., Schenker, S., Skutella, M., and Strunk, T. (2016). Polyscip. In Greuel, G.-M., Koch, T., Paule, P., and Sommese, A., editors, *Mathematical Software - ICMS 2016, 5th International Conference, Berlin, Germany, July 11-14, 2016, Proceedings*, volume 9725, pages 259 – 264.

Borodina, I. and Nielsen, J. (2014). Advances in metabolic engineering of yeast saccharomyces cerevisiae for production of chemicals. *Biotechnology journal*, 9(5):609–620.

Boulton, R. B., Singleton, V. L., Bisson, L. F., and Kunkee, R. E. (1999). Yeast and biochemistry of ethanol fermentation. In *Principles and practices of winemaking*, pages 102–192. Springer.

Burgard, A. P., Nikolaev, E. V., Schilling, C. H., and Maranas, C. D. (2004). Flux coupling analysis of genome-scale metabolic network reconstructions. *Genome research*, 14(2):301–312.

Burgard, A. P., Pharkya, P., and 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.

Burgard, A. P., Vaidyaraman, S., and Maranas, C. D. (2001). Minimal reaction sets for escherichia coli metabolism under different growth requirements and uptake environments. *Biotechnology progress*, 17(5):791–797.

Cambiaghi, A., Ferrario, M., and Masseroli, M. (2017). Analysis of metabolomic data: tools, current strategies and future challenges for omics data integration. *Briefings in Bioinformatics*, 18(3):498–510.

Case, A., Lutz, J. H., and Stull, D. M. (2016). Reachability problems for continuous chemical reaction networks. In *International Conference on Unconventional Computation and Natural Computation*, pages 1–10. Springer.

Casey, G. P. and Ingledew, W. M. (1986). Ethanol tolerance in yeasts. *CRC Critical Reviews in Microbiology*, 13(3):219–280.

Caspi, R., Billington, R., Ferrer, L., Foerster, H., Fulcher, C. A., Keseler, I. M., Kothari, A., Krummenacker, M., Latendresse, M., Mueller, L. A., Ong, Q., Paley, S., Subhraveti, P., Weaver, D. S., and Karp, P. D. (2016). The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Research*, 44(D1):D471.

Chagoyen, M. and Pazos, F. (2011). Mbrole: enrichment analysis of metabolomic data. *Bioinformatics*, 27(5):730–731.

Chong, J., Soufan, O., Li, C., Caraus, I., Li, S., Bourque, G., Wishart, D. S., and Xia, J. (2018). MetaboAnalyst 4.0: towards more transparent and integrative metabolomics analysis. *Nucleic Acids Research*, 46(W1):W486–W494.

Christensen, C. D., Hofmeyr, J.-H. S., and Rohwer, J. M. (2015). Tracing regulatory routes in metabolism using generalised supply-demand analysis. *BMC systems biology*, 9(1):89.

Cottret, L., Frainay, C., Chazalviel, M., Cabanettes, F., Gloaguen, Y., Camenen, E., Merlet, B., Heux, S., Portais, J.-C., Poupin, N., et al. (2018). Metexplore: collaborative edition and exploration of metabolic networks. *Nucleic acids research*, 46(W1):W495–W502.

Cottret, L. and Jourdan, F. (2010). Graph methods for the investigation of metabolic networks in parasitology. *Parasitology*, 137(9):1393.

Cottret, L., Milreu, P. V., Acuña, V., Marchetti-Spaccamela, A., Martinez, F. V., Sagot, M.-F., and Stougie, L. (2008). Enumerating precursor sets of target metabolites in a metabolic network. In *International Workshop on Algorithms in Bioinformatics*, pages 233–244. Springer.

Covert, M. W. and Palsson, B. O. (2003). Constraints-based models: regulation of gene expression reduces the steady-state solution space. *Journal of theoretical biology*, 221(3):309–325.

D'amore, T., Panchal, C. J., Russell, I., and Stewart, G. (1989). A study of ethanol tolerance in yeast. *Critical reviews in biotechnology*, 9(4):287–304.

D'Amore, T. and Stewart, G. G. (1987). Ethanol tolerance of yeast. *Enzyme and Microbial Technology*, 9(6):322–330.

Dantzig, G. B. (1998). *Linear programming and extensions*, volume 48. Princeton university press.

Dantzig, G. B. and Thapa, M. N. (2006). *Linear programming 1: introduction*. Springer Science & Business Media.

Dombek, K. and Ingram, L. (1986). Magnesium limitation and its role in apparent toxicity of ethanol during yeast fermentation. *Applied and environmental microbiology*, 52(5):975–981.

Durot, M., Bourguignon, P.-Y., and Schachter, V. (2008). Genome-scale models of bacterial metabolism: reconstruction and applications. *FEMS microbiology reviews*, 33(1):164–190.

Edwards, J. S., Ibarra, R. U., and Palsson, B. O. (2001). In silico predictions of escherichia coli metabolic capabilities are consistent with experimental data. *Nature biotechnology*, 19(2):125–130.

Edwards, J. S. and Palsson, B. O. (2000). Robustness analysis of the escherichiacoli metabolic network. *Biotechnology progress*, 16(6):927–939.

Ehrgott, M. (2005). *Multicriteria Optimization*. Springer Science & Business Media.

Feist, A. M. and Palsson, B. O. (2010). The biomass objective function. *Current opinion in microbiology*, 13(3):344–349.

Fiehn, O. (2002). Metabolomics—the link between genotypes and phenotypes. In *Functional genomics*, pages 155–171. Springer.

Frainay, C., Aros, S., Chazalviel, M., Garcia, T., Vinson, F., Weiss, N., Colsch, B., Sedel, F., Thabut, D., Junot, C., et al. (2019). Metaborank: network-based recommendation system to interpret and enrich metabolomics results. *Bioinformatics*, 35(2):274–283.

Frainay, C. and Jourdan, F. (2017). Computational methods to identify metabolic sub-networks based on metabolomic profiles. *Briefings in Bioinformatics*, 18(1):43–56.

Gass, S. I. (2003). *Linear programming: methods and applications*. Courier Corporation.

Ginsburg, H. (2009). Caveat emptor: limitations of the automated reconstruction of metabolic pathways in plasmodium. *Trends in parasitology*, 25(1):37–43.

Gleixner, A., Eifler, L., Gally, T., Gamrath, G., Gemander, P., Gottwald, R. L., Hendel, G., Hojny, C., Koch, T., Miltenberger, M., Müller, B., Pfetsch, M. E., Puchert, C., Rehfeldt, D., Schlösser, F., Serrano, F., Shinano, Y., Viernickel, J. M., Vigerske, S., Weninger, D., Witt, J. T., and Witzig, J. (2017a). The SCIP Optimization Suite 5.0. Technical report, Optimization Online.

Gleixner, A., Eifler, L., Gally, T., Gamrath, G., Gemander, P., Gottwald, R. L., Hendel, G., Hojny, C., Koch, T., Miltenberger, M., Müller, B., Pfetsch, M. E., Puchert, C., Rehfeldt, D., Schlösser, F., Serrano, F., Shinano, Y., Viernickel, J. M., Vigerske, S., Weninger, D., Witt, J. T., and Witzig, J. (2017b). The SCIP Optimization Suite 5.0. ZIB-Report 17-61, Zuse Institute Berlin.

Goel, A., Wortel, M. T., Molenaar, D., and Teusink, B. (2012). Metabolic shifts: a fitness perspective for microbial cell factories. *Biotechnology letters*, 34(12):2147–2160.

Goodell, E. (1985). Recycling of murein by escherichia coli. *Journal of bacteriology*, 163(1):305–310.

Gottstein, W., Olivier, B. G., Bruggeman, F. J., and Teusink, B. (2016). Constraint-based stoichiometric modelling from single organisms to microbial communities. *Journal of the Royal Society Interface*, 13(124):20160627.

Gurobi Optimization, L. (2020). Gurobi optimizer reference manual.

Hädicke, O. and Klamt, S. (2011). Computing complex metabolic intervention strategies using constrained minimal cut sets. *Metabolic engineering*, 13(2):204–213.

Handl, J., Kell, D. B., and Knowles, J. (2007). Multiobjective optimization in bioinformatics and computational biology. *IEEE/ACM Transactions on computational biology and bioinformatics*, 4(2):279–292.

Heavner, B. D., Smallbone, K., Barker, B., Mendes, P., and Walker, L. P. (2012). Yeast 5–an expanded reconstruction of the saccharomyces cerevisiae metabolic network. *BMC systems biology*, 6(1):55.

Heirendt, L., Arreckx, S., Pfau, T., Mendoza, S. N., Richelle, A., Heinken, A., Haraldsdóttir, H. S., Wachowiak, J., Keating, S. M., Vlasov, V., et al. (2019). Creation and analysis of biochemical constraint-based models using the cobra toolbox v. 3.0. *Nature protocols*, 14(3):639–702.

IBM (2016). IBM ILOG CPLEX 12.6 User Manual IBM Corp.

IBM (2019). IBM ILOG CPLEX Optimization Studio CPLEX.

Ivanisevic, J. and Want, E. J. (2019). From samples to insights into metabolism: Uncovering biologically relevant information in lc-hrms metabolomics data. *Metabolites*, 9(12):308.

Julien-Laferrière, A. (2016). *Models and algorithms applied to metabolism: From revealing the responses to perturbations towards the design of microbial consortia*. PhD thesis, Université Claude Bernard Lyon 1.

Kanehisa, M. and Goto, S. (2000). KEGG: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30.

Karp, P. D., Billington, R., Caspi, R., Fulcher, C. A., Latendresse, M., Kothari, A., Keseler, I. M., Krummenacker, M., Midford, P. E., Ong, Q., et al. (2019). The biocyc collection of microbial genomes and metabolic pathways. *Briefings in Bioinformatics*, 20(4):1085–1093.

Karp, P. D., Ong, W. K., Paley, S., Billington, R., Caspi, R., Fulcher, C., Kothari, A., Krummenacker, M., Latendresse, M., Midford, P. E., et al. (2018). The ecocyc database. *EcoSal Plus*, 8(1).

Kelk, S. M., Olivier, B. G., Stougie, L., and Bruggeman, F. J. (2012). Optimal flux spaces of genome-scale stoichiometric models are determined by a few subnetworks. *Scientific reports*, 2(1):1–7.

Kim, M. K. and Lun, D. S. (2014). Methods for integration of transcriptomic data in genome-scale metabolic models. *Computational and structural biotechnology journal*, 11(18):59–65.

King, Z. A., Dräger, A., Ebrahim, A., Sonnenschein, N., Lewis, N. E., and Palsson, B. O. (2015a). Escher: a web application for building, sharing, and embedding data-rich visualizations of biological pathways. *PLoS Comput Biol*, 11(8):e1004321.

King, Z. A., Lu, J., Dräger, A., Miller, P., Federowicz, S., Lerman, J. A., Ebrahim, A., Palsson, B. O., and Lewis, N. E. (2015b). BiGG Models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Research*, 44(D1):D515–D522.

Klamt, S. (2006). Generalized concept of minimal cut sets in biochemical networks. *Biosystems*, 83(2-3):233–247.

Klamt, S. and Gilles, E. D. (2004). Minimal cut sets in biochemical reaction networks. *Bioinformatics*, 20(2):226–234.

Klamt, S., Hädicke, O., and von Kamp, A. (2014). Stoichiometric and constraint-based analysis of biochemical reaction networks. In *Large-scale networks in engineering and life sciences*, pages 263–316. Springer.

Klamt, S., Haus, U.-U., and Theis, F. (2009). Hypergraphs and cellular networks. *PLOS Computational Biology*, 5(5):1–6.

Klamt, S., Saez-Rodriguez, J., and Gilles, E. D. (2007). Structural and functional analysis of cellular networks with cellnetanalyzer. *BMC systems biology*, 1(1):1–13.

Klamt, S. and von Kamp, A. (2011). An application programming interface for cellnetanalyzer. *Biosystems*, 105(2):162–168.

Kuo, T.-C., Tian, T.-F., and Tseng, Y. J. (2013). 3omics: a web-based systems biology tool for analysis, integration and visualization of human transcriptomic, proteomic and metabolomic data. *BMC Systems Biology*, 7(1):64.

Lacroix, V., Cottret, L., Thébault, P., and Sagot, M.-F. (2008). An introduction to metabolic networks and their structural analysis. *IEEE/ACM transactions on computational biology and bioinformatics*, 5(4):594–617.

Lam, F. H., Ghaderi, A., Fink, G. R., and Stephanopoulos, G. (2014). Engineering alcohol tolerance in yeast. *Science*, 346(6205):71–75.

Lee, S., Phalakornkule, C., Domach, M. M., and Grossmann, I. E. (2000). Recursive milp model for finding all the alternate optima in lp models for metabolic networks. *Computers & Chemical Engineering*, 24(2-7):711–716.

Lewis, N. E., Hixson, K. K., Conrad, T. M., Lerman, J. A., Charusanti, P., Polpitiya, A. D., Adkins, J. N., Schramm, G., Purvine, S. O., Lopez-Ferrer, D., et al. (2010). Omic data from evolved e. coli are consistent with computed optimal growth from genome-scale models. *Molecular systems biology*, 6(1):390.

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

Lin, Y. and Tanaka, S. (2006). Ethanol fermentation from biomass resources: current state and prospects. *Applied microbiology and biotechnology*, 69(6):627–642.

Mahadevan, R. and Schilling, C. (2003). The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metabolic engineering*, 5(4):264–276.

Mahadevan, R., Von Kamp, A., and Klamt, S. (2015). Genome-scale strain designs based on regulatory minimal cut sets. *Bioinformatics*, 31(17):2844–2851.

Maia, P., Rocha, M., and Rocha, I. (2016). In silico constraint-based strain optimization methods: the quest for optimal cell factories. *Microbiol. Mol. Biol. Rev.*, 80(1):45–67.

Marco-Ramell, A., Palau-Rodriguez, M., Alay, A., Tulipani, S., Urpi-Sarda, M., Sanchez-Pla, A., and Andres-Lacueva, C. (2018). Evaluation and comparison of bioinformatic tools for the enrichment analysis of metabolomics data. *BMC bioinformatics*, 19(1):1.

Milreu, P. V., Klein, C. C., Cottret, L., Acuña, V., Birmelé, E., Borassi, M., Junot, C., Marchetti-Spaccamela, A., Marino, A., Stougie, L., Jourdan, F., Crescenzi, P., Lacroix, V., and Sagot, M.-F. (2014). Telling metabolic stories to explore metabolomics data: a case study on the yeast response to cadmium exposure. *Bioinformatics*, 30(1):61–70.

Mukhopadhyay, A. (2015). Tolerance engineering in bacteria for the production of advanced biofuels and chemicals. *Trends in microbiology*, 23(8):498–508.

Mussatto, S. I., Dragone, G., Guimarães, P. M., Silva, J. P. A., Carneiro, L. M., Roberto, I. C., Vicente, A., Domingues, L., and Teixeira, J. A. (2010). Technological trends, global market, and challenges of bio-ethanol production. *Biotechnology advances*, 28(6):817–830.

Nielsen, J., Larsson, C., van Maris, A., and Pronk, J. (2013). Metabolic engineering of yeast for production of fuels and chemicals. *Current opinion in biotechnology*, 24(3):398–404.

Orth, J. D., Conrad, T. M., Na, J., Lerman, J. A., Nam, H., Feist, A. M., and Palsson, B. Ø. (2011). A comprehensive genome-scale reconstruction of escherichia coli metabolism—2011. *Molecular systems biology*, 7(1):535.

Orth, J. D., Fleming, R. M., and Palsson, B. O. (2010a). Reconstruction and use of microbial metabolic networks: the core escherichia coli metabolic model as an educational guide. *EcoSal plus*.

Orth, J. D., Thiele, I., and Palsson, B. Ø. (2010b). What is flux balance analysis? *Nature biotechnology*, 28(3):245–248.

Palsson, B. (2000). The challenges of in silico biology. *Nature biotechnology*, 18(11):1147–1150.

Pandey, V., Hadadi, N., and Hatzimanikatis, V. (2019). Enhanced flux prediction by integrating relative expression and relative metabolite abundance into thermodynamically consistent metabolic models. *PLoS computational biology*, 15(5):e1007036.

Park, J. T. and Uehara, T. (2008). How bacteria consume their own exoskeletons (turnover and recycling of cell wall peptidoglycan). *Microbiology and Molecular Biology Reviews*, 72(2):211–227.

Patané, A., Jansen, G., Conca, P., Carapezza, G., Costanza, J., and Nicosia, G. (2019). Multi-objective optimization of genome-scale metabolic models: the case of ethanol production. *Annals of Operations Research*, 276(1-2):211–227.

Pearcy, N., Crofts, J. J., and Chuzhanova, N. (2014). Hypergraph models of metabolism. *International Journal of Biological, Veterinary, Agricultural and Food Engineering*, 8(8):752–756.

Perez de Souza, L., Alseekh, S., Brotman, Y., and Fernie, A. R. (2020). Network based strategies in metabolomics data analysis and interpretation: from molecular networking to biological interpretation. *Expert Review of Proteomics*, (just-accepted).

Pramanik, J. and Keasling, J. (1997). Stoichiometric model of escherichia coli metabolism: Incorporation of growth-rate dependent biomass composition and mechanistic energy requirements. *Biotechnology and bioengineering*, 56(4):398–421.

Ranganathan, S., Suthers, P. F., and Maranas, C. D. (2010). OptForce: an optimization procedure for identifying all genetic manipulations leading to targeted overproductions. *PLoS Comput Biol*, 6(4):e1000744.

Reznik, E., Mehta, P., and Segrè, D. (2013). Flux imbalance analysis and the sensitivity of cellular growth to changes in metabolite pools. *PLOS Computational Biology*, 9:1–13.

Roessner, U. and Bowne, J. (2009). What is metabolomics all about? *Biotechniques*, 46(5):363–365.

Rohwer, J. M. and Hofmeyr, J.-H. S. (2008). Identifying and characterising regulatory metabolites with generalised supply–demand analysis. *Journal of theoretical biology*, 252(3):546–554.

Romero, P., Wagg, J., Green, M. L., Kaiser, D., Krummenacker, M., and Karp, P. D. (2005). Computational prediction of human metabolic pathways from the complete human genome. *Genome biology*, 6(1):R2.

Rosato, A., Tenori, L., Cascante, M., Carulla, P. R. D. A., dos Santos, V. A. M., and Saccenti, E. (2018). From correlation to causation: analysis of metabolomics data using systems biology approaches. *Metabolomics*, 14(4):37.

Sajitz-Hermstein, M., Töpfer, N., Kleessen, S., Fernie, A. R., and Nikoloski, Z. (2016). iremet-flux: constraint-based approach for integrating relative metabolite levels into a stoichiometric metabolic models. *Bioinformatics*, 32(17):i755–i762.

Schilling, C. H., Letscher, D., and Palsson, B. Ø. (2000). Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *Journal of theoretical biology*, 203(3):229–248.

Schrijver, A. (1998). *Theory of linear and integer programming*. John Wiley & Sons.

Schuster, S. and Hilgetag, C. (1994). On elementary flux modes in biochemical reaction systems at steady state. *Journal of Biological Systems*, 2(02):165–182.

Segrè, D., Vitkup, D., and Church, G. M. (2002). Analysis of optimality in natural and perturbed metabolic networks. *Proceedings of the National Academy of Sciences*, 99(23):15112–15117.

Sendín, O. H., Vera, J., Torres, N. V., and Banga, J. R. (2006). Model based optimization of biochemical systems using multiple objectives: a comparison of several solution strategies. *Mathematical and Computer Modelling of Dynamical Systems*, 12(5):469–487.

Sevin, D. C., Kuehne, A., Zamboni, N., and Sauer, U. (2015). Biological insights through nontargeted metabolomics. *Current Opinion in Biotechnology*, 34:1 – 8. Systems biology • Nanobiotechnology.

Stanstrup, J., Broeckling, C. D., Helmus, R., Hoffmann, N., Mathé, E., Naake, T., Nicolotti, L., Peters, K., Rainer, J., Salek, R. M., et al. (2019). The metarbolomics toolbox in bioconductor and beyond. *Metabolites*, 9(10):200.

Taymaz-Nikerel, H., De Mey, M., Baart, G., Maertens, J., Heijnen, J. J., and van Gulik, W. (2013). Changes in substrate availability in escherichia coli lead to rapid metabolite, flux and growth rate responses. *Metabolic engineering*, 16:115–129.

Tepper, N. and Shlomi, T. (2009). Predicting metabolic engineering knockout strategies for chemical production: accounting for competing pathways. *Bioinformatics*, 26(4):536–543.

van Heijenoort, J. (2011). Peptidoglycan hydrolases of escherichia coli. *Microbiology and Molecular Biology Reviews*, 75(4):636–663.

Van Uden, N. (1985). Ethanol toxicity and ethanol tolerance in yeasts. In *Annual reports on fermentation processes*, volume 8, pages 11–58. Elsevier.

Van Vleet, J. and Jeffries, T. W. (2009). Yeast metabolic engineering for hemicellulosic ethanol production. *Current Opinion in Biotechnology*, 20(3):300–306.

Vera, J., De Atauri, P., Cascante, M., and Torres, N. V. (2003). Multicriteria optimization of biochemical systems by linear programming: application to production of ethanol by saccharomyces cerevisiae. *Biotechnology and bioengineering*, 83(3):335–343.

von Kamp, A. and Klamt, S. (2014). Enumeration of smallest intervention strategies in genome-scale metabolic networks. *PLoS Comput Biol*, 10(1):e1003378.

von Kamp, A., Thiele, S., Hädicke, O., and Klamt, S. (2017). Use of cellnetanalyzer in biotechnology and metabolic engineering. *Journal of biotechnology*, 261:221–228.

Waschina, S., D'Souza, G., Kost, C., and Kaleta, C. (2016). Metabolic network architecture and carbon source determine metabolite production costs. *The FEBS journal*, 283(11):2149–2163.

Xia, J., Sinelnikov, I. V., Han, B., and Wishart, D. S. (2015). Metaboanalyst 3.0—making metabolomics more meaningful. *Nucleic Acids Research*, 43(W1):W251–W257.

Xia, J., Sinelnikov, I. V., and Wishart, D. S. (2011). Metatt: a web-based metabolomics tool for analyzing time-series and two-factor datasets. *Bioinformatics*, 27(17):2455–2456.

Xia, J. and Wishart, D. S. (2010a). Metpa: a web-based metabolomics tool for pathway analysis and visualization. *Bioinformatics*, 26(18):2342–2344.

Xia, J. and Wishart, D. S. (2010b). Msea: a web-based tool to identify biologically meaningful patterns in quantitative metabolomic data. *Nucleic acids research*, 38(suppl_2):W71–W77.

Yeung, M., Thiele, I., and Palsson, B. Ø. (2007). Estimation of the number of extreme pathways for metabolic networks. *BMC bioinformatics*, 8(1):363.

Zanghellini, J., Ruckerbauer, D. E., Hanscho, M., and Jungreuthmayer, C. (2013). Elementary flux modes in a nutshell: properties, calculation and applications. *Biotechnology journal*, 8(9):1009–1016.

Ziska, I., Andrade, R., Ferrarini, M., Julien-Laferrière, A., Duchemin, L., César Jr., R. M., Mary, A., Vinga, S., and Sagot, M.-F. (2020). Totoro: Identifying active reactions during the transient state for metabolic perturbations. *Bioinformatics*. The paper was submitted in September 2020.

Appendix

Supplementary material for Chapter 2

Id	Biomass	Ethanol	Toxicity
1	0.131931	17.6169	37.0072
2	0.138936	17.6120	36.0071
3	0.145084	17.5654	35.0066
4	0.157094	17.6384	26.0035
5	0.159635	17.5740	29.0040
6	0.165196	17.5216	26.0035
7	0.168954	17.4082	28.0047
8	0.141714	16.0393	43.0142
9	0.180441	16.6031	37.0120
10	0.185103	16.7232	35.0105
11	0.187788	16.6035	36.0115
12	0.154335	15.4269	44.0196
13	0.185612	15.0541	43.0187
14	0.242456	15.2165	37.0185
15	0.248702	15.2362	36.0177
16	0.266104	15.0928	29.0171
17	0.125938	14.1871	49.0216
18	0.232248	14.9783	39.0199
19	0.103374	13.4770	52.0291
20	0.105248	13.8536	51.0233
21	0.180364	13.0074	49.0285
22	0.373706	12.0860	37.0402
23	0.382106	12.0263	36.0399
24	0.394393	12.0021	29.0387
25	0.122139	11.3336	54.0767
26	0.154353	11.0129	54.0684
27	0.317404	11.6915	44.0432
28	0.330150	11.5924	43.0425
29	0.333643	11.3673	43.0431
30	0.360746	11.8979	39.0410
31	0.371885	11.6655	38.0411
32	0.385956	11.7410	36.0409
33	0.387230	11.8536	35.0402
34	0.400338	11.8970	26.0387
35	0.134482	10.5433	54.0787
36	0.156743	10.9324	54.0688
37	0.169067	10.1430	54.0708
38	0.190924	10.5895	54.0542
39	0.195540	10.4200	54.0543

40	0.219911	10.9508	52.0439
41	0.224613	10.3661	53.0514
42	0.229225	10.1968	53.0515
43	0.254288	10.1605	52.0486
44	0.278927	10.4040	50.0478
45	0.293279	10.3044	49.0469
46	0.332376	10.1183	45.0455
47	0.347631	10.2221	43.0447

Table A.2: *Computed tradeoffs between biomass production, ethanol production and toxicity score.* Only the extreme points in the Pareto front were computed. Lower bounds for biomass production was set to 0.1, lower bounds for ethanol production to 10. The list is sorted by descending ethanol production.

Reaction Id	Reaction name	Group
r_0074	4PP-IP5 pyrophosphorylation to 4-5-PP2-IP4	1
r_0083	5-diphosphoinositol-1-2-3-4-6-pentakisphosphate synthase	1
r_0090	6-phosphofructo-2-kinase	2
r_0093	6PP-IP5 pyrophosphorylation to 5-6-PP2-IP4	1
r_0099	acetyl-CoA ACP transacylase	1
r_0142	adenosine kinase	1
r_0195	alpha-alpha-trehalose-phosphate synthase (UDP-forming)	1
r_0226	ATP synthase	2
r_0227	ATPase (cytosolic)	1
r_0231	C-14 sterol reductase	2
r_0243	C-8 sterol isomerase	1
r_0358	diphosphoinositol-1-3-4-6-tetrakisphosphate synthase	1
r_0396	fatty acyl-ACP synthase (n-C8:0ACP)	1
r_0423	fatty-acyl-ACP synthase (n-C10:0ACP)	1
r_0424	fatty-acyl-ACP synthase (n-C12:0ACP)	1
r_0425	fatty-acyl-ACP synthase (n-C14:0ACP)	1
r_0426	fatty-acyl-ACP synthase (n-C14:1ACP)	1
r_0427	fatty-acyl-ACP synthase (n-C16:0ACP)	1
r_0428	fatty-acyl-ACP synthase (n-C16:1ACP)	1
r_0429	fatty-acyl-ACP synthase (n-C18:0ACP)	1
r_0430	fatty-acyl-ACP synthase (n-C18:1ACP)	1
r_0431	fatty-acyl-ACP synthase (n-C18:2ACP)	1
r_0511	glycogen phosphorylase	1
r_0568	inorganic diphosphatase	2
r_0667	isopentenyl-diphosphate D-isomerase	2
r_0721	malonyl-CoA-ACP transacylase	1
r_0770	NADH dehydrogenase	2
r_0858	phosphatidylethanolamine methyltransferase	2
r_0900	phospholipid methyltransferase	2
r_0901	phospholipid methyltransferase	2
r_0908	phosphoribosyl amino imidazolesuccinocarbozamide synthetase	1
r_0916	phosphoribosylpyrophosphate synthetase	2
r_0967	riboflavin synthase	1
r_0974	ribonucleotide reductase	2
r_0976	ribonucleotide reductase	2
r_0978	ribonucleotide reductase	2
r_0980	ribonucleotide reductase	2
r_1040	threonine aldolase	1
r_1051	trehalose-phosphatase	1
r_1172	glycerol transport via channel	2
r_1260	spermidine transport	2

Table A.3: *Critical reactions for resistance against ethanol in yeast.* Reactions in group 1 are associated to genes whose knock out resulted in a biomass production between 50% and 70% of the wild type biomass production under ethanol stress. Reactions in group 2 are connected to genes whose knock out lead to less than 50% of the wild type biomass production.

Sbml Id	Reaction name	Gene associations	#
r_1110	ADP/ATP transporter	(YBL030C or YBR085W or YMR056C)	18
r_0766	NAD kinase	YPL188W	17
r_1112	AKG transporter mitochonrial	YMR241W	17
r_0489	glycerol-3-phosphatase	(YER062C or YIL053W)	14
r_0324	D-sorbitol reductase	YHR104W	13
r_0149	adenylate kinase	YER170W	12
r_1623	5-formethyltetrahydrofolate cyclo-ligase		12
r_1183	L-alanine transport	(YBR068C or YCL025C or YDR046C or YKR039W or YOL020W or YOR348C or YPL265W)	11
r_2037	reduced thioredoxin transport		11
r_0174	aldehyde dehydrogenase	YOR374W	10
r_0658	isocitrate dehydrogenase	(YNL037C and YOR136W)	10
r_1239	oxaloacetate transport	YKL120W	10
r_0713	malate dehydrogenase	YKL085W	9
r_1118	aspartate-glutamate transporter	YPR021C	9
r_0111	acetyl-CoA hydrolase	YBL015W	8
r_0416	fatty-acyl-ACP hydrolase	(YKL182W and YPL231W)	8
r_0940	proline oxidase (NAD)	YLR142W	8
r_1096	(R)-mevalonate transport		8
r_0415	fatty-acyl-ACP hydrolase	(YKL182W and YPL231W)	7
r_0552	hydrogen peroxide reductase	((YGR209C and YLR109W) or (YLR043C and YLR109W))	7
r_1117	aspartate transport	YPR021C	7
r_1777	fatty acid transport		6
r_0113	acetyl-CoA synthetase	YAL054C	5
r_0300	citrate synthase	(YNR001C or YPR001W)	5
r_1030	tetrahydrofolate aminomethyltransferase	(YAL044C and YBR221C and YDR019C and YER178W and YMR189W)	5
r_1780	fatty-acyl-ACP transport		5
r_0012	1-pyrroline-5-carboxylate dehydrogenase		4
r_0104	acetyl-CoA C-acetyltransferase	YPL028W	4
r_0217	aspartate transaminase	YKL106W	4
r_0470	glutamate dehydrogenase	YDL215C	4
r_0731	methylenetetrahydrofolate dehydrogenase	YKR080W	4
r_1781	fatty-acyl-ACP transport		4
r_1888	L-glutamate 5-semialdehyde dehydratase		4
r_1905	L-proline transport		4
r_1997	pantetheine-phosphate adenylyltransferase		4
r_0025	2-isopropylmalate synthase	YNL104C	3
r_0062	3-methyl-2-oxobutanoate decarboxylase	(YGR087C or YLR044C or YLR134W)	3
r_0404	fatty-acid–CoA ligase (hexadecenoate)	YER015W	3
r_1224	L-valine transport	(YBR068C or YBR069C or YCL025C or YDR046C or YKR039W)	3
r_0403	fatty-acid–CoA ligase (hexadecenoate)	(YIL009W or YMR246W or YOR317W)	2

r_0419	fatty-acyl-ACP hydrolase	(YKL182W and YPL231W)	2
r_0442	FMN reductase	YLR011W	2
r_0455	fumarate reductase FMN	YEL047C	2
r_0560	hydroxymethylglutaryl CoA synthase	YML126C	2
r_0716	malate synthase	(YIR031C or YNL117W)	2
r_0819	ornithine transaminase	YLR438W	2
r_0837	palmitoyl-CoA desaturase	YGL055W	2
r_1138	D-lactate/pyruvate antiport		2
r_1265	succinate-fumarate transport	YJR095W	2
r_2034	pyruvate transport		2
r_0003	(R-R)-butanediol dehydrogenase	YAL060W	1
r_0052	3-hydroxyacyl-CoA dehydrogenase	YKR009C	1
r_0159	alcohol acetyltransferase (ethanol)	(YGR177C or YOR377W)	1
r_0161	alcohol acetyltransferase (isobutyl alcohol)	(YGR177C or YOR377W)	1
r_0307	CTP synthase (NH3)	(YBL039C or YJR103W)	1
r_0334	dephospho-CoA kinase	YDR196C	1
r_0401	fatty-acid–CoA ligase (hexadecanoate)	(YIL009W or YMR246W or YOR317W)	1
r_0402	fatty-acid–CoA ligase (hexadecanoate)	YER015W	1
r_0414	fatty-acid–CoA ligase (tetradecenoate)	YER015W	1
r_0428	fatty-acyl-ACP synthase (n-C16:1ACP)	(YBR026C and YER061C and YHR067W and YKL055C and YKL192C and YOR221C)	1
r_0547	homoserine dehydrogenase	YJR139C	1
r_0648	IPS phospholipase C	YER019W	1
r_0678	L-aminoadipate-semialdehyde dehydrogenase	(YBR115C and YGL154C)	1
r_0817	ornithine decarboxylase	YKL184W	1
r_1105	acetate transport	YCR032W	1
r_1106	acetate transport	YCR010C	1
r_1216	L-proline transport	(YKR039W or YOR348C)	1
r_1238	orntithine transport	(YEL063C or YKR039W)	1
r_1596	3-methyl-2-oxopentanoate transport		1
r_1633	acetaldehyde transport		1
r_1708	D-erythrose 4-phosphate transport		1
r_1717	D-sorbitol transport		1
r_1986	Oleoyl-CoA desaturase		1

Table A.5: *Number of reaction occurrences in 19 different knockout sets.*

Supplementary material for Chapter 3

Network model	Metabolite Id	$\Delta_X^{\min}$	$\Delta_X^{\max}$
S + L	M_g6p_c	1.790	2.130
S + L	M_f6p_c	0.473	0.547
S + L	M_6pgc_c	0.183	0.237
S + L	M_fdp_c	3.672	3.816
S + L	M_pep_c	-1.060	-0.988
S + L	M_pyr_c	2.632	2.776
S + L	M_cit_c	0.760	0.860
S + L	M_succ_c	7.255	12.605
S + L	M_fum_c	0.880	1.616
S + L	M_mal__L_c	0.488	0.728
S + L	M_adp_c	-0.269	-0.003
S + L	M_gln__L_c	-2.036	-1.492
S + L	M_glc__D_e	-50.000	0
S + L	M_o2_c	-50.000	0
S + L	M_biomass_c	0	50.000
L	M_man6p_c	0.205	0.255
L	M_val__L_c	-0.300	0
L	M_his__L_c	-0.105	-0.015
L	M_phe__L_c	0.106	0.290
L	M_tyr__L_c	-0.084	-0.020
L	M_gly_c	-0.772	-0.228
L	M_trp__L_c	-0.019	-0.007
L	M_pro__L_c	-0.121	0
L	M_asp__L_c	-1.344	-0.672
L	M_asn__L_c	-0.132	0
L	M_thr__L_c	-0.307	0
L	M_ile__L_c	-0.087	0

Table A.6: *Calculated variations interval for glucose pulse experiment.* The network model indicates if the metabolite was present in the *E. coli* core model (S) or in the *E. coli* iJO1366 model (L). The interval for each metabolite was calculated based on the baseline concentration and the fold change for the pseudo-steady state taking into account the given small variations for the baseline measurements. To add glucose as a source, $\Delta_{glucose}^{\min}$ was set to -50. Oxygen was added as a source in the same way. Biomass was added as a sink.

Network model	Metabolite Id	$\Delta_X^{\min}$	$\Delta_X^{\max}$
S + L	M_g6p_c	-1.045	-0.915
S + L	M_f6p_c	-0.252	-0.228
S + L	M_6pgc_c	-0.296	-0.275
S + L	M_fdp_c	-0.723	-0.703
S + L	M_pep_c	-1.060	-0.988
S + L	M_cit_c	1.560	1.680
S + L	M_icit_c	1.560	1.680
S + L	M_fum_c	0	0.272
S + L	M_mal__L_c	0.341	0.571
S + L	M_atp_c	-1.180	0
S + L	M_adp_c	-0.269	-0.003
S + L	M_glu__L_c	-4.232	-3.908
S + L	M_gln__L_c	-1.464	-0.888
S + L	M_pyr_c	-50.000	0
S + L	M_o2_c	-50.000	0
S + L	M_biomass_c	0	50.000
L	M_man6p_c	-0.335	-0.309
L	M_ala__L_e	1.986	2.172
L	M_val__L_c	0	0.480
L	M_leu__L_c	0.087	0.249
L	M_phe__L_c	-0.514	-0.410
L	M_tyr__L_c	-0.095	-0.035
L	M_gly_c	-0.556	0
L	M_trp__L_c	-0.012	0
L	M_lys__L_c	-0.240	-0.164
L	M_asp__L_c	-1.806	-1.218
L	M_asn__L_c	-0.184	-0.040
L	M_thr__L_c	-0.394	0

Table A.7: *Calculated variations interval for pyruvate pulse experiment.* The network model indicates if the metabolite was present in the *E. coli* core model (S) or in the *E. coli* iJO1366 model (L). The interval for each metabolite was calculated based on the baseline concentration and the fold change for the pseudo-steady state taking into account the given small variations for the baseline measurements. To add pyruvate as a source, $\Delta_{pyruvate}^{\min}$ was set to -50. Oxygen was added as a source in the same way. Biomass was added as a sink.

Network model	Metabolite Id	$\Delta_X^{\min}$	$\Delta_X^{\max}$
S + L	M_g6p_c	-0.505	-0.335
S + L	M_f6p_c	-0.107	-0.073
S + L	M_6pgc_c	-0.252	-0.228
S + L	M_fdp_c	0.051	0.093
S + L	M_pep_c	-0.185	-0.071
S + L	M_pyr_c	0.847	0.921
S + L	M_cit_c	3.160	3.320
S + L	M_fum_c	23.720	33.592
S + L	M_mal__L_c	25.919	27.889
S + L	M_adp_c	-0.269	-0.003
S + L	M_glu__L_c	-4.232	-3.908
S + L	M_succ_c	-100.000	0
S + L	M_o2_c	-100.000	0
S + L	M_biomass_c	0	100.000
L	M_man6p_c	-0.155	-0.121
L	M_val__L_c	-0.350	-0.010
L	M_leu__L_c	-0.102	0
L	M_his__L_c	-0.087	0
L	M_phe__L_c	-0.204	-0.060
L	M_trp__L_c	-0.014	-0.001
L	M_asp__L_c	3.738	5.334
L	M_asn__L_c	0	0.140
L	M_thr__L_c	0.0410	0.547

Table A.8: *Calculated variations interval for succinate pulse experiment.* The network model indicates if the metabolite was present in the *E. coli* core model (S) or in the *E. coli* iJO1366 model (L). The interval for each metabolite was calculated based on the baseline concentration and the fold change for the pseudo-steady state taking into account the given small variations for the baseline measurements. To add glucose as a source, $\Delta_{succinate}^{\min}$ was set to -100. Oxygen was added as a source in the same way. Biomass was added as a sink. Higher values for sources and sinks were chosen due to larger calculated $\Delta^{\min}$ and $\Delta^{\max}$ than in the other two experiments.

Active reactions in the E. coli core model

The following figures show the results for the three pulse experiments for different parameters. The metabolites that were given as input are highlighted in blue if the corresponding input deltas were below zero and red if they were above zero. Reactions that are highlighted in orange were chosen in almost all of the enumerated solutions. Reactions that are yellow were chosen only in around half of the solutions. White reactions were not chosen in any solution.

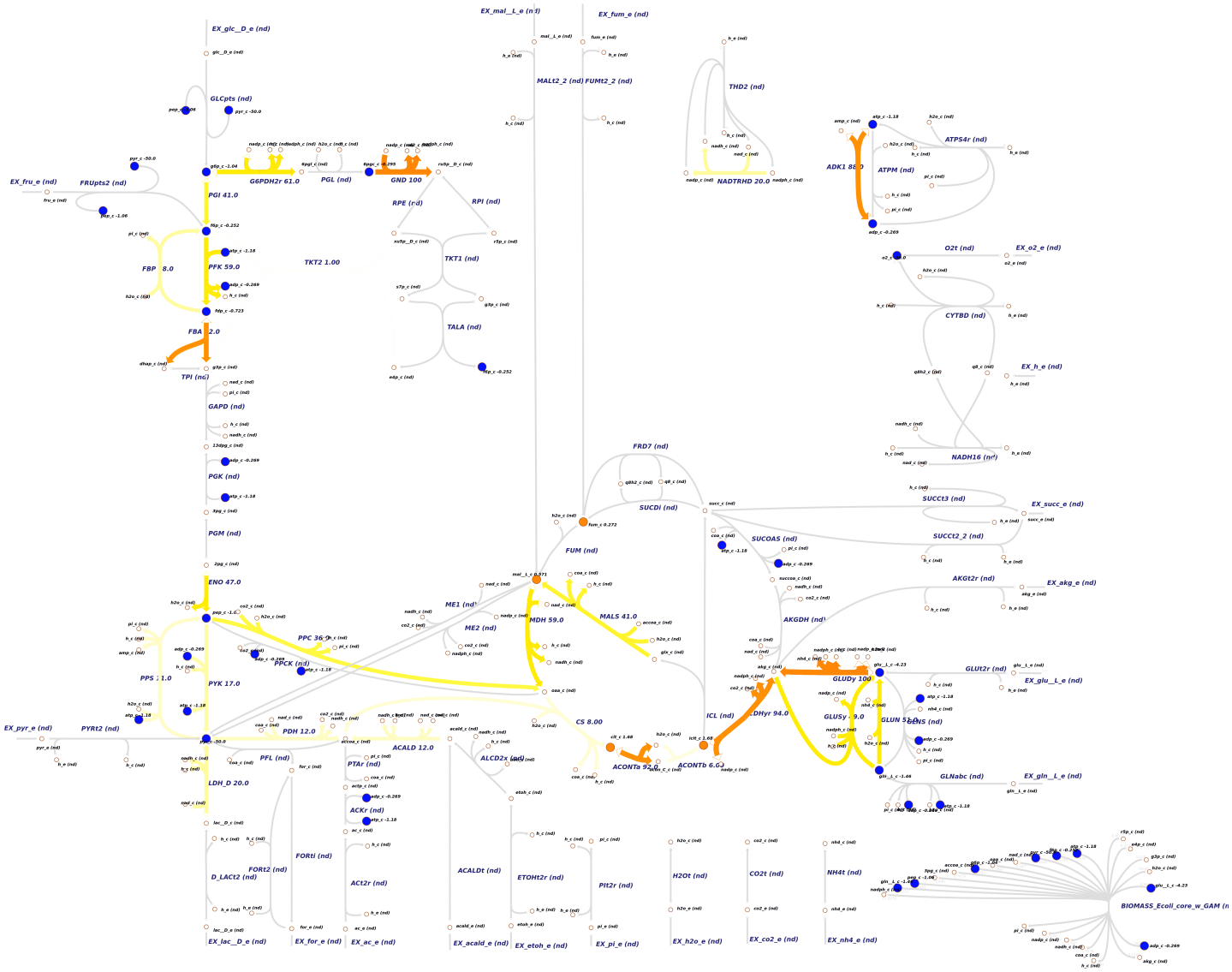


Figure A.1: *E. coli core model - Results for pyruvate pulse ($\lambda = 0.9$, $\epsilon = 10$).* The active reactions are disconnected. Since $\lambda = 0.9$, the optimization prioritizes the minimization of the number of active reactions, fewer active reactions are thus chosen in total and the accumulation/depletion of unmeasured metabolites is higher. The figure was created using *Escher* (King et al., 2015a).

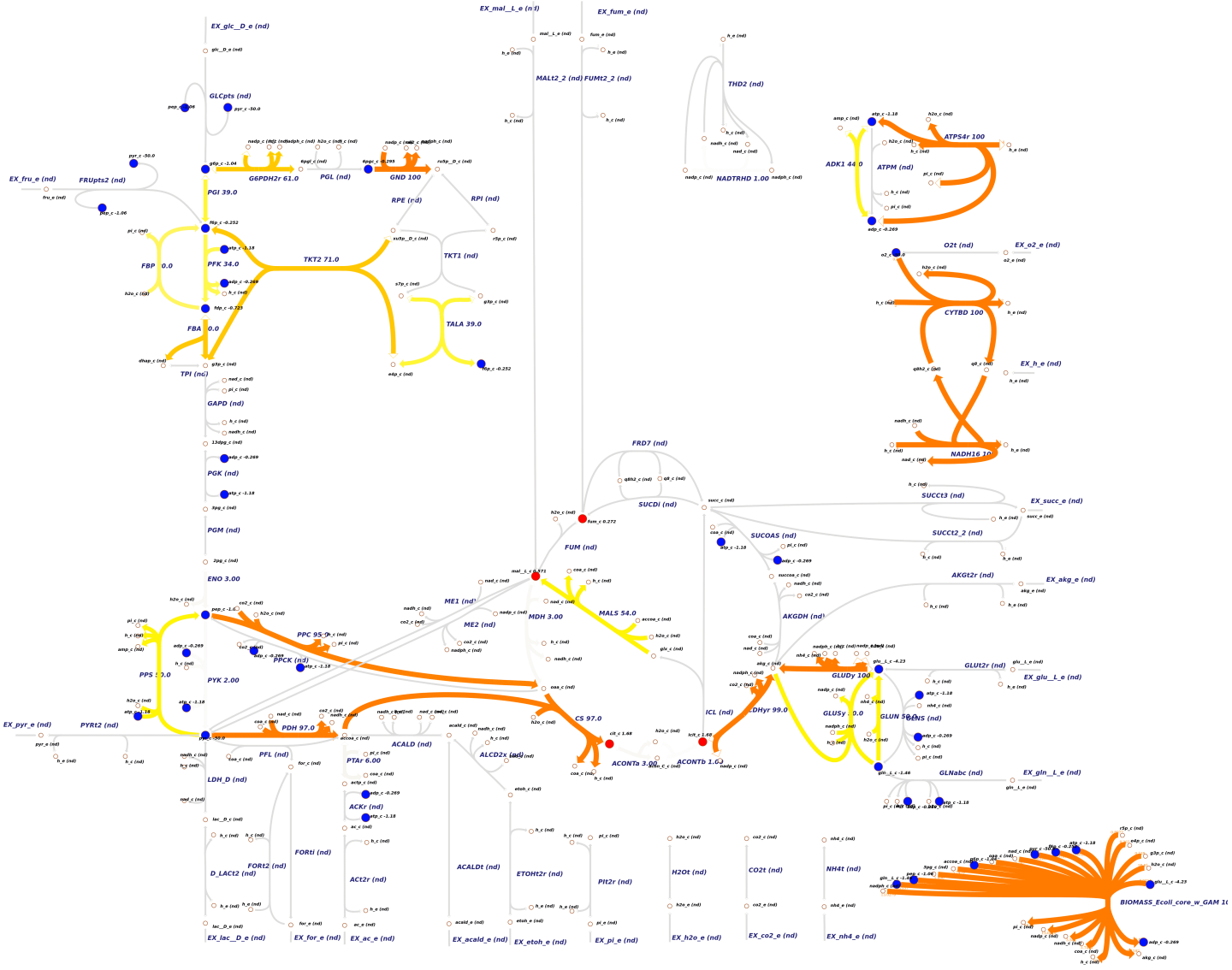


Figure A.2: *E. coli core model - Results for pyruvate pulse ($\lambda = 0.9$, $\epsilon = 5$).* Similar reactions are active as for $\lambda = 0.9$ and $\epsilon = 10$. An important difference is that the biomass reaction is part of the solution. The figure was created using *Escher* (King et al., 2015a).

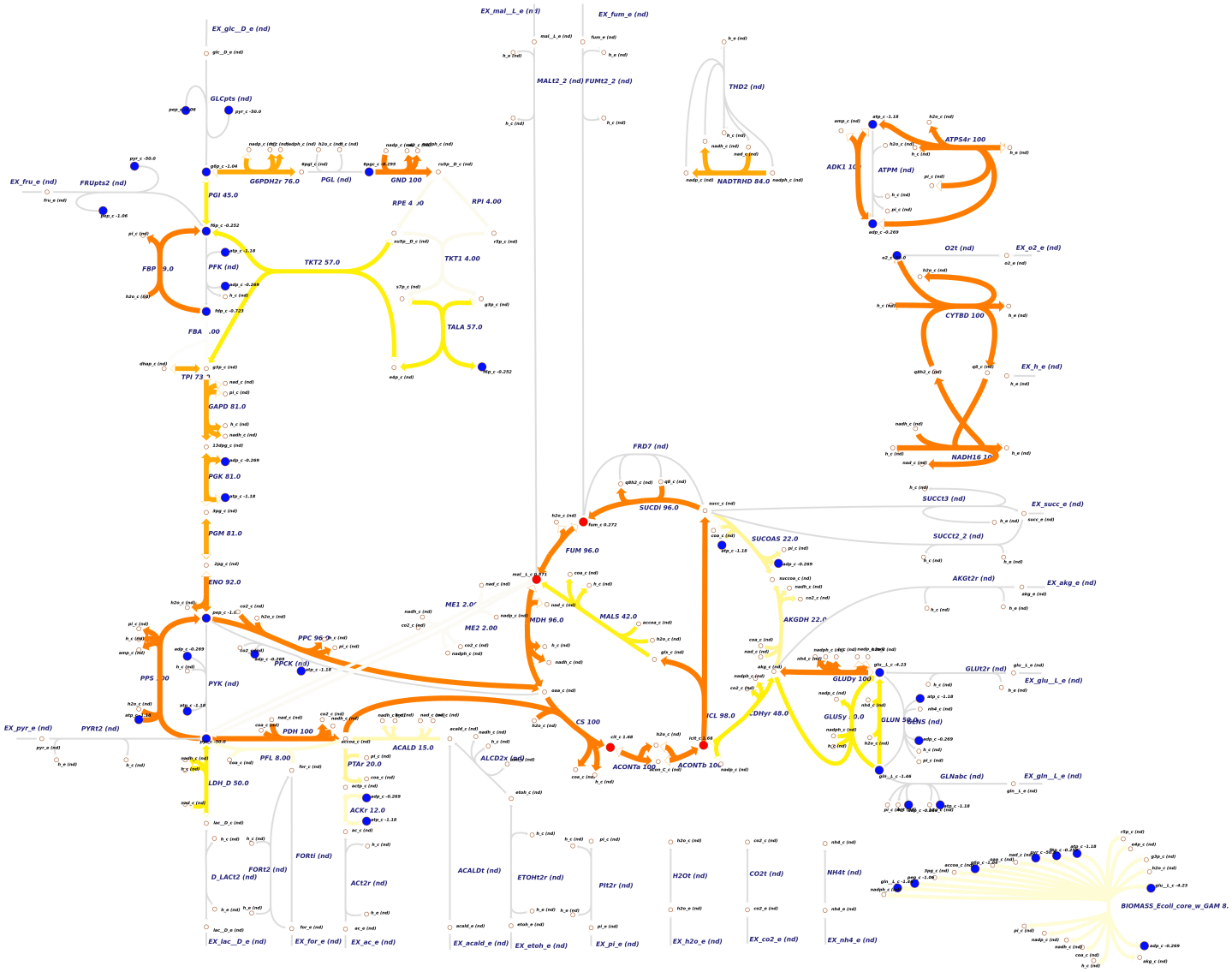


Figure A.3: *E. coli core model - Results for pyruvate pulse ($\lambda = 0.5$, $\epsilon = 10$).* After lowering λ to 0.5, the solutions already contain more active reactions and we are able to see connected pathways that are active during the metabolic shift. The biomass reaction is only chosen in 8 out of the 100 solutions. The figure was created using *Escher* (King et al., 2015a).

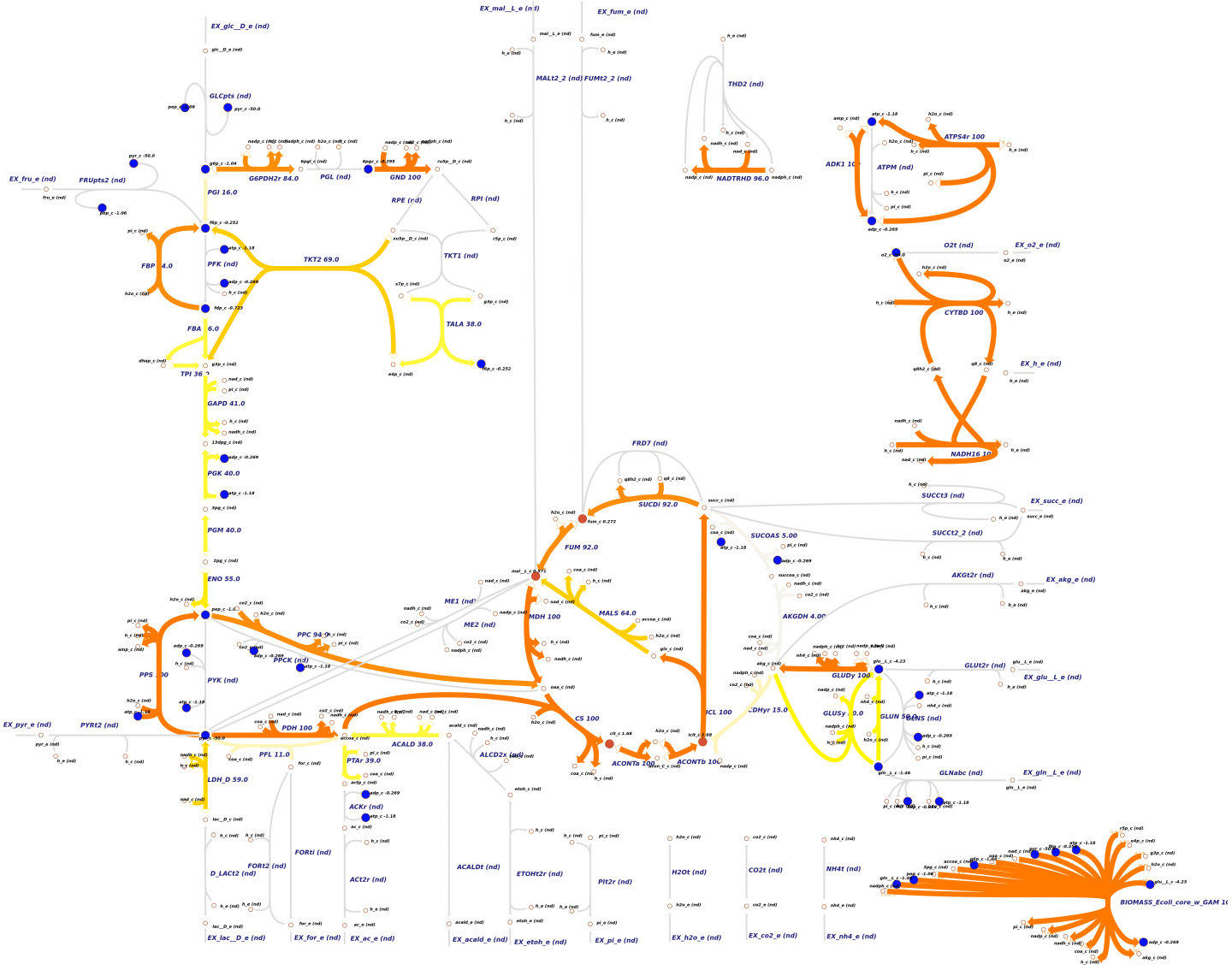


Figure A.4: *E. coli core model - Results for pyruvate pulse ($\lambda = 0.5$, $\epsilon = 5$).* The results are similar to $\lambda = 0.5$ and $\epsilon = 10$. However, again the biomass reaction is active in all solutions. The figure was created using *Escher* (King et al., 2015a).

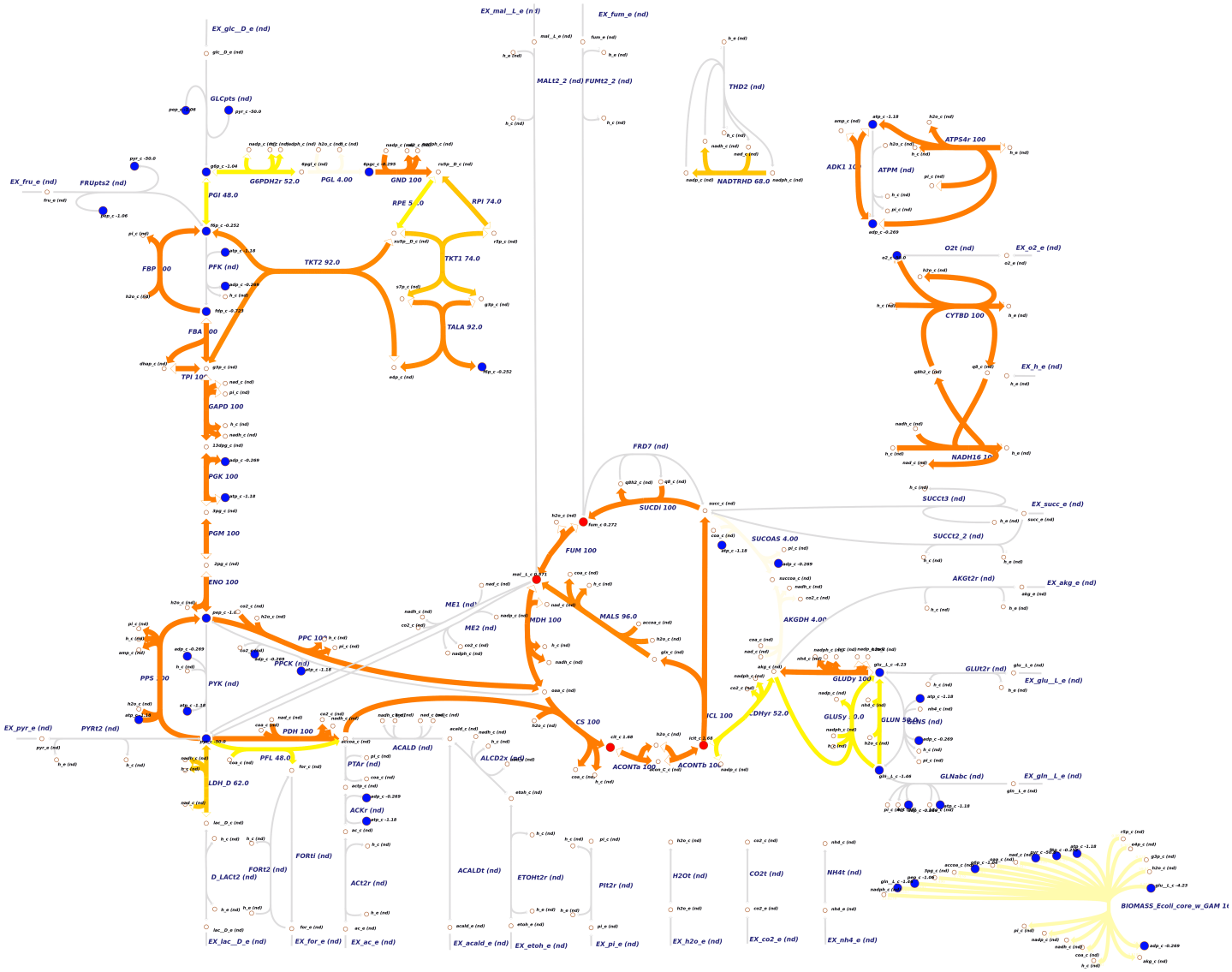


Figure A.5: *E. coli core model - Results for pyruvate pulse ($\lambda = 0.1$, $\epsilon = 10$).* We can see connected pathways. The expected reactions of the gluconeogenesis and part of the TCA cycle are active in all 100 solutions. The reversible reactions of the gluconeogenesis are chosen in the correct direction. The figure was created using *Escher* (King et al., 2015a).

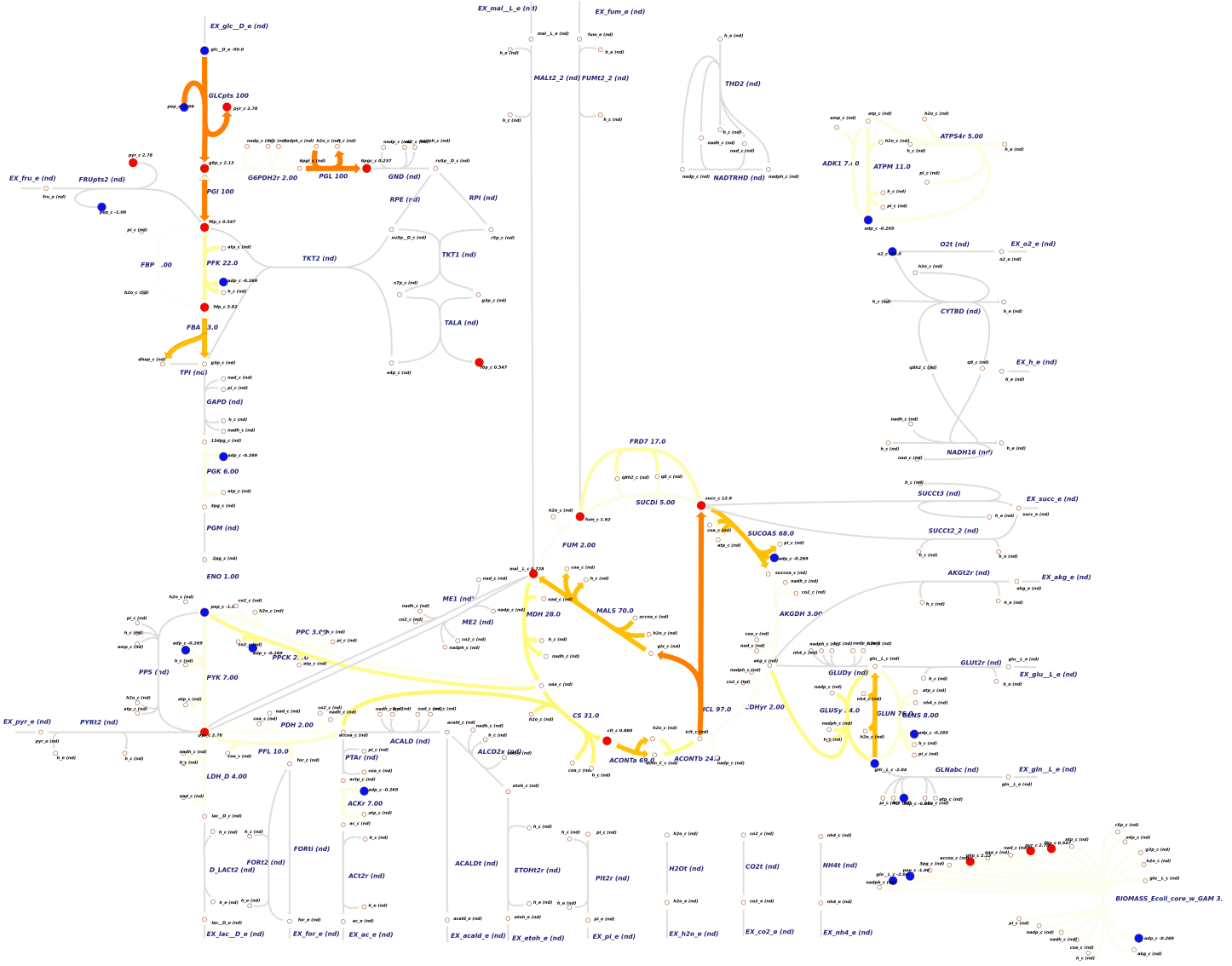


Figure A.6: *E. coli core model - Results for glucose pulse ($\lambda = 0.9$, $\epsilon = 5$).* Similar to the results of the pyruvate pulse for $\lambda = 0.9$, the active reactions in the solutions are disconnected. The figure was created using *Escher* (King et al., 2015a).

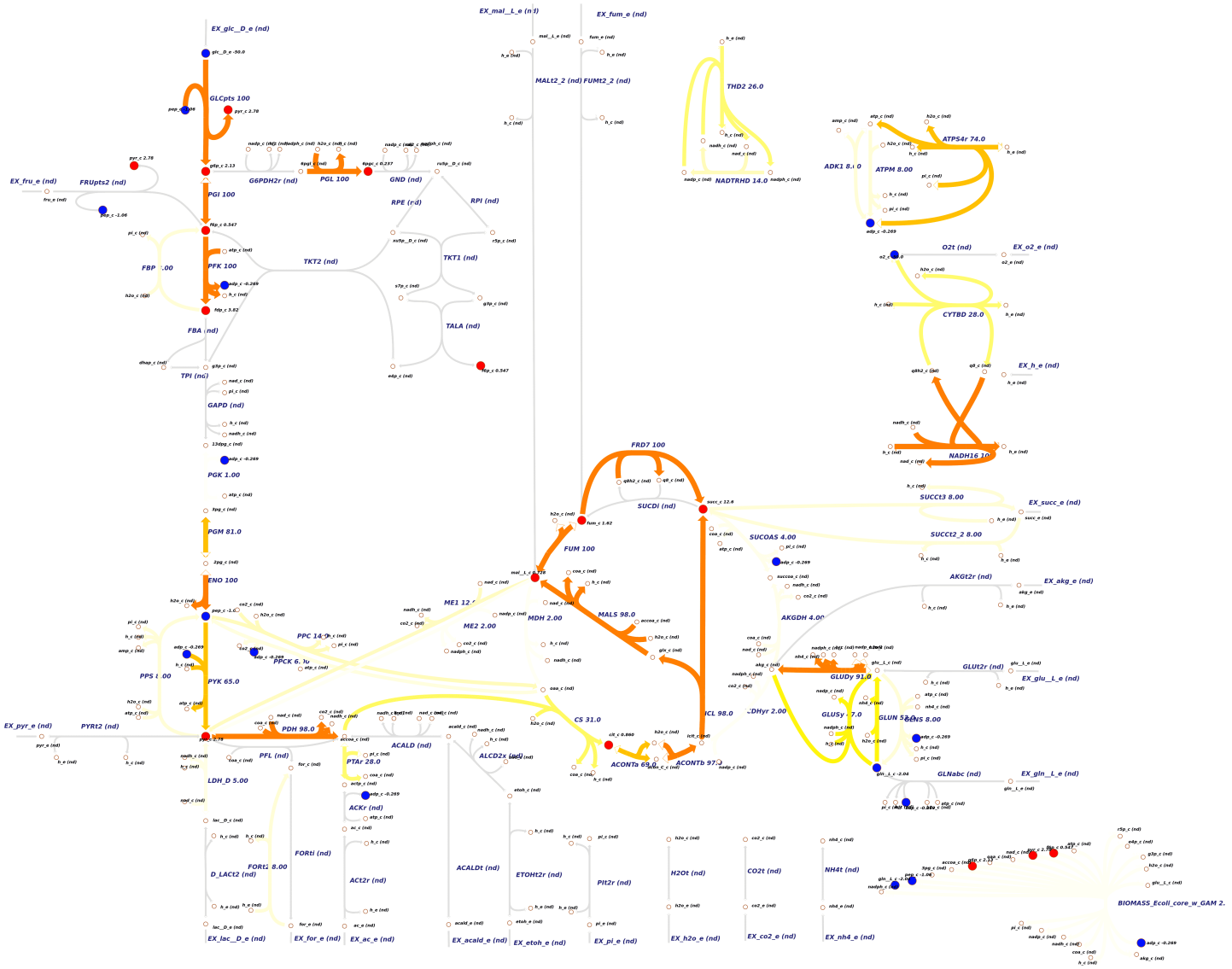


Figure A.7: *E. coli core model - Results for glucose pulse ($\lambda = 0.5$, $\epsilon = 5$).* The result is still similar to $\lambda = 0.9$ although more reactions get chosen more frequently in the different solutions. The figure was created using *Escher* (King et al., 2015a).

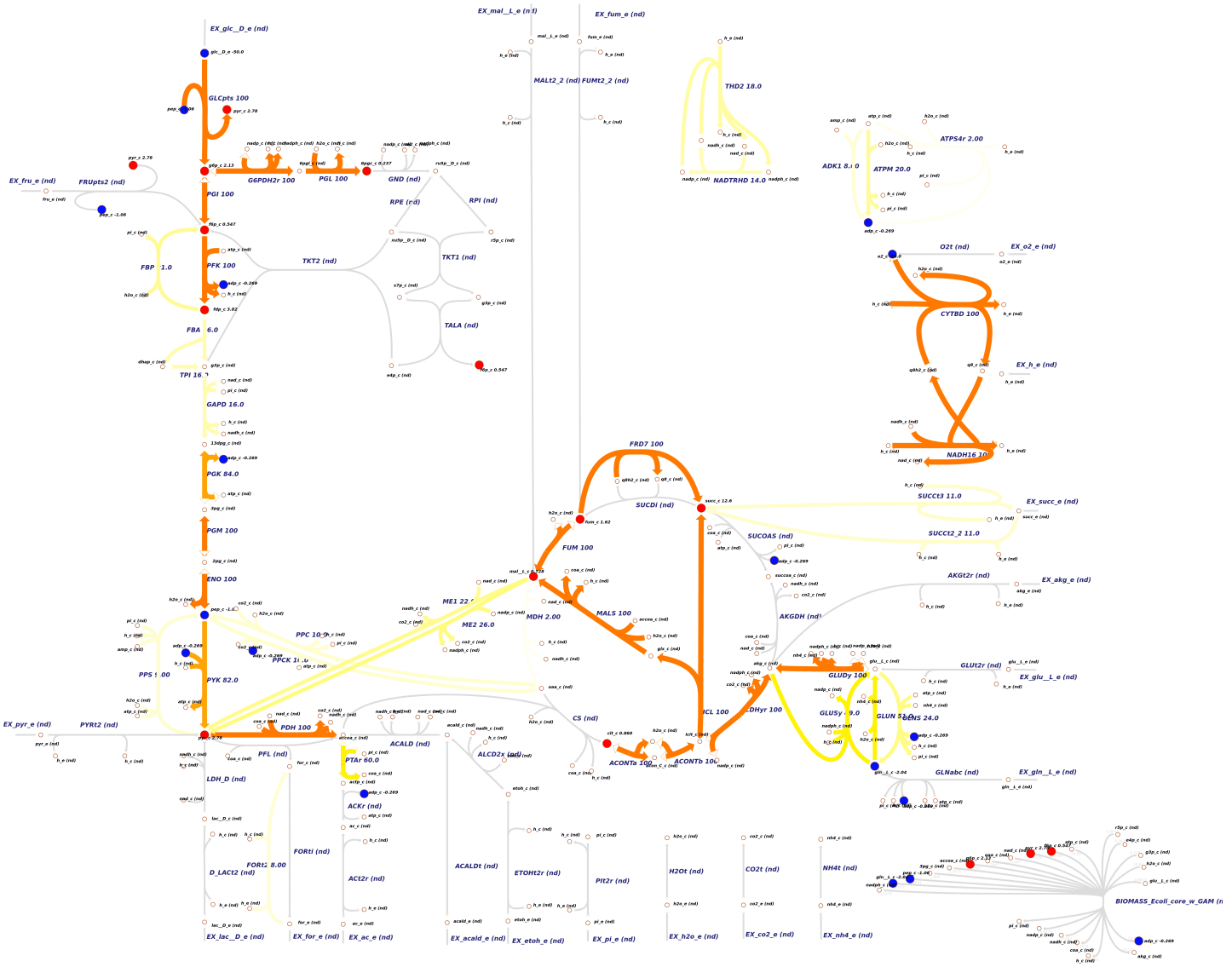


Figure A.8: *E. coli core model - Results for glucose pulse ($\lambda = 0.1$, $\epsilon = 5$).* Even for $\lambda = 0.1$, we are not able to see that the expected reactions of the glycolysis and parts of the TCA cycle are active in most of the solutions. Therefore, λ was decreased further to see if it is possible to improve the results and to obtain connected pathways. The figure was created using *Escher* (King et al., 2015a).

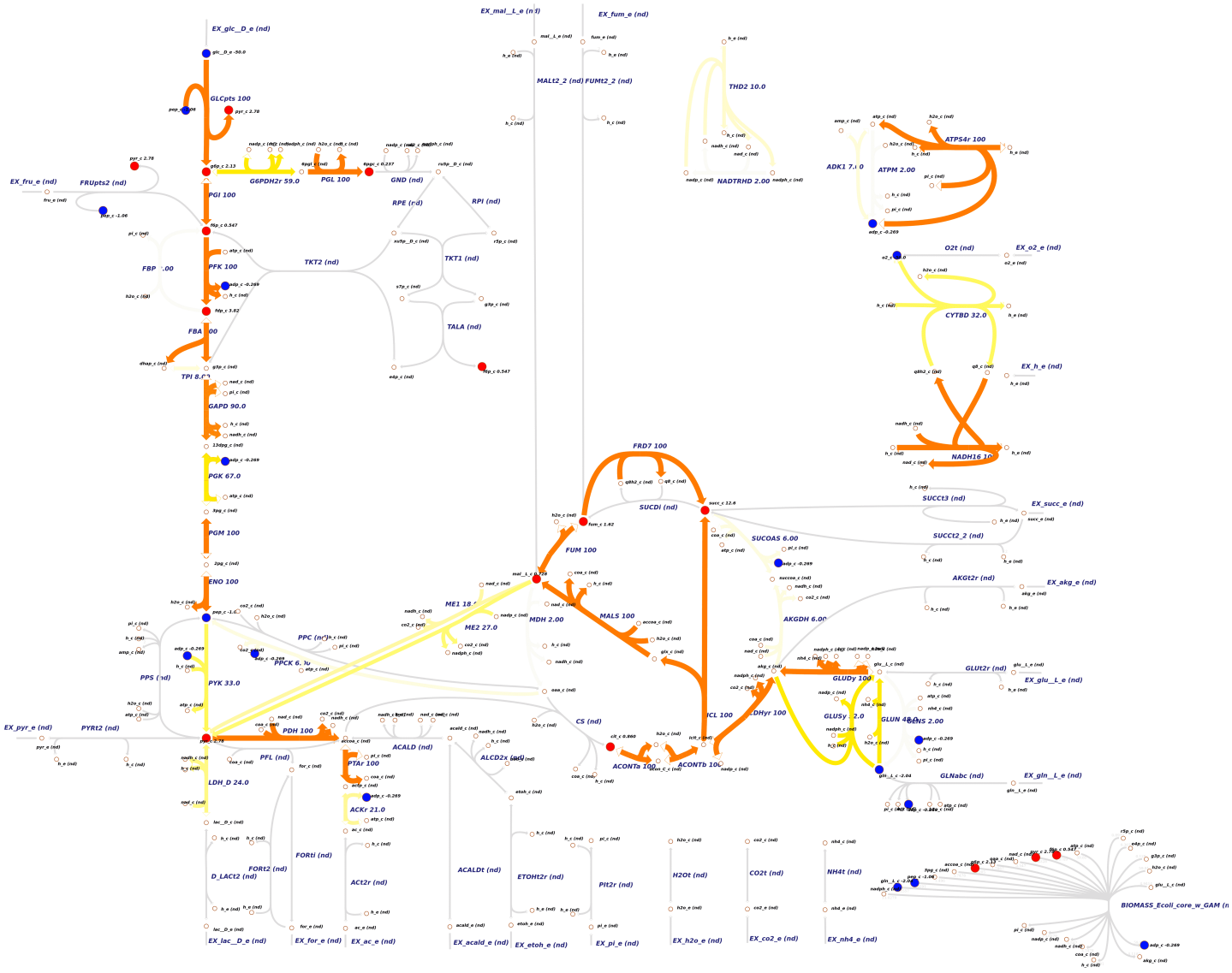


Figure A.9: *E. coli core model - Results for glucose pulse ($\lambda = 0.1$, $\epsilon = 2$).* After decreasing λ to 2, more reactions of the glycolysis were active more frequently in the 100 solutions. The figure was created using *Escher* (King et al., 2015a).

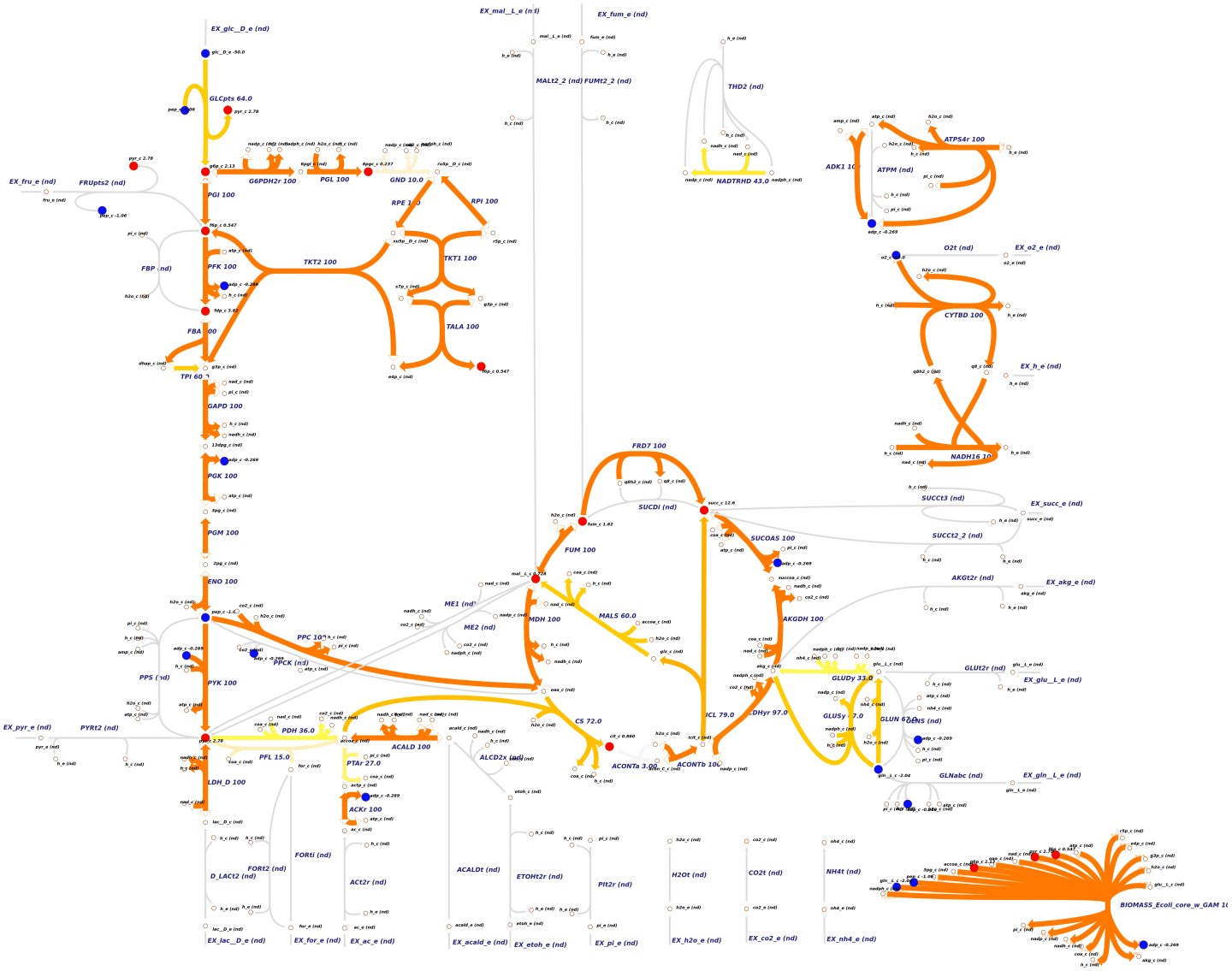


Figure A.10: *E. coli core model - Results for glucose pulse ($\lambda = 0.1$, $\epsilon = 1.2$).* The previous result could be improved even further by decreasing λ to 1.2. Reactions of the glycolysis and the TCA cycle are active in all 100 solutions. The active reactions of the glycolysis pathway that are reversible are chosen in the correct direction. Furthermore, in contrast to the previous results, the biomass reaction is part of the solution. However, the reaction that transports glucose is only active in 64 out of 100 solutions. Lowering λ to 1.1 rendered the optimization problem infeasible. The figure was created using *Escher* (King et al., 2015a).

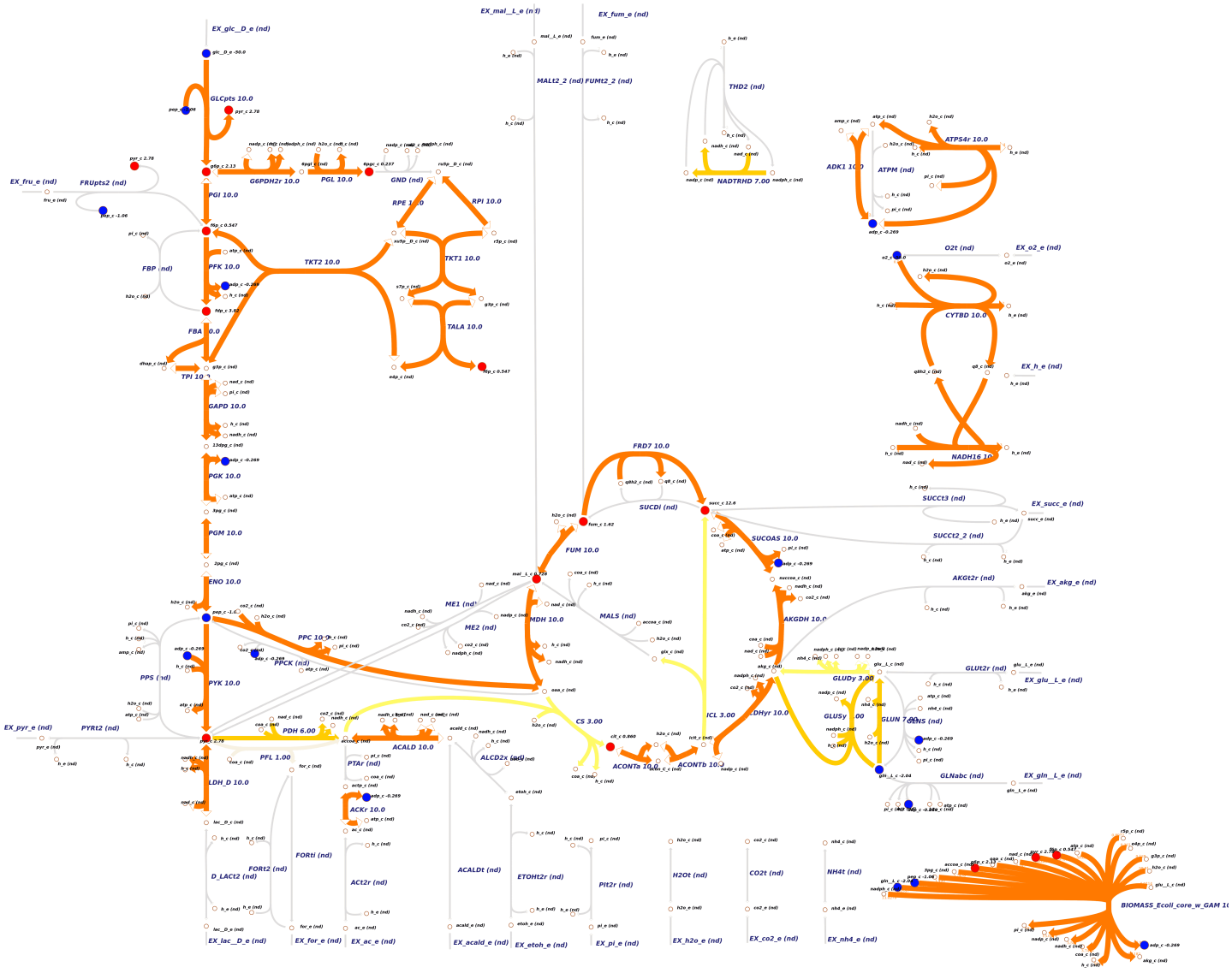


Figure A.11: *E. coli core model - First ten solutions for glucose pulse ($\lambda = 0.1$, $\epsilon = 1.2$).* In the first ten solutions, the transport reaction for glucose is always active. The other active reactions are very similar to the results for 100 solutions. The figure was created using *Escher* (King et al., 2015a).

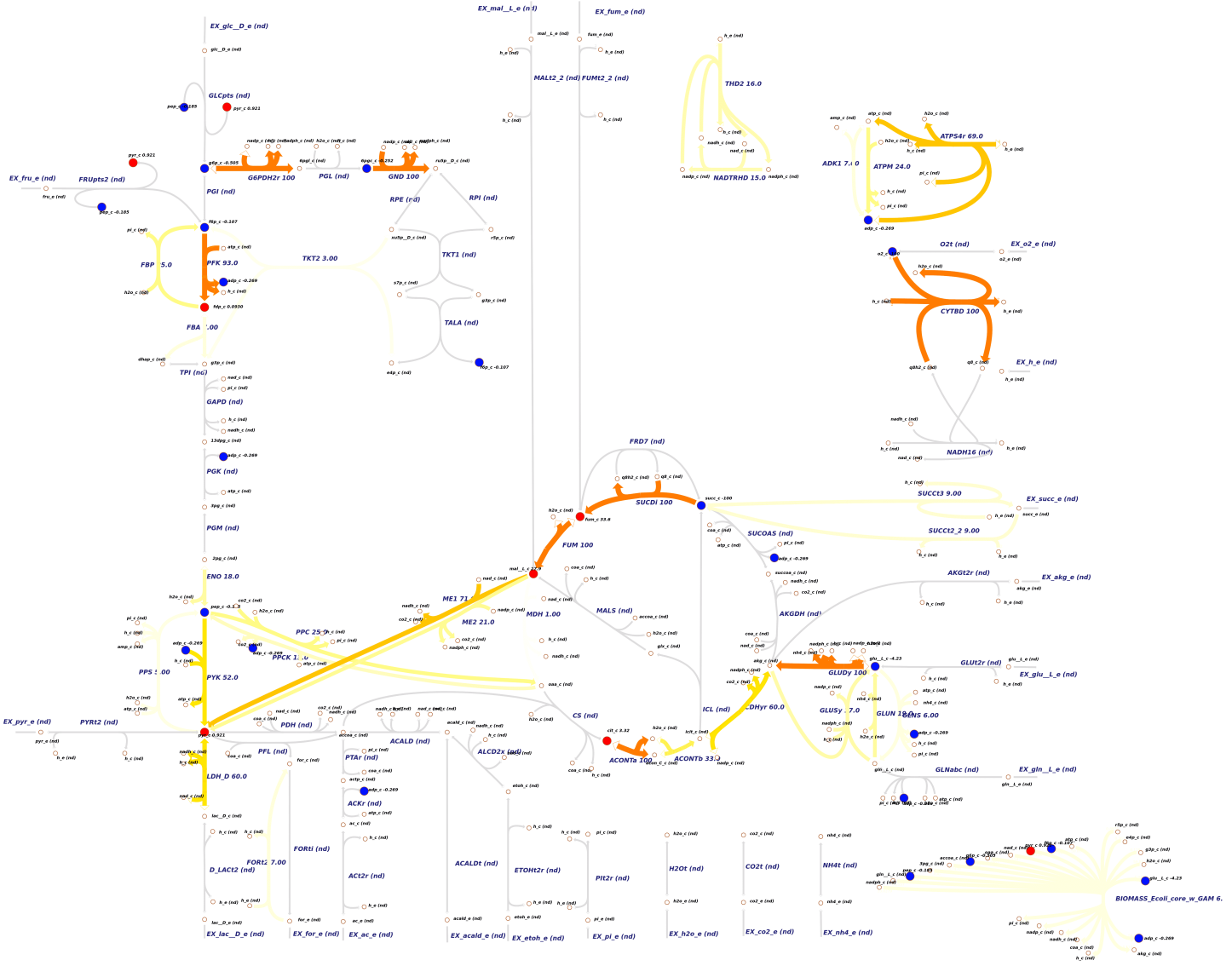


Figure A.12: *E. coli core model - Results for succinate pulse ($\lambda = 0.9$, $\epsilon = 10$).* We can see similarities to the results of the other pulse experiments. For $\lambda = 0.9$ the active reactions are disconnected. The figure was created using *Escher* (King et al., 2015a).

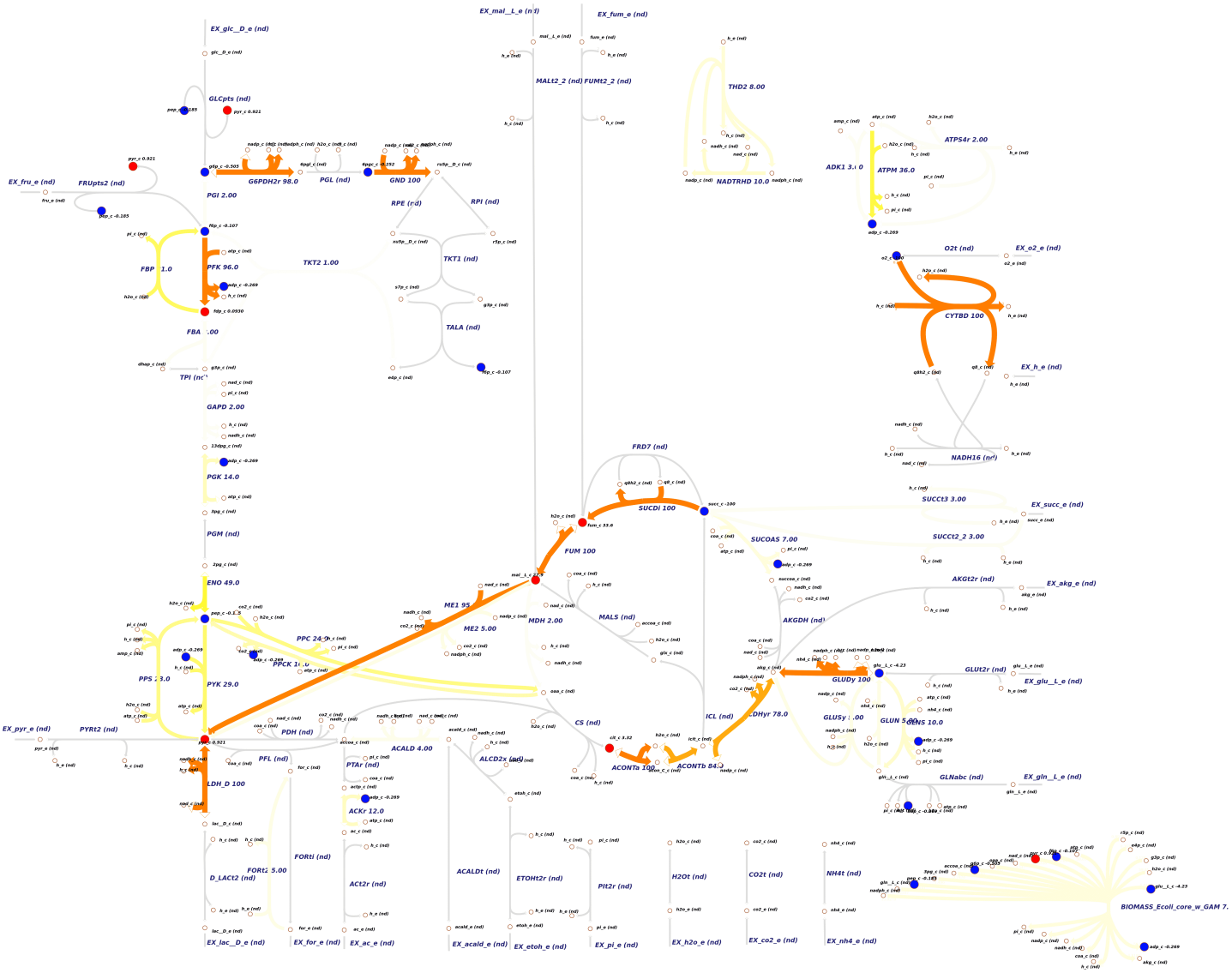


Figure A.13: *E. coli core model - Results for succinate pulse ($\lambda = 0.9$, $\epsilon = 5$).* For $\lambda = 0.9$ the active reactions are disconnected. The figure was created using *Escher* (King et al., 2015a).

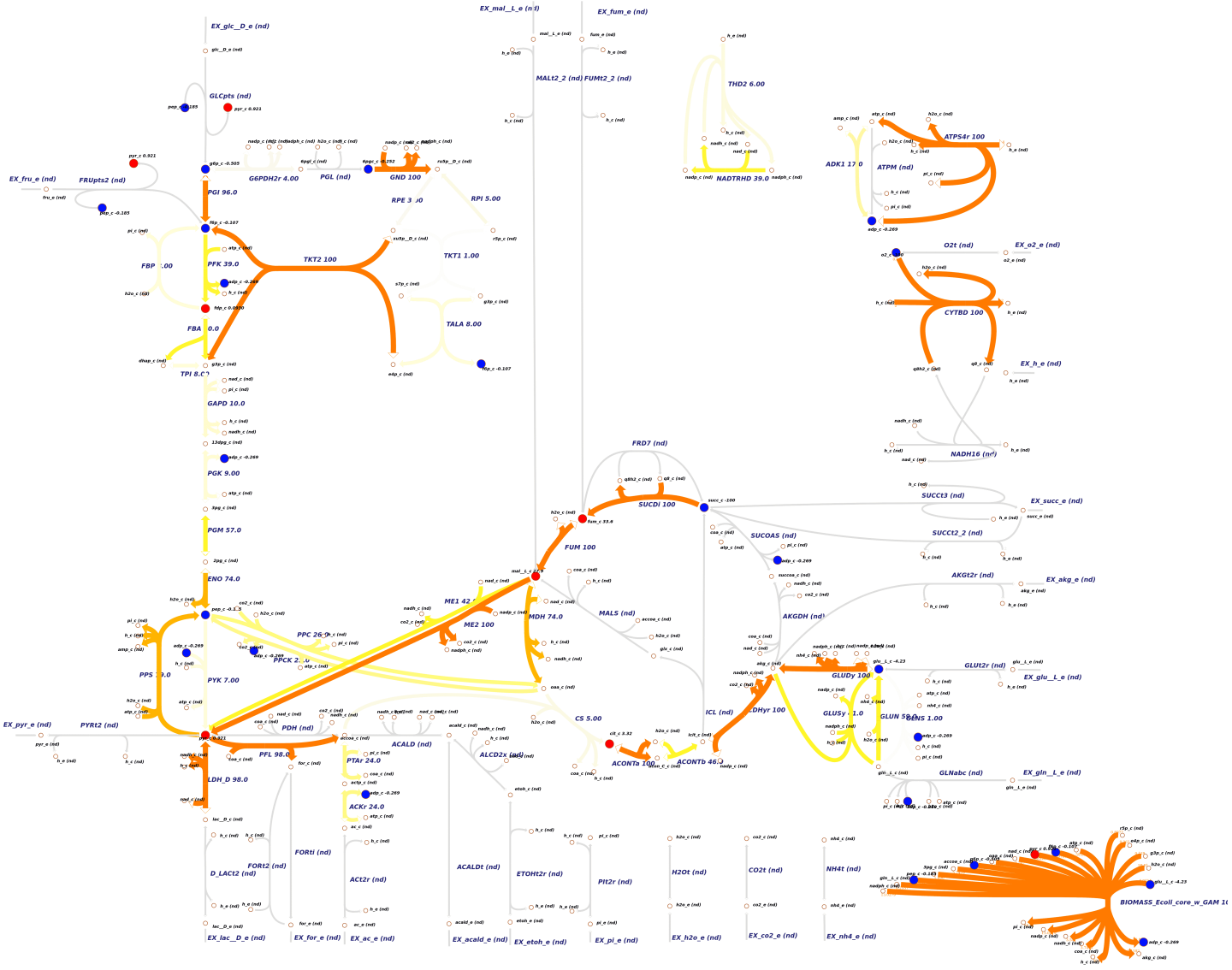


Figure A.14: *E. coli core model - Results for succinate pulse ($\lambda = 0.5$, $\epsilon = 10$).* The results are already more connected than for $\lambda = 0.5$ but we are not able to obtain the expected gluconeogenesis pathway. The figure was created using *Escher* (King et al., 2015a).

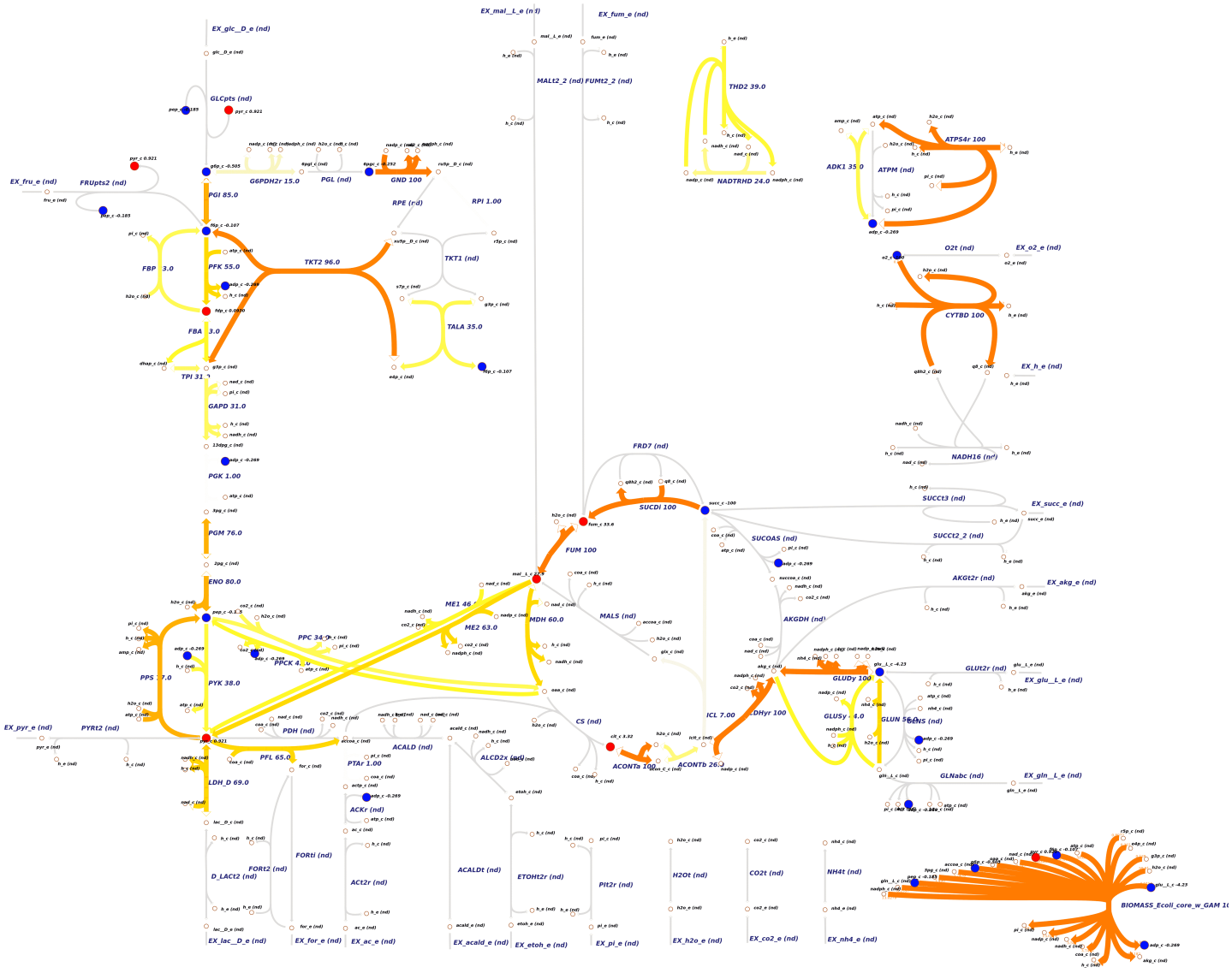


Figure A.15: *E. coli core model - Results for succinate pulse ($\lambda = 0.5$, $\epsilon = 5$).* The results are already more connected than for $\lambda = 0.5$ but we are not able to obtain the expected gluconeogenesis pathway. The figure was created using *Escher* (King et al., 2015a).

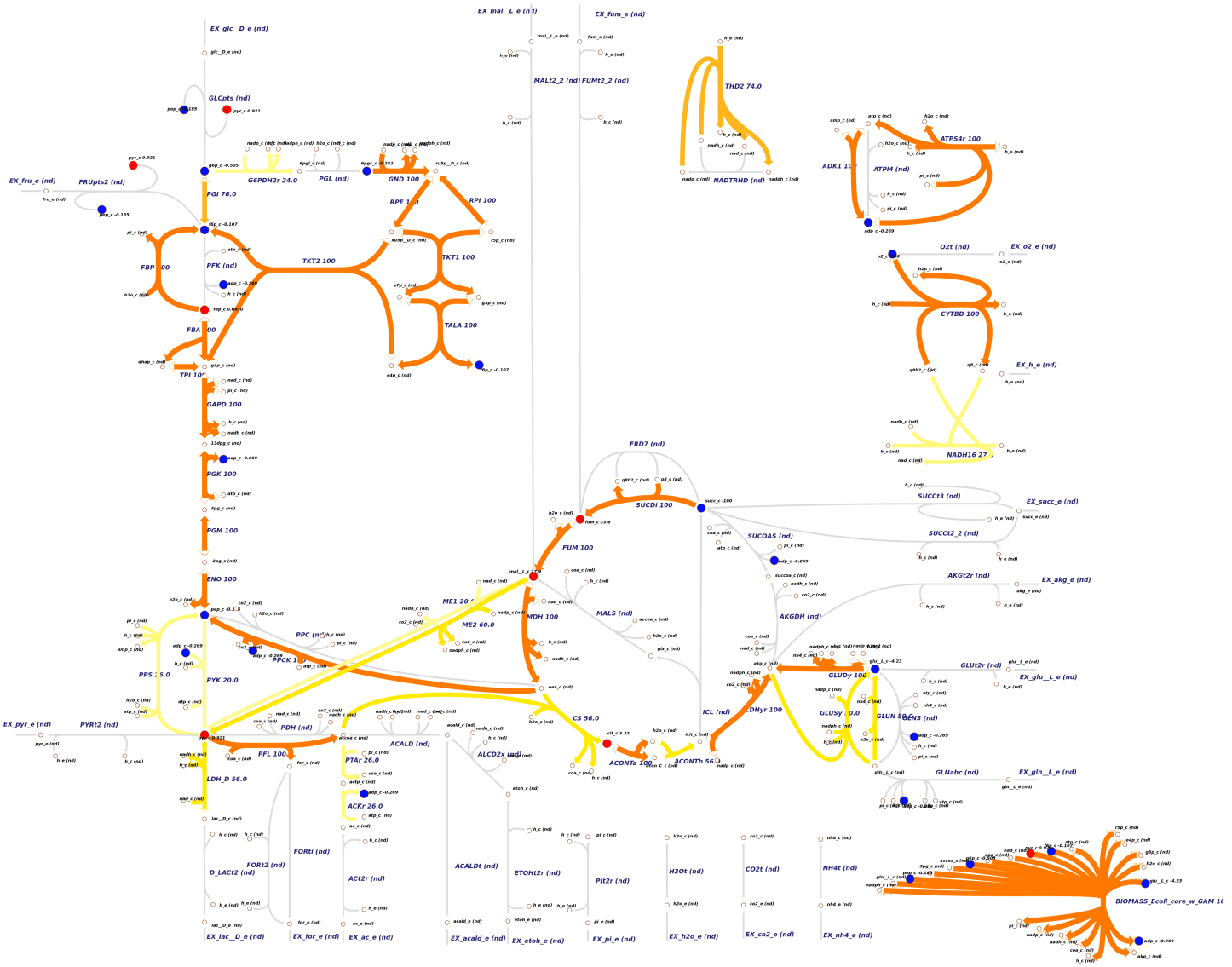


Figure A.16: *E. coli core model - Results for succinate pulse ($\lambda = 0.1$, $\epsilon = 10$).* The reactions of the gluconeogenesis pathway and the reactions that transform succinate in the TCA cycle and subsequently into pyruvate are active in all 100 solutions. The figure was created using *Escher* (King et al., 2015a).

TITRE en français

Modèles et algorithmes pour étudier et exploiter le métabolisme des micro-organismes

RESUME en français

Dans cette thèse, j'ai travaillé sur deux approches différentes pour étudier le métabolisme des micro-organismes. La première méthode identifie les knock-outs qui augmentent la production d'une métabolite cible dans un micro-organisme dans le cas où le métabolite cible est toxique pour le micro-organisme utilisé. Dans la première partie de l'approche, un problème d'optimisation multi-objectifs est formulé qui calcule des compromis entre la production de biomasse, la production du métabolite cible et un score qui mesure la résistance possible à la toxicité du métabolite cible. Dans la deuxième partie, des knock-outs sont calculés sur la base de l'identification et de la séparation des sous-réseaux qui peuvent atteindre les valeurs de production souhaitées identifiées dans la première partie. L'approche est applicable aux réseaux métaboliques à l'échelle du génome, comme est montré dans l'étude de cas sur la production d'éthanol dans la levure.

La deuxième méthode, appelée TOTORO, a été développée pour l'analyse des changements métaboliques. Elle intègre les concentrations internes de métabolites qui ont été mesurées avant et après une perturbation dans les réseaux métaboliques. Il prédit les réactions qui étaient actives pendant l'état transitoire qui s'est produit après la perturbation. TOTORO est une approche basée sur les contraintes qui prend en compte la stœchiométrie du réseau. La méthode est appliquée à trois expériences d'impulsions dans *Escherichia coli* pour montrer qu'elle peut récupérer des voies actives connectées et prédire des directions distinctes pour des réactions réversibles qui sont conformes aux observations biologiques connues. TOTORO est applicable aux réseaux métaboliques à l'échelle du génome.

MOTS-CLEFS en français

métabolisme; modélisation des réseaux métaboliques; énumeration; knock-outs; changement métabolique; hypergraphes dirigés; programmation sous constraintes

Title in english

Models and algorithms for investigating and exploiting the metabolism of microorganisms

Abstract in english

In this thesis, I worked on two different constraint-based approaches to study the metabolism of microorganisms. The first method identifies knockouts that increase the production of a target chemical in a microorganism in the scenario where the produced target metabolite is toxic for the utilized microorganism. In the first part of the approach, a multi-objective optimization problem is formulated that computes tradeoffs between biomass production, target production and a score that measures the possible toxicity resistance against the toxic target metabolite. In the second part, promising knockout sets are computed based on identifying and cutting off subnetworks that can lead to the desired production values identified in the first part. The approach is applicable to genome-scale metabolic networks which is shown in the case-study of ethanol production in yeast.

The second method which is called TOTORO was developed for the analysis of metabolic shifts. It integrates internal metabolite concentrations that were measured before and after a perturbation into genome-scale metabolic networks. It predicts reactions that were active during the transient state that occurred after the perturbation. TOTORO is a constraint-based approach that takes the stoichiometry of the network into account. The method is applied to three pulse experiments in *Escherichia coli* to show that it can retrieve connected active pathways and predict distinct directions for reversible reactions that are in accordance with known biological observations. TOTORO is applicable to genome-scale metabolic networks.

Keywords in english

metabolism; metabolic network modeling; enumeration; knockouts; metabolic shift; directed hypergraphs; constraint-based programming