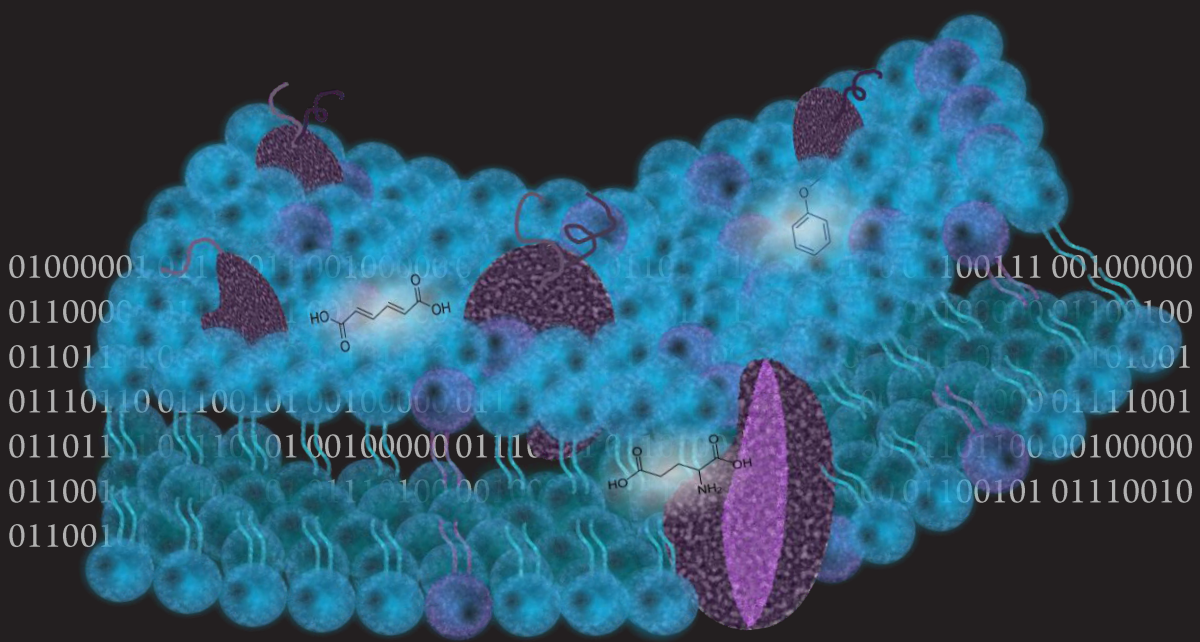
Evaluating and deploying genome-scale metabolic models for microbial cell factories



Nhung Pham

Propositions

1. Ambiguous naming systems in genome-scale metabolic models hamper their potential.
   (this thesis)

2. The use of design-build-test-learn cycles facilitates research across and beyond disciplines.
   (this thesis)

3. Discrepancies between observations and model predictions can be a gold mine.

4. Confusing correlation with causation is a known but still common bias in data-driven research.

5. Solutions for new problems do not need to be new.

6. Empathy is an important skill we need to succeed in many aspects of life.

Propositions belonging to the thesis, entitled

Evaluating and deploying genome-scale metabolic models for microbial cell factories

Nhung Pham

Wageningen, 25 January 2021

Evaluating and deploying genome-scale metabolic models for microbial cell factories

Nhung Pham

Thesis committee

Promotor

Prof. Dr Vitor A.P. Martins dos Santos
Professor of Systems and Synthetic Biology
Wageningen University & Research

Co-promotors

Dr. Peter J. Schaap
Associate professor, Laboratory of Systems and Synthetic Biology
Wageningen University & Research

Dr. Maria Suarez-Diez
Associate professor, Laboratory of Systems and Synthetic Biology
Wageningen University & Research

Other members

Prof. Dr. Dick de Ridder, Wageningen University & Research
Prof. Dr. Jeroen Hugenholtz, Wageningen University & Research
Dr. Ronan Fleming, Leiden University
Dr. Isabel Rocha, Universidade Nova de Lisboa, Oeiras, Portugal

This research was conducted under the auspices of the Graduate School VLAG (Advanced studies in Food Technology, Agrobiotechnology, Nutrition and Health Sciences).

Evaluating and deploying genome-scale metabolic models for microbial cell factories

Nhung Pham

Thesis

submitted in fulfilment of the requirements for the degree of doctor
at Wageningen University
by the authority of the Rector Magnificus,
Prof. Dr A.P.J. Mol,
in the presence of the
Thesis Committee appointed by the Academic Board
to be defended in public
on **Monday 25 January 2021**
at **4 p.m.** in **the Aula**.

Nhung Pham
Evaluating and deploying genome-scale metabolic models for microbial cell factories,
236 pages.

PhD thesis, Wageningen University, Wageningen, the Netherlands (2021)
With references, with summary in English

ISBN: 978-94-6395-655-0
DOI: 10.18174/537210

Contents

1 Introduction 3

2 Genome-scale metabolic modelling underscores the potential of *Cutaneotrichosporon oleaginosus* ATCC 20509 as a cell factory for biofuel production 19

3 Design of pathways for chemical production in *Pseudomonas Putida* KT 2440 53

4 Consistency, inconsistency, and ambiguity of metabolite names in biochemical databases used for genome-scale metabolic modelling 79

5 Systematic evaluation of gap-filling algorithms in GEMs 111

6 General discussion 139

7 Summary 173

References 227

I don't know anything, but I do know that everything is interesting if you go into it deeply enough [1].

Richard Feynman

1 Introduction

Nhung Pham

In 1994, a panel assembled by NASA gave a definition of life as *'life is a self-sustained chemical system capable of undergoing Darwinian evolution'*[2, 3]. Before NASA, many definitions had been proposed [3, 4]. Robert Morison (1782-1834), an academic Anglo Scottish sinologist, lexicographer and translator, said *'Life is not a thing or a fluid any more than heat is. What we observe are some unusual sets of objects separated from the rest of the world by certain peculiar properties such as growth, reproduction, and special ways of handling energy. These objects we elect to call 'living things'* [4]. A scientist, environmentalist and futurist, James E. Lovelock (1919-), said *life is something edible, lovable, or lethal* [4]. Another definition presents life as *a chemical entity that consists of bounded micro-environments in chemical disequilibrium with their environment, capable of maintaining a low entropy state by energy and environment transformation, and capable of information encoding and transfer* [5, 6]. The most recent one, from 2019, defines life as *Life is a far-from- equilibrium self-maintaining chemical system capable of processing, transforming and accumulating information acquired from the environment* [6]. These definitions can be formulated differently but they share a foundation that 'life' is self-replicating and self-maintaining [3, 6]. It means plants and animals are examples of 'life', but bicycles and water, although nurturing 'life', are not 'life' themselves. What makes the difference between 'life' and inorganic matter? This is the key question that many generations of scientists have tried to answer, from Charles Darwin and Jean-Baptiste Lamarck with the evolution theory to James Watson and Francis Crick with the discovery of the double helix structure of DNA.

The study of life gave birth to a scientific discipline 'biology'. For long time, scientists have studied nature the way James Watson and Francis Crick did. We start with knowledge of a general phenomenon and we start breaking down the system to pieces until we characterized the smallest component. This is called reductionism which will be discussed in more details in the next section. Reduction-

ism paradigm led researchers in biology, especially molecular biology, to discover genes, molecules, and biological processes and to gather a tremendous amount of data about each part of an organism. These data will continue to bloom as technology improves. We just do not know quite well how we can 'solve life' with these data. This is a perfect moment to assemble these data on each part of a living system to study interactions among these parts. This is the foundation of a holistic approach to studying biological complex systems, the so-called 'Systems biology'. It is still uncertain whether holistic approaches, such as Systems biology, are sufficient to enable understanding of the emergent properties of complex biological puzzles. There is nothing certain in science but the quest for knowledge certainly never ends.

1.1 Holism and reductionism in Biology

Reductionism was first introduced in 1637 by Rene Descartes (1596-1650) in his Discourse V as *"Method consists entirely in the order and arrangement of those things upon which the power of the mind is to be concentrated in order to discover some truth. And we will follow this method exactly if we reduce complex and obscure propositions step by step to simpler ones and then try to advance by the same gradual process from the intuitive understanding of the very simplest knowledge of all the rest"*. His notion simply stated that a complex system can be studied by reducing it to more manageable pieces, studying them and reassembling the whole from its parts (Figure 1.1.1).

The reductionist approach, therefore, focuses on describing a system as its constituent parts. Reductionism allows us to draw conclusions such as "this mutation in aminoacyl-tRNA synthetases can lead to neurodegeneration" [7]. It is undeniable that many ground-breaking findings could not have been made without reductionist approaches. However, the reassembled data does not give rise to the whole as

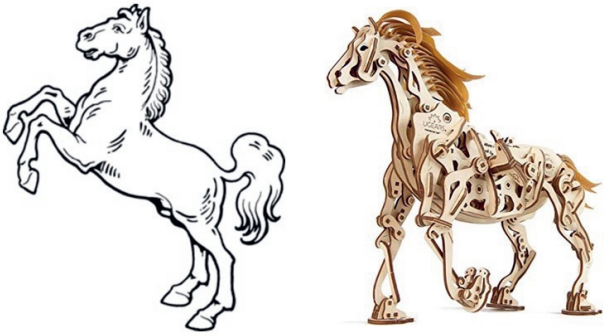


Figure 1.1.1: Holism vs Reductionism. Automata horse model obtained from https://www.amazon.com/Automata-Wooden-Mechanical-Miniature-Kinetic/dp/B07KMHL622 on August 2020. Sketch of horse obtained from http://www.clker.com/clipart-rearing-horse.html on August 2020.

we miss many interactions between these parts when separating them. This is when holism came to light.

Much debate has been raised on whether Aristotle is the antecedent of the holism paradigm with his statement *"The whole is something over and above its parts and not just the sum of them all"* . The first one to give it the term 'holism' is Smuts in 1926 in his book 'holism and evolution', where he said *"Taking a plant or an animal as a type of a whole, we notice the fundamental holistic characters as a unity of parts which is so close and intense as to be more than the sum of its parts"* [8]. Holism prioritizes the study of the whole over that of the parts. Holism does not reduce the whole to

its parts and study them. Holism had been the pillar of science until the 17^{th} century and faded away when biology developed [9]. But it now has reappeared and lead to the birth of Systems biology, in which living systems are studied as a whole instead of each part independently.

With the emergence of Systems Biology, a majority of scientists believe that reductionism and holism are in fact interdependent and complementary [10–12]. A holistic view is required to connect molecular parts learnt using reductionist approaches to higher biological phenomena. Interpreting observations from holistic studies may require mechanistic insights gained from previous reductionistic work or may generate hypotheses that are amenable to testing through reductionistic approaches.

As Smuts said *"The whole is in the parts and the parts are in the whole, and this synthesis of whole and parts is reflected in the holistic character of the functions of the parts as well as of the whole"* , holism and reductionism should be interplayed and integrated. Which one is useful is not the question. The question is how to combine them.

1.2 What is Systems biology?

Before Systems biology was defined, molecular biologists have been applying system approaches to study the molecular components and logic behind the cellular processes on a small scale. Such an example is the discovery of the feedback inhibition of amino acid biosynthesis pathways in 1975, or the discovery of *lac* operon, an autonomous functioning unit consisting of different parts that together responsible for the transport and metabolism of lactose in 1967 [13].

The term 'Systems biology' was coined in 1968 by Mihajlo Mesarovic [14, 15]. Many publications suggest Systems biology is a holistic paradigm, a modern approach to replace reductionism [15]. Although it aims to study organisms beyond

the molecular level, Systems biology is not exclusively holistic [10]. Systems biology predicts systemic responses when altering individual components. In addition, Systems biology critically relies on the availability of the experimental data obtained at the element level to assemble the whole. Reductionist approaches using in molecular and cell biology will still grow to expand our knowledge on cellular components, the critical building blocks for future Systems biology [16, 10]. Without the reductionist approach, mechanistic insights into the phenomena described by Systems biology cannot be explained.

Many attempts have been made to define Systems biology [16]. In this thesis, Systems biology is defined as a collection of quantitative and qualitative modelling approaches to study living organism with the focus on the interplay of three main networks - metabolic, regulatory, and signalling networks (Figure 1.2.1). The potential of the system to regulate itself is put in a prominent and central position in Systems biology [17]. One of the main goals of Systems biology is to test the consistency between our understanding of complex biological processes and the observed experimental data [18].

1.3 Genome-scale constraint-based metabolic models (GEMs)

The fate of any living organisms critically relies on nutrient availability and on the environmental conditions they are in. Living organisms uptake nutrients from the environment and convert them into energy and essential building blocks to sustain growth. This conversion process is termed metabolism. Ideally, when describing what is going on in the cell, one would need to quantify all the metabolic changes over time in the cell because regulation of metabolism is a key mechanism to control

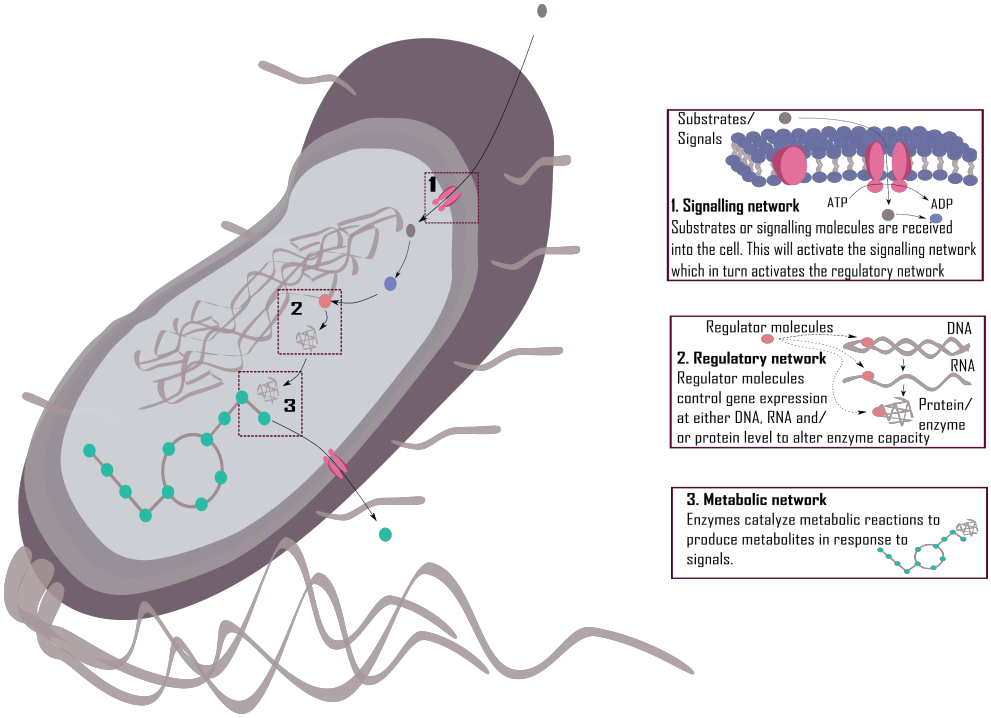


Figure 1.2.1: Phenotypes result from the interconnection of the signalling, regulatory and metabolic networks.

cell growth and death [19]. The changes in the cell rely on the rate of matter conversion. The rates at which nutrients are converted are called fluxes. These fluxes are metabolic reaction rates and are greatly affected by enzyme activity and metabolites concentration. This information is not yet available for all reactions that form the whole metabolic network, and it is likely that it will never be. Therefore, most of the modelling techniques that require kinetic parameters can only be used to describe small networks. Genome-scale constraint-based metabolic modelling (GEM) is an approach that balances the detail level and the scale it covers [20]. This type of models describes the metabolic network based on topology and fluxes. Despite the lack of kinetic information, this approach provides a quantitative tool to study the flux distribution underlying the condition specific phenotype. GEM is a comprehensive collection of known metabolic functions in the organism of interest.

Building a GEM is a tedious process requiring intensive manual curation [21]. Briefly, the construction process starts with the identification of metabolic functions from the genome of the organism in question. These functions are represented as chemical reactions giving raise to a network of metabolic processes. The metabolic network is represented in a form of a stoichiometic matrix which includes metabolites and reactions that produce and consume them. Functional genome annotation is still facing many challenges, hence missing functions in the annotate genome are expected. The network resulting from missing annotations is thus incomplete. These missing annotations result in so-called gaps that need to be solved in a subsequent stage of the reconstruction process. In the next step, the model is validated by verifying to what extent it reflects experimental data. At this stage the model is ready for the desired simulation.

There is always a limit on nutrient uptake, for example the uptake rate of glucose, resulting in bounds (or constrains) for fluxes for all reactions in the metabolism. At

exponential growth, the turn-over rates of metabolites are faster than cell growth and division, which means all intracellular metabolites will be produced and consumed at the same rates [22] and there will be no net accumulation or depletion, leading to a steady-state. This steady-state and the constraints form the core of the network analysis approach for genome-scale constraint-based metabolic models. One common technique to simulate GEMs is Flux Balance Analysis (FBA) (Figure 1.3.1). FBA is a quantitative prediction method that only requires a GEM, growth condition (i.e. availability of substrates), and an objective function (i.e. biomass synthesis) as input [22]. The system is assumed at a steady state. The stoichiometric mass-balance yields a system of linear equations, describing all fluxes within the system. The system in this method is usually underdetermined since there are more unknowns (fluxes) than equations (reactions). Such an underdetermined system gives a vastness of possible results. Metabolism is limited by several constraints which in practice will decrease the size of the solution space, that is the set of fluxes that are compatible with the imposed constraints. There are three types of constraints that can be added to GEMs [22]. The first one is thermodynamic constraints, which limit certain reactions to be irreversible. The second type of constraints relates to enzyme capacities, which restricts the upper limit of reaction rates, so-called upper bounds of certain reactions. The last type of constraints are environmental limitations. Due to the limited availability of nutrients in the environment and the limited availability of the cell to uptake them, this type of constraints will determine the upper and lower bound of certain reactions, for instance the reactions for the uptake of carbon sources.

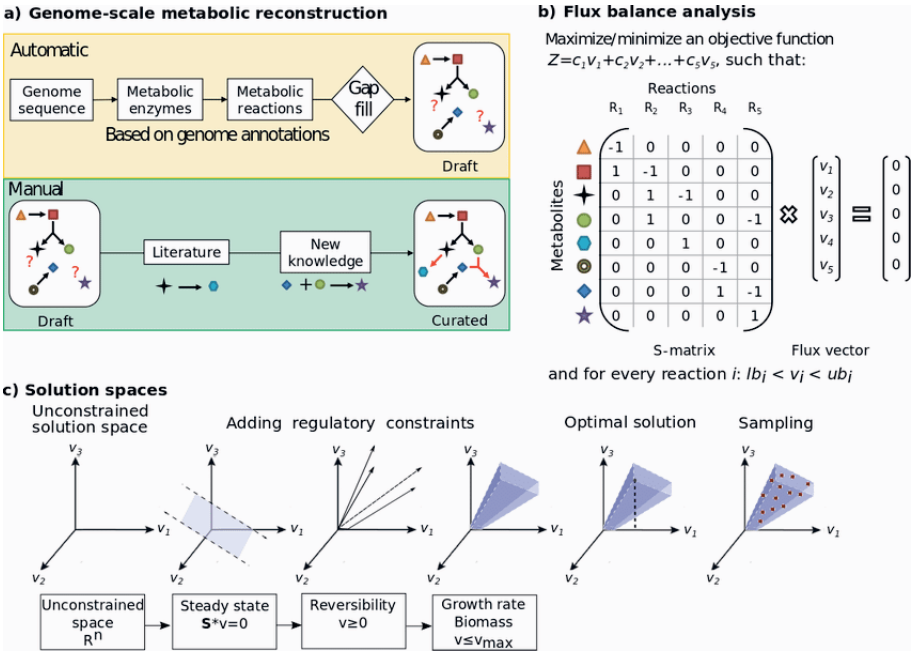


Figure 1.3.1: Key concepts in GEMs. Figure was adapted from [23]

The objective function in GEMs represents the desired phenotype that the modeler wants to optimize. Possible objective functions in GEM are to optimize biomass or ATP production, substrate consumption, and the redox potential [24]. The most commonly used objective function in GEMs is to maximize growth or biomass synthesis. This is also best to describe reality since cells have been selected by evolution for an optimal growth [25]. In a real cell, new biomass is created through a multitude of different processes that produce all molecules that needed for cell growth. In GEMs, biomass synthesis reaction is an artificial reaction which consumes the molecules and energy necessary for building new cells in an experimentally measured stoichiometric ratio [26].

GEMs can be used for several applications: (i)- GEMS are an excellent tailor-

made knowledge base for the target organism. GEM contains state-of-the-art knowledge about the metabolism in the target organism; (ii)- GEMs provide a simulation platform to quickly scan through the metabolic capacities of the target organism; (iii)- GEMS are an excellent tool to calculate maximum theoretical yields of native and non-native pathways; (iv)- Platform for contextualizing 'omics' data and studying internal fluxes which are difficult to study otherwise; (v)- Guiding metabolic engineering [27]. GEMs can predict phenotypes such as the production of metabolites, change in the growth rate or cell death when perturbing the internal environment for instance, genetic modification or external environment for instance the change of growth media. They have been used for industrial and medical applications owning to their power in hypothesis-driven discovery and in guiding metabolic engineering [28, 29]. Such successful applications are its recent use to triple the hyaluronan yield in *Lactococcus lactis* [30] or to improve antibiotic production in *Actinomycetes* [31].

However, GEMs are not accurate for processes with non-linear relationships. When regulation and signalling networks control the process, the model gives discrepancies with observation. Besides, the constraints in GEMs usually represents estimated ranges of fluxes and sometime due to the lack of data many of internal fluxes are left unbounded. This results in a huge solution space with many alternative solutions that can be biologically unfeasible [32].

1.4 Synthetic biology and Design Built Test Learn (DBTL) cycles

Synthetic biology is *an application of science, technology and engineering to facilitate and accelerate the design, manufacture and/or modification of genetic materials in living*

organisms such as microbes [33].

Microbes have been employed to produce chemicals for more than thousands of years with significant impact for instance the introduction of beverages, cheeses, bread, pickled foods and vinegar in the ancient time [34]. These early applications were mainly done without understanding how microbes arose [34]. The discovery of fermentation process by Louis Pasteur has revolutionized the use of microbes and made microbiology a distinct field [34]. Some of the significant examples are the production of lactic acid in high quantity [35], citric acid *Aspergilus niger* [35] or the establish of acetone-butanol-ethanol fermentation from *Clostridium acetobutylicum* at industrial scale in 1916 [36].

These early processes employed microbes when they were not well-characterized. Since then, with the development of high-throughput technology, our understanding about microorganism have been expanded significantly lead together metabolic engineering. In recent years numerous examples of microbial cell factories have been established for many targets from biofuels to chemicals such as amino acids, vitamins, and organic acids [37–39].

Nowadays, the emerge of Synthetic biology holds the promise to revolutionise the production of natural and non-natural bioproducts [40–42]. First successful industrial applications have reported, being some noteworthy examples the production and commercialization of semi-synthetic artemisinin, an antimalarial drug precursor from *Saccharomyces cerevisiae* [43] and biofuel precusor 1,4-butanediol from *Escherichia coli* [44]. The development of model design and prediction in Systems biology combined with advanced tools in Synthetic biology have the potential to allow large-scale modification and reprogramming non-model organisms [45–47]. This will enable expanding the list of potential microbial factories and bioproducts that can be brought to the market, contributing thereby to shift a petrochemical-based to

a more bio-based economy.

The Design - Build - Test - Learn (DBTL) cycle provides a framework for the development of tailor-made microbes and speed up the innovation process. The DBTL cycle is a recursive loop used to obtain a design that satisfies the desired specifications (Figure 1.4.1). It is a framework that helps systematize bioengineering and increase its efficacy and generalizability. It usually takes more than one DBTL cycle to achieve the desired product [48].

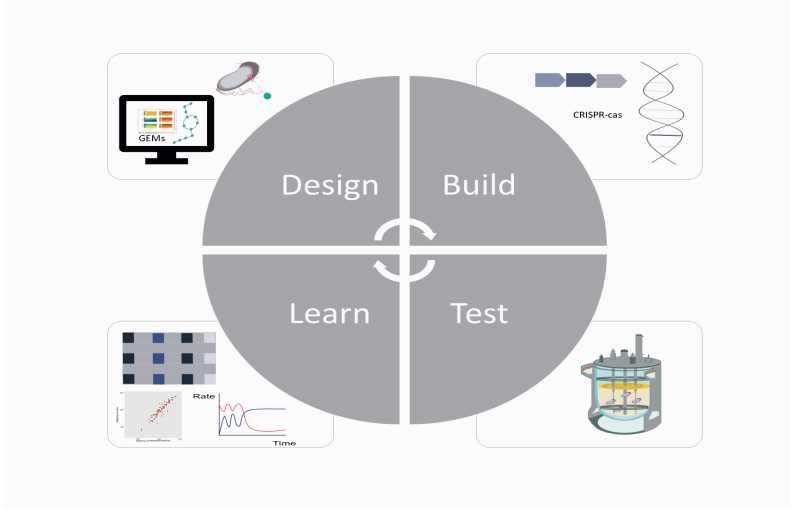


Figure 1.4.1: DBTL cycles

DBTL cycle has been used successfully in many processes such as in shorting the time for the market release of antimalarial drugs, the semi-synthetic Artemisinin in less than a decade making the re-engineering yeast strain the most profitable in the market [43, 49, 50]. DBTL cycle is shown to be also useful in plant engineering as demonstrated by the recent success in synthesis and accumulation of energy-dense plant storage lipids in vegetative tissues such as root, leaves and stems [50].

A successful DBTL cycle depends on four criteria: the speed, productivity, quality of each step and the number of cycles to obtain the final product. How to speed up, to improve each step and to link individual cycles to make a consistent solution is, therefore, a bottleneck [51]. To this end, it is crucial to ensure there are appropriate tools and methods that can be integrated into this workflow [52, 48, 53]. To address this problem in bio-engineering an automated workflow has been suggested [54]. The workflow allows quick prototyping and optimization of synthetic pathways in a target microbial chassis. Being a cycle, the DBTL workflow can technically start anywhere in the four steps of the design-build-test-learn sequence. Most often, when starting a new project, the workflow starts by assembling all available knowledge on the process at stake (molecular and physiological data, process information, etc.) and use it as input for the Design phase where pathway design tools are used to design pathways for a product of interest. These pathways are ranked and *in silico* screened for promising candidates. The next step is to Build the DNA constructs followed by the Test phase where all candidate pathways are tested with different configurations. In the next step, the Learn phase, statistical tools and machine learning (among other) are used to select the best configuration. The output is then subjected to a refined Design for further optimization in the next cycle. The final outputs of this recursive loop are optimal pathways and genetic constructs aiming to produce the target compound(s) in the microbial chassis.

1.5 Thesis objective and outline

The objectives of this thesis are to deploy GEMs for selected microbial cell factories, to evaluate key technical limitations of GEMs and to propose possible solutions to overcome these.

One of the most used applications of GEMs is their use towards understanding

the metabolism of the organism in consideration. In **Chapter 2** I constructed a GEM for *Cutaneotrichosporon oleaginosus* to model its lipid production. *C. oleaginosus* is a fast-growing oleaginous yeast that can grow in a wide range of low-cost carbon sources. I constructed a GEM to increase our understanding of this yeast and provide a knowledge base for further industrial use.

Living organisms are minuscule chemical factories where carbon in different forms is converted to thousands of valuable compounds. Producing chemicals from living cells has been considered a sustainable approach to life. However, biosynthesis of many natural compounds is still limited due to the lack of efficient synthesis routes. As a showcase of how GEMs can assist in designing pathways for chemical production in microbes, in **Chapter 3** I deployed GEMs to design pathways for cis,cis-muconic acids, anisole, aniline, 3-methylmalate and geranic acid production in *Pseudomonas putida* in the context of implementation of DBTL cycles.

A critical step in constructing GEMs is to manually curate them by integrating information from independent (organism-specific) sources to provide a comprehensive representation of what is presently known about the metabolism of the modelled organism. Combining this precious information from individual GEMs to make a consensus model of the organism is essential. Using models from different species as a foundation to construct a new model can help to avoid repeating the same time-consuming manual curation step. In addition, GEMs need to be updated continuously since new knowledge is coming in short order. However, such simple tasks cannot be done easily due to a simple reason: inconsistent namespace. GEMs constructed for different organisms by different researchers often use different naming conventions depending on which database was selected for model construction. While mapping between namespaces seems like the only fair solution, it involves a high risk of mismatch and may invalidate the model. I evaluated this problem and

proposed solutions to overcome them in **Chapter 4**.

The lack of accurate functional annotations often renders GEMs incomplete, giving rise to missing reactions, the so-called 'gaps' in the network. Gap-filling becomes important during model construction not only to make a functional model but also to generate new knowledge on protein function. To assist gap-filling, many algorithms have been published. To be able to use GEMs effectively, these methods should allow the model to be as accurate as possible, preferably also in a user-friendly manner so that they become available to many researchers. However, gap-filling algorithms vastly differ in their objectives, implementation platforms, and input data requirements. These differences imply a variety in their usability and accuracy. In **Chapter 5** I conducted an extensive evaluation of these algorithms from a user's perspective.

Finally, in **Chapter 6** I will discuss the two main limitations, namely the lack of standard in namespace and gap-filling tools in a broader context. Other limitations and recommendations to improve them will also be discussed in **Chapter 6**.

2

Genome-scale metabolic modelling underscores the potential of *Cutaneotrichosporon oleaginosus* ATCC 20509 as a cell factory for biofuel production

Pham Nhung, Maarten Reijnders, Maria Suarez-Diez, Bart Nijsse, Jan Springer, Gerrit Eggink, and Peter Schaap.

Accepted for publication in Biotechnology for Biofuels

ABSTRACT

Cutaneotrichosporon oleaginosus ATCC 20509 is a fast-growing oleaginous basidiomycete yeast that is able to grow in a wide range of low-cost carbon sources including crude glycerol, a byproduct of biodiesel production. When glycerol is used as a carbon source, this yeast can accumulate more than 50% lipids (w/w) with high concentrations of mono-unsaturated fatty acids. To increase our understanding of this yeast and to provide a knowledge base for further industrial use, a FAIR re-annotated genome was used to build a genome-scale, constraint-based metabolic model containing 1553 reactions involving 1373 metabolites in 11 compartments. A new description of the biomass synthesis reaction was introduced to account for massive lipid accumulation in conditions with high carbon to nitrogen (C/N) ratio in the media. This condition-specific biomass objective function is shown to better predict conditions with high lipid accumulation using glucose, fructose, sucrose, xylose, and glycerol as sole carbon source. Contributing to the economic viability of biodiesel as renewable fuel, *C. oleaginosus* ATCC 20509 can effectively convert crude glycerol waste streams in lipids as a potential bioenergy source. Performance simulations are essential to identify optimal production conditions and to develop and fine tune a cost-effective production process. Our model suggests ATP-citrate lyase as a possible target to further improve lipid production.

2.1 Introduction

Microbial lipids produced by oleaginous yeasts are promising sources for oleochemical replacements of hazardous petrochemicals in fuels and chemicals [55, 56]. For the establishment of an economical bio-based utilization, cost-effective production is key. Of the fewer than 30 known oleaginous yeasts, the top five most studied species are *Yarrowia lipolytica*, *Rhodotorula glutinis*, *Rhodosporidium toruloides*, *Cutaneotrichosporon oleaginous*, and *Lipomyces starkeyi* [55]. The profile of lipids and fatty acids produced by these yeasts varies but, under natural conditions they can, on average, accumulate lipids up to 40 % of their weight [57, 58]. A lipid content of up to 70 % can be obtained if in the presence of a carbon source, an essential nutrient is depleted [58]. Under such conditions, excess carbon will be re-routed to storage compounds, being lipids in oleaginous yeasts [59, 57]. Nitrogen limitation, often referred to as a high C/N ratio has been shown to be the most efficient inducer of such lipid accumulation [58].

As input materials are one of the main contributors to production cost [60], for an economically feasible process, a natural capacity for high lipid biosynthesis may not be enough. Oleaginous yeasts are able to use a range of alternative sugars for lipid production (Table 2.1.1). Among them, *C. oleaginosus* appears to be one of the most accommodating and is able to grow in a wide range of industrially interesting operational conditions such as in food waste and municipal wastewater streams [61], whey permeate [62], office paper production waste streams [63, 64], spent yeast lysate from brewery industry and crude glycerol from biodiesel production [65, 66]. Lipid production by this yeast has been studied for at least two decades [58, 62, 67–70] and when growing on crude glycerol, *C. oleaginosus* can produce more lipid content than many other yeast, microalgae or mold (Table 2.1.1). Owing to these advantages, *C. oleaginosus* is flagged as one of the most cost-effective and versatile cell factories for *de*

novo lipid production [55, 71]. Especially when the inexpensive waste product from biodiesel production, crude glycerol, is becoming abundantly available, this organism could play a major role in further upcycling of the biodiesel process, as lipids derived from *C. oleaginosus* grown on glycerol have high concentrations of monounsaturated fatty acids (MUFA) [72]. MUFAs are excellent biodiesel components due to their low temperature fluidity and oxidative stability [72].

Table 2.1.1: Lipid yields obtained by oleaginous yeasts.

Organism	Yield*	Carbon source	Reference
Yarrowia lipolytica	0.27	Glucose	[73]
Yarrowia lipolytica	0.10	Crude glycerol	[74]
Rhodosporidium toruloides	0.29	Lignocellulosic hydrolysates	[75]
Rhodotorula glutinis	0.18	Molasses	[76]
Lipomyces starkeyi	0.24	Glucose	[77]
Cutaneotrichosporon oleaginosus	0.22	Glucose	[78]
Cutaneotrichosporon oleaginosus	0.29	Whey permeate	[62]
Cutaneotrichosporon oleaginosus	0.27	Crude glycerol	[72]
*g-lipid/g-substrate			

C. oleaginosus is a basidiomycete yeast of the *Tremellomycetes* class and recently added to the *Cutaneotrichosporon* genus [79]. Taxonomically, it has been reclassified and renamed several times as *Apiotrichum curvatum*, *Cryptococcus curvatus*, *Trichosporon cutaneum*, *Trichosporon oleaginosus*, and *Cutaneotrichosporon curvatum* [65, 66]. In this study, we will refer to it as *Cutaneotrichosporon oleaginosus* ATCC20509 [80, 81]. The yeast can metabolize a wide range of oligo- and monomeric sugars such as cellobiose, xylose, sucrose, lactose, and glucose [82]. Xylose is efficiently metabolized via the phosphoketolase pathway and partly via the pentose phosphate pathway [65, 83]. Both pathways produce pyruvate as intermediate for further metabolic

processes [65].

Despite many efforts spent on studying this yeast, its use for the production of lipids is still far from optimized [55, 64, 84]. Recently, a response surface method was used to design experiments to optimally explore the relationship between the carbon to nitrogen ratio in the medium and lipid production and to guide the design of optimal production media for *C. oleaginosus* ATCC20509 [85]. However, the translation from genotype to (selected) phenotype [i.e. lipid production], is typically a multi-factorial process depending on the growth medium, culture conditions, strain specificity and the interplay among these factors. Hence, a predictive constraint-based, genome-scale model of metabolism (GEM), along with genetic accessibility tools [86] will provide new avenues towards reaching the full potential of *C. oleaginosus* ATCC 20509 as a lipid producer [65].

By drawing upon a thorough functional re-annotation of its genome, we have built a GEM for *C. oleaginosus* ATCC 20509. The model is named *i*NP636_*Coleaginosus*_ATCC20509, expanding the usual naming convention for GEMs [87] by including information on the organism considered to enhance recognition. Subsequently, the model was used to investigate optimal lipid production in glycerol.

2.2 Results and Discussion

2.2.1 Annotation

One of the major bottlenecks in eukaryotic genome annotation is the identification of exon-intron boundaries. In this regard, transcriptome data can provide a good basis for predicting introns. We therefore collected transcriptome data (RNAseq) of

C. oleaginosus ATCC 20509 from two conditions and used it to structurally annotate genome sequence MATS00000000.1 of *C. oleaginosus* ATCC 20509 [88].

BRAKER1 [89] predicted 7861 protein coding genes. Of these, 7474 genes are directly supported by RNAseq with more than 50 read counts per million (CPM). Among the protein-coding genes, 5621 proteins with functional protein domains (Pfam release 31) and 2358 with a full unique Enzyme Commission (EC) number could be predicted. A summary is provided in Table 2.2.1. A complete annotation is provided in Additional file 1.

2.2.2 Lipid synthesis pathways

C. oleaginosus ATCC 20509 metabolizes sugars by using standard central metabolic pathways including glycolysis, pentose phosphate pathway and the citric acid (TCA) cycle. The yeast metabolizes xylose via the phosphoketolase pathway and partly via the pentose phosphate pathway [65, 83]. These pathways provide the precursors and energy required for lipid biosynthesis. Lipid biosynthesis can be divided into three steps: formation of fatty acids, synthesis of triacylglyceride (TAG), and synthesis of phospholipids (Figure 2.2.1).

Formation of fatty acids

In yeasts, fatty acids can derive from either a *de novo* synthesis pathway or from hydrolysis of complex lipids and delipidation of proteins, and from hydrolysis of external fatty acids sources [90]. *De novo* fatty acid synthesis generally occurs in the cytosol [58], and in some cases, in the mitochondrion [91]. This produces saturated fatty acids of up to 16 C atoms while further elongation and desaturation takes place in the endoplasmic reticulum (ER) [58, 92]. The process is catalyzed by the

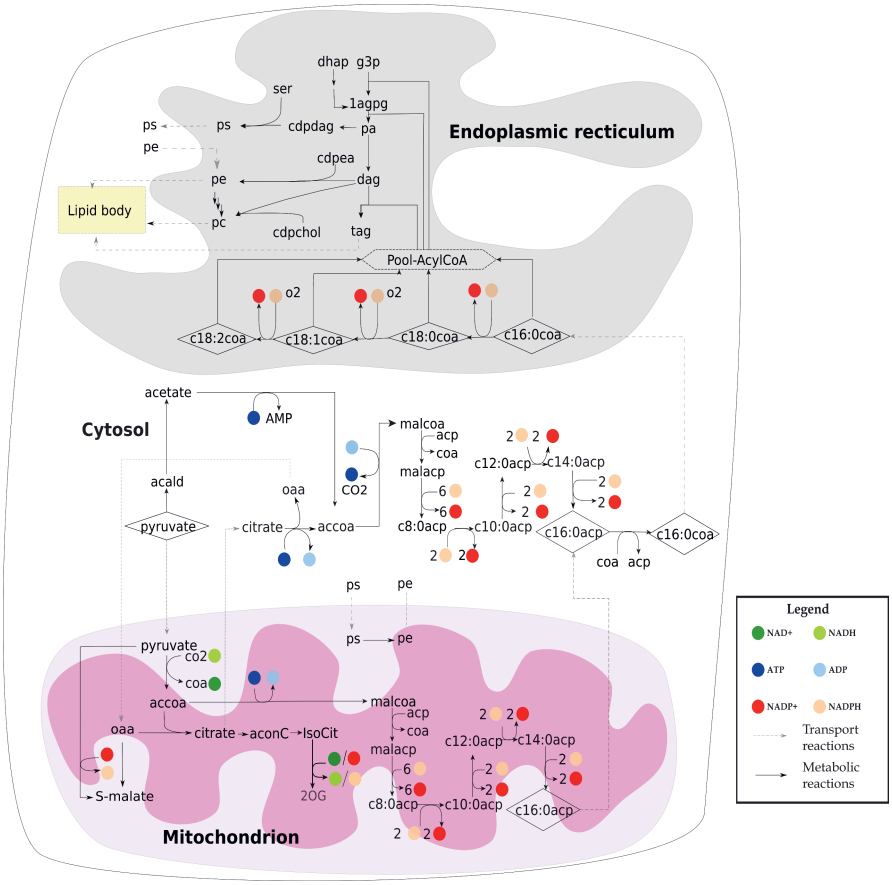


Figure 2.2.1: Lipid synthesis pathway in *C.oleaginosus* ATCC 20509. **1agpg** - 1-Acyl-sn-glycero-3-phosphoglycerol; **2OG** - 2-Oxoglutarate; **acald** - acetaldehyde; **accoa** - acetyl-CoA; **aconC** - cis-Aconitate; **acp** - acyl carrier protein; **c8:0acp** - octanoyl acyl carrier protein; **c10:0acp** - decanoyl acyl carrier protein; **c12:0acp** - dodecanoyl acyl carrier protein; **c14:0acp** - tetradecanoyl acyl carrier protein; **c16:0acp** - Hexadecanoyl acyl carrier protein; **c16:0coa** - Hexadecanoyl-coa; **c18:0coa** - Octadecanoyl-coa; **c18:1coa** - oleoyl-CoA ; **c18:2coa** - linoleoyl coA; **cdpdag** - CDP-Diacylglycerol; **cdpea** - CDP-ethanolamine; **cdpchol** - CDP-choline; **coa** - coenzyme A; **dag** - diacylglycerol; **g3p** - glyceraldehyde 3-phosphate; **oaa** - oxaloacetate; **IsoCit** - Isocitrate; **malacp** - malonyl acyl carrier protein; **malcoa** - malonyl-CoA; **pa** - phosphatidate; **pc** - phosphatidylcholine; **pe** - phosphatidylethanolamine; **ps** - phosphatidylserine; **ser** - serine; **tag** - triacylglycerol.

multi-enzyme fatty acid synthetase complex (FAS) [58]. We found multiple genes, g2870.t1, g5734.t1, g570.t1 and g5733.t1, that together encode this enzyme complex in *C. oleaginosus* ATCC 20509. The overall process of fatty acid synthesis in *C. oleaginosus* ATCC 20509 (Figure 2.2.1) can be simplified as follows:

- **ATP- citrate lyase (ACL)**. Citrate + ATP $\rightarrow$ oxaloacetate + acetyl-CoA + ADP + P_i
- **Acetyl-CoA Carboxylase (ACC)**. acetyl-coA + CO_2 + ATP $\rightarrow$ malonyl-CoA + ADP + P_i
- **Fatty acid synthetase (FAS)**. acetyl-CoA + 7 malonyl-CoA + 14 NADPH + 14 H^+ $\rightarrow$ palmityl-CoA + 14 $NADP^+$ + 7 CoA + 7 CO_2

For the formation of unsaturated fatty acids (C16:1, C18:1, and C18:2) a fatty acid desaturase is required [93]. A single gene, g3345.t1, was predicted to encode this enzyme in *C. oleaginosus* ATCC 20509.

Synthesis of triacylglyceride and phospholipids

Like other oleaginous yeast, the process of triacylglyceride (TAG) synthesis in *C. oleaginosus* ATCC 20509 also starts with the formation of phosphatidic acid (PtdOH) from glycerol-3-phosphate either through the glycerol-3-phosphate or the dihydroxyacetone phosphate pathway [91, 94] (Figure 2.2.1). PtdOH is subsequently converted to diglyceride which later with the addition of one acyl-coa becomes triacylglyceride.

The main phospholipids in *C. oleaginosus* ATCC 20509 are phosphatidylcholine, phosphatidylethanolamine , and phosphatidylserine [95]. They are synthesized

from the CDP-diacylglycerol (CDP-DAG) and the Kennedy (or CDP-choline) pathways [96, 92] (Figure 2.2.1).

We provide more details of the reconstructed *C. oleaginosus* lipid synthesis pathway in Additional file 2.

2.2.3 Features of the Model

The GEM for *C. oleaginosus* was constructed using the well-curated GEM iNL895 [97] of the oleaginous model organism *Y. lipolytica* as template. A template based approach is often more efficient than starting from scratch however, as the use of a template could limit the scope of the specific GEM, we did an in depth *C. oleaginosus* specific curation of the here important target pathways, i.e. the fatty acid and lipid synthesis. Of the 895 genes underlying the *Y. lipolytica* model, *de novo* genome annotation followed by manual curation led to the identification of 636 orthologs genes in *C. oleaginosus* ATCC 20509 that were used to generate the *i*NP636_*Coleaginosus*_ATCC20509 GEM. A full list of orthologs is provided in Additional file 3. Both models cover the central carbon and lipid metabolism but, accounting for the differences in lipid and fatty acid profiles in these two organisms, in lipid formation there are differences in the number of isoenzymes involved. A comparison of enzymes involved lipid metabolism of *C. oleaginosus* ATCC20509, *Y. lipolytica* and the non-oleaginous model yeast, *Saccharomyces cerevisiae* is shown in Table 2.2.2.

Compared to *S. cerevisiae* there are few differences. *S. cerevisiae* lacks an ATP-citrate lyase and does not have the gene encoding for a Δ12 Fatty acid desaturase, which introduces the second double bond in the biosynthesis of 18:3 fatty acids. In *S. cerevisiae*, acetyl-CoA is produced from Acetyl-coenzyme A synthetase encoded

Table 2.2.1: Genome annotation results for *Cutaneotrichosporon oleaginosus* ATCC 20509.

Annotation features	Results
Genome size (Mbp)	19.86
No. of protein coding genes	7861
Protein length (median no. of amino acids)	409
Gene length (median bp)	1708
Transcript length (median bp)	2460
No. of genes with intron	6891
Proteins with at least one functional domain assigned	5621
No. of predicted (partial) EC's	627
No. of predicted (full) unique EC's	1072
Proteins with a predicted (full) EC's	1778

by two distinct genes *ACS1* and *ACS2* representing the "aerobic" and "anaerobic" forms of acetyl-coenzyme A synthetase, respectively [99]. In *C. oleaginosus* ATCC 20509 and *Y. lipolytica*, there is only one acetyl-coA synthase gene, similar to *ACS2* in *S.cerevisiae*. After curation, the final GEM (*i*NP636_*Coleaginosus*_ATCC20509) contains 1553 reactions, 1373 metabolites, 636 genes, and 11 compartments: cytoplasm, Golgi apparatus, cell envelope, endoplasmic reticulum, mitochondrion, nucleus, peroxisome, vacuolar membrane, vacuole, lipid particle, representing lipid droplets, and extracellular (Table 2.2.3 and Figure 2.2.2).

Biomass synthesis reaction

The biomass synthesis reaction included in the model represents the formation of the main building blocks required for growth of the target organism [100, 101]. Application of growth-limiting nutrients, however, may induce large variations

Table 2.2.2: Enzymes involved in lipid metabolism in *Saccharomyces cerevisiae* model, iNL800 [98], *Yarrow lipolytica* model, iNL895 [97] and *Cutaneotrichosporon oleaginosus* ATCC 20509 model, *i*NP636_*Coleaginosus*_ATCC20509 (this study). Y indicates the presence of the enzyme-encoding gene, (-) indicates the absence of the enzyme-encoding gene. Number of isoenzymes is indicated in brackets.

EC	Function	*S.cerevisiae* (iNL800)	*Y. lipolytica* (iNL895)	*C. oleaginosus* ATCC 20509
EC 6.2.1.1	Acetyl-coenzyme A synthetase 1	Y	-	-
EC 6.2.1.1	Acetyl-coenzyme A synthetase 2	Y	Y	Y
EC 1.3.1.9	Fatty acid synthase subunit beta	Y	Y (2)	Y
EC 2.3.1.86	Fatty acid synthase subunit alpha	Y	Y (2)	Y
EC 2.7.7.41	Phosphatidate cytidylyltransferase	Y	Y (2)	Y
EC 2.7.8.11	CDP-diacylglycerol–inositol 3-phosphatidyltransferase	Y	Y	Y
EC 2.7.8.8	CDP-diacylglycerol–serine O-phosphatidyltransferase	Y	Y	Y
EC 2.7.1.30	Glycerol kinase	Y	Y	Y (2)
EC 1.1.1.8	Glycerol-3-phosphate dehydrogenase (NAD(+))	Y (2)	Y	Y (2)
EC 2.3.1.51	Probable 1-acyl-sn-glycerol-3-phosphate acyltransferase	Y	Y (2)	Y
EC 2.3.1.20	Diacylglycerol O-acyltransferase	Y	Y	Y
EC 2.3.1.158	Phospholipid:diacylglycerol acyltransferase	Y	Y	Y
EC 3.1.1.3	Triacylglycerol lipase	Y (3)	Y (2)	Y (2)
EC 2.3.1.26	Acyl-CoA:sterol acyltransferase	Y	Y	Y
EC 1.14.19.1	Acyl-CoA desaturase	Y	Y	Y
EC 1.14.19.6	Δ12 Fatty acid desaturase	-	Y	Y
EC 1.3.3.6	Acyl-coenzyme A oxidase	Y	Y (3)	Y
EC 2.3.1.16	3-ketoacyl-CoA thiolase, peroxisomal	Y	Y	Y (2)
EC 2.3.3.8	ATP-citrate lyase, subunit a	-	Y	Y
EC 2.3.3.8	ATP-citrate lyase, subunit b	-	Y	Y
EC 1.1.1.38	NAD-dependent malic enzyme, mitochondrial	Y	Y	Y
EC 6.4.1.2	Acetyl-CoA carboxylase	Y	Y	Y

Table 2.2.3: Characteristics of *i*NP636_*Coleaginosus*_ATCC20509 model. Unique metabolites indicate species regardless of compartment.

Categories	Features
Total reactions	1553
Gene associated reactions	1142 (84%)
Exchange reactions	189
Transport reactions	486
Metabolic reactions	878
Total metabolites	1373
Unique metabolites	786
Genes	636
Compartments	11

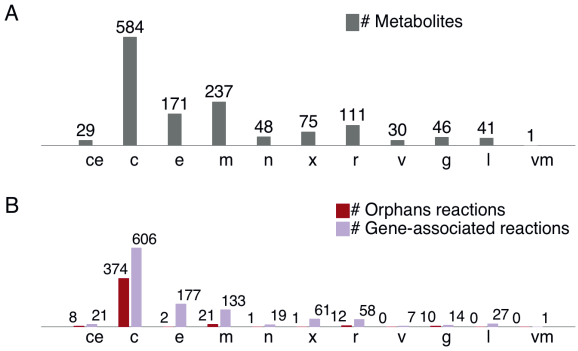


Figure 2.2.2: Distribution of (A) metabolites and (B) reactions among compartments in *i*NP636_*Coleaginosus*_ATCC20509. Orphan reactions are exchange reactions, transport reactions, spontaneous reactions and reactions with no associated catalyzing genes. c- cytosol, ce- cell envelope, e- extracellular, g- Golgi, l-lipid particle, m-mitochondrion, n-nucleus, r-endoplasmic recticulum, v- vacuole, vm- vaculor membrane, x-peroxisome.

in biomass composition. Model flux distributions can be very sensitive to such changes, compromising the predictive accuracy of the metabolic model [101].

The biomass composition of *C. oleaginosus* ATCC 20509 was shown to vary along with the C/N ratio in the medium [62, 85, 102], as in nitrogen limiting conditions excess carbon is converted in lipids.

Experimental data show an increase in lipid content with increasing C/N ratio [85] until a maximum is observed at C/N ratio of 120 g/g (Figure 2.2.3). The link between lipid content and C/N ratio can be approximated by a quadratic relationship, as shown in Figure 2.2.3.

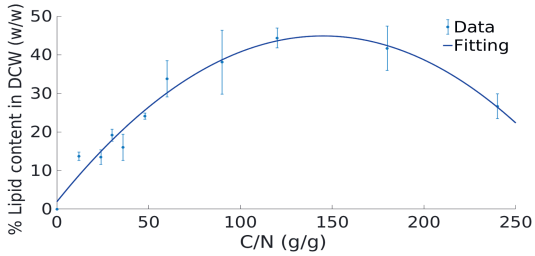


Figure 2.2.3: Lipid content in *C. oleaginosus* ATCC 20509 dry cell weight (DCW) at different C/N ratio. *In-vitro* data was obtained from [85]. Fitting line corresponds to $y = -0.002 * x^2 + 0.59 * x + 1.9$

In addition to lipids, the biomass content of protein and carbohydrate also varies with the C/N ratio [102]. Here we model weight fraction of biomass that corresponds to carbohydrates, proteins and total lipids in the biomass using:

$$0.11 biomass_{Carbohydrate} + biomass_{Protein} + biomass_{totalLipid} + 0.05 biomass_{other} = biomass \quad (2.1)$$

The remaining 5 % of the biomass weight is assigned to RNA, DNA, minerals and co-

factor content. As these represent minor quantities in the biomass, their coefficients are assumed to be constant. Upon nitrogen starvation, the yeast cells start accumulating intracellular sugars [102] as short-term energy storage [65, 103]. These intracellular sugars will be then converted to long-term energy storage in form of lipid droplet [102]. Furthermore, nitrogen depletion leads to a decrease in protein content as proteins are used as nitrogen source. No changes in carbohydrate profile in the cell wall under nutrient shortage conditions has been reported [65]. Therefore, we assume that nitrogen depletion will lead to a maximum carbohydrate content in the cell and excess carbon will be rerouted for lipid synthesis. Data from [62] at a relatively low C/N ratio (2.8) suggest 11% as a reasonable and conservative estimate for this weight fraction. Combining this expression and the relationship in Figure 2.2.3, a biomass synthesis reaction for nitrogen starvation can be dynamically built for any C/N ratio (Additional file 4).

The amount of lipids in the biomass reaction varies along with the C/N ratio, however the lipid composition does not change. TAGs still make up 90 % of total lipid in *C. oleaginosus* ATCC 20509 [58] and phospholipids for the remaining 10% [58]. The phosholipids, phosphatidylserine, phosphatidylethanolamine and phosphatidylcholine are added with equal weights. Finally, the fatty acid content of lipids (25 % hexadecanoic (C16:0), 10 % octadecanoic acid (C18:0), 57 % oleic acid (C18:1), and 7 % linoleic acid (C18:2) [59, 58]) can also considered to be stable.

2.2.4 Lipid production and growth in *C. oleaginosus* ATCC 20509

Effect of the C/N ratio on lipid production

We compared simulation results from our model, *i*NP636_*Coleaginosus*_ATCC20509, with simulations obtained from the response surface method [85] using either a fixed standard or a condition specific biomass objective function. The results are presented in Figure 2.2.4.

When the condition-specific biomass objective function is applied (Figure 2.2.4B) GEM predictions are better aligned with predictions obtained with the response surface method in [85] (Figure 2.2.4A) underpinning the crucial role of high C/N ratios in lipid production.

Effect of the carbon source on lipid production and growth

Carbon sources have been shown to have different effects on growth and lipid production in oleaginous yeast [55, 62]. *C. oleaginosus* ATCC 20509 is able to grow on glycerol, sucrose, glucose, fructose, ethanol or xylose as sole carbon source [65, 85] and *in silico* growth was evaluated on these sources (Figure 2.2.5).

Overall, except for ethanol, growth was predicted in all tested carbon sources. In our *in silico* experiment, uptake rates were adjusted for each carbon source to guarantee the same C-mol was provided. On all tested carbon sources, the model predicted favorable growth in rich nutrient-conditions. Comparable growth rates were obtained in sucrose, glucose, fructose and xylose (Figure 2.2.5). Lower growth rate was obtained for glycerol while no growth was obtained when ethanol was used as sole carbon source.

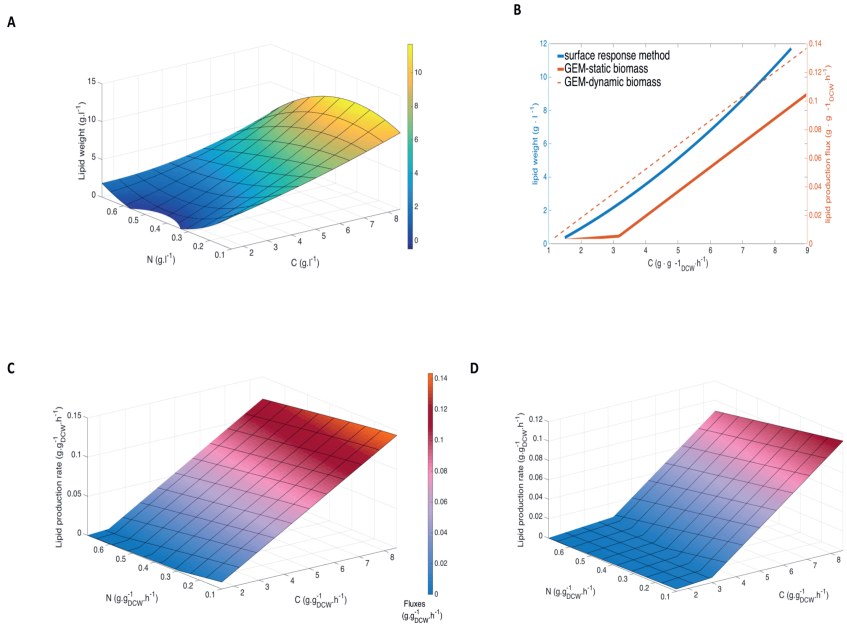


Figure 2.2.4: Simulations of impact of C/N ratio on lipid production by *C. oleaginosus* ATCC 20509. Gluscose is used as a carbon source as in the response method in [85] for (B) and (C).

A: Simulation using the surface response model [85],
B: A comparison of lipid production at a fixed N concentration at 0.3 (g/l) between surface response method in [85] and *i*NP636_*Coleaginosus*_ATCC20509 using a standard biomass (static) and condition-specific (dynamic) biomass.
C: *i*NP636_*Coleaginosus*_ATCC20509 simulation using the proposed condition-specific biomass objective function,
D: *i*NP636_*Coleaginosus*_ATCC20509 simulation using a standard biomass objective function.

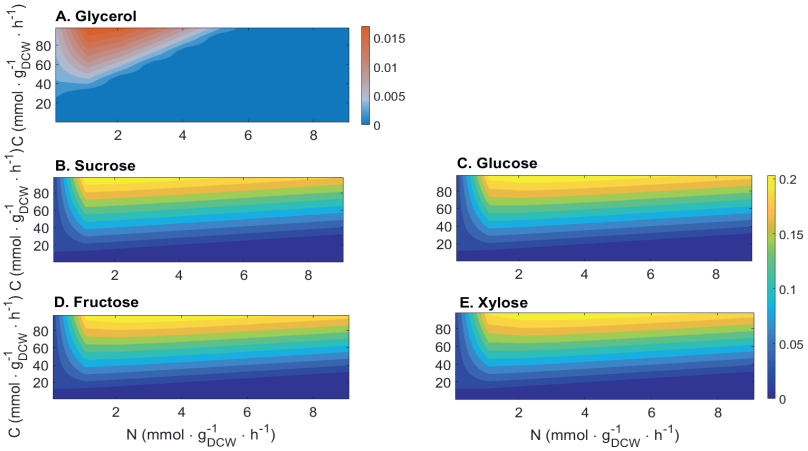


Figure 2.2.5: *In – silico* growth rate when using different carbon sources. X and Y axis start from 0.1. The color bars indicate growth rate (h^{-1}).

Effects of 5-carbon sugars, i.e. xylose, 6-carbon sugars, i.e. glucose and fructose, and of sucrose on growth in *C. oleaginosus* ATCC 20509 have been studied extensively, and results varies among these studies. According to [85], comparing fructose, glucose, xylose and sucrose, *C. oleaginosus* ATCC 20509 grows the fastest in fructose, the slowest in sucrose while there is no significant difference between glucose or xylose. According to [86] xylose is favored over glucose for biomass generation. These differences can be due to various factors such as pH, temperature, oxygen, dilution rate, and fermentation modes across experiments. When growing in different fermentation modes, i.e. batch, fed batch, and continuous fermentation, the microorganisms are subjected to differences in environment, substrate availability and by-product concentration [104]. In addition, different carbon sources may have different uptake rates. These factors can result in different growth rates, biomass and by-product accumulation. In this study, we simulated the process in continuous fermentation and assumed the same uptake rate for all carbon sources.

As for growth, a similar trend was predicted for lipid production on different carbon sources (Figure 2.2.6). For all tested carbon sources, the model predicted highest lipid production at high C/N ratios. Model prediction for lipid production in glycerol is noticeably different from that of other carbon sources. This is consistent with findings in [105] who reported a maximum growth rate and lipid production of *C. oleaginosus* ATCC 20509 on glycerol in a fed-batch fermentation mode at $16g/l$ glycerol and $0.27g/l$ NH_4Cl, corresponding to a C/N ratio of 100 mol/mol.

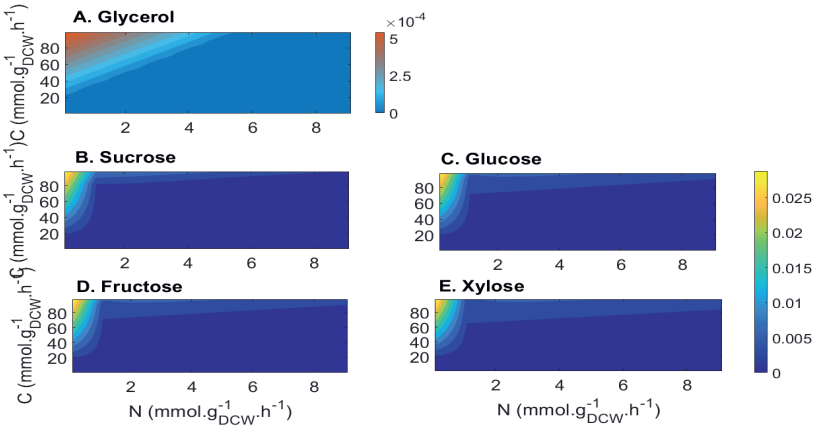


Figure 2.2.6: *In — silico* lipid production rate when using different carbon sources. X and Y axis started from 0.1. The color bars indicate lipid production rate $(mmol \cdot g_{DCW}^{-1} \cdot h^{-1})$.

The model predicted the highest lipid production rate in sucrose, glucose, fructose, and xylose. Glycerol gave lower lipid production rate. Similar to growth, literature also captured contrasting findings on lipid production on different carbon sources. Across various carbon sources, xylose was found the most suitable sugar source for lipid yield in batch and chemostat cultures [82]. On the other hand, a

lower lipid production on xylose compared to glucose were reported in other studies [85, 106]. This disagreement between studies can be due to influence of other factors such as temperature, oxygen and fermentation mode [65].

Acetyl-CoA source for lipid production in *C. oleaginosus* ATCC 20509

Lipid synthesis requires a constant supplement of fatty acid and fatty acid synthesis in turn requires a continuous supplement of acetyl-CoA [92]. In non-oleaginous yeast such as *Saccharomyces cerevisiae*, the main source of acetyl-CoA is from the ligation of acetate and coenzyme A by acetyl-coA synthase [92]. While, oleaginous yeast such as *Y. lipolytica*, do not have the gene encoding for acetyl-CoA synthase [92].

The main source for acetyl-CoA in oleaginous yeast is believed to be from the cleavage of citrate to release acetyl-CoA and oxaloacetate in the cytosol by ATP:citrate lyase [92]. It implies that there is a continuous export of citrate from the mitochondria to the cytosol. Our model with the assumed chemostat cultivation also predicts this. The flux of the citrate transport reaction increases positively with ATP:citrate lyase whose flux also increases sharply after passing C/N ratio of 10 g/g (Figure 2.2.7). Fluxes through acetyl-coA pool and lipid synthesis reaction also surged after passing the same C/N ratio (Figure 2.2.7). The large standard deviations in Figure 2.2.7 represent alternative flux distributions that are compatible with the set constraints. This variability reflects both the metabolic flexibility of this organism and the lack of sufficient data to fully constrain the model, a common problem in GEM model analysis.

As reported in [107], after passing the critical C/N of 11 g/g, when the nitrogen concentration is limiting further growth, the yeast starts to accumulate more

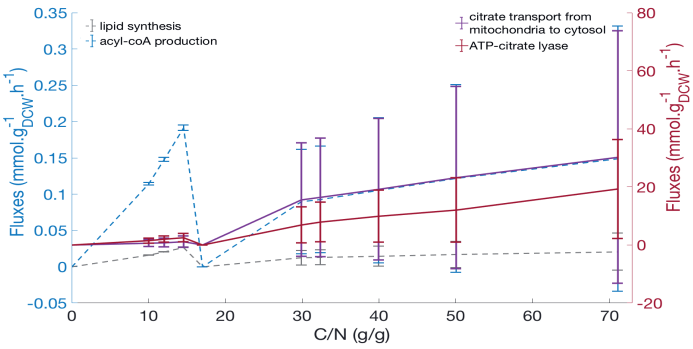


Figure 2.2.7: *In − silico* flux analysis of acetyl-CoA source for lipid synthesis. The citrate transport from mitochondria to cytosol and ATP-citrate lyase which catalyzes the reaction ATP + citrate + Coenzyme-A → acetyl-CoA + oxaloacetate + P_i + ADP in cytosol follow the red Y-scale; acyl-coA production (an artificial reaction represents acyl-coA Pool in the model); and lipid synthesis (an exchange reaction of lipid) follow the blue Y-scale. Bars indicate the standard errors of means of fluxes through each reaction. The C/N (g/g) refers to the ratio between the uptake rates of carbon and ntirogen sources.

lipid. In order to sustain cellular functioning, the cell degrades AMP to inosine monophosphate and ammonium ions [92, 57]. A decreased AMP concentration in turn down-regulates the activity of isocitrate dehydrogenase [92, 108, 109, 57]. This enzyme converts citrate to isocitrate. Its down-regulation, therefore, leads to the accumulation of citrate in mitochondria. Accumulated citrate is then exported to cytosol where it is hydrolysed to acetyl-CoA and oxaloacetate by ATP:citrate lyase [92, 57]. This process provides more acetyl-CoA for fatty acid synthesis which further enhances lipid production in the cell [57]. Although FBA analysis does not account for regulation, the same trend was observed in our simulations, that clearly indicate the association between increased flux through ATP:citrate lyase reaction and lipid production (Figure 2.2.7). Furthermore, model simulations show no alternative lipid production pathway as *in-silico* growth is inhibited when simulating a knock out of this enzyme. Our model suggests ATP-citrate lyase as the main source for acetyl-CoA suggesting that overexpression of ATP-citrate lyase can help to further improve lipid production. This strategy has been successfully implemented in *Y. lipolytica* [110].

Lipid metabolism regulation

The effect of nitrogen limitation on lipid production was studied by analyzing the effect of the C/N ratio on (i) the *in-silico* flux distribution and (ii) the transcriptional landscape of *C. oleaginosus* ATCC 20509 grown on glycerol.

(i) *in-silico* flux distribution: We tested lipid production at different C/N ratios while keeping the carbon concentration constant at either 16, 24 or 32 g/g DCW (Figure 2.2.8) as for the same C/N ratio the absolute amount of carbon supplied has been shown to greatly affect lipid production [62]. The model predicted

that for the higher C/N ratios, more carbon is required to sustain lipid production. With 16 g/g DCW carbon no lipid could be produced at a C/N ratio of 240 (g/g). Likewise, with 24g carbon, no lipid formation was predicted at a C/N ratio of 300 (g/g). Only with 32 g/g DCW carbon, lipid accumulated at the complete range of C/N ratio's tested.

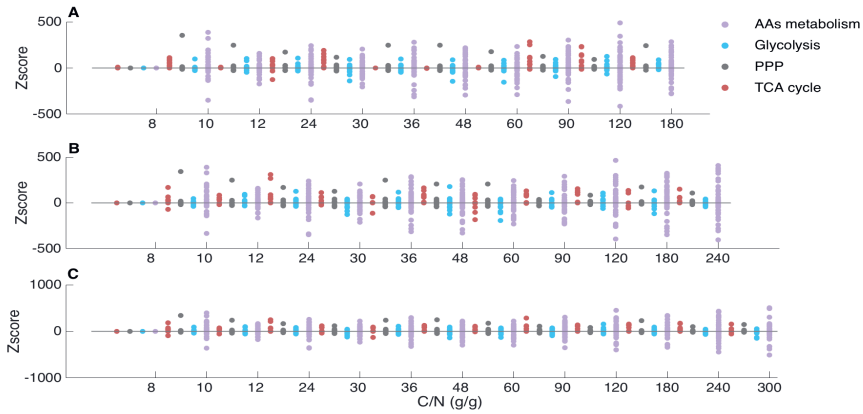


Figure 2.2.8: *In – silico* flux changes in *C. oleaginosus* ATCC 20509 at different C/N ratio with (A) 16g carbon; (B) 24g carbon; (C) 32g carbon using glycerol as a sole carbon source. A C/N ratio of 6 g/g was used as reference point to calculate Zscore for each C/N ratio. The C/N (g/g) refers to the ratio between the uptake rates of carbon and nitrogen sources. Zscore > 0 indicates an increase in flux compare to that at reference point; Zscore < 0 indicates a decrease in flux compare to that at reference point. PPP - Pentose Phosphate Pathway; AAs metabolism - Amino acids metabolism. Each dot in the graph represents a reaction in the corresponding pathway.

For the three tested carbon concentrations, the same trends in flux distribution were obtained (Figure 2.2.8). When increasing C/N ratio, a majority of reactions in TCA and PPP have their fluxes increased. This could be due to the high de-

mand of reducing power, i.e. NADPH, of lipid production. Fluxes through glycolysis are greatly diverse when changing glycerol concentration. Reactions related to glucose catabolism such as hexokinase (D-glucose:ATP) and glucose-6-phosphate isomerase (PGI) have their fluxes reduced. Down regulation of PGI was reported to lead to the accumulation of intracellular sugar which is later converted to lipid in the nitrogen-depletion stage [102]. Downstream reactions in glycolysis such as 6-phosphofructo-2-kinase, pyruvate kinase and acetyl-CoA synthetase have their fluxes increased. Upregulation of these enzymes can be a result from a high demand of precursors for lipid accumulation.

As already mentioned in [111] we also observed flux fluctuations in amino acid metabolism (Figure 2.2.8). Fluxes through enzymes in amino acid degradation pathways, i.e. argininosuccinate lyase, L-hydroxyproline dehydrogenase (NAD), and L-glutamate 5-semialdehyde dehydratase increase at a high C/N ratio. This is expected since amino acid degradation provides an alternative source for nitrogen upon limitation. Reactions in amino acids synthesis such as glutamine synthetase and ornithine decarboxylase, on the other hand, had their fluxes reduced.

(ii) Nitrogen limitation induced transcriptional changes: RNAseq data was obtained from *C. oleaginosus* ATCC 20509 when growing in a glycerol medium with initial C/N (g/g) ratio of 28 and 2.8 respectively. Nitrogen is significantly depleted at the time of sampling (Table 2.4.1). There were 7272 genes expressed in high C/N ratio medium and 7246 genes expressed in low C/N ratio medium ($>$ 50 Counts Per Million). When comparing low C/N ratio to high C/N ratio medium, 75 genes were found to be up-regulated and 26 were down-regulated (see Additional file 5). Interestingly, the majority of these genes code for unknown protein functions. No genes involved in primary metabolism were found to have significant different expression level in either low or high C/N ratio.

In response to nitrogen starvation the gene expression levels of many genes in the lipid synthesis pathway were reported to fluctuate in *Y. lipolytica* [112], in contrast Kerhoven E.J. et al [111] reported no significant change in transcription level of these genes under nitrogen limitation. Using xylose as carbon source the Acetyl-CoA carboxylase (ACC) gene was found to be upregulated in *Trichosporon oleaginosus* strain IBC0246 under nitrogen limitation [113] and the same authors also reported significant upregulation of fatty acid synthetase (FAS1 and FAS2), malic enzyme and ATP-citrate lyase (ACL) under these conditions.

In our case, upon growth in glycerol, RNAseq analysis showed no difference in transcription level of genes involved in lipid synthesis pathway in *C. oleaginosus* ATCC 20509. The model however, was able to predict lipid production at different C/N ratio qualitatively consistent with experimental data. This suggests that in glycerol *C. oleaginosus* ATCC 20509 lipid metabolism is not regulated at the transcriptional level. Pathway flux is controlled by simultaneous multisite modulation through action on a number of enzymes [114]. This suggests that other regulatory effects, such as regulation of translation or allosteric effects may dominate in *C. oleaginosus*. In *Lipomyees starkeyi*, an oleaginous yeast, and *Aspergillus niger*, a citric producing yeast, ATP:citrate lyase, the key enzyme in lipid synthesis is controlled by the energy charge and fatty acid acyl CoA esters [115]. While human ATP:citrate lyase activity has been reported to be regulated by *in-vitro* allosteric effects via phosphorylation [116]. Little is know on the regulation of this enzyme in *C. oleaginosus*.

2.3 Conclusions

In this study, we introduced the first GEM for *C. oleaginosus* ATCC 20509 and as such *i*NP636_*Coleaginosus*_ATCC20509 represents a valuable platform to integrate, interpret and combine many decades of experimental efforts since its first iso-

lation from a dairy farm in 1978 [117, 67]. The model gave qualitative predictions at different C sources consistent with experimental data, highlighted the lipid production lifestyle of *C. oleaginosus* ATCC 20509 and pinpointed ATP-citrate lyase as a target to further improve lipid production. Analysis of RNAseq revealed that lipid production in *C. oleaginosus* ATCC 20509 in glycerol does not appear to be regulated at the transcriptional level.

C. oleaginosus is known to have a great potential for lipid production due to its efficient growth on inexpensive carbon sources such as glycerol. Our simulations show that its potential has not yet been fully explored and can be optimized further. The predictive accuracy of *i*NP636_*Coleaginosus*_ATCC20509 renders its great potential for future studies to guide metabolic engineering for the production of high value industrial compounds such as polyunsaturated plant-like fatty acids.

2.4 Materials and Methods

2.4.1 *C. oleaginosus ATCC* 20509 experimental data collection

The strain was cultivated in the same basal medium as described in [62] except for the glycerol and NH_4Cl concentration which was adapted in order to achieve the chosen C/N ratio. A C/N ratio of 28 was obtained by adding 16 g/l glycerol and 1 g/l NH_4Cl (medium A), while in other sample, 8 g/l glycerol and 5 g/l NH_4Cl was added to make a C/N ratio of 2.8 (medium B). The C/N ratios were taken from [62], which shows *C. oleaginosus* grows at a C/N ratio of less than 5, and lipid production for a ratio between 20 and 40 carbon / nitrogen.

Two biological replicates for each condition were inoculated from a freshly prepared YPD-agar plate in 50 ml of YPD medium and grown O/N in a 100 ml Erlemeyer flask at 30 ° C and 225 rpm. The culture was divided in two 25 ml portions

and centrifuged (10 min. 300 rpm) to collect the cells. The cell pellets were resuspended in 30 ml medium A or medium B. 4 ml of the resuspended cells was used to start duplicate cultures in medium A and B which were incubated for 18 hours at 30 °C and 225 rpm. Each culture was divided in two equal portions and the cells were harvested by centrifugation and the wet pellet frozen in liquid and used for RNA extraction.

We measured the concentration of glycerol and NH4Cl in the medium at the initial condition and at the sampling point (Table 2.4.1). Glycerol and NH4Cl were measured with HPLC analysis and NH4 chemical analysis, respectively.

Table 2.4.1: Glycerol and NH4Cl levels obtained from HPLC analysis & NH4 chemical analysis

	OD 650		glycerol (g/L)		NH4Cl (g/L)		C/N (mol/mol)	CDW* (mg/ 50 ml)
	T=0h	T=18h	T=0h	T=18h	T=0h	T=18h	T = 18h	
A1	2.75	12.2	20.5	11.3	1.1	0.022	896	404
A2	2.9	11.8	20.5	11.4	1.1	0.015	1325	365
B1	2.8	10.4	10.2	1.6	4.7	2.9	0.96	445
B2	2.85	10.4	10.2	2.5	4.7	3.6	1.2	445

* Cell dry weight

2.4.2 RNA extraction procedure

RNA was extracted using an acidic hot phenol extraction procedure. Briefly, the cell pellet was ground in liquid nitrogen and mixed with 4 volumes of pre-warmed (60°C) phenol + extraction buffer (1% SDS, 10 mM EDTA, 0,2 M NaAc (pH 5)) after this 2 volumes of chloroform were added and mixed thoroughly. After centrifugation the buffer layer was washed once with chloroform. RNA was precipitated from the buffer layer by adding 8 M LiCl to and end concentration of 2M. After centrifugation the pellet was washed once with 2M LiCl and twice with 70%

ethanol. The remaining pellet was resuspended in RNase free water. Total RNA extract, RNA sequencing, and RNAseq data processing were performed as described in [62]. Samples were sequenced by NovoGene using Total RNA.

2.4.3 RNAseq analysis

Raw read counts in two C/N ratios, 2.8 and 28 (mol C/mol N), were obtained with the RNA-seq aligner STAR (v2.6.0b) [118] using the parameter "–quantMode GeneCounts", the public genome sequence MATSoooooooo.1 of *C. oleaginosus* ATCC 20509 and the GTF file obtained from BRAKER1. Read count data were then analyzed using DESeq2 [119] to identify genes that have different expression when changing the C/N ratio. Two biological replicates for each condition were provided. The statistical significance of gene expression differences was evaluated using a false discovery rate (FDR) < 0.05 and $|\log_2 (\text{fold change})| \geq \log_2 1.5$ as a threshold.

2.4.4 Genome sequence

The genome sequence MATSoooooooo.1 from *Cutaneotrichosporon oleaginosus* ATCC 20509 reported by [88] was annotated and used to build the model. The genome sequence has 19.86 Mbp and a GC content of 60.7%.

2.4.5 Genome annotation

Unsupervised RNA-Seq-based gene prediction of *C. oleaginosus* ATCC 20509 was performed with BRAKER1 v1.10 [89] in combination with HISAT2 (v2.1.0) [120] using all the RNAseq datasets combined.

The genome, predicted gene structures and their proteins sequences were directly stored in the SAPP semantic (RDF) database [121] using the GBOL ontology [122]. Protein signature prediction was done with a standalone version of InterProScan v5.24.64.0 [123] using the default databases. EnzDP [124] was used to assign EC numbers to Proteins. This is with a confidence score cut-off of 0.2. Both tools were used in direct interaction with the previous mentioned SAPP database.

2.4.6 Construction of *i*NP636_*Coleaginosus*_ATCC20509 model

Software environment

The model was read, modified and analyzed in MATLAB (version R2015b) [125], using COBRA toolbox 3.0 [23] and GLPK [126] as a linear solver.

Construction of the draft model

A draft model was constructed using the scaffold-based method described in [97]. A GEM of *Y. lipolytica* , considered as a model for oleaginous organism [92, 56] was chosen as a reference scaffold. There are 5 published models for *Y. lipolytica* iNL895 [97], iYL619 [127], iMK735 [128], iYALI4 [111], iYLI647 [129]. *Y. lipolytica* iNL895 model [97] was used as a scaffold because it contains the most reactions and genes and was also constructed based on the *Saccharomyces cerevisiae* model, iIN800 [98] which was specialized for lipid synthesis.

To find ortholog proteins from *Y. lipolytica* to *C. oleaginosus* ATCC 20509, the enzyme-coding-genes obtained from *Y. lipolytica* iNL895 model were functionally annotated in the same manner as *C. oleaginosus* ATCC 20509 and stored in the SAPP database. A combination of the protein signatures, EC prediction, BLAST and manual curation was used to find the orthologues.

If an ortholog gene was found in *C. oleaginosus* ATCC20509 , the associate reaction in the scaffold iNL895 was kept. In addition, exchange and non-enzymatic transport reactions for the medium were kept. Spontaneous and growth essential orphan reactions from the scaffold were also preserved. This step resulted in a draft model for further curation.

Curation of the draft model

In order to build a working GEM the draft model expanded and refined in the following manner:

(i) The lipid synthesis pathway was curated based on KEGG [130], literature [92, 58, 59, 57] as well as experimental data in [62, 105].

(ii) The central metabolic network, including glycolysis, pentose phosphate pathway and TCA cycle were manually curated based on literature [131, 132].

(iii) Growth associated maintenance energy (GAM) was adopted from *Y. lipolytica* model, iNL895. Non-growth associated maintenance energy in *C. oleaginosus* ATCC 20509 is known to be relatively low in comparison with other yeasts [62], in the model this value was set as $1\ mmol \cdot g_{DCW}^{-1} \cdot h^{-1}$.

(iv) The draft model was further curated by removing gaps, irrelevant reactions and infeasible energy production cycles. Method described in [133] was employed to identify infeasible energy production cycles in our model. In short, we added energy dissipation reactions for ATP, CTP, GTP, UTP, NADH, NADPH, FADH2 and proton with unconstrained bounds. Fluxes through all network reactions, except the added energy dissipation reactions were constrained to range [-1,1] for reversible and [0,1] for irreversible reactions. No uptake nutrients were allowed. Each energy dissipation reaction was maximized to identify the presence of infeasible loops.

Development of a condition specific biomass function

The four major macro-molecules of living cells are proteins, carbohydrates, nucleic acids and lipids [134]. The ratio between them are assumed to be different in different conditions. We assumed lipid, protein and carbohydrate makeup 95 % of the cell dry weight. Depending on the C/N ratio in the medium, the ratio between them will vary. Nucleic acids and other cofactors and mineral only make up a small fraction of the biomass, and kept constant. Using data from literature, we parametrized the relationship between the biomass and carbohydrates, proteins and lipids under nitrogen starvation using:

$$0.11 biomass_{Carbohydrate} + biomass_{Protein} + biomass_{totalLipid} + 0.05 biomass_{other} = biomass \tag{2.2}$$

This assumes that under nitrogen starvation, 11% of the cellular biomass corresponds to carbohydrates, 5% to nucleic acids and other components and the remaining fraction correspond to proteins and total lipids. We used the experimental data in

[85] to model the C/N ratio in the media and lipid accumulation using a quadratic regression (Figure 2.2.3) with a correlation coefficient of 0.98. This enables the estimation of the contribution of lipids w_{TL} to the biomass. For this, C and N uptake rates were used to compute the ratio between both components as we assumed a simulation scenario (chemostat) where not net accumulation of either one happens. Using this approach, we can generate specific biomass reaction at any C/N ratio using carbon source and nitrogen source uptake rates as the sole inputs. Details regarding components and their coefficients in the biomass reaction at normal condition, i.e. when there is no nitrogen depletion, can be found in (Additional file 6). Finally, the biomass equation was standardized to have a molecular weight of $1 \cdot g/mmol$.

The main lipid building-blocks are fatty acid residues. The majority of fatty acid in *C. oleaginosus* ATCC 20509 is oleic acid (C18:1) [58, 59]. When growing on glucose, the composition of main fatty acids in *C. oleaginosus* ATCC 20509 are 25 % hexadecanoic (C16:0), 10 % octadecanoic acid (C18:0), 57 % oleic acid (C18:1), and 7 % linoleic acid (C18:2) [59, 58]. As specific information about each fatty acid in lipid molecules is not available for *C. oleaginosus* ATCC 20509, in the model, an artificial acyl-CoA pool for lipid synthesis was formulated. A reaction representing the acyl-CoA pool was introduced:

$$0.24952C_{16:0} + 0.096712C_{18:0} + 0.55233C_{18:1} + 0.067963C_{18:2} \rightarrow Acyl - CoA_{pool}$$

Coefficients of fatty acids in the acyl-CoA pool reaction represent their weight percentages in the lipid of *C. oleaginosus* ATCC 20509 according to data in [59, 58].

Growth simulation

Model accuracy was validated using flux balance analysis (FBA) implemented in COBRA Toolbox 2.0.6 [135] in MATLAB environment. Minimum defined medium was used. Unlimited uptake rates of CO_2, H_2O, H^+, O_2, $Iron^{2+}$, phosphate, potassium, sodium, sulphate, and NH_4 were allowed. This entailed setting the lower bounds of the corresponding exchange reactions to -1000, as we used the usual convention of writing the exchange reactions in such way that production corresponds to positive fluxes and consumption to negative ones. These constraints were kept for all simulations.

The gold standard validation technique in GEMs is to compare model prediction to experimental data. *In-silico* growth simulation in the presence of different carbon sources was carried out, for this uptake rate of the corresponding carbon source was constrained to -10 $mmol \cdot g_{DCW}^{-1} \cdot h^{-1}$. Biomass reaction with experimentally determined content at nitrogen abundant condition was used.

2.4.7 Investigation of lipid synthesis in *C. oleaginosus* ATCC 20509

Simulations of growth and lipid production

We conducted *in-silico* experiments to assess the effect of C/N ratio on lipid production in *C. oleaginosus* ATCC 20509. To compare our prediction with simulation from the response surface method [85], we mimic the experimental set up in [85].

To generate different C/N ratios, C mmol and N mmol were calculated from the data in [85] where nitrogen was set up in the range of [0.1:0.01:0.8] g, carbon was in [1.5 : 0.05 : 8.5] g with urea and glucose as nitrogen and carbon source, respectively. We assumed a constant uptake of carbon and nitrogen. As reported in [107] after passing a critical C/N of 12.83 (mol/mol) or 11 (g/g) the biomass reaches the

maximum value of 0.20 $h^{-}1$. Thus, to simulate lipid production we fixed the growth rate for subsequent optimizations. If the *in-silico* growth rate at the tested C/N ratio was higher than 0.2 $h^{-}1$, we fixed the biomass lower bound and upper bounds to $[0.2 \cdot 0.9, 0.2]$. For growth rates smaller than 0.2 $h^{-}1$, we fixed the biomass to the maximal predicted values at the corresponding C/N ratio. A specific biomass reaction for each C/N ratio was used. Lipid formation happens when the cell was subjected to sudden depletion of other nutrients such as nitrogen after growing maximally [59]. To mimic this process, in our simulation, biomass function was constrained to the set values and exchange reaction of lipid body was maximized. We did not constraint biomass when simulating for growth.

To study effects of different carbon sources on growth and lipid production, lower bound and upper bound of each exchange reaction for glucose, fructose, xylose, sucrose, ethanol and glycerol was constraint in each study. To generate different C/N ratio the uptake rate was increased gradually in the range of -[0.1: 5: 100] $mmol \cdot g_{DCW}^{-1} \cdot h^{-1}$ for carbon source and of -[0.1 : 1: 10] $mmol \cdot g_{DCW}^{-1} \cdot h^{-1}$ for nitrogen.

Sampling the solution space when shifting C/N ratio

To study how flux distribution change when changing C/N ratio, we sampled the solution space at steady state for each C/N ratio. Based on [107, 85] we selected the C/N ratio as [6,8,10, 12,24,30,36,48,60,90,120,180,240]. To study the effect of carbon concentration on lipid synthesis we simulated lipid production at 3 different C (g) as [16,24,32] for the same C/N range. Minimal medium was used. The solution space at steady-state for each C/N ratio when optimizing for lipid production with constrained biomass (see section "Simulations of growth and lipid production") was sampled using gpSampler [136] implemented in COBRA toolbox 2.0.6 [135]. The

sample was taken for 5000 sample points with no bias, "maxtime" was 10 minutes, "maxsteps" was set to 10^{10} and 1 thread was used. Sampling results were analyzed as described in [137]. In short, means and standard deviations were calculated from the sampling results to obtain Zscores for each reaction in the central metabolic network. A C/N ratio of 6 (g/g) was used as the reference point to calculate Zscores for fluxes in other C/N ratio.

To study the main source of acetyl-coA for lipid synthesis in *C. oleaginosus* ATCC 20509, we sampled the solution space at steady state for each C/N ratio. The C/N ratio data from experiment in [107] were used for the simulation. To mimic their experimental set up, uptake rate of nitrogen, in form of urea, was fixed at -25 $mmol \cdot g_{DCW}^{-1} \cdot h^{-1}$. Carbon was gradually increased to generate the desired C/N ratio.

Additional Files

Additional files of this work can be found online at https://www.researchsquare.com/article/rs-26414/v1

3

Design of pathways for chemical production in *Pseudomonas Putida* KT 2440

Pham Nhung, Peter J. Schaap, Maria Suarez-Diez and Vitor A. P. Martins Dos Santos

ABSTRACT

Advances in metabolic engineering and synthetic biology make microbial production a promising alternative over chemical synthesis. However, long innovation time, unfamiliarity with new biocatalysts, costly scale-up and non-flexible industrial set up hamper the shift from petro-based to bio-based chemical production. Pathways design and ranking are critical steps to facilitate the biosynthesis production because any pathway design coming downstream in the pipeline will be customized for the chosen pathways. Speeding up the design phase saves time and effort for engineering phases. In this study, we used a combination of tools based on retrosynthesis and genome-scale, constraint-based metabolic models to design biosynthesis pathways for five different classes of compounds of industrial interest: cis,cis-muconic acid, aniline, anisole, geranic acid, and 3-methylmalate in *Pseudomonas putida*. We established a general, systematic workflow to rank these pathways based on thermodynamic feasibility, enzyme sequence availability, and maximum theoretical yield. Using our approach, we discovered pathways that have not been accounted for before to produce these compounds. We illustrate this in detail for cis,cis-muconic acid, a well-characterised platform chemical for which we identified 2 fully new pathways despite of the wealth of information previously available. We have thus shown here a successful approach to quickly design and select potential chemical production pathways by combining systematically retrosynthesis and genome-scale models.

3.1 Introduction

Chemicals play an undeniable significant role in our life. These compounds are primarily synthesized relying on petrochemical feedstocks. Although efficient, this petro-based approach is considered unsustainable and raises lots of environmental issues. On the quest for a more sustainable production approach, microbial cell factories are gaining attention. The biobased chemical production is considered more sustainable due to the use of available renewable biomass instead of fossil resources [138, 139]. In addition, this approach also produce less greenhouse gas and operate at mild conditions such as low temperature and pressure [138, 139]. Using microbes for chemical production also bring other advantages such as:(i)- microorganism can grow low-cost on renewable biomass which can be waste products, hence a stable production cost is ensured [140]; (ii)- microbes produce high yields of valuable chemicals with less byproduct [141]; (iii)-the flexible of the microbial systems allows the production of a wider range of chemicals that may not be possible or are difficult to make chemically [140].

Microbes have been employed to produce chemicals for more than thousands of years with significant impact for instance the introduction of beverages, cheeses, bread, pickled foods and vinegar in the ancient time [34]. These early applications were mainly done without understanding how microbes arose [34]. The discovery of fermentation process by Louis Pasteur has revolutionized the use of microbes and made microbiology a distinct field [34]. The oldest industrial application of fermentation was to produce lactic acid in high quantity from fermented milk in 1841 [35]. The microbe that carried out this fermentation was later characterized as lactic yeast in 1858 [35]. A few years later, in 1893 the first industrial process to produce citric acid from fungi, which is recently known as *Aspergilus niger*, was established [35]. Another significant achievement of chemical production from microbes was the es-

tablish of acetone-butanol-ethanol fermentation from *Clostridium acetobutylicum* at industrial scale in 1916 [36]. This process was developed by the chemist Weizmann who became the first president of Israel some years later [34].

These early processes employed microbes when they were not well-characterized. In recent years numerous examples of microbial cell factories have been established for many targets. Recent thorough reviews feature an extensive metabolic map with various pathways for the production of more than 435 chemicals and materials [38] and cellulosic ethanol, the second-generation bioethanol [39]. Many chemicals such as amino acids, vitamins, organic acids, and other compounds have been successfully produced from microbes at commercial scale [37].

Still we have not reached the maximum potential of microbes and the shift from petro-based synthetic chemical to bio-based production is incredibly moderate [48, 42]. Long innovation time, costly scale-up, and non-flexible industrial set up make it difficult to implement conceptual laboratory research to actual industrial production [140, 42]. The majority of natural compounds of interest fall into four main groups: alkaloids, flavonoids, terpenoids, and polypeptides [142]. Choosing a target for chemical production involves the consideration of various aspects in relation to techno-economic and life cycle analysis such as operation cost, product estimated price, and environmental impact [142]. Establishing a new chemical is expensive and difficult, implementation of biosynthesis of natural compounds is still limited to only standard groups of compounds such as alcohols, organic acids, and amino acids [143, 37].

In order to connect research and industry, there is a need for a product-independent standardized workflow that can shorten the innovation time and reduce the implementation cost. This is the core of the design-build-test-learn (DBTL) paradigm in synthetic biology, an iteration cycle to design production

strategies that fit implementation requirements [144]. This cycle has long been applied in engineering but has just recently adopted in synthetic biology [145, 146, 54]. An automatic workflow has been suggested to optimize the cycle [54]. The workflow allows quick prototyping and optimization of synthetic pathways in a target microbial chassis. The process starts with the Design of pathways for the biosynthesis of the product of interest. These pathways are ranked and screened for promising candidates. The next step is to Build the DNA constructs followed by the Test phase where all candidate pathways are tested with different configurations. In the next step, the Learn phase, statistical analysis and machine learning methods are deployed to select the best configuration. Then the cycle is iterated for further optimization. The final output of this recursive loop is optimum pathways and plasmid constructs to produce the target chemical in a micro-host of choice.

Designing routes to produce the chemical compound of interest is the first step in the biosynthesis approach. It is usually done manually based on expert knowledge or literature. This restricts the number of pathways one can find and is even harder to design pathways for less-studied compounds. Many computational methods have therefore been developed to allow the exploration of a bigger knowledge pool and hence ensure higher chances to find production routes. These techniques employ different algorithms to predict pathways such as graph topology [147], stoichiometric matrix [148] or retrosynthetic search [149]. The later approach, retrosynthesis, is unique cause it is based on molecular signatures of substrates and products to predict enzyme promiscuity, a new substrate for the known enzyme hereafter new reactions and pathways. Retrosynthesis approach has been used in chemistry for ages but very recent in pathway design. Its unique prediction ability allows the discovery of new knowledge, hence is more suitable when finding new pathways for well-studied compounds or designing pathway for uncommon or even non-natural chemicals.

The retrosynthesis approach can predict numerous potential pathways. We wanted to study the performance of retrosynthesis methods to identify pathways for natural compounds. To asset the validity of the method, we include well-studied compounds to serve as positive control. To expand our exploration over standard compounds of interest, in addition to the well-known chemical, we also selected compounds that have little literature about their biosynthesis production. In this study, we aimed to design pathways for cis,cis-muconic acid, geranic acid, anisole, aniline, and 3-methylmalate (Table 3.1.1). Cis,cis-muconic acid, a six-carbon diacid, is of great interest because it can be converted to adipic acid, an important industrial platform chemical for the synthesis of various plastics and polymers. Many biosynthesis pathways and patents such as anthranilte [150], shikimate [151], 2,3-dihydroxybenzoic [152], phenol [153], tyrosine [153] and chorismate pathway [154] have been introduced. Nevertheless, it is still mainly produced chemically from benzoate which is not a sustainable feedstock.

Target	PubChem CID	Class	Application
Cis,cis-muconic acid	5280518	Alkaloid	Precursor for plastic and polymers
Anisole	7519	Alkaloids	Agriculture, consumer & pharmaceutical products
Aniline	6115	Alkaloids	Agriculture & consumer-use products
3-methylmalate	558882	Polyketides	Cosmetics, pharmaceutical products & biodegradable plastics
Geranic acid	5275520	Terpenoids	Agriculture

Table 3.1.1: Target chemicals for pathway design

Anisole, a monomethoxybenzene, is a plant metabolite that has been widely used in industry and healthcare as precursors for fragrances, insect pheromones, and pharmaceuticals [155]. Research has been done for isolating -anisoles related compounds such as brominated phenols and anisoles from natural sources, i.e. marine

worms [156, 157], algae [158] and sponges [159]. Yet no effort has been spent in implementing biosynthesis of anisole, it is still mainly produced chemically by alkylation of phenol with methanol [160].

Aniline is the simplest aromatic amine with a phenyl group attached to an amino group. The chemical is commercially important for making a wide variety of products such as polyurethane foam, agricultural chemicals, synthetic dyes, antioxidants, stabilizers for the rubber industry, herbicides, varnishes, and explosives [161]. The oldest way to produce aniline was from the distillation of indigo, a natural blue substance extracted from plants [162]. Nowadays, aniline is mainly produced chemically by the reduction of nitrobenzene and the amination of phenol with ammonia [163].

3-methylmalate - also known as 3-methylmalic acid or 2-Hydroxy-3- methylsuccinate, is classified as a member of the hydroxy fatty acids. Due to their unique properties, hydroxy fatty acids are excellent materials for many applications from cosmetics, pharmaceutical products to precursors for plastics and the unique family of biodegradable plastic, polyhydroxyalkanoates [164, 165]. However, the production of hydroxy fatty acids is difficult and expensive [164]. The limitation in chemical catalysts to make hydroxylation of fatty acids [166] and the lack of a low-cost biosynthesis pathway for hydroxy fatty acids make their commercial production unavailable [167]. Although some attempts to make biosynthesis production of hydroxy fatty acids was introduced but mainly from the conversion of vegetable oils that strongly depend on oil crops [167].

Geranic acid is a plant terpenoids with strong inhibitory activity against pathogenic fungi [168]. Geranic acid has been successfully produced in maize by cloning a geraniol synthase. In 2014, the first *de novo* biosynthesis of geranic acid in *P. putida* was reported. This pathway involves the introduction of a truncated geran-

iol synthase (GES) from *Ocimum basilicum* and the complete mevalonate pathway from *Myxococcus xanthus* [169].

The next step after pathway design is to rank these pathways and find enzymes that catalyze reactions in these pathways. Although pathway design has been studied intensively, only a few studies has focused on the pathway ranking [170, 171]. In this study we aim to establish a systematic pathway design and evaluation to integrate into the DBTL cycle. We will combine retrosynthesis and genome-scale models to design and enumerate pathways.

Genome-scale model is a comprehensive metabolic knowledgebase of a target organism. These models are linear, constraint-based models that enables simulating metabolism at steady-state [21]. They have been used to guide metabolic engineering and to serve as platform for contextualizing 'Omics' data [27, 28]. In the context of pathway ranking, GEMs are excellent tools to calculate maximum theoretical yields of native and non-native pathways [172].

The next step is to select enzymes for reactions in the production pathways. Selenzyme is a unique tool for this task as it allows to search for enzymes for reactions that are not in the database [173].This is feasible because Selenzyme bases on SMARTS reaction rule to mine for enzymes that can act on the same reaction centers.

Selecting suitable hosts for chemical production is also an essential element to facilitate the biosynthesis of chemicals. Over the traditional workhorses such as *Escherichia coli* and *Saccharomyces cerevisiae, Pseudomonas putida* emerges as a more suitable host for chemical production due to their outstanding solvent tolerant capacity while producing fewer or no by-product [174]. The biocatalysis with *P. putida* in organic systems make downstream product removal simpler cause hydrophobic compounds can be directly isolated [175–177]. *P. putida* is also certified as HV1 which means the bacteria is safe to work with [178]. Due to these advantages, we

employed *P. putida* as production chassis for cis,cis-muconic acid, anisole, aniline, geranic acid, and 3-methylmalate.

In this study we employed a retrosynthesis tool, RetroPath2.0 [179] to design pathways. RetroPath2.0 can design pathways, yet, identifying the best candidates are not covered in the algorithm. In this study, we establish a general ranking system based on pathway capabilities such as maximum theoretical yield and thermodynamic feasibility using genome-scale metabolic model to extract meaningful output from the numerous results obtained from RetroPath2.0.

3.2 Results

In this work, we designed biosynthesis pathways for five compounds, cis,cis-muconic acid, aniline, anisole, geranic acid, and 3-methylmalate. The workflow is summarised in Figure 3.2.1. In the first step, RetroPath2.0 was used to design possible routes of metabolic conversions from internal metabolites to products. Here GEM was used as a knowledge base to provide the known metabolic pool of the target organism, in our case *P. putida*. In the second step, pathways resulted from RetroPath2.0 was added to two existing high quality GEMS of *P. putida*, namely iJP962 and ijN1411, to select pathways that produce high theoretical yields. Selected routes will be converted in biochemical feasible pathways through enzyme selection in the next step. In the final step, selected pathways will be checked for thermodynamic feasibility using eQuilibrator. Note that the ranking method we proposed will discover already existing solutions. These are deliberately included to serve as a positive control.

3.2.1 RetroPath2.0 and the required input

The required input for RetroPath2.0 is International Chemical Identifiers (InChI) of internal metabolites in the target organism, a so-called sink metabolites. In this study, we obtained InChI structures for 792 out of 1390 unique metabolites in iJN1411 due to the presence of many organism specific compounds such as membrane lipids or fatty acids with hydrocarbon tails of specific carbon numbers that are not identified in public databases. Although we can only map 56.9% of metabolites in the *P. putida* model iJN1411, RetroPath2.0 was able to find pathways with the input we provided.

Using RetroPath2.0, pathway enumerator rp2path, an algorithm to enumerate RetroPath2.0 output into pathways and a ranking system based on pathway length, theoretical yield, the availability of enzymes and thermodynamic feasibility which is the max-min driving force of the whole pathway, we designed a total of 16 synthetic pathways for five compounds cis,cis-muconic acid, aniline, anisole, geranic acid, and 3-methylmalate in *P. putida*. These pathways are depicted in Figure 3.2.1 and an overview is provided in Table 3.2.2, that also includes model estimates of maximal theoretical yield on glucose. Selected candidate genes associated to the enzymes are provided in Table 3.2.3.

Table 3.2.1: The number of pathways predicted after each step in the designing and ranking workflow

The workflow	Tools	Number of pathway predicted after each step				
		Cis,cis-muconic acid	Aniline	Geranic acid	Anisole	3-methylmalate
Pathway design	RetroPath2.0	150	02	04	150	52
Theoretical yield	iJP962	54	01	02	0	0
	iJN1411	54	01	02	89	15
Known enzyme available ?	Promiscuity score Literature Uniprot Selenzyme	17	01	02	03	04
Thermodynamic feasible?	eQuilibrator	08	01	02	03	02
Final pathways		08	01	02	03	02

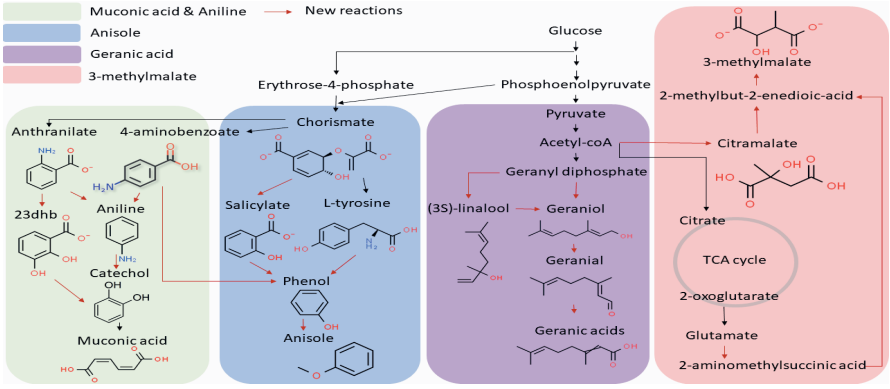


Figure 3.2.1: Predicted pathways for biosynthesis of target compounds in *P. putida* from glucose. For cis,cis- muconic acid, only novel pathways are included. 23dhb-2,3-dihydroxybenzoate

Table 3.2.2: *In-silico* design for production of cis,cis-muconic acids, aniline, geranic acid, anisole, and 3-methylmalate by *P.putida*. EC numbers are included to characterize the corresponding enzymes in the reaction steps. The first compound in each pathway is found in the metabolic pool of *P. putida*. Glucose is used as substrate in all pathways. ⇒ indicates reactions that are already in *P. putida*

Targets Pathways & EC numbers	Theoretical yield (mol P/mol glc)	References
Cis,cis-muconic acid		
protocatechual (pca) ⇒ 1.14.12.10 ⇒ catechol ⇒ target	0.81	[180], in *P. putida* [181]
4-hydroxybenzoate → 4.1.1.61 → phenol → 1.14.13.7 → catechol ⇒ target	0.7	[153]
chorismate → salicylate synthase → salicylate → 1.14.13.1 → catechol ⇒ target	0.65	in *E. coli* [151], patent [154]
2-aminobenzoate → 1.14.13.35 → 2,3-dihydroxybenzoate → 4.1.1.46 → catechol ⇒ target	0.64	This study
4 (or 2)-aminobenzote → 4.1.1.24 → aniline → 1.14.- → catechol ⇒ target	0.62	This study
chorismate → salicylate synthase → salicylate → 4.1.1.91 → phenol (+ 02)→ 1.14.13.7 → catechol ⇒ target	0.54	[153]
2-aminobenzoate → 1.14.12.1 → catechol ⇒ target	0.17	[150]
benzoate ⇒ 1.14.12.10 ⇒ catechol ⇒ target	0.14	[182], in *P. putida* [181]
Aniline		
4 (or 2)-aminobenzote → 4.1.1.24 → target	0.68	This study, patent [183]
Geranic acid		
Geranyl diphosphate → 4.2.3.25 → (3S) – linalool → 5.4.4.4 → geraniol → 1.1.1.347 → geranial → 1.2.1.86 → target	0.42	This study
Geranyl diphosphate → 3.1.7.3 → geraniol → 1.1.1.347 → geranial → 1.2.1.86 → target	0.42	This study
Anisole		
L_tyrosine → 4.1.99.2 → phenol → 2.1.1.25 → target	0.55	This study
4-hydroxybenzoate → 4.1.1.61 → phenol → 2.1.1.25 → target	0.55	This study
Chorismate → salicylate synthetase → salicylate → 4.1.1.91 → phenol → 2.1.1.25 → target	0.55	This study
3-methylmalate		
Pyr + acetyl-coA → 2.3.1.182 → Citramalic acid → 4.2.1.35 → 2- methylbut-2-enedioic acid → 4.2.1.35 → target	1.25	This study
L-glutamate → 5.4.99.1 → 2- aminomethylsuccinic acid → 4.3.1.2 → 2-methylbut-2- enedioic acid → 4.2.1.35 → target	1.19	This study

Table 3.2.3: Enzymes and selected candidate genes for the designed pathways

Enzymes	Genes	Organism	References
Cis,cis-muconic acid and Aniline			
1.14.12.10	*benC*	*Pseudomonas Putida* KT2440	[181]
4.1.1.61	*edcC*	*Escherichia coli* O157:H7	[184]
1.14.13.7	*ph*KLMNOP	*pseudomonas stutzeri* OX1	[153]
1.14.13.1	*catA*	*Pseudomonas putida* KT2440	[153]
1.14.13.35	–	*Aspergillus niger*	[185]
	–	*Trichosporon cutaneum*	[186]
4.1.1.46	GenBank: AM270011.1	*Aspergillus niger*	[187]
4.1.1.24	GenBank: EEQ92869.1	*Ochrobactrum intermedium* LMG 3301	NCBI
	GenBank: ACO02919.1	*Brucella melitensis* ATCC 23457	NCBI
	GenBank: EEH13339.1	*Brucella ceti* str. Cudo 7	NCBI
	GenBank: EEP61496.1	*Brucella abortus* str. 2308 A	NCBI
1.14.-	*tdnQ*, *tdnA1*, *tdnA2*, *tdnB*, and *tdnR*	*Pseudomonas putida* mt-2 (UCC22)	[188]
4.1.1.91	*Sdc*	*Trichosporon moniliiforme* WU-0401	[189]
Salicylate synthase	*mbtI*	*Mycobacterium tuberculosis* H37Rv	[190]
1.14.12.1	*paantABC*	*Pseudomonas aeruginosa*	[150]
Geranic acid			
4.2.3.25	*TPS14*	*Arabidopsis thaliana*	[191]
5.4.4.4	*Ldi*	*Castellaniella defragrans*	[192]
1.1.1.347	*geoA*	*Castellaniella defragrans*	[193]
1.2.1.86	*geoB*	*Castellaniella defragrans*	[193]
3.1.7.3	*GES*	*Ocimum basilicum*	[194]
Anisole			
4.1.99.2	*tpl*	*Erwinia herbicola*	[195]
2.1.1.25	–	Mammal	[196]
3-methylmalate			
2.3.1.182	*cimA*	*Sulfolobus acidocaldarius*	[197]
4.2.1.35	*leuC MJ0499* and I	*Methanocaldococcus jannaschii*	[198]
5.4.99.1	*glmE* and *glmS*	*Clostridium cochlearium*	[199]
4.3.1.2	*mal rrnAC0687*	*Haloarcula marismortui*	[200]

3.2.2 Muconic acid

RetroPath2.0 predicted a total of 150 pathways describing 72 different reactions and 62 different compounds that produce cis,cis-muconic acid from glucose. They were divided into 4 pathways of length 2, 14 of length 3, and 132 of length 4.

54 out of 150 pathways produce higher than 30 % of theoretical yields on glucose when adding to both genome-scale models of *P. putida* iJP962 and iJN1411. Of these 54 pathways, 17 have known enzymes that catalyze all reactions in the conversion route. 8 pathways are thermodynamic feasible in the next step. Using these three criteria, namely theoretical yields, known enzyme available and thermodynamic feasible, we selected eight pathways to produce cis,cis-muconic acid in *P. putida* (Tables 3.2.1 and 3.2.2). Among them, six pathways have been reported in literature and implemented in either *P. putida* or *E. coli*, while the remaining two pathways have not been reported before (Table 3.2.2).

All predicted pathways converge to same last step where catechol is converted to cis,cis-muconic acid. This has been reported in all natural and synthesis pathways for cis,cis-muconic acid production and is also patented in [201]. In *P. putida* this reaction is catalysed by the action of catechol 1,2-dioxygenase encoded by the gene *PP3713*.

In the previous step catechol is produced from either protocatechual, phenol, salicylate, 2,3-dihydroxybenzoate, aniline, 2-aminobenzoate, or benzoate (Table 3.2.2). These compounds either has already been produced in *P. putida* or can be produced in *P. putida* by introducing heterologous genes when growing on glucose (Table 3.2.2). Production of catechol from protocatechual was predicted from the model to give the highest cis,cis-muconic acid yield on glucose, 0.81 (mol product/ mol glucose). This is similar to findings in [181] for *P. putida*. This is the first biosynthesis pathway discovered to produce cis,cis-muconic acid [180]. Production of catechol from phenol was predicted to give the second highest yield with 0.70 (mol product/ mol glucose). Phenol has been reported as an effective substrate for catechol and later muconic acid in [153]. In this study, phenol is produced by the action of enzyme 4-hydroxybenzoate decarboxylase (4.1.1.61) encoding by *edcC* in *E. coli* (Ta-

ble 3.2.3). The other effective substrate for catechol is benzoate [202]. This pathway has been implemented in *E. coli* and *P. putida* [181]. Benzoate is supplemented as substrate in the medium. Cis,cis-muconic acid producing from chorismate gives the third highest yield. This pathway has been patented and implemented in *E. coli* with 16.3% *in-vitro* yield [151]. In *P. putida* chorismate is produced from pyruvate and 4-hydroxybenzoate by the action of probable chorismate pyruvate-lyase encoded by *PP5317*. The other two pathways via aniline and 2,3-dihydroxybenzoate are novel and have not reported in literature.

[**new pathway 1**] **4-aminobenzoate -> aniline -> catechol:**

In this pathway 4-aminobenzoate is catalyzed by aminobenzoate decarboxylase EC 4.1.1.24 to produce aniline. Aniline is then degraded to catechol. The degradation of aniline to catechol is the first step in the aromatic degradation pathway in many bacteria [203]. *Pseudomonas putida* KT2440 does not have genes in this pathway but *Pseudomonas putida* UCC22 [188] and *Pseudomonas sp.* AW-2 [204] are known to possess genes to degrade aniline via the *meta*-cleavage pathway. This multistep reaction is catalyzed by three enzymes, glutamine synthetase (GS)-like enzyme, glutamine amidotransferase like enzyme, and an aniline dioxygenase [203]. Five genes in a plasmid: tdnQ, tdnA1, tdnA2, tdnB,and tdnR have been shown to be essential for the conversion of aniline to catechol in *P. putida* [188].

Production of cis,cis-muconic acid from the degradation of aromatic compounds such as benzoate, phenol, salicylate, and benzene has been widely discussed in [205, 151, 153]. However, degradation of aniline to produce cis,cis-muconic acid has not been reported.

[**new pathway 2**] **2-aminobenzoate -> 2,3-dihydroxybenzoate -> catechol:** The last step in this pathway has been mentioned in [152] in which 2,3-dihydroxybenzoate is converted to catechol by the action of 2,3-dihydroxybenzoate

carboxy-lyase EC 4.1.1.46. 2,3-dihydroxybenzoate has been reported to be produced from isochorismate in *E. coli* [152]. However, the production of 2,3-dihydroxybenzoate from 2-aminobenzoate has not been used in the production pathway of cis,cis-muconic acid. This can be done by cloning an anthranilate 3-monooxygenase (deaminating) (1.14.13.35), an iron protein from *Aspergillus niger* [185] or a flavoprotein (FAD) from the yeast *Trichosporon cutaneum* [186]. However, these enzymes have no known protein sequences.

3.2.3 Aniline

RetroPath2.0 predicted two pathways to produce aniline (Table 3.2.1). Of these, only one is thermodynamic feasible with a yield of 0.66 (mol product/mol glucose). This pathway is the first step in the above described pathway for the production of cis,cis-muconic acid. Only one reaction, the conversion of either 4 or 2-aminobenzoate, to aniline is needed to produce aniline in *P. putida*. This reaction is catalyzed by the action of enzyme aminobenzoate decarboxylase EC 4.1.1.24. Many species in *Brucella* family have been validated to encode for this enzyme. They are GenBank: EEQ92869.1 from *Ochrobactrum intermedium LMG 3301*, GenBank: ACO02919.1 from *Brucella melitensis ATCC 23457*, GenBank: EEH13339.1 from *Brucella ceti str. Cudo*, and GenBank: EEP61496.1 from *Brucella abortus str2308 A*. This pathway has not been reported in scientific literature but has been patented in [183].

3.2.4 Geranic acid

Four pathways for the production of this compound were predicted by RetroPath2.0 (Table 3.2.1). After manual inspection of thermodynamic feasibility and enzyme availability, two of them, of lengths 3 and 4 respectively, were selected, both with

the same predicted yield of 0.42 (mol product/ mol glucose). These two pathways are similar in the last two steps where geraniol is converted to geranial by geraniol dehydrogenase EC 1.1.1.347 encoded by *geoA* from *Castellaniella defragrans* [193] (Table 3.2.2 and 3.2.3). Geranoil is then converted to geranic acid by geranial dehydrogenase EC 1.2.1.86 encoded by *geoB* from *Castellaniella defragrans* [193].

In the three-step pathway, geraniol is produced from geranyl diphosphate by the action of enzyme monoterpenyl-diphosphatase EC 3.1.7.3 encoded by *GES* in the plant *Ocimum basilicum* (Sweet basil) [194]. GES has been successfully expressed in *P. putida* to produce geranic acid [169].

In the four-step pathway, instead of going from geranyl diphosphate to geraniol, there is an intermediate step to (3S)-linalool. This is feasible due to two enzymes S-linalool synthase EC 4.2.3.25 which converts geranyl diphosphate to (3S)-linalool and geraniol isomerase EC 5.4.4.4 which converts (3S)-linalool to geraniol. S-linalool synthase is encoded by *TPS14* from *Arabidopsis thaliana* [191]. Geraniol isomerase is encoded by *Ldi* from *Castellaniella defragrans* [192].

3.2.5 Anisole

Of 150 pathways predicted from RetroPath2.0 for Anisole synthesis, 89 pathways generated non-zero yield when added to the genome-scale model of *P. putida* iJN1411. No pathway leads to the production of 3-methylmalate when adding to iJP962 because the sink metabolites were obtained from iJN1411. Thermodynamic feasibility and enzyme availability screening eliminated a majority of pathways. Only three pathways qualified for further inspection (Table 3.2.1). All of them give a similar yield of 0.55 mol product/ mol glucose. These predicted pathways share the last step where anisole is formed from phenol by the action of phenol O-methyltransferase EC 2.1.1.25. The enzyme has been studied since 1968

[196]. It has been known to be in mammals such as human [206], yet there is no known coding-sequence available for phenol O-methyltransferase. Using SeleEnzyme, we found another enzyme, eugenol O-methyltransferase EC 2.1.1.146, encoded by gene *EOMT1* from *Ocimum basilicum* [207] that can possess promiscuity on phenol. The enzyme transfers methyl group from S-adenosyl-L-methionine to a hydroxyl group on the benzene ring of isoeugenol to produce isomethyleugenol (Figure 3.2.2B). Our target enzyme, phenol O-methyltransferase EC 2.1.1.25, also transfers the methyl group from S-adenosyl-L-methionine to the hydroxyl group on the benzene ring of phenol (Figure 3.2.2A). Since the reaction centers and cofactors used by these two enzymes are similar, Selenzyme predicted that EC 2.1.1.146 can have promiscuity on phenol to produce anisole.

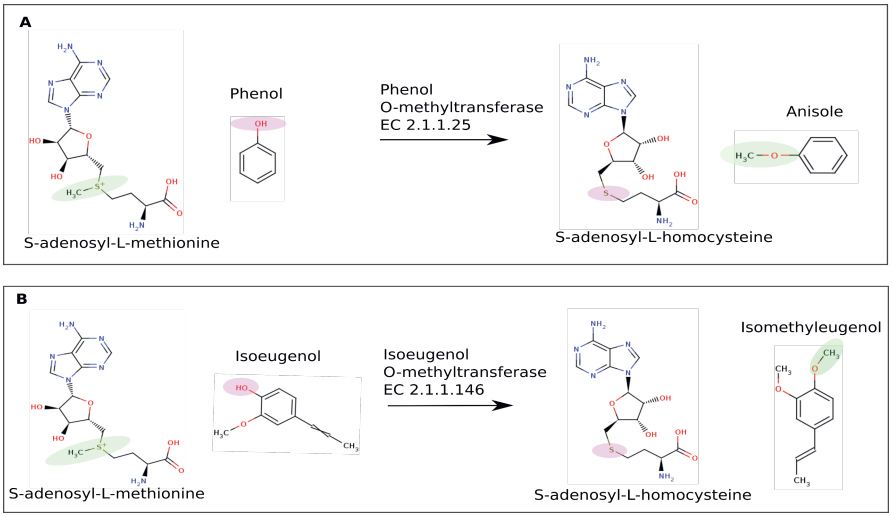


Figure 3.2.2: Phenol o-methyltransferase (A) and Isoeugenol o-methyltransferase (B) both transfer methyl group from S-adenosyl-L-methionine to the hydroxyl group on a benzene ring, hence we hypothesize that isoeugenol o-methyltransferase can also act on phenol to convert it to anisole.

These three pathways differ in the phenol generation step. In the first pathway, phenol is produced from L-tyrosine by the action of tyrosine phenol-lyase, EC 4.1.99.2. The enzyme is found in many organisms. In this study we selected one gene *tpl* from *Erwinia herbicola* because it had been successfully implemented in *E. coli* [195]. In the second pathway, phenol is produced from 4-hydroxybenzoate by 4-hydroxybenzoate decarboxylase, EC 4.1.1.61. The enzyme can be encoded in many organisms. In this study we chose gene from *E. coli edcC* [184]. In the last pathway, phenol is produced from salicylate by the action of salicylate decarboxylase, EC 4.1.1.91. Gene *sdc* from *Cutaneotrichosporon moniliiforme (Yeast) (Trichosporon moniliiforme)* [189] is known to encode salicylate decarboxylase. Salicylate is in turn produced from chorismate by salicylate synthetase. There is only one gene known to encode salicylate synthetase which is *mbtl* from *Mycobacterium tuberculosis* [190].

3.2.6 3-METHYLMALATE

Of the 52 pathways predicted by RetroPath2.0, 15 pathways produce 3-methylmalate in the model iJN1411 (Table 3.2.1). No pathway leads to the production of 3-methylmalate when adding to iJP962 because the sink metabolites were obtained from iJN1411. Among these 15 pathways, four of them have known enzymes and only two are thermodynamically feasible. In these two pathways, 3-methylmalate was produced from 2-methylbut-2-enedioic acid by the action of enzyme (R)-2-methylmalate dehydratase EC 4.2.1.35 (Table 3.2.2). The enzyme has two subunits which are encoded by *leuC* and *leuD* from *Methanocaldococcus jannaschii* (strains ATCC 43067 / DSM 2661/ JAL-1 / JCM 10045 / NBRC 100440) [198] (Table 3.2.3).

In the first pathway, the same enzyme, (R)-2-methylmalate dehydratase EC 4.2.1.35, also works on citramalate to produce 2-methylbut-2-enedioic acid.

Citramalate is produced from pyruvate and acetyl-coA by the action of (R)-citramalate synthase EC 2.3.1.182. This enzyme is encoded by cimA from *Sulfolobus acidocaldarius* [197]. In the second pathway, l-glutamate is converted to 2-aminomethylsuccinic acid by glutamate mutase EC 5.4.99.1, a 2-subunit enzyme found in *Clostridium tetanomorphum* encoded by *Glutamate mutase epsilon* and *sigma* subunit, *glmE* and *glmS* [199]. 2-aminomethylsuccinic acid is then catalyzed by 3-methylaspartase EC 4.3.1.2 to produce 2-methylbut-2-enedioic acid. The enzyme is known to synthesis from *mal* rrnAC0687 from *Haloarcula marismortui* (strain ATCC 43049 / DSM 3752 / JCM 8966 / VKM B-1809) or CHY_0484 or CHY_0582 from *Carboxydothermus hydrogenoformans* [200].

3.3 Discussion

In this work, we used a combination of tools based on retrosynthesis and genome-scale metabolic models to design biosynthesis pathways for five compounds, cis,cis-muconic acid, aniline, anisole, geranic acid, and 3-methylmalate. To design pathways we employed RetroPath2.0 which was developed explicitly for this purpose. It uses generalized reaction rules and to some extend enzyme promiscuity to look for different substrates for the same enzyme. Like this, new reactions and hence pathways can be predicted. RetroPath2.0 requires the International Chemical Identifiers (InChI) of all metabolites in the host organism as input. Although InChI is classified as an international standard identifier, generating them is labour intensive as it cannot be automatized. The inconsistent namespace of the metabolites in genome-scale models and the use of general names in these models makes it difficult to map with public databases that provide InChI structures. For instance, cis,cis-muconic acid, and muconate are two states of the same chemical. In microbes such as *P. putida*, cis,cis-muconic acid exists in its reduced form, muconate. This is also applied for most of

other acids, for instance, succinic acid, an important intermediate in TCA cycle is found in its anion form as succinate in the cell. Since these compounds and their reduced forms are one hydrogen different, their InChI structures are also slightly different. However, these two forms of metabolites are often considered as one in genome-scale models. It may imply mismatch and redox imbalance if many compounds are used in the incorrect forms. As we observed in this study, RetroPath2.0 suggested many reactions just to produce hydrogen and water. Introducing these pathways into GEMs will generate extra energy and lead to high unrealistic theoretical yields. Water and hydrogen are basic components that already in *P. putida* if not all microbes. These reactions need to be removed from RetroPath2.0's result.

The use of inconsistent and ambiguous namespaces for metabolites is a well-known source of complaints. We demonstrated in our paper [208] how ambiguity and inconsistency can hamper the use of GEMs. To reduce the risk of mismatch when mapping between namespaces, we also suggested as a good practice, genome-scale models should include InChI for metabolites to avoid ambiguity. This is not yet common for GEMs. We provided in this study, the updated model of *P. putida* iJN1411-InChI with InChI for metabolites.

While RetroPath2.0 allows the exploration in a bigger search space, it is certainly a challenge to extract meaningful candidates from the numerous output it produces. Pathway selection is one important challenge that needs to be addressed in metabolic engineering. Many general criteria have been proposed such as pathway length, the number of interventions, thermodynamic, and theoretical yield [209, 171]. To shorten the tedious innovation process, we need a quick system to eliminate infeasible pathways, yet ensure no promising pathways are discarded during the screening step. In this study, we prioritize pathways that: (i) have high theoretical yield for economic values, (ii) involve known enzymes to limit the risk of synthesizing and testing

new enzymes and (iii) are thermodynamic feasible. With this system, we identified all known and patented pathways for cis,cis-muconic acid from RetroPath2.0 output. This shown that the criteria and ranking system we proposed although simple, it is effective, sufficient and reliable.

In addition to these criteria, toxicity of intermediate metabolites is also important for pathway design since they can hamper cell survival and growth [209]. The toxicity of metabolites can be obtained from experimental data or can be estimated based on structure-activity relationship models [210, 209, 211]. In this study we did not include the toxicity since *P. putida* has shown to have high tolerance towards toxic compounds [212, 213, 169, 214]. Its tolerance has been tested toward various compounds, for instance aromatic compounds such as phenol [215] and p-hydroxybenzoate [216], o-cresol [217], monoterpenoid such as geranic acids [169], or aniline and catechol [218, 219].

Cis,cis-muconic acid has been studied extensively, yet, we found two new pathways to produce it with competitive maximum theoretical yield when comparing to the most effective pathways that have been reported. Our ability to discover new designs is certainly not at its limit, using the traditional expert-based approach will constraint us from expanding our knowledge, especially when designing pathways for obscure compounds.

To demonstrate the simple yet effective approach to design pathway for less-studied compounds, we employed RetroPath2.0 for anisole, aniline, 3-methylmalate, and geranic acid. Little to no effort has been spent on developing pathways for these compounds, to the best of our knowledge, except for geranic acid and aniline, we are the first to propose pathways to synthesize them in microbes.

All the pathways we found for the five target compounds require a few metabolic conversions from central carbon metabolites. They all shared common precursors

such as phosphoneolpyruvate and chorismate and many interchangeable precursors, for example, 4-aminobenzoate can lead to both aniline, cis,cis-muconic acid, and anisole. While geranic acid and 3-methylmalate both have the same important precursors such as acetyl-coA. Although there are many more different networks in metabolism, we can already employ central carbon metabolism to produce many compounds of interest. This allows the use of general precursors over-producers to increase the yield of diverse targets.

Many enzymes in production pathways predicted for anisole, 3-methylmalate, geranic acid are from plants. This is expected because these compounds are plant terpenoids. They are produced in plants in small quantities. These enzymes are known in literature but no pathway was reported for the production of these compounds in microbes. The pathways we predicted used plant enzymes, transferring them to microbes can imply high risk with the post-translational modification that microbes do not possess. Nevertheless, adapting plant pathways in microbes is not new, and has been done before with great success [220–222].

In this work, our aim was to speed up the pathway design and selection phase to save time and efforts for the downstream optimization steps. We provided a showcase of how automatic pathway designing tool such as RetroPath2.0, a general ranking system with the help of genome-scale models can be effectively used in the context of design-test-build-learn cycles. Using these tools, pathway design can be done in a quick and more standardized fashion.

3.4 Methods

3.4.1 *In-silico* pathway prediction

In this study we used RetroPath2.0 [179] to design pathways for cis,cis-muconic acid, aniline, anisole, geranic acid and 3-methylmalate. RetroPath2.0 is an automated open-source workflow implemented in KNIME [223]. The tool allows users to design biosynthetic routes to connect metabolites in a known chassis to the desired target using retrosynthesis and enzyme promiscuity rules. In metabolic engineering, in a forward manner metabolites in the chassis strain is the source, the target compound is a so-called sink. This is reversible in a reverse manner. In RetroPath2.0, source is the target chemical one wishes to produce and sink is the metabolites in a known chassis host.

RetroRules was obtained from https://retrorules.org/dl on Sep 2019. The input configuration is set up for a maximum pathway length of 4 steps and diameter at 16 bonds around the reaction center. rp2path were then used to enumerate pathways from RetroPath2.0 output. This list was used in the next step - pathway ranking.

3.4.2 Models and InChI

To design and rank pathways, the two most comprehensive models for *P. putida*, iJP962 [224] and iJN1411 [225] were used. RetroPath2.0 required the International Chemical Identifier code (InChI) [226] as input. To generate the InChI list, BiGG [227] identifiers of unique metabolites in iJN1411 were translated to KEGG [130] by mapping with MetaNetX/MNXref [228]. KEGG identifiers were in turn mapped to wikidata [229] to extract InChI structure. The metabolites left unmapped after the first step, were mapped using the Chemical Translation Service (http://cts.fiehnlab.ucdavis.edu/batch), a web-based tool that allows the con-

version between chemical identifiers [230].

3.4.3 Pathway ranking

Pathways predicted by RetroPath2.0 were added to the genome-scale model of *P. putida*, iJP962 and iJN1411 to calculate the maximum theoretical yield. Flux balance analysis was used to optimize product formation from glucose while constraint biomass to 10 % of the maximum growth rate. Pathways that give non-zero production rate in either model were further checked for available enzyme coding sequences in the second step. This probability to retrieve enzyme sequences for a reaction were computed in RetroPath2.0 as 'scores' [179]. High scores implied a high penalty for enzyme sequence availability. Reactions with high scores were therefore removed. The remaining pathways were checked for their thermodynamic feasibility using eQuilibrator [231] at pH 7.0 and the default concentration range of $1\mu M - 10mM$. eQuilibrator received pathways in an SBtab file as input and calculated the feasibility of the pathway by Max-min Driving Force (MDF) [232]. Pathways with negative MDF were not feasible and discarded. If no candidate pathway was found after this step, Selenzyme [173] was used to find promiscuous alternatives that act on the same reaction centers for pathways with unknown enzymes in the second step. The thermodynamic feasibility of new candidates was then computed. The final list was manually inspected to identify gene sequences. Enzyme-coding-gene sequences were retrieved from UniProt [233]. We selected gene sequences with a high score on UniProt which means they are manually curated with experimental functional tests.

4

Consistency, inconsistency, and ambiguity of metabolite names in biochemical databases used for genome-scale metabolic modelling

Pham Nhung, Ruben GA van Heck, Jesse CJ van Dam, Peter J. Schaap, Edoardo Saccenti, and Maria Suarez-Diez

Published in "**Pham, Nhung**, Ruben GA van Heck, Jesse CJ van Dam, Peter J. Schaap, Edoardo Saccenti, and Maria Suarez-Diez. "Consistency, inconsistency, and ambiguity of metabolite names in biochemical databases used for genome-scale metabolic modelling." Metabolites 9, no. 2 (2019): 28. "

ABSTRACT

Genome scale metabolic models (GEMs) are manually curated repositories describing the metabolic capabilities of an organism. GEMs have been successfully used in different research areas, ranging from systems medicine to biotechnology. However, the different naming conventions (namespaces) of databases used to build GEMs limit model reusability and prevent the integration of existing models. This problem is known in the GEM community but its extent has not been analyzed in depth. In this study, we investigate the name ambiguity and the multiplicity of non-systematic identifiers and we highlight the (in)consistency in their use in eleven biochemical databases of biochemical reactions and the problems that arise when mapping between different namespaces and databases. We found that such inconsistencies can be as high as 83.1%, thus emphasizing the need for strategies to deal with these issues. Currently, manual verification of the mappings appears to be the only solution to remove inconsistencies when combining models. Finally, we discuss several possible approaches to facilitate (future) unambiguous mapping.

4.1 Introduction

Genome scale metabolic models (GEMs) combine available metabolic knowledge of an organism in a consistent and structured way that allows prediction and simulation of metabolic phenotypes [234]. GEMs have been successfully used in different research areas, ranging from biotechnology to systems medicine, often resulting in new insights on metabolic processes in living organisms [235–238]. GEMs may differ in content and scope, and can contain anything from a few hundred to a few thousand reactions and metabolites. However, the structure of the model remains similar regardless of the application: the main components are metabolites, metabolic reactions, enzymes and the corresponding encoding genes.

The construction of a GEM includes three main steps [21, 239]. First, the genome of the organism considered is functionally annotated in order to identify enzymes and the associated reactions and metabolites. Second, the list of enzymes and reactions is converted into a mathematical model, a so-called draft model, in the form of a stoichiometric matrix to which constraints are added to account for reaction reversibility and uptake and secretion of metabolites. Last, the model is manually curated using experimental data (such as growth data), information from literature and/ or expert knowledge. Manual curation involves human workload and entails the verification of each reaction in the model and its corresponding constraints, which is a very time-consuming task. Tools and pipelines (such as, for example, the SEED [240], Pathway Tools [241], and the Raven toolbox [242]) have been developed to automatize the annotation, draft the reconstruction and to aid high-throughput creation of genome scale draft models [243].

The tools for automated draft reconstruction rely on biochemical databases that are used to find reactions associated to the enzymes identified in the genome through annotation. In general, different tools use different databases. For instance: the

SEED uses its own naming system [240], Pathway Tools [241] uses MetaCyc [244], and Raven [242] uses KEGG [245]. Every database uses its own namespace which is a particular set of identifiers (such as numerical tags or names) for metabolites and reactions: because of this, it can happen that the same metabolites and reactions have different naming conventions when different tools are used to generate draft GEMs. To complicate the matter further, researchers often tend to use their own naming conventions such as custom abbreviations for metabolites or consecutive numbering for metabolites and reactions and this adds up to the observed heterogeneity of names and identifiers found in GEMs available in the literature [246]: the use of unique identifiers, independent from the particular databases used, such as InchI [247, 226], or references to interlink different namespaces, have been suggested as an essential and fundamental part of GEM [248] but this is seldom implemented.

GEMs are manually curated knowledge repositories integrating information from independent (organism-specific) sources and thereby provide a comprehensive representation of what is presently known about the metabolism of the modelled organism. There is often the need to combine the information stored in individual GEMs to arrive to a consensus metabolic model for a given organism [249, 250]. The use of different namespaces limits the reusability of a GEM and often makes it impossible, or extremely laborious, to combine two GEMs. Further, it often hampers model expansion, which is the addition of new reactions and/or metabolites to an existing model because if different namespaces are used the same metabolite can be added many times with different names and, as a consequence, considered as different chemical entities which can, in the worst case, invalidate the model. In principle, different GEMs can be combined into a community model (partially) representing the different organisms present in a microbial community, with the aim of modelling community metabolic interactions such as cross feeding or substrate

competition [251].

Since mapping manually different namespaces is highly laborious and practically unfeasible for large models [249], the only viable solution to integrate different GEMs has often been to rebuild *de novo* the required models [252, 253]. However, while this approach leads to models that can be easily combined, it causes the loss of all the expert knowledge introduced in the manual curation process.

Naive direct comparison of names using string algorithms is often insufficient [254] and to help mapping among different namespaces in a more systematic way tools for consensus model generation and for automatic translation have been introduced [255, 250], together with databases like MNXRef from MetaNetX [256] and MetRxn [257], developed to provide cross-linking among the identifiers in the namespaces of different databases.

As a matter of fact, mapping different namespaces using metabolite or reaction identifiers is not a trivial task because researchers often refer to compounds with many different names and abbreviations and the namespaces reflect this (Figure 1A). Often in GEMs different chemical entities (like, for instance, citrate and citric acid) are used as exchangeable names and may end up in databases like BIGG (which harvest reactions which have been used in metabolic modelling) resulting in imprecise, misleading and sometimes incorrect synonyms. Similarly, GEMs are often built featuring reactions using generic compound classes (such as 'Lipids' or 'Protein'). When these are included in GEMs databases they cause the same compound to be linked to different identifiers.

Internal database inconsistency is also often caused by ambiguous abbreviations, with the same shorthand used for different compound (Figure 1B). To make the matter worse, the same abbreviation can refer to different compounds in different databases (see Figure 1C).

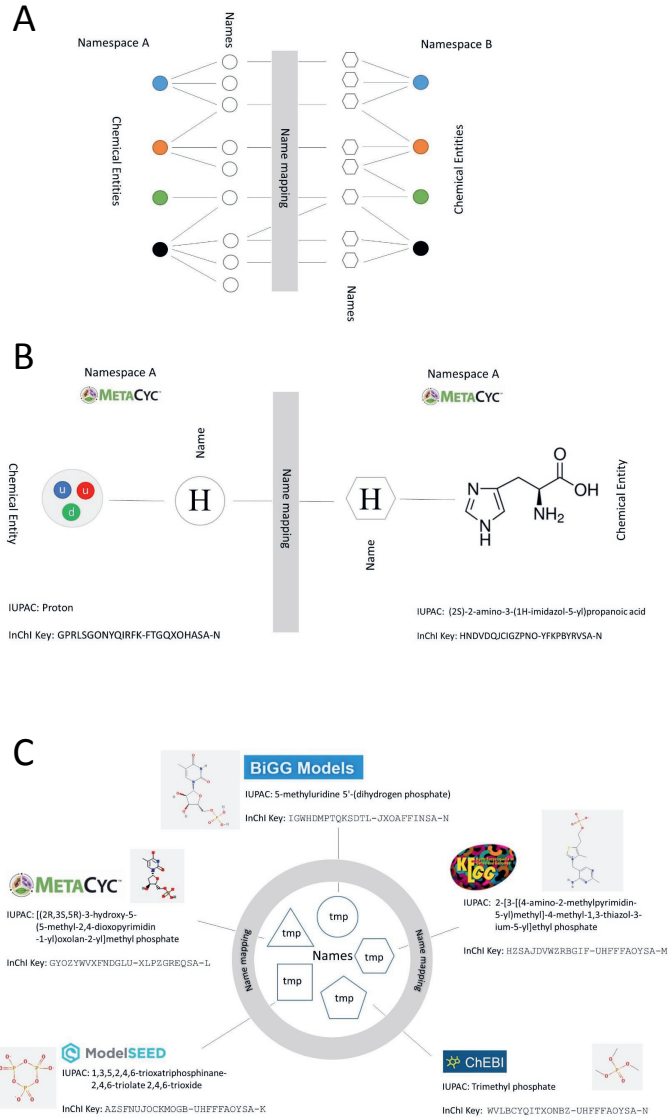


Figure 4.1.1: Overview of namespace mapping problems. (**A**) The same chemical entities (colored nodes) link to different names (colorless nodes) in different namespaces: names in namespace A may link to different chemical entities in namespace B; (**B**) Example of inconsistency within the same namespace: the same name links to different chemical compounds; (**C**) Example of inconsistency between different namespaces: the same name links to different compounds in different databases. Chemical entities are represented with colored nodes, names are represented with colorless nodes

The problems deriving from the inconsistency and the ambiguity in the namespaces of reaction databases used to build GEMs have been mentioned before [258–261] and are a well-known source of complaints in the modelling community. However, since the extent of the namespace mapping problem has not been so far analyzed in depth, we investigate the level of inconsistency and ambiguity encountered when *i*) mapping metabolites within a database and *ii*) mapping metabolites between two databases. To this task, we analyzed and compared naming and identifier conventions in eleven biochemical databases commonly used for metabolic modelling and metabolomics data analysis. Similar research has been done for small molecule databases that have been used in pharmaceutical research but did not consider databases used for metabolic modelling [262]. With this work we aim at raising awareness on this problem within the modelling community; provide a framework for evaluating when (or whether) GEMs and databases can be combined, suggest practices for dealing with this issue on the short term and outline a strategy for a long term solution.

4.2 Results

To avoid ambiguity, we explicitly define the specific terms used in this study as follows:

- *Identifier (ID)*: Identifiers are strings of alpha-numeric characters used to identify uniquely a metabolite or a reaction in a database. Examples are `C00001` in KEGG or `ATPM` in BIGG.
- *Name*: Here we use name to refer not only to the chemical name, but also to the set of aliases, synonyms and abbreviations that are often included in

a database as other names of the compound. For instance the KEGG ID C00001 is associated to the name 'water'.

- *Multiplicity*: describes the case on which a single ID is linked to multiple names. For instance, The KEGG ID C00001 is associated to the names 'water' and 'H2O'; therefore we state that this ID has a multiplicity of 2.

- *Ambiguous.* The Merriam-Webster dictionary defines ambiguous (second entry) as 'capable of being understood in two or more possible senses or ways'. Here, we use ambiguous (and its derivatives) to refer to the case on which the same name links to more than one ID in the same database. An example is shown in Figure 4.1.1 B, where the name 'H' links to the MetaCyc IDs 'PROTON' and 'HIS', associated to 'hydrogen ion' and 'L-histidine', respectively.

- *Consistency*: We use consistency (and its derivatives like consistent) to refer to mappings on which a molecular entity is mapped to itself. It follows that inconsistency is used to indicate a mapping or a database on which a molecular entity is associated to a different one.

We have analyzed eleven biochemical databases for their consistency and we have performed pairwise comparisons to investigate the degree of inter-database consistency. These databases were chosen for this study, primarily, because they were integrated in MetaNetX which facilitates data retrieval. Many of them (BiGG, KEGG, SEED, HMDB, ChEBI and MetaCyc) are commonly often used for metabolic model reconstruction [246, 263]; HMDB is the reference database for metabolomics studies.

4.2.1 Mappings within the same database

Name ambiguity

We calculated the average number of IDs per compound name for each of the eleven databases: results are summarized in Table 4.2.1.

Table 4.2.1: Ambiguity in biochemical database: number of compound names associated to more than one identifier (ID.) s.d. stands for standard deviation. Blue boxes are used to highlight highest numbers.

Database	#Name	Average number of IDs per name ± s.d.	% Ambiguous names	# Ambiguous names	Highest number of IDs per name
BiGG	5102	1.0141 ± 0.126	1.31	67	3
ChEBI	388505	1.3846 ± 1.52	14.8	57497	413
enviPath	11648	1.0804 ± 0.325	7.38	860	10
HMDB	101101	1.0377 ± 3.865	1.67	1686	921
KEGG	59682	1.1461 ± 0.422	13.3	7936	16
LIPID MAPS	77457	1.0113 ± 0.33	0.62	478	63
MetaCyc	55823	1.0058 ± 0.103	0.5	279	13
Reactome	6972	1.7902 ± 2.458	29.43	2052	34
SABIO-RK	11475	1.0008 ± 0.031	0.07	8	3
SEED	47410	1.0108 ± 0.106	1.06	503	4
SLM	1218750	1.0782 ± 0.321	6.72	81894	9

With ChEBI and Reactome as exceptions, in most databases the average ID number is around 1: however there is a low consistency. Reactome has the lowest consistency: nearly 30% of compounds are associated with more than one ID, metabolites with generic descriptive names like 'secretory granule lumen proteins', 'secretory granule membrane proteins', and 'ficolin-rich granule lumen proteins' associate to 34 different IDs; there are also more specific names, like 'hydron', 'water' and 'ATP' associated to 21, 14 and 11 IDs, respectively. In the latter cases the cause is that different IDs are used to indicate the same metabolite in different subcellular compart-

ments, although they all get assigned to the same name, for example the ID 5278291 indicates water in the cytoplasm while water in extracellular compartment is identified as 109276.

Overall, the most ambiguous metabolite name is 'lecithin', which is associated to 921 different IDs in the Human Metabolome database (HMDB). In this database, the most ambiguous names are general compound classes such as 'diacylglycerol', 'PPP' and 'pyridin-3-ylboronic acid'.

The overall consistency of HMDB is very high, as only 1.7% of names are linked to multiple IDs, followed by ChEBI and KEGG, where 14.8% and 13.3% of names map to multiple IDs; also in ChEBI 'lecithin' is the most ambiguous compound, linked to 413 IDs; other ambiguous names are, again, generic names such as 'Diglyceride', 'Diacylglycerol', 'Triglyceride' and 'Triacylglycerol' (see Figure 4.2.1A). Also in KEGG the most ambiguous names refer to generic compounds like 'DS-18' with 16 corresponding IDs. Furthermore this compound shares ID with 'Chondroitin 4-sulfate' which is a sulfated glycosaminoglycan while DS-18 generally refers to glycan, which further complicates metabolite characterization, as shown in Figure 4.2.1 B.

EnviPath and SLM databases have also relatively low consistency with 7% names being ambiguous. SLM is the largest database considered ($> 1.2 \times 10^6$ entries) and the most ambiguous name refers to 'Triacylglycerol'. In enviPath the most ambiguous compound is 'compound 0044249', with SMILES representation CC1=CC=C(C=C1O)O that corresponds to 4-methyl-1,3-benzenediol. In this database, many metabolites are renamed with numbers, *i.e.* 'P06','M320I23', or 'compound 869', which makes it cumbersome to the human user to identify them.

Other databases, namely, SABIO-RK, MetaCyc and LIPID MAPS are highly consistent, with SABIO-RK containing only 8 metabolites with ambiguous names.

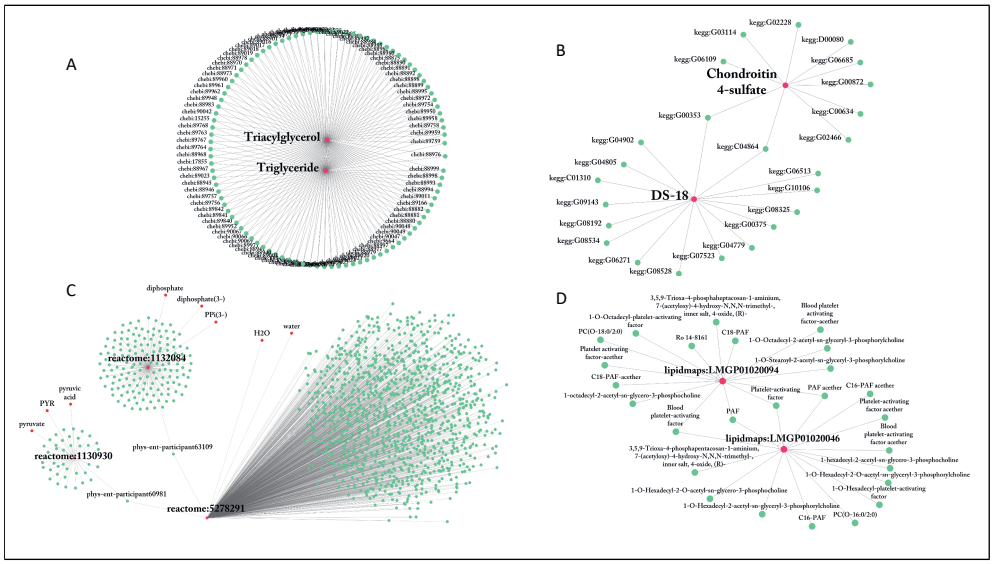


Figure 4.2.1: *Intra* database consistency. Edges indicate a link between a metabolite name and a database ID. Database name has been added to the ID (denoted as database names followed by ':', *i.e.* kegg:C00228). (**A**) Examples of metabolite names associated with multiple IDs in ChEBI. (**A**) Examples of metabolite names associated with multiple IDs in KEGG . (**C**) Examples of metabolite IDs associated with multiple names in Reactome. (**D**) Examples of metabolite IDs associated with multiple names in LIPID MAPS.

ID multiplicity and use of synonyms

In an effort to increase readability of entries in the database, often multiple names are linked to the same ID, *i.e.* IDs have a multiplicity larger than 1. Note that multiplicity is different from ambiguity as defined at the start of the Results section. Multiplicity increases human readability and is beneficial, as long as the alias, names and synonyms describe the same metabolite. Table 4.2.2 presents the average ID multiplicity for the eleven databases considered.

Table 4.2.2: ID multiplicity in each database: number of IDs in each database, average number of names per ID (average multiplicity), percentage and number of IDs that associate to more than one name, and highest number of names an ID links to; s.d. stands for standard deviation. Blue boxes are used to highlight highest numbers.

Database	#ID	Average multiplicity ± s.d.	% of IDs with multiplicity >1	# of IDs with multiplicity >1	Highest multiplicity in database
BiGG	5174	1.0 ± 0.0	0.0	0	1
ChEBI	123835	4.344 ± 3.588	97.74	121034	57
enviPath	12306	1.0226 ± 0.229	1.6	197	10
HMDB	43179	2.4297 ± 0.512	99.71	43052	8
KEGG	40256	1.6991 ± 1.231	38.93	15671	31
LIPID MAPS	40772	3.9213 ± 0.962	100.0	40772	23
MetaCyc	17159	3.2722 ± 1.984	99.75	17116	98
Reactome	5344	2.3355 ± 16.65	47.46	2536	1106
SABIO-RK	7683	1.4947 ± 1.193	24.17	1857	21
SEED	27693	1.7305 ± 1.311	39.83	11031	28
SLM	505004	2.602 ± 0.611	99.87	504333	9

BiGG is the only exception to this rule. Every metabolite identifier is associated to one and only one metabolite name, but, as shown in Table 4.2.1, the contrary does not hold true. BiGG is the smallest database here considered (with only 5102 metabolite names and 5174 metabolite IDs), although it should be stressed that this database has been built by integrating reactions and metabolites appearing in several published and manually curated genome-scale metabolic networks.

All other databases have some extent of multiplicity: in ChEBI, HMDB, Metacyc, SLM and LIPID MAPS nearly 100% of IDs are linked to more than 1 name. The use of multiple names is intended to increase usability of the database. However, inconsistencies might arise when ambiguous names are linked to IDs with high multiplicity, as illustrated in Figure 4.2.1 C and D. This can result in errors and mismatches when identifying compounds. A most extreme case is Reactome identifier `reactome:5278291` which is linked to 1106 difference names (see Fig-

Table 4.2.3: Example of compound names and IDs with high ambiguity and multiplicity.

Metabolite name	# Associate IDs	Metabolite ID	# Associated names
lecithin	922	reactome:5278291	1106
diacylglycerol	812	reactome:1131511	266
Lecithin	417	reactome:1236709	266
Diglyceride	317	reactome:1132345	180
Diacylglycerol	317	reactome:1132084	155
Triacylglycerol	106	reactome:1132304	140
Triglyceride	103	reactome:5278409	123
PPP	66	reactome:5278317	107
Cer[NS]	63	metacyc:PARATHION	98

ure 4.2.1C), among them 'H2O', 'water', 'phys-ent-participant60981' and 'phys-ent-participant63109'. The latter two names are linked to identifiers pointing to 'diphosphate' and 'pyruvate', which means that within this database is possible to map 'water' to 'pyruvate'. Other striking examples can be found in Table 4.2.3. When mapping with these compounds extra care needs to be taken.

Database mapping to IDs from MNXRef

MNXRef is a common namespace derived from MetaNetX and has been developed to combine namespaces from multiple databases and provides links between compounds (and identifiers) from different databases, the overarching goal is to enable bringing together GEMs.

We found that, each of the IDs in the 11 databases link to a MNXRef ID, however, as shown in Table 4.2.4, one MNXRef ID can connect to several IDs within a database resulting in a multiplicity larger than 1. This happens, for instance, when

MetaNetX associates one ID to several compound synonyms. This might be due to conscious modelling-specific decisions. For instance, it would make sense to combine citrate/citric acid identifiers in different databases to deal with protonation state differences. Thus linking several IDs to the same MNXRef ID addresses the multiplicity present in the database. However, this also generates errors if the ID links to ambiguous names. The most striking case is observed when mapping Reactome and MetaNetX: 2058 MetaNetX IDs are associated to Reactome IDs and 41.93% of them link to more than one Reactome ID.

Table 4.2.4: Number of IDs (#ID) in each database, number of MNXRef IDs (#MNXRef ID) linking to each database, multiplicity of MNXRef IDs when mapping to IDs in the corresponding database, and average and highest number of MNXRef ID per database ID; s.d. stands for standard deviation. Blue boxes are used to highlight highest numbers, while red boxes are for lowest numbers.

Database	#ID	#MNXRef ID	average #ID per MNXRef ID ± s.d.	% of IDs with multiplicity >1	# of IDs with multiplicity >1	Highest ID multiplicity >1
BiGG	5174	5062	1.0221 ± 0.165	1.96	99	4
ChEBI	123835	96746	1.28 ± 1.005	11.93	11541	30
enviPath	12306	11087	1.1099 ± 0.44	8.14	902	9
HMDB	43179	42354	1.0195 ± 0.176	1.63	691	12
KEGG	40256	37722	1.0672 ± 0.293	6.14	2316	12
LIPID MAPS	40772	40546	1.0056 ± 0.083	0.51	207	6
MetaCyc	17159	16985	1.0102 ± 0.115	0.9	153	5
Reactome	5344	2058	2.5967 ± 3.895	41.93	863	34
SABIO-RK	7683	7512	1.0228 ± 0.154	2.2	165	3
SEED	27693	26894	1.0297 ± 0.181	2.79	749	4
SLM	505004	504881	1.0002 ± 0.016	0.02	119	3

4.2.2 Namespace mapping between databases

To study namespace consistency between databases, we performed a pairwise mapping of the 11 databases. We performed the mapping using both the names in the corresponding database and MNXRef identifiers.

Mapping between databases using metabolite names

Table 4.2.5 shows the results of pairwise database mapping using metabolite names. Here, we map IDs in the databases using associated names. The databases have different metabolite coverages, for instance SLM contains 1218750 names while BiGG only 5102, this is because some are specific for a certain class of compounds (like SLM for lipids) while others aim to be comprehensive and do not describe all compound classes in exhaustive details (like HMDB for lipids). The difference in coverage and multiplicity of names associated to IDs (previously presented in Table 4.2.1 and Table 4.2.2) can cause the mapping between two databases not to be symmetric as evident from Table 4.2.5 .

In all comparisons, the fraction of compounds sharing the same name is rather limited. Overall, except for mapping from SEED to KEGG and ChEBI with 60.1 % and 57.2% overlap, respectively, all databases have less than 50% of compound names in common. The namespace of ChEBI has the largest overlap with other namespaces: around 40% towards MetaCyc, Reactome, and KEGG can be mapped to ChEBI. The namespaces of SLM, enviPath, and LIPID MAPS have the smallest overlap with other namespaces, which is most likely because these are very specific databases. The low ratios in Table 4.2.5, indicate that mapping using string algorithms is not effective since trivial differences in the names (such as the use of underscore and hyphen) can results in mismatches.

Ambiguous naming, *i.e.* one name associated to more than one ID, can result in mapping inconsistencies where one ID in the first database, gets mapped to multiple IDs in the second database. The fraction of non-univocal mappings is indicated in Table 4.2.6. Hence, although 40.2% of the Reactome IDs can be mapped to ChEBI (see Table 4.2.5), 81.3% of the successfully mapped Reactome IDs are ambiguously

Table 4.2.5: Number of IDs in one database (column) that map to IDs in the database in the corresponding row using database names as a bridge for mapping. Percentages indicate fraction of the initial database. Blue boxes indicate highest overall mapping. Red boxes are used to highlight the lowest numbers

Database	BiGG	ChEBI	enviPath	HMDB	KEGG	LIPID MAPS	MetaCyc	Reactome	SABIO-RK	SEED	SLM
BiGG	–	5097 (4.1%)	150 (1.2%)	702 (1.6%)	1489 (3.7%)	158 (0.4%)	210 (1.2%)	361 (6.8%)	839 (10.9%)	1829 (6.6%)	61 (0.0%)
ChEBI	1303 (25.2%)	–	816 (6.6%)	9178 (21.3%)	16013 (39.8%)	4662 (11.4%)	7209 (42.0%)	2146 (40.2%)	2552 (33.2%)	15837 (57.2%)	4336 (0.9%)
enviPath	142 (2.7%)	2284 (1.8%)	–	304 (0.7%)	1111 (2.8%)	55 (0.1%)	31 (0.2%)	90 (1.7%)	300 (3.9%)	983 (3.5%)	6 (0.0%)
HMDB	643 (12.4%)	15749 (12.7%)	310 (2.5%)	–	4745 (11.8%)	4078 (10.0%)	1693 (9.9%)	877 (16.4%)	1268 (16.5%)	3868 (14.0%)	14007 (2.8%)
KEGG	1286 (24.9%)	30098 (24.3%)	1050 (8.5%)	3922 (9.1%)	–	1725 (4.2%)	731 (4.3%)	928 (17.4%)	2604 (33.9%)	16646 (60.1%)	84 (0.0%)
LIPID MAPS	149 (2.9%)	7832 (6.3%)	54 (0.4%)	4200 (9.7%)	1862 (4.6%)	–	622 (3.6%)	311 (5.8%)	377 (4.9%)	1893 (6.8%)	13478 (2.7%)
MetaCyc	212 (4.1%)	20183 (16.3%)	31 (0.3%)	1967 (4.6%)	851 (2.1%)	648 (1.6%)	–	1266 (23.7%)	340 (4.4%)	7703 (27.8%)	326 (0.1%)
Reactome	156 (3.0%)	5833 (4.7%)	41 (0.3%)	620 (1.4%)	588 (1.5%)	254 (0.6%)	717 (4.2%)	–	368 (4.8%)	542 (2.0%)	146 (0.0%)
SABIO-RK	864 (16.7%)	10413 (8.4%)	324 (2.6%)	1456 (3.4%)	3127 (7.8%)	390 (1.0%)	342 (2.0%)	781 (14.6%)	–	2692 (9.7%)	55 (0.0%)
SEED	1824 (35.3%)	32212 (26.0%)	1020 (8.3%)	4971 (11.5%)	18489 (45.9%)	1915 (4.7%)	7580 (44.2%)	985 (18.4%)	2641 (34.4%)	–	233 (0.0%)
SLM	55 (1.1%)	4964 (4.0%)	4 (0.0%)	12354 (28.6%)	94 (0.2%)	10634 (26.1%)	289 (1.7%)	225 (4.2%)	44 (0.6%)	211 (0.8%)	–

mapped to multiple ChEBI IDs.

In some cases, more than 50% of the mappings are non-unique. The highest fractions of non unique ID mapping occurs when mapping to ChEBI, although when mapping from ChEBI to the other databases, this fraction reduces significantly. When considering Reactome, both mappings to and from this database lead to relatively high number of non-univocal assignments. SLM and SABIO-RK have a significant low ambiguity when mapping from other databases, although as shown in Table 4.2.5, only a small fraction in these databases can be mapped from other databases.

Table 4.2.6: Percentage of IDs in the database in the column that get mapped to more than one ID in the database in the corresponding row using database names as a bridge. Blue boxes are used to highlight highest numbers. While red boxes indicate lowest numbers.

Database	BiGG	ChEBI	enviPath	HMDB	KEGG	LIPID MAPS	MetaCyc	Reactome	SABIO-RK	SEED	SLM
BiGG	–	2.9	1.3	3.0	3.6	3.2	1.4	0.6	2.9	2.7	1.6
ChEBI	76.3	–	67.0	38.1	38.3	34.3	58.7	81.3	78.7	37.3	26.9
enviPath	6.3	6.5	–	8.2	6.1	0.0	0.0	12.2	7.7	4.6	0.0
HMDB	10.7	11.5	6.8	–	7.3	4.3	13.2	22.8	12.8	7.4	0.7
KEGG	17.0	15.2	11.1	28.5	–	10.2	18.5	34.5	19.6	12.4	33.3
LIPID MAPS	8.7	9.8	1.9	1.8	3.2	–	4.2	13.2	4.5	3.2	0.8
MetaCyc	0.5	3.9	0.0	2.5	3.9	2.0	–	6.0	4.1	1.5	0.6
Reactome	42.3	41.4	51.2	49.0	51.4	24.4	38.9	–	49.5	43.2	47.9
SABIO-RK	0.0	4.5	0.0	0.0	3.8	1.0	3.8	2.2	–	3.3	1.8
SEED	3.0	6.0	0.9	2.0	2.4	2.2	3.1	8.9	5.3	–	1.7
SLM	7.3	37.2	25.0	12.3	18.1	22.3	10.4	24.4	20.5	9.5	–

Mapping between databases using MNXRef ID

Another approach to map IDs from different databases is to use MetaNetX/MNXRef as a bridge. Table 4.2.7 shows the fraction of IDs in each database pair that can be mapped through MetaNetX/MNXRef. Again the

Table 4.2.7: Number of IDs in one database (column) that map to IDs in the database in the corresponding row using MetaNetX as a bridge. Percentages indicate fraction of IDs in the initial database. Blue boxes are used to highlight highest numbers. While red boxes indicate lowest numbers.

database	BiGG	ChEBI	enviPath	HMDB	KEGG	LIPID MAPS	MetaCyc	Reactome	SABIO-RK	SEED	SLM
BiGG	–	2064 (2.1%)	232 (2.1%)	1469 (3.5%)	1781 (4.7%)	533 (1.3%)	1715 (10.1%)	609 (29.6%)	1180 (15.7%)	2652 (9.9%)	221 (0.0%)
ChEBI	2064 (40.8%)	–	1424 (12.8%)	8775 (20.7%)	19244 (51.0%)	5464 (13.5%)	9019 (53.1%)	1242 (60.3%)	3252 (43.3%)	17649 (65.6%)	3848 (0.8%)
enviPath	232 (4.6%)	1424 (1.5%)	–	549 (1.3%)	1093 (2.9%)	166 (0.4%)	733 (4.3%)	120 (5.8%)	377 (5.0%)	1123 (4.2%)	23 (0.0%)
HMDB	1469 (29.0%)	8775 (9.1%)	549 (5.0%)	–	5028 (13.3%)	5387 (13.3%)	3283 (19.3%)	788 (38.3%)	1804 (24.0%)	5021 (18.7%)	9870 (2.0%)
KEGG	1781 (35.2%)	19244 (19.9%)	1093 (9.9%)	5028 (11.9%)	–	2397 (5.9%)	7030 (41.4%)	926 (45.0%)	2651 (35.3%)	16791 (62.4%)	375 (0.1%)
LIPID MAPS	533 (10.5%)	5464 (5.6%)	166 (1.5%)	5387 (12.7%)	2397 (6.4%)	–	2056 (12.1%)	325 (15.8%)	719 (9.6%)	2807 (10.4%)	10076 (2.0%)
MetaCyc	1715 (33.9%)	9019 (9.3%)	733 (6.6%)	3283 (7.8%)	7030 (18.6%)	2056 (5.1%)	–	877 (42.6%)	2538 (33.8%)	11502 (42.8%)	655 (0.1%)
Reactome	609 (12.0%)	1242 (1.3%)	120 (1.1%)	788 (1.9%)	926 (2.5%)	325 (0.8%)	877 (5.2%)	–	705 (9.4%)	1006 (3.7%)	200 (0.0%)
SABIO-RK	1180 (23.3%)	3252 (3.4%)	377 (3.4%)	1804 (4.3%)	2651 (7.0%)	719 (1.8%)	2538 (14.9%)	705 (34.3%)	–	2915 (10.8%)	253 (0.1%)
SEED	2652 (52.4%)	17649 (18.2%)	1123 (10.1%)	5021 (11.9%)	16791 (44.5%)	2807 (6.9%)	11502 (67.7%)	1006 (48.9%)	2915 (38.8%)	–	647 (0.1%)
SLM	221 (4.4%)	3848 (4.0%)	23 (0.2%)	9870 (23.3%)	375 (1.0%)	10076 (24.9%)	655 (3.9%)	200 (9.7%)	253 (3.4%)	647 (2.4%)	–

differences in coverage between the databases cause this table to be non-symmetric.

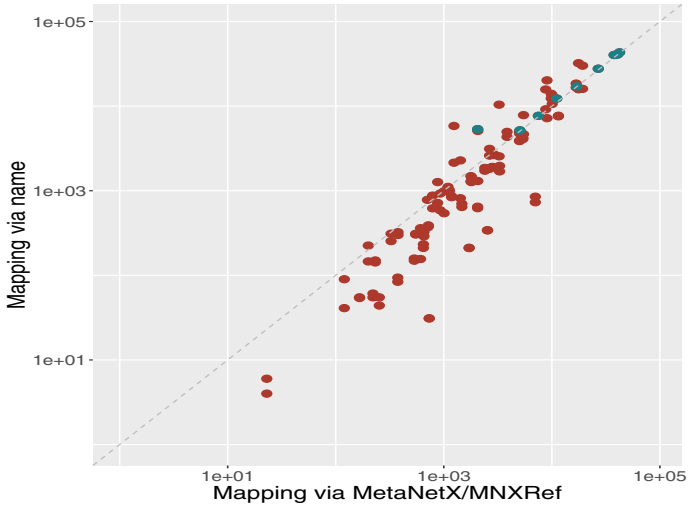


Figure 4.2.2: Comparison of number of mapping between the two approaches: The x axis shows the mapping resulted using MNXRef ID as a bridge; the y axis shows the number of mapping via name. Each red dot indicates mapping between a pair of database, diagonal points (in blue) indicate mapping from the database to itself. Mapping results from/to SLM are not shown in the plot as the number of matches outside the range considered.

Figure 4.2.2 shows that mapping via MNXRef ID results in more identified mappings than the previous approach that used names. Nevertheless, the overall map is also not high. None of tested databases maps higher than 70 % either to or from other databases. The highest match is 67.7 % when mapping MetaCyc to SEED. SEED can be mapped fairly well from BiGG, Reactome and KEGG with more than 40% match. Note that these are all databases specialized in reactions and metabolic pathways. There is almost no overlap between SEED and SLM, the latter specialized in lipids. Databases with overall good match are ChEBI, KEGG and MetaCyc. Among them,

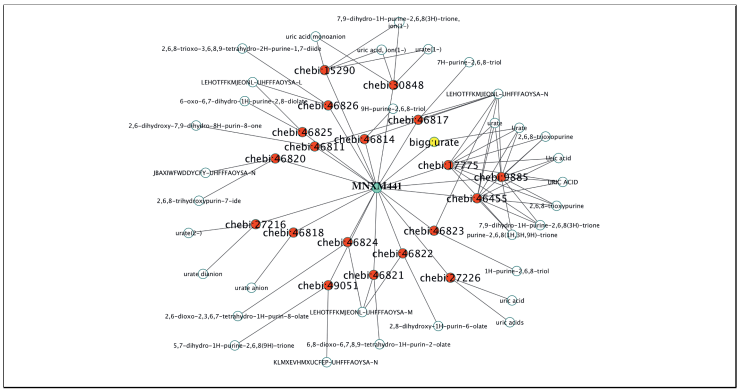


Figure 4.2.3: Visualization of the *inter* database inconsistency. An ID from BiGG (in yellow) can link to many other IDs in CheBI (red) when using MetaNetX ID (green) for the mapping.

ChEBI has the highest overlap with other databases. Almost 50 % of IDs in SEED, Reactome, MetaCyc, KEGG, SABIO-RK and BiGG can be mapped to ChEBI. However, there is not so much overlap when mapping enviPath (12.8%), LIPID MAPS (13.5%) and especially SLM (0.8%) to ChEBI. The remaining databases have a significant low overlap percentage when mapping via IDs. Especially SLM, there is just a minor part of the database that can be mapped to other databases.

This approach also results in instances of one ID from the first database associated to more than one ID in the target database, an example is provided in Figure 4.2.3 and Table 4.2.8 summarizes the identified cases.

Table 4.2.8: Percentage of IDs in the database in the column that get mapped to more than one ID in the database in the corresponding row using database MetaNetX as a bridge. Blue boxes are used to highlight highest numbers. While red boxes indicate lowest numbers.

Database	BiGG	ChEBI	enviPath	HMDB	KEGG	LIPID MAPS	MetaCyc	Reactome	SABIO-RK	SEED	SLM
BiGG	–	3.9	5.2	3.5	3.9	3.2	4.0	3.8	4.5	3.2	2.7
ChEBI	83.1	–	56.2	39.7	36.4	37.8	64.7	76.8	72.2	39.4	27.8
enviPath	9.9	10.6	–	12.0	8.1	8.4	7.6	14.2	11.1	8.1	8.7
HMDB	19.1	6.8	12.6	–	9.3	5.1	12.7	26.4	17.2	9.7	1.6
KEGG	15.0	10.0	11.0	22.1	–	8.4	9.6	19.7	17.5	11.2	14.7
LIPID MAPS	10.5	2.8	6.0	2.5	4.5	–	4.6	14.5	7.0	4.2	0.5
MetaCyc	3.6	1.4	3.4	2.1	1.7	0.9	–	4.3	2.8	1.1	0.5
Reactome	42.7	33.3	45.0	32.4	37.1	28.9	36.7	–	41.7	37.2	35.0
SABIO-RK	8.1	4.7	5.6	6.1	5.3	3.8	5.6	9.2	–	5.1	4.7
SEED	8.4	3.5	4.2	4.6	3.7	3.6	5.7	12.1	9.5	–	5.1
SLM	5.0	1.1	0.0	0.2	5.6	0.4	2.7	6.0	5.1	3.2	–

Name ambiguity and non-unique ID mapping between databases can lead to inconsistencies (different metabolites being considered to be equivalent) and included as such in the metabolic model. Table 4.2.9 lists some illustrative examples. These examples show that automatic mapping (manual mapping is impossible for large scale models) of compounds between or within databases can lead to introduction of unrealistic reactions that can potentially reduce the accuracy of the predictions of the model.

Table 4.2.9: Examples of mapping inconsistencies

Abbreviation	Database	IDs in Database	MetaNetX ID	Compound(s)
suc	MetaCyc	SUC	MNXM25	succinate
suc	Reactome	188980	MNXM167	sucrose
H	MetaCyc	PROTON	MNXM1	proton
H	MetaCyc	HIS	MNXM134	L-histidine
tmp	BiGG	tmp	MNXM87343	TMP
tmp	ChEBI	10529	MNXM257	Thymidine monophosphate
tmp	KEGG	C01081	MNXM662	Thiamine monophosphate
tmp	MetaCyc	CPD-610	MNXM88031	cyclo-triphosphoric acid
PPP	Reactome	1475054	MNXM3109	triphosphate ion
PPP	MetaCyc	2-PHENYL-2-1-PIPERDINYLPROPANE	MNXM150634	2-phenyl-2-1piperdinylpropane

4.3 Discussion

GEMs aim to be comprehensive representations of the metabolism of one organism. They are often built based on more than one database. As explained, the initial step of model constructions is typically automated model drafting. Tool selection will determine which name space the model is associated to. For instance, modelSEED uses SEED as a reference reaction database while Pathway Tools uses MetaCyc. In the next step in the model building process - manual curation - gap filling is possibly the most important task. Tools for gap-filling often systematically explore the GEM to identify possible gaps [264], Other methods rely on additional experimental data such as measured metabolites to identify the gaps [265]. In this step, researchers may use different sources and databases to identify reactions and associated metabolites. Errors might arise due to inconsistencies in this mapping.

A second application of GEMs is the integration and contextualization of 'omics' data such as transcriptomics, proteomics, metabolomics and/or fluxomics data. These applications often require a mapping of metabolite identifiers to match the namespace of the model and that of the database that has been used in the data gen-

eration process. Both applications may imply potential problem(s) caused by ambiguous names or identifiers.

Among the eleven explored databases, KEGG, BiGG, ChEBI, MetaCyc, HMDB and SEED are the most commonly used in metabolic modeling. We calculated the ambiguity of names and the multiplicity of non-systematic identifiers within and between eleven databases. Within the same database, the percentage of identifiers with multiplicity larger than one varies from 0 % to 100 %, whereas the ambiguity of names ranges from 0.07 % to 29.4 %. When mapping between databases, these ambiguities and multiplicities lead to larger inconsistencies, and this agree with previous observations regarding small molecules databases [262, 266]. The inconsistencies when mapping using metabolite names range from 0 % to 81.2 %. Similar results are obtained while mapping via MNXRef ID, between databases, as the number of inconsistencies varies from 0 % to 83 %, however on average better results are obtained. Mapping with the databases with the highest number of ambiguous names also results in higher number of inconsistencies than when mapping between other databases. Among the eleven tested databases, Reactome, HMDB, ChEBI, and KEGG are those that show the highest *intra-* and *inter-*database ambiguity.

Most of the ambiguous names are associated to general compounds such as triacylglycerol, glycan or protein. These names and IDs represent classes of compounds rather than metabolites with defined structures and are included in metabolic models as they have a clear biological interpretation. However, care should be taken when introducing them in databases and these names should not be included in the list of synonyms for specific compounds, as mentioned in [266]. Using abbreviations to refer to compounds is also highly ambiguous as the same abbreviations can represent different compounds in the same or in different databases.

Our findings show that compound names or IDs cannot be clearly mapped au-

tomatically. Even if we use non-ambiguous identifiers, many mappings are still inconsistent because they can link to ambiguous names. MetaNetX solved some of the issues as shown in Table 4.2.9. However, not all compounds in the eleven tested databases can be mapped with MNXRef. Mapping from MetaCyc to SEED, SEED to ChEBI, and SEED to KEGG using MNXRef give the highest number of matches, but still only around 60 % of compounds matched. Other databases show much lower coverage.

In order to use metaNetX/MNXRef ID to map compounds in a GEM, the namespace of the model needs to be related to at least one of the eleven databases considered. However, many models uses custom made naming conventions [267]. For these models, mapping through name is the only option.

Ambiguous namespaces also hamper the (re)use of models from different research labs or organisms. Due to a low level of interoperability, in practice it is impossible to directly compare models, as metabolites can hardly be cross-mapped, which in turn makes it impossible to compare reactions in both models, see examples in Table 4.2.9. Nevertheless, comparing models is important and necessary: it helps to reduce the time to build models for closely related species; to combine efforts from different research groups that study the same organism; and to study the metabolic differences between different organisms. In addition, microbial communities are notoriously difficult to characterize. While transcriptomics and proteomics measurements can be associated (to a great extent) to the originating microorganism, it is not possible to do this for metabolites. Therefore, there is a need for models that can help combine both types of measurements. As a result, there are on-going efforts to define modelling frameworks, based on combining GEMs of individual organisms, to characterize the behaviour of the community [268, 269, 252, 253]. Enabling unambiguous mapping will be required to take full advantage of these on-

going developments.

Below we have enumerated a number of recommendations that may increase the level of interoperability of GEMs, facilitating unambiguous mapping

- Limit the use of aliases, *i.e.* compound classes or abbreviations, as synonyms in databases. These aliases increase human readability, but should be clearly distinguished from names and synonyms in the databases and should not be used for mapping.

- In the context of metabolic modelling it is frequent and desirable to use compound classes to identify generic compounds [259]. Compounds like 'biomass' or 'lipid' are often used in GEMs; this does not affect the use of the model, except when predicting or simulating the production (of a specific component) of generic compounds, *i.e.* when 'lipids' are the main focus of the model. In fact, it is often better to use generic compounds whenever a specific compound is not needed, as they can be universal. For instance, 'biomass' has been used as a standard among the modelling community as an artificial compound that represents the growth objective of the cell [270, 248]. Another reason is that often the precise identity of the compounds is not needed and there is a lack of experimental data for their characterization. Therefore, when using generic compounds, it is desirable to add extensive annotation to the model to clearly state which compounds they represent and for which purpose they are used in the model. These generic compounds are among the most ambiguous entities in the eleven analyzed databases and we therefore advise to exclude them from any automatic mapping process.

- Avoid using highly ambiguous names as the sole description of the compound in the model. When referring to these compounds, clear annotation needs to

be included to prevent mismatches and inconsistencies.

- In addition to human-readable identifiers and database-dependent identifiers, include database-identifiers, such as InChI [247, 226], whenever possible for compounds with defined structures. Using InChI can help to fully automate the mapping [259]. However, it should be taken into account that mismatches and errors can also happen because identifiers can also link to incorrect InChI as shown in [271, 266, 260].

- Model mapping only based on metabolite information can imply certain mismatch due to differences in namespaces, even if systematic identifiers were used. Hence, different mapping strategies, *i.e.* mapping through encoding genes and network topology [250], should be used to complement name or identifier based mapping.

- GEMs also need to have a unique standard annotation so that they generate the same output even when different tools are used for the simulation. Neal *et al.* [272] suggest that semantic annotation can help to store and combine models, but these models need to stick to a unique standard annotation format.

Simply deciding a standard database/identifier/annotation to represent metabolites in models will also not help to improve the situation, as they will limit the available model construction tools. Nevertheless, while increasing the level of interoperability none of the presented approaches above can by itself ensure automated mapping without errors. Different approaches need to be combined when translating between namespaces. Manual curation is still required, at least for compounds with highly ambiguous names.

We did analyse the (in)consistency of databases (commonly) used in metabolic modelling but we did not analyse the (in)consistency of GEMs built using different databases. However, since every metabolite in a GEM is usually associated with at least one identifier from biochemical databases such as KEGG, BiGG, SEED or MetaCyc, every GEM can be considered as a small subset of the identifiers and names from those database(s). Hence, the ambiguity of the compound names in GEMs can be considered to be equivalent the ambiguity of the compound names in the tested databases. Moreover, it should be noted that some databases (like BIGG) aggregate compounds names used in deposited GEMs and thus mapping of these databases against other databases provide an overall, direct measure of the ambiguity of the compound names in GEMs. In addition to solving mapping inconsistencies, GEM namespace translation can be further improved by using tools that analyse the consistency of the generated models [19].

Finally, our analysis has some limitations. It should be noted that the list of inconsistencies provided represents just an upper bound to the number of possible errors when changing namespaces. We have only studied non-systematic identifier and names. We did not use structure data such as MOL files, we cannot evaluate how many of the consistent mappings are actually correct. We have not included such information in our analysis because it is not often found in metabolic models. In any case, the inconsistencies here described pertain automatic mapping and most (or all) of them should be fixable upon manual curation. Comparing names between databases is not trivial due to heterogeneity issues: our approach may be over simplified, which may reflect in the results shown. It should be noted that in some databases, synonyms are clearly differentiated, in this case, the inconsistency will not arise. However in many databases considered in this study, synonyms are not well distinguished. For instance H in MetaCyc belong to the synonyms list of

both proton and L-histidine. This is one of the primary causes of ambiguous mapping. In addition, MetaNetX data that was downloaded at the moment of conducting this study contained data from the originating databases that was produced in 2017 and some in 2016 (see section 4.4 for more detail). As databases change over time, a similar analysis with the most recent database updates might lead to different results. *Stat Roma pristina nomine, nomina nuda tenemus.*

4.4 Methods

4.4.1 Data collection and preprocessing

Data about compound identifiers and synonyms were downloaded from MetaNetX[256]. MetaNetX is a repository of GEMs and biochemical pathways. It contains entries from some of the most relevant databases that have been used in GEMs construction and simulation such as KEGG, BiGG , MetaCyc and SEED [273]. The platform (http://www.metanetx.org/) allows access to these databases as well as provides tools to map/translate them. In this study, The chem_xref.tsv file was downloaded from the MetaNetX website on 31st October, 2018. In the following, we provide a brief description of the content of these databases.

Biochemical, Genetic and Genomic models (BiGG) [274] is a knowledge database of genome scale metabolic models (GEMs). Currently, it contains 85 high-quality, manually curated GEMs, 24311 reactions, and 7339 metabolites (data retrieved on 30th, Nov, 2018 from http://bigg.ucsd.edu/). In BiGG, the metabolite is identified as the abbreviation of its name. For example, '10fthf' for 10-Formyltetrahydrofolate. MetaNetX obtained data from BiGG on 2017/04/11.

Model SEED [275] is a platform to construct GEMs that uses its own database

for metabolites and reactions. This database combines information from KEGG and existing metabolic models in a non-redundant set of reactions. In this database, metabolite identifiers start with "cpd" and followed by a 5 digits number. For example, D-Glucose-1-Phosphate is cpd00089. The database can be downloaded from https://github.com/ModelSEED/ModelSEEDDatabase/tree/master/Biochemistry. MetaNetX obtained data from SEED on 2017/04/13.

ChEBI [276]. (http://www.ebi.ac.uk/chebi/aboutChebiForward.do) is a database of Chemical Entities of Biological Interest [276] and is a repository for small chemical compounds. In ChEBI, metabolites are named by 5 digit numbers. For example, Alpha-D-glucose-1-phosphate(2-) is 58601. File can be downloaded from ftp://ftp.ebi.ac.uk/pub/databases/chebi/Flat_file_tab_delimited/. ChEBI data in MetaNetX are from the release version 150.

enviPath [277]. (https://envipath.org/). Is a database to store and predict the microbial biotransformation of organic environmental contaminants. Data in MetaNetX were downloaded on 2017/04/12.

HMDB [278]. (http://www.hmdb.ca). Is a comprehensive and curated collection of human metabolite and human metabolism data. Data in MetaNetX was obtained on 2017/04/12.

KEGG [245]. (http://www.KEGG.jp). The Kyoto Encyclopedia of Genes and Genomes is a resource that provides information about pathways and reactions in organisms. In KEGG, metabolites started with a letter 'C' (compound) and followed by 5 digit numbers. For example, D-Glucose-1-Phosphate is identified as C00103. Data in MetaNetX were obtained on 2017/04/12.

LIPID MAPS [279]. (http://www.lipidmaps.org). Is a database that contains structures and annotations of biologically relevant lipids. Data in MetaNetX were obtained on 2017/04/13.

MetaCyc [244]. (http://metacyc.org). Is a curated database of metabolic pathways. All data in MetaCyc are experimentally validated. The metabolite is identified by its full name. For example, D-glucose-1-Phosphate is D-glucose-1-phosphate. The database can be downloaded here http://bioinformatics.ai.sri.com/ptools/flatfile-format.html. Data in MetaNetX were obtained on 2017/04/13.

Reactome [280]. (http://www.reactome.org). Is a curated and peer-reviewed database of human biological processes. Data in MetaNetX were obtained on 2017/04/13.

SABIO-RK [281]. (http://sabiork.h-its.org/). Is a database containing comprehensive information about biochemical reactions and their kinetic properties. Data in MetaNetX were obtained on 2016/05/27.

SwissLipids (SLM) [282] (http://www.swisslipids.org/) contains curated data about lipid structures and metabolism. Data in MetaNetX were obtained on 2017/04/13.

The original data file was modified prior to analyzing. The modification includes the removal of the description part, of IDs starting by bigg:M as they are not real compound ID in BiGG, and the removal of 'biomass' compounds. Data from MetaNetX were organized in four columns in this order: compound ID in original database with database indicator in front, for example bigg:1ofthf, corresponding compound IDs in MetaNetX, evidence and description (name).

4.4.2 *Intra*-database analysis

For *intra*-database consistency analysis, the first, the second and the last column of the MetaNetX data file were used for mapping. Name ambiguity was calculated as the number of ID each name links to. Similarly, the name multiplicity of each ID was

calculated as the number of names it refers to.

4.4.3 *Inter*-database analysis

We mapped compound IDs between databases. A direct map between IDs in the database is not possible. The tested databases use different system for compound identifiers. For instance, in KEGG, the compound ID is a capital 'C' following by a 5 digit numbers, *i.e.* 'C00002' for ATP. In contrast, in BiGG, the compound is identified as abbreviation of its name, for example, 'atp' for ATP. Therefore, to map from one ID in database 1 to other ID in database 2, we used either the associated compound name or the associated MNXRef ID. That is also what MNXRef is meant for, as a link between databases.

Mappings via name were done by link from name to name in one database to the other. We first identified all compound names from one database, *i.e.* database A. From this list, we counted the number of IDs in the second database, *i.e.* database B, that link to each name in the database A. It means in this case, we did not use any string processing algorithm, *.i.e* processing case sensitive, underscore, or brackets, the name was mapped as exact match. Ambiguous names were treated as normal name in the database. In other words, we did not distinguish ambiguous names from un-ambiguous names from the mapping.

5

Systematic evaluation of gap-filling algorithms in GEMs

Pham Nhung, Peter J. Schaap, Vitor A. P. Martins Dos Santos and Maria Suarez-Diez

ABSTRACT

Microbes have been increasingly used to provide solutions for global healthcare, agricultural and environmental challenges. Development of microbial factories is often a costly, time-consuming and uncertain process, and is best done with the assistance of mathematical modeling. Genome-scale, constraint-based metabolic models (GEMs) are among the most common methods to explore microbial metabolism, as they provide a comprehensive metabolic repository of an organism that enables simulating impact of genetic modifications. However, the lack of accurate functional annotations often renders such models incomplete, giving rise to missing reactions in the network ('gaps'). Hence, gap-filling is an essential step in model development. Thus far, 18 algorithms have been published to assist in the gap-filling process. Their usability and accuracy vary widely due to differences in their objectives, implementation platforms, and input data. Hence, we carried an extensive review and evaluation of these algorithms from a user's perspective. We found that a majority of the tools do not have workable implementations available. From those for which an implementation is readily available, we selected SMILEY, FASTGAPFILL and Meneco to further investigate their performances. As for recall and precision, SMILEY is the best among three algorithms for small-scale degradation. When applied to highly degraded networks, all three algorithms perform poorly. Gap-filling algorithms could be great resource to improve a GEM but are hardly used in modelling as they are hampered by the lack of workable implementations and inconsistencies between the model namespace and the required reference databases. In order to improve the situation, there should be workable implementation for these algorithms and the inconsistency between the model namespace and the required reference databases need to be solved.

5.1 Introduction

Microorganisms are minuscule chemical factories that have the ability to naturally produce a wide range of valuable compounds. Producing chemicals from these living cells has been considered an alternative sustainable approach over the tradition petro-based chemical production due to the use of available renewable biomass [139, 283]. Using microbes for chemical production also bring various advantages such as a stable production cost due to the use of low-cost renewable biomass as substrate [140], the production of less byproduct [141], and the production of a wider range of chemical due to the flexible of the microbial systems [140].

Microorganisms have been employed to produce chemicals from the ancient time with significant impact for instance the introduction of beverages, cheeses, bread, pickled foods and vinegar [34]. These early processes employed microbes when they were not well-characterized. Nowadays, advances in genetic sequencing and high-throughput technologies have fostered the development of Synthetic biology and Systems biology [284] leading to the establishment of numerous examples of microbial cell factories for many targets such as chemicals, materials [37, 38] and biofuels [39].

Synthetic biology is *an application of science, technology and engineering to facilitate and accelerate the design, manufacture and/or modification of genetic materials in living organisms such as microbes* [33]. Systems biology is a collection of quantitative and qualitative modelling approaches to study living organism as a whole [284]. Mathematical modelling in Systems biology is an essential part in Synthetic biology to guide rational design and to predict outcomes of potential genetic and environmental implementations [45, 285]. One of the most common used modelling techniques in metabolic engineering is genome-scale metabolic modeling (GEM), a linear, constraint-based model that enables simulating metabolism at steady-state.

It contains a comprehensive inventory of metabolic reactions that are predicted or known to occur in an organism. GEMs have been shown to be useful for industrial and medical applications owning to their power in hypothesis driven discovery and in guiding metabolic engineering [28, 29].

Model organisms such as *Escherichia coli* and *Saccharomyces cerevisiae* have been extensively researched and thoroughly curated models are available [286, 287]. However, there is a plethora of microorganis ms that have great potential for applications and are not as well characterized. Oleaginous yeasts such as *Cutaneotrichosporon oleaginosus* and *Yarrow lipolytica* can accumulate at least 50 % lipid in their biomass, implying great potential for lipid production[288]. Other promising organisms are bacteria with high solvent tolerance or redox potential, such as *Pseudomonas putida* [289], or prokaryotes that can thrive in extreme conditions such as thermophiles and acidophiles that can produce chemicals with higher productivity and lower maintenance cost [290].

Affordable sequencing technologies and easy-to-use pipelines for genome annotation such as SAPP [291], Prokka [292] or DFAST [293] have largely expanded the interest in non-model organisms. This leads to a concomitant increased interest in building GEMs and in deploying modeling tools for such microbes. In addition, there is a growing interest in modelling microbial communities, enabled by the availability of metagenomics/metatranscriptomics technology [294, 295]. Thus users need to have good modelling strategies to design the possible interventions.

The construction of GEMs critically depends on the identification of metabolic functions encoded in the genome of the organism in question. During model construction intensive manual curation is required to remove gaps in the network. Gap is missing reactions to either produce or consume a metabolite, the metabolite becomes a dead-end. Reactions involve dead-end metabolites will not be able to carry

flux, a so-called block reactions. It violates the steady state assumption of the simulation techniques used in GEMS, which requires metabolites to be produced and consumed in equal amounts. As a result, the model is rendered unable to simulate the network.

The causes for gaps are twofold. First, the organism is indeed missing the enzyme for the reaction [296]. For instance, *Akkermansia muciniphila* one of the most abundant mucosal bacteria in humans is threonine auxotroph, a consequence of the adaptation in the mucosal environment where threonine is abundant [297]. The organism in this case does not possess genes to encode enzymes for threonine synthesis. This is represented in the model as a missing reaction to produce threonine, although it is consumed in biomass and protein synthesis reactions. This renders the model unable to simulate growth unless threonine is *in silico* supplemented to the medium, correctly reproducing the auxotrophy of this organism. These are biological gaps and they should not be filled. Instead, this gap will be solved by entailing identification of additional biological mechanism, such as enabling threonine uptake from the environment in the case of *Akkermansia muciniphila*.

The second cause of gaps is due to the limitation in our knowledge, a so-called knowledge gaps resulting from missing annotation. This type of gap needs to be solved to allow fluxes through the network. These knowledge gaps are targets for gap-filling. GEM has well-defined scope, and is an open system by itself. In this study, we do not consider metabolite whose production or consumption is in networks outside the scope of the model as gap.

Gap-filling is the process of finding reactions to connect dead-end metabolites to the network to allow flux through the objective function. The gap-filling process starts with the identification of candidate reactions to restore network connectivity. In this step a list of reactions are generated without genomic evidence. In the

manual curation step candidate genes encoding enzymes for suggested reactions are identified. Reactions added to the model for which no genome evidence is found are termed orphan reactions. These reactions can be either spontaneous or have unknown enzyme-encoding-genes.

The identification of gaps depends on the objective function of the model, very often the biomass synthesis reaction. It decides the scope of a GEM. Any inconsistency between model simulation and observation may relate to the objective function, whether the model capture correctly all pathways that make up this function. For instance if the objective function of a model is designed to specifically account for different types of membrane lipids then it might end up showing gaps on membrane lipid synthesis pathways, whereas a model with an objective function that only contain a generic type of lipids will not have these gaps. In addition, the identification of gaps also depends on growth phenotypes of the organism in question in the selected medium. Gaps are difference between simulation in rich media and minimum media. When the model is simulated in rich media where microbes are supplied with external biomass precursors, pathways for energy generation and co-factor regeneration will be checked while simulating in minimum media GEM is more challenged exposing possible gaps. In this case, beside energy and co-factor generation pathways, pathways for synthesis of biomass precursors are also checked.

A typical GEM often contains from a few hundred to few thousands reactions. During its construction there are usually a few to dozens of dead-end metabolites rendering manual gap-filling in many cases impractical. To address this issue, various algorithms have been published to assist gap-filling. Pan and Reed [298] divide gap-filling algorithms in two broad categories: reaction addition algorithms and gene assignment algorithms. Reaction addition algorithms identify gaps and then suggest changes in the model content to fill them. In this category, the list of suggested candi-

dates are orphan reactions that need to be manually assigned enzyme-coding-genes for. In the second category, gene assignment algorithms attempt to match reactions with gene sequences. In this study we define gap-filling algorithms as reaction addition algorithms. These algorithms apply the same principle, they aim to facilitate flux to produce a target phenotype, most often biomass production, by adding as low number of reactions as possible from a reference database. The main differences among these algorithms are how they identify gaps, for instance purely based on network topology or also based on stoichiometric balance. Another difference among these gap-filling tools is the database where they draw candidate reactions from.

Algorithms explicitly designed for gap-filling have been introduced either as stand-alone tools or as built-in functions in automatic reconstruction tools such as ModelSEED [299], CoreCo [300], Pathway Tools [301], RAVEN Toolbox [302], CarveMe [303] and AureMe [304]. Some of these algorithms have been tested and compared in previous reviews. Latendresse and Karp [305] evaluated the performance of the two modes of the gap-filler implemented in the Pathway Tools MetaFlux sofware [306]. They randomly deleted flux-carrying reactions from the EcoCyc-20.0-GEM of *Escherichia coli* [307], and assessed how accurately the model was restored after gap-filling, using MetaCyc [244] as reference database. They used two different solvers (CPLEX and SCIP) and reported precision and recall statistics for each gap-filling variant. They found that in the most accurate variant, 13% of the gap-filled reactions were incorrect and 39% of the gap-filled reactions were not found. In this case, the gap-filler was designed to be used in Pathway Tools and was tested on a model that was built from the MetaCyc library.

More generic stand-alone tools have been designed for gap-filling since 2006 [298]. Most of them are mainly based on network topology either in bottom-up and top-down manner. In bottom-up approach the network is enriched until the objec-

tive is achieved, i.e. GapFill [308] and FASTGAPFILL [309]. Top-down starts from adding all reactions and eliminate each of them iteratively, i.e. Christian et al [310] and MIRAGE [311]. Despite the continuous publication of these algorithms, they are not independent, many of them are built based on or as a modification of previous published algorithms. For instance, GrowMatch [312] and SMILEY [313] used GapFill [308] as foundation. BoostGAPFILL [314] can be used to produce input for FASTGAPFILL. While likelihood-based gap filling workflows [315] employed part of the ProbAnnoWeb/ProbAnnoPy algorithm [316].

A few gap-filling tools have been reviewed and tested. Oyetunde et al. [314] used random deletion approach to test their algorithm, BOOSTGAPFILL. They introduced artificial gaps on several models and compared the performance of BOOSTGAPFILL against previously developed algorithms, FASTGAPFILL and GrowMatch, using GapFind [308] to calculate the number of gaps before and after gapfilling. The reference database in their study is KEGG [130]. Their results on BOOSTGAPFILL show more than 60% precision and recall which is double that of other tested algorithms.

When Prigent et al. [317] introduced Meneco they generated 3600 randomly degraded networks of metabolic model iJR904 of *E. coli*, at different degrees of degradation, and tested their algorithm on each, with MetaCyc as the database of candidate reactions. Their results show that Meneco was able to find essential reactions missing in networks at high degradation rates.

New gap-filling algorithms have been continuously developed and many of them have not yet reviewed and evaluated. The highly interdisciplinary nature of GEM development brings together computational scientists, mathematicians, physicians and molecular biologists in the same field. Since more researchers have been using GEMs [318], in this study we aim to give an extensive review and evaluation of the

performance and the practical usability of recently published stand-alone gap-filling tools for all users. We do not cover gap-filling algorithms embedded in model reconstruction tools since many of them use the same algorithms as stand-alone tools and they do not allow gap-filling without building the model from scratch.

5.2 Overview of gap-filling algorithms

Growth phenotypes either in one or multiple conditions depending on the intended purpose of modelling are the basic required input for all gap-filling algorithms. Additionally, some algorithms also require experimental data such as gene expression data, gene essentiality data, or metabolic flux data. In this study, we have divided gap-filling algorithms into two groups based on their required inputs: algorithms that only require growth data and those that need additional experimental data. Beside these two groups, there are other approaches such as machine learning techniques that are not classified as gap-filling algorithms but have been used for gap-filling in GEMs.

5.2.1 Gap-filling algorithms requiring only growth data

SMILEY (2006) [313] The algorithm poses a mixed-integer linear programming (MILP) problem reconciling experimental observations and *in-silico* simulations of growth on minimal media. It aims to find a minimum set of reactions from a universal database, KEGG, to add to the GEM to rescue *in-silico* false negative growth predictions. We were not able to find an implementation of this algorithm. Yet, there is a very similar gap filling implementation [319] in COBRApy toolbox, we therefore considered this as SMILEY algorithm.

GapFind/GapFill (2007) [308] These MILP algorithms identifies metabolites

that cannot be produced or consumed (GapFind) and searches for candidate reactions from MetaCyc database to connect the gaps to the network (GapFill) either by reversing reaction direction, adding reaction from other organism, adding exchange reactions or adding intracellular transport reactions.

Christian et al (2009)[310] The algorithm uses a similar approach to that of GapFill and SMILEY [313]. First all reactions in a reference database are added to the draft network. Each of these new reactions is removed to check for their essentiality to produce the target metabolites. Non-essential reactions are removed and the remaining reactions form the gap-filling set.

FASTGAPFILL (2014) [320] The first algorithm that computes near minimal set of added reactions for compartmentalized models. While other gap-filling algorithms focus on facilitate flux to simulate a desire phenotype, i.e. biomass production, the objective of FASTGAPFILL is to unblock as many gaps as possible. Hence, FASTGAPFILL does not guarantee the smallest solution size of additional reactions. The algorithm implements FASTCORE algorithm [320] to expand a core subset of the network with reactions from a universal database (KEGG) until all dead-end metabolites in the model are eliminated. The price is that the solution set will contain transport and exchange reactions.

Likelihood-based gap filling workflows (2014) [315] The algorithm predicts alternative functions for genes by calculating their likelihood scores based on sequence homology. Maximum-likelihood pathways for gap-filling are found by solving an MILP problem. This approach is genome-specific cause it provides reaction for gap-filling with gene-protein-reaction association information and confidence scores for each suggestion. It is available via API or command-line web interface as part of the DOE Systems Biology Knowledgebase (KBase), an automated metabolic network reconstruction framework [321].

DEF (2016) [322] The algorithm mimics the endosymbiosis event in microbes. In this process, mitochondria, a predecessor of prokaryotes capable of aerobic respiration, was engulfed by primitive eukaryotes unable to consume oxygen. As a result, the eukaryote adopts the most efficient pathways to consume oxygen. Similarly, DEF solves a linear programming (LP) problem that looks for reactions in an external database to maximize the consumption or production of dead-end metabolites in the original model.

BoostGAPFILL (2016) [314] The algorithm leverages machine learning and constraint-based methods to find reactions for gap-filling. The incomplete stoichiometric matrix of a constraint-based model, a subject for gap-fill, is converted to an incomplete adjacency matrix A. This adjacency matrix A is completed using matrix factorization. An integer least square optimization is used to select reactions from the universal database of choice that best match the completed A. This step resulted in a ranking of all reactions. This ranking set can be used as the database input for FASTGAPFILL in mode 2 in BoostGAPFILL. Or it can be used for gap-filling based on network topology with or without extra biological constraints in mode 1 and 3, respectively.

Meneco (2017) [317] or **Me**tabolic **Ne**twork **Co**mpletion is a topological gap-filling approach that allows stoichiometric constraints to be violated and does not rely on phenotypic or taxonomic information. This approach is specially designed for new, less-studied organism whose experimental data is scarce.

Hybrid Metabolic Network Completion (2017) [323] The algorithm combines answer set programming with linear stoichiometry constraints for network completion. By doing this, it avoids self-activated cycles resulting from flux balance based method. It is claimed to offer a better solution for restoring highly degraded models.

ProbAnnoWeb/ProbAnnoPy (2018) [316] ProbAnnoWeb/ProbAnnoPy first rank the reaction for gap-fill based on likelihoods of gene functions. Similar to likelihood-based gap filling workflows, these functions are based on sequence homology with a trusted annotation database. The rank organism-specific reactions are next subjected for gap-filling candidates. In this way, reaction databases for gap-filling in the likelihood-based approach is customized for the specific organism in question.

OptFill (2020) [324] The most recent gap-filling algorithm, a Novel Optimization-Based Tool to Automate Infeasible Cycle-Free Gapfilling of Genome-Scale Metabolic Models, aims to remove thermodynamic infeasible cycles in GEM. This whole model gap-filling algorithm identifies a minimum number of reactions to connect as many metabolites to the network as possible while avoiding introducing thermodynamic infeasible cycles.

5.2.2 Gap-filling algorithms requiring experimental data

OMNI (Optimal Metabolic Network Identification) (2006) [325] Is the first algorithm proposed for gap-filling in GEMs. The algorithm poses a bilevel MILP to find the optimal reaction set to match *in-vivo* and *in-silico* metabolic flux data. In this case, the outer optimization problem is to find reactions to add to the model, while the inner problem finds flux distributions for the optimal solution for a particular model structure.

GrowMatch (2009) [312] An optimization-based framework that predicts reactions to suppress or to restore growth to match with experimental observations. This algorithm uses *in-vitro* determined gene essentiality data to identify incorrect model predictions. When the model predicts growth while no-growth is observed *in-vitro* (false positives), the algorithm poses a bilevel optimization problem to sup-

press growth. In this case, the outer problem minimizes biomass formation with a pre-defined number of reactions to suppress while the inner problem maximizes biomass formation when redirecting metabolic fluxes to biomass precursors, uptake and ATP maintenance. For *in-silico* non growth and growth *in-vitro* mismatch, the algorithm based on network topology to find reactions from an external multi-organism database such as MetaCyc to connect with the network in the model.

MIRAGE (2012) [311] MetabolIc Reconstruction via functionAl GEnomics (MIRAGE) identifies gaps by integrating metabolic flux analysis and functional genomics data. MIRAGE use a two-step procedure where functional genomics data is used in the first step to calculate the probability of adding a reaction from a reference database of choice into the model. Enzymes' phylogenetic profiles and gene expression profile of the target organism are used in this step to calculate phylogenetic weight of each reaction in the reference database. In this manner, MIRAGE also includes gene assignment in the gap-filling procedure. In the second step, metabolic flux analysis is used to identify the set of high-weight gap-filling reaction whose addition will restore the desire phenotype.

GlobalFit (2016) [326] While other gap-filling algorithms fill gaps per each condition iteratively, GlobalFit reformulated the MILP problem to a bi-level linear problem to look for a single global optimal network to match *in-silico* prediction to all experimental observations simultaneously. In addition to a global set of changes, it also suggests subsets of solutions to solve false positive prediction per each observation. GlobalFit is integrated with the sybil [327] toolbox for constraint-based analyses.

SONEC (2016) [328] SOrting by NEtwork Completion (SONEC) approach fills in network gaps based on analysing bins of contigs from metagenomics samples. The algorithm aims to complete metabolic networks in a microbes community. Fragments from metagenomics samples can be mapped to metabolic functions,

this results in various metabolic networks representing different species. The unassigned sequence fragments will be assigned to the network such that it can eliminate as many as dead-end metabolite as possible. This means it will be assigned to a network that have the highest connectivity scores of metabolites. These connectivity scores indicate the number of dead-end metabolites that can be consumed or produced in the parent network with the addition of the unassigned reaction.

GAUGE(2017)[329] The algorithm finds gaps based on co-expression data of genes. Inconsistency between *in-vitro* coupled gene expression and *in-silico* flux coupled reactions indicate missing reactions in the network. GAUGE aims to identify the minimum set of reactions to match coupled gene expression observations and flux couple reaction predictions by solving an MILP problem. However, the algorithm only works on the subset of the model where reactions have clear gene-protein-association.

EnsembleFBA (2017) [330] The algorithm aims to limit the bias of gap-filling order in which the end network when gap-filling model for growth on glucose and then sucrose, for instance, is different from that when gap-filling for growth on sucrose first and then on glucose. EnsembleFBA was developed based on FASTGAPFILL and FastGapFilling algorithms. It then compiles individual networks resulted from gap-filled for each growth medium with a random order into an ensemble network.

5.2.3 Other techniques

There are many other algorithms that do not suggest candidates for gap-filling but can support gap-detection and gap-filling procedures. Recently, Martyushenko and Almaas [331] (2019) published ErrorTracer. As implied in its name, ErroTracer allows to identify inconsistencies, classify them and inspect their origins. However,

the algorithm does not suggest candidate reactions for gap-filling. Another older technique, **M**odel-**e**nable **g**ene **s**earch (MEGS) [332] is a combination of computational and experimental approaches to identify gaps by functional genomics analysis and to assign genes to reactions to fill them. Another approach that combines computational and experimental power to identify gaps and gap-filling candidates is from [333]. The authors used GEMs to identify unconnected modules in the network and validate their essentiality in the lab.

Many machine learning techniques such as naïve Bayes, decision trees, and logistic regression have been used for pathway prediction [334, 335] and can also be used for gap-filling. Recently, Medlock and Papin [336] published a framework for guiding model refinement using machine learning called automated metabolic model ensemble-driven elimination of uncertainty with statistical learning (AMMEDEUS). AMMEDEUS used unsupervised learning to identify inconsistent between *in-silico* and *in-vitro* observation. Supervised learning is then performed to suggest modification to reconcile model prediction and experimental data.

5.3 Results

5.3.1 Usability evaluation

In this study we evaluated gap-filling algorithms from a user's perspective. Hence, we focus on the usability of these tools without the need of writing code. It means they are implemented either as a web portal, a command-line tool or a graphical user interface. As shown in Table 5.3.1, except for GrowMatch, Christian et al [310] and hybrid Metabolic Network Completion algorithms, all others have accessible implementation. They are obtainable either from the supplementary files in the original papers or from code repositories such as github, e.g. OMNI and MIRAGE, or are

available as web applications e.g. DEF and Prob AnnoWeb or as built-in packages in common used toolboxes such as COBRA e.g. FASTGAPFILL and BoostGAPFILL.

The next important criterion is if they are easy to install. Whether it is easy or difficult to install a software critically depends on user's experience. This can be subjective for gap-filling algorithms since target audiences of these tools are mostly in GEMs community with diverse backgrounds ranging from computer science to biology. Therefore, to establish a common ground for evaluation, we considered most used platforms in GEMs community such as MATLAB, PYTHON and R and solvers such as Gurobi, GPLK and CPLEX are easy to obtain for gap-filling algorithms users. We graded the user friendliness of an installation based on how many dependencies the algorithm in question required and how easy it was to get these dependencies. Algorithms with additional dependencies will therefore be considered to be more tedious to install. Most of the algorithms that we tested in this study do not required extra packages or compilers, with the exception of OMNI, GapFind/GapFill and OptFill that require GAMS. Similar to MATLAB, GAMS is a programming language and an optimization tools [337]. It has been used in GEMs community but not as commonly as MATLAB. MATLAB is more wide-spread in many other disciplines while GAMS is more specialized and has a narrower user community. This is evident by the high number of users for toolboxs using MATLAB such as COBRA toolbox [338]. In addition, it is more difficult to get access to GAMS than MATLAB even for academic users.

Upon installation, we tested if these algorithms are easy to run which mean they have clear documentation and can be executed without errors. Our first attempt was to reproduce the results in the original publications using example data and codes provided along with the algorithms or if they have a tutorial. With the exception of SMILEY, FASTGAPFILL, Meneco and ProbAnnoWeb/ProAnnoPy, other algo-

rithms are not well-documented. Some do not provide clear instruction on what parameters they require or how to format input data. Among algorithms with available implementation, FASTGAPFILL, BoostGAPFILL and GAUGE required debugging in order to run them. These algorithms are implemented and used some functions in COBRA toolbox, their errors mainly cause by the incompatible issues with the new updates in this toolbox.

We also tested if these gap-filling algorithms are well-maintained which mean if they are up-to-date and/or have an active technical support community. Most of the algorithms published earlier are not maintained. They have the last commit date around the time their original papers were published. ProbAnnoWeb/ProbAnnoPy is the only one have active updates. SMILEY, FASTGAPFILL and Likelihood-based gap filling workflows although are not updated frequently but they have active communities for technical support.

In addition, namespace requirement is also an important criterium that decides the usuability of an algorithm. If there is mismatch in namespace between model and that gap-filling algorithm requires, users will need to translate the namespace in the model. This will imply errors as we indicated in our study [208]. Among tested gap-filling algorithms, FASTGAPFILL and DEF require the model to have metabolites and reactions in KEGG namespace. Likelihood-based gap filling workflows, EnsembleFBA, and ProbAnnoWeb/ProbAnnoPy require ModelSEED namespace. All other gap-filling algorithms do not require metabolite and reaction identifiers in the model to follow a specific format.

Algorithms	Year	Platform	Additional input	Namespace	Accessible	Well-documented	Well-maintained	Easy to install	Easy to run
OMNI	2006	MATLAB & GAMS & CPLEX	*In vitro* flux data	any	script			need GAMS	
SMILEY [313]	2006	GAMS & CPLEX or COBRApy	no	any	script				
GapFind/GapFill [308]	2007	GAMS platform	no	any	script			need GAMS	
GrowMatch	2009	-	*In vitro* gene essentiality data	any					
Christian et al	2009	-	-						
MIRAGE	2012	MATLAB	Enzyme phylogenetic & gene expression profile	any	script				
FASTGAPFILL	2014	MATLAB & CPLEX	no	KEGG	script				
Likelihood-based gap filling workflows	2014	API or command-line on KBase		ModelSEED	website				
GlobalFit	2016	CPLEX API, Sybil & R	Experimental growth and non-growth data	any	script				
DEF	2016	Webserver	-	KEGG	website				
BoostGAPFILL	2016	MATLAB	no	any	script				
SONEC	2016	MATLAB, R & Python	Metagenomic	any	script				
EnsembleFBA	2017	MATLAB	*In-vitro* gene essentiality data	ModelSEED	script				
Meneco	2017	Python	Growth medium & targets to produce	any	script				
Hybrid Metabolic Network Completion	2017	Python, C++, ASP, Rust, CLIPS	no	any					
GAUGE	2017	COBRA toolbox MATLAB, F2C2, & CPLEX	Gene expression data	any	script				
ProbAnnoWeb/ProbAnnoPy	2018	Web service (ProbAnnoWeb) and standalone python package (ProbAnnoPy)	no	ModelSEED	website & script				
OptFill	2020	GAMS, Perl, Python, & CPLEX	-	any	script			need GAMS	

Table 5.3.1: The usability assessment of gap-filling algorithms. Links to get codes for these algorithms are provided in the supplementary file. Blue - good, orange - moderate, red- bad. Blank - not available or not tested.

5.3.2 Performance evaluation

In order to evaluate the performance of gap-filling algorithms, we used an approach similar to the one described in [314] with some modification. Shortly, we started with a reference GEM that can grow on minimum medium with glucose as sole carbon source when simulated with flux balance analysis. In this GEM, we introduced artificial gaps by randomly removing 1, 5, 10, and 100 essential reactions, that is reactions whose deletion will prevent growth. *Escherichia coli* GEM, *i*ML1515 and *Saccharomyces cerevisiae* GEM, *i*MM904 were chosen to represent single and multi-compartment models. Degraded models were then subjected to gap-filling with the selected algorithms. The gap-filling sets of reactions suggested by the gap-filling algorithms were compared to the deletion set to calculate recall and precision. We repeated the experiment 100 times for each model and each algorithm.

The basic required inputs for all gap-filling algorithms are a subject for gap-filling, i.e. a GEM, a reaction database to draw gap-filling candidates from, i.e. BiGG [339], KEGG [130] or MetaCyc [340] and growth phenotype(s) depending on the intended purpose of modelling. Some algorithms also require extra inputs such as *in-vitro* flux data or *in-vitro* gene expression data (Table 5.3.1). To conduct performance evaluation of gap-filling algorithms, we only chose those that do not require in-house codes or heavy debugging and do not require *in- vitro* data as input since we intend them to be usable for all users. Of 18 algorithms only SMILEY, FASTGAPFILL and Meneco suite our requirements.

The overall accuracy and precision of the three algorithms fulfilling our usability criteria on *E. coli* and *S.cerevisiae* GEMS (iML1511 and iMM940 respectively) with deletions of sizes 1, 5, 10 and 100 are reported in Table 5.3.2.

Overall, all algorithms that could be tested perform better on the single compart-

Model	Deletion size	Mean addition size ± std	Running time (minutes)	Feasible (%)	Recall (%)	Precision (%)
SMILEY						
E. coli	1	1.04 ± 0.2	37.7 ± 32.9	85	73	70
	5	5.04 ± 0.2	184.6 ± 241.6	22	83	82
	10	-	-	0	-	-
	100	-	-	0	-	-
S. cerevisiae	1	1.03 ± 0.2	22 ± 11.1	38	70	70
	5	-	-	0	-	-
	10	-	-	0	-	-
	100	-	-	0	-	-
FASTGAPFILL						
E. coli	1	366.4 ± 75.6	20.8 ± 5.7 / 2.8 ± 1.0	96	0	0
	5	387.6 ± 15.8	30.9 ± 11.9/ 4.2 ± 2.3	100	0	0
	10	388.9 ± 21.1	28.8 ± 13.9 / 3.03 ± 1.6	100	0	0
	100	366.4 ± 75.6	20.8 ± 5.7/2.7± 1.0	96	0	0
S. cerevisiae	1	463.4 ± 14	41.1 ±12.3/ 503.1 ± 251.2	100	0	0
	5	479.3 ± 15.5	56.8 ± 20.7/ 541.8 ± 377.4	100	0	0
	10	479.3 ± 15.5	56.8 ± 20.7/541.8 ± 377.4	100	0	0
	100	562.6 ± 63.3	136.1 ±50.8/342.3 ±211.9	99	0	0
Meneco						
E. coli	1	2.0 ± 0.0	0.5 ± 0.1	100	0	0
	5	2.05 ± 0.2	0.9 ± 0.4	57	0	0
	10	2.16 ± 0.4	1.1 ± 0.3	26	0.4	0.19
	100	6.03 ± 1.5	1.5 ± 0.5	31	3	48
S. cerevisiae	1	-	-	0	-	-
	5	-	-	0	-	-
	10	1.0 ± 0.0	1.1 ± 0.1	3	0	0
	100	1.0 ± 0.0	1.0 ± 0.1	7	0	0

Table 5.3.2: Performance evaluation results. *Feasible* represents the number of experiments for which the gap-filling algorithm could find results. Recall and precision were calculated over the total feasible experiments. FASTGAPFILL includes two steps: gap-fill preparation and gap-fill. Running time for each step for this algorithm is listed in the table as time for preparation/ time for gap-fill.

ment model, *E. coli*, than on the multi-compartment model, *S. cerevisiae*. Although both models have a similar number of reactions (2712 in *E. coli* model and 1577 in *S. cerevisiae* model), the yeast model is more complex with 8 compartments whereas *E. coli* model only has 2 compartments. SMILEY had the highest precision and recall, more than 70 % of reactions were correctly recovered. It works best for small degree of network degradation. Meneco gave slightly better performance at large scale dele-

tions than the other algorithms, but overall precision and recall are very poor for all experiments. FASTGAPFILL did not recover any reactions that we removed. This is partly due to the inconsistencies in the namespace between the model and KEGG. Although the namespace was translated, there are various metabolites that cannot be mapped and still in KEGG namespace. It means FASTGAPFILL suggests reactions in different namespace, as a result they look different from our deletion set.

The solution size of FASTGAPFILL is several orders of magnitude larger than our removed reaction set and is comparable among different deletion sizes. SMILEY, on the other hand tends to give the same number of reactions for gap-filling as the size of the removed set of reactions. Finally, Meneco gives smaller solution size than removed set size.

On a remote server with a 2x Intel(R) Xeon(R) CPU E5-2650 v4, 256 G of random access memories operating under Ubuntu 16.04.6, Meneco is the fastest algorithm, its running time is in seconds. SMILEY took longer to complete a gap-fill cycle, its running time is in the order of minutes and proportional to the degradation level. Of the three algorithms tested, FASTGAPFILL is the one that needs the most time to fill gaps. It took hours for the models we tested, and the running time increased proportionally to the degradation level. The preparation times for FASTGAPFILL are independent of the deletion size. This step conducts the addition of all reactions from KEGG to the model, this is similar for all degradation degrees. FASTGAPFILL is slower than SMILEY and Meneco because it aims to solve all the gaps in the model, unlike Meneco and SMILEY whose objective is to only fill gaps that can rescue growth.

5.4 Discussion

In this study, we reviewed gap-filling algorithms and evaluated the performance of SMILEY, FASTGAPFILL and Meneco algorithms. Regarding recall and precision, SMILEY performs best for small-scale degradation. For highly degraded networks, all three algorithms have a marginal performance. These algorithms are implemented in different platforms and used different dependencies. In addition, each algorithm uses different reaction databases for candidate reactions, for instance, FASTGAPFILL use KEGG while SMILEY and Meneco uses user-defined database, in our study it is BiGG. Differences in performance may not arise from the algorithm themselves but from the implementation and database they use. We used models in BiGG namespace as subjects for gap-filling. Every reaction we removed from the BiGG models should be in the BiGG database, this could explain the higher recall and precision from SMILEY. From this perspective, KEGG simply worsens the signal to noise ratio of available reactions. We did not find high quality GEM in KEGG namespace to test FASTGAPFILL and our evaluation therefore can be biased. Similar studies also capture low precision and recall from FASTGAPFILL [314]. Although having low performance in accuracy metrics, candidate reactions proposed by FASTGAPFILL provide a multitude of potential directions for discovery. These reactions can represent the organism's potential metabolism that has not yet been experimentally confirmed. Of course, in many cases these reactions are just simply resulted from the mismatch in namespace between reference database and GEM for gap-fill. In either case, the objective reconciliation technique used by SMILEY and Meneco performed better at predictions of essential reactions needed for the desire phenotype, while the topological expansion technique used by FASTGAPFILL is more suitable for discovery of unknown metabolism.

Of 18 gap-filling algorithms, we only evaluated three due to the different required

inputs, implemented platforms, and availability of these tools. In other studies [314, 317, 305], also only a few gap-filling algorithms could be evaluated. Most of these gap-filling algorithms do not have workable implementations publicly available. When publishing an algorithm, authors should provide off-the-shelf codes to ensure reproducibility.

Writing and optimizing code to execute such algorithms are time consuming task and end-users of these algorithms should not be expected to make their own in-house codes to use them. In addition, an executable implementation is also useful to check if the algorithm solve what they claim prior to their publication. Unfortunately, it seems this is not yet a common practice in GEM community evident by the lack of workable implementation of the gap-filling algorithms that we reviewed. Similar trend was reported for other algorithms used in GEMs such as construction tools [341] and optimizing algorithms [342]. Beside having ready-to-use implementation, a golden rule should be established to test new algorithms before publishing them. Similar to tools used for dynamic models, these tools can be tested using the same standard model, database and platform [343]. This will prevent the bias when comparing their performance.

Gap-filling algorithms have certain limitations. Many algorithms try to find minimum set of reaction to complete the network without considering genomics evidence. These reactions are plausible hypotheses that still need to be validated / manually curated. In such cases where organism specific data such as fluxes and gene expression profile are available, it is better to use algorithms that can integrate them. In addition, smallest solution set make manual inspection easy but it is not necessary the most biological relevant solution. While algorithms that aim to solve as many gaps as possible can run into over-fitting problem. In addition, many algorithms impose time or size constraint on the solution set, they will stop searching after one or a

user-defined number of solution is/are found. This will limit the change to find biological relevant solution. Except for GrowMatch and GlobalFit, other gap-filling algorithms cannot address false positive simulation where the model predicts growth while the organism do not grow *in vitro*.

Gap-filling has not been used very extensively in the past [296], the situation remains the same 10 years later. This is shown by the lack of technical support community for these algorithms. This is due to the cumbersome of applying these algorithms such as the lack of executable implementation, unclear input data, and namespace inconsistencies between GEM for gap-fill and reference databases. Most of the tools we evaluated are still in prototype or beta version. In order to get the most from these gap-filling algorithms, it is necessary to optimize them.

Nowadays, the use of GEMs for metabolic network analysis becomes more popular [318], for example the increase interest in using GEMs for analyzing metabolomic data [344] and for constructing microbe community models [345, 346]. This bring together scientists from a diversity background. Usability is one of the most important factor determining software quality [347]. Regardless of differences in programming experience among users, a functionality and usability tool is still the most desirable.

Despite these limitation, one cannot deny the potential of these gap-filling algorithms in assisting us to fill in the metabolic puzzle. This is evidenced by the continuous growing number of publication for gap-filling algorithms since 2006. With a foreseeable explosion of organism specific data, those algorithms that do gap-fill accounting for genetic and metabolic properties such as GrowMatch and GAUGE will become handy. Pure network topology gap-filling algorithms also have their own strength such as lower computational cost, relatively easy to use and modify. Depend on what type of data and resource available and the namespace the model

is in, each of these gap-filling algorithms will have their own advantages.

5.5 Methods

5.5.1 Usability performance

We evaluated the usability of gap-filling tools based on 6 criteria: model namespace, accessible, well-documented, well-maintained, easy to install and easy to run.

- Model namespace. If the algorithm requires the model to be gap-filled to be in a certain namespace.
- Accessible. If there is command line interface, graphical user interface and/or web browsers to implement the algorithm
- Well-documented. If the algorithm comes with a clear documentation highlighting the required input format, the expected behavior of the functions, and how to configure and execute them.
- Well-maintained. If the algorithm is updated or has active technical support
- Easy to install. If it is tedious to get all the required dependencies for the algorithm
- Easy to run. If the algorithm can be executed without debugging using the example dataset from the publication.

5.5.2 Performance evaluation

Selection of algorithms and software environment

In this study we chose algorithms that are accessible, ease to use, no required custom-made input, i.e. metagenomics or gene expression data. Three algorithms were thus tested: FASTGAPFILL [309] implemented in COBRA Toolbox version 2.20.1 on MATLAB 2017b, SMILEY implemented in COBRApy as gapfiller in Python 3.5, and Meneco implemented as a Python package in Python 3.5. IBM CPLEX was used as solver for LP and MILP problems. These algorithms were run on a remote server with a 2x Intel(R) Xeon(R) CPU E5-2650 v4, 256 G of random access memories operating under Ubuntu 16.04.6.

Selection of models and reference databases

GEMs of *Escherichia coli* and *Saccharomyces cerevisiae* were chosen to represent single-compartment and multi-compartment models. BiGG was chosen as a model source because its models are manually curated and of sufficient quality to be used as references. In BiGG, iML1515 is the most comprehensive up-to-date model for *E.coli* with high accuracy for gene essentiality and cover wide range of carbon metabolism [29]. While iMM904 is among the most studied model for *S. cerevisiae* available in BiGG, it has been used as template to build GEMs for other organisms.

For FASTGAPFILL, we used the default reference database, KEGG and the default dictionary to translate BiGG to KEGG. For Meneco and SMILEY, universal reactions from BiGG was used as reference database.

Selection of metrics for algorithm evaluation

Follow [348] we also chose two metrics, recall and precision to evaluate algorithms performance

- Sensitivity or recall: the total number of correct positive results that were retrieved. $Recall = TP/(TP + FN)$ a high recall rate can help to reduce the potential of missing correct reactions
- Accuracy or precision: the number of positive results that actually are positive instances $Precision = TP/(TP + FP)$ the higher precision the more likely we will get the right reaction

Where

TP: the number of reaction that are identical with what has been removed.

FN: the number of reaction that has been removed but was not recovered

FP: the number of reaction that suggested from the algorithms but not in the removed list

Evaluation workflow

We applied the same *in-silico* experiments on all algorithms. The workflow is based on previous evaluations [305, 314, 333] with some modification. Shortly, we started with a reference GEM that can grow on default medium with glucose as sole carbon source when simulated with flux balance analysis. Original setup from the model were used, we did not impose extra constraints on them.

In the first step, we introduced artificial gaps in the model by deleting essential reaction(s) set A. Random deletions were made by choosing a specified number of randomly selected reactions from the organism's essential reaction pool (excluding

exchange, transport, orphan and non-essential reactions). Essential reactions are those whose removal lead to growth rate falls under 1e-06. We conducted experiments for 1, 5, 10 and 100 reaction deletions for 100 replicates each.

In the second step, we applied gap filling algorithms on the degraded model. The gap filling tool predicts a set of added reactions to fill the model; this set of reactions is the solution set, A'. Using the removed set A and the solution A', we calculated precision and recall.

Integer threshold for SMILEY was set to the default value, 1e-06 for *E. coli* model and 1e-09 for *S. cerevisae* model. The incompatible between the new updated version of COBRA Toolbox and FASTGAPFILL raised errors when running the code. Debugging is provided in the supplementary.

To escape infinite run from solving MILP problem, we impose time constraint on our experiment. If an experiment does not return gap-fill result within two weeks, we will consider it as infeasible. In addition, Meneco and FASTGAPFILL also return infeasible error when they do not find result for a test case.

6

General discussion

Nhung Pham

It has been 31 years since the first genome-scale constraint-based model (GEM) was introduced in 1989. Until 2019, 6239 organisms in all domains of life have had their GEMs constructed [29]. Of these, 180 organisms have well-curated GEMs that are used for different purposes [29]. GEMs have been applied successfully in many applications especially in guiding metabolic engineering and contextualizing 'Omics' data. However, these models also have limitations. The aim of this thesis is to deploy GEMs for microbial cell factories and evaluate their main technical limitations.

GEMs were deployed as a knowledge-base in **Chapters 2** and **3**. In **Chapter 2**, I constructed the GEM for *Cutaneotrischoporon oleaginosus* to study its lipid metabolism and its potential for biofuel production. In **Chapter 3**, I employed GEMs in the context of the design-built-test-learn cycle to design and rank pathways for chemical production in *Pseudomonas putida*.

While constructing and using GEMs in **Chapters 2** and **3**, two main problems have recurred. The first problem is the use of inconsistent namespaces among GEMs. In **Chapter 2**, a GEM was built following a scaffold-based approach. *Yarrow lipolytica* was chosen as a reference template and there were five published GEMs for this reference organism. It was difficult to compare the content of these GEMs because each had different identifiers for metabolites and reactions. The scope of the constructed GEM for *C. oleaginosus* could be further expanded if the model were to be incorporate information from all available GEMs for the reference organism. However, this task was prevented due to the use of different namespaces and the lack of efficient mapping methods to link these namespaces. In **Chapter 3**, two GEMs of *P. putida* were used to design and rank pathways for chemical production. Although the two GEMs describe the metabolism of the same microbe, comparisons between them were hampered by the lack of interoperability, in particular in their respective

namespaces. In addition, the pathway design tool, RetroPath2.0, used in **Chapter 3** requires InChI structures or keys for all metabolites in the target host. This task could not be fully achieved because iJP962 uses a custom-made namespace that cannot be fully mapped to InChI keys. In addition, only 56.9% of metabolites in iJN1411, the other *P. putida* model used in that chapter, could be linked to InChI identiers. The use of different namespaces in GEMs was a well-known problem in the community, yet how effective the mapping and the risk of introducing mismatch when translating namespaces had not been evaluated before. To that end I evaluated in **Chapter 4** the mapping efficiency among 11 databases commonly used for building GEMs.

The second problem that has emerged relates to the efficiency of gap-filling tools. GEM construction is an intensive and time-consuming process because manual work is required to curate and remove gaps in the network. Although many tools have been developed to assist in gap-filling, at that point I could not use any of these published tools due to three main reasons: many described tools were not available, tools that were available were not workable because of unclear documentation and/or operational errors, workable tools were not efficient or require models to be in a specific namespace. Therefore, gap-filling for the GEM in **Chapter 2** was manually conducted. The purpose of the pathway design in **Chapter 3** also aligns with that of the gap-filling algorithms as pathway design can be seen as a problem of restoring the connectivity in the network, yet none of these algorithms could be readily used for this task. Although some papers have noticed the lack of workable gap-filling algorithms [314], their performance had not been evaluated. To that end, in **Chapter 5** I made an extensive review and evaluation on the performance of stand-alone algorithms for gap-filling.

The findings of the individual works have been discussed in the previous chapters (**Chapter 2, 3, 4, 5**). Across these chapters, three significant themes stood out: 1)

the lack of standards in namespaces, tool development, and guidelines for model evaluation, 2) the need to improve models and computational tools, for instance to account for uncertainty in the biomass synthesis reaction or to improve gap-filling algorithms, and 3) the potential contribution of GEMs to the DBTL cycle. More in-depth discussion of each of these themes will be provided in this chapter.

6.1 The lack of standards in GEMs

6.1.1 The lack of standards in the namespaces used to develop GEMs

As I demonstrated in **Chapter 4**, GEMs built by different research groups or from different construction tools have their metabolites and reactions in different namespaces. I highlighted in the **Chapter 4** how an inconsistent namespace can lead to mismatches when mapping to another namespace. This is a well-known issue in the community, yet a unify standard is still lacking [246, 349, 350, 338].

Currently, GEMs do not have a common namespace because of three main reasons. First, different construction tools generate GEMs in different namespaces. For instance, GEMs constructed from the SEED will be in the SEED namespace [240], GEMs built using Pathway Tools [241] will be in the MetaCyc namespace [244] and GEMs from Raven [242] will be in the KEGG [245]. It is difficult to convert them to a common namespace as there is no efficient mapping system and not all metabolites can be accurately converted to one database. Currently, there are eleven databases frequently used for GEMs. Each of them has different coverage. For example, BiGG [328] is a curated albeit small database and does not cover all metabolites in the metabolism. KEGG is a non-curated database for more general biological processes [130]. MetaCyc is for experimentally validated pathways [340]. ChEBI is for small chemical compounds [276]. HMDB is for human metabolites [278]. LIPID MAPS

[279] and SwissLipids [282] mainly contains lipids. There are overlaps among these databases but not all of them are covered in each other. In **Chapter 4**, I demonstrated that we can only map a maximum of 60% of a database to any other database. Each database has their own advantages and coverage, therefore it is infeasible to decide on one standard namespace among these 11 databases for GEMs. This has also been noticed in the metabolomics community as there is currently no single database that can cover all metabolomes [344].

The second reason for GEMs not having a common namespace and sometimes even using a custom-made namespace is the presence of specific compounds such as tautomers and/or polymers that are not identified or complicated to identify in standard databases [351, 338].

The third reason for the lack of a common namespace is that the choice of namespace does not influence the prediction ability of GEMs. When GEMs are used within their intended scopes, the most important requirement for GEMs is to correctly reflect the intended objective of the modellers. As a result, previous published procedure focuses more on model performance rather than their namespace. This is reflected by the publication of GEMs in different namespace or even a majority in custom-made namespace [246]. Nowadays, with the increase demands on (i) building microbe community models [345, 346], consensus or community models [352, 353]; (ii) using existing GEMs as templates for constructing new GEMs [97]; (iii) facilitating the use of tools [338] and (iv) using GEMs as platform for metabolomics analysis [344], GEMs annotation and reusability get more attention and are recently stressed [350, 338].

In this thesis, I highlighted that the lack of standardization in namespace for GEMs is a real problem that needs to be addressed. This issue prevents the use of tools, for instance gap-filling tools presented in **Chapters 2 and 5**; It prevents the

comparison and combination of GEMs, for example to compare template GEMs in **Chapter 2** or GEMs for *P. putida* in **Chapter 3**; It prevents the mapping of parameters important to evaluate and to use GEMs such as InChI identifiers for pathway design in **Chapter 3** and mass, charge or metabolite formulas for checking stoichiometry consistency and mass balances.

At the moment, MetaNetX is a community bridge that has been developed to reconcile namespaces in GEMs [354]. However, I demonstrated in **Chapter 4** that this approach is not yet efficient. First, not all metabolites can be mapped from one namespace to the other via names or via MetaNetX identifiers. Second, there is a high degree of inconsistency when mapping between namespace due to the ambiguity of names and the multiplicity of identifiers. The findings from **Chapter 4** imply that we need a standard for namespaces and a more efficient mapping system.

The standard should be to have GEMs in a namespace that can be mapped. This would imply the use of identifiers from published databases and no custom-made namespace should be used. For specific metabolites that are not identified in public databases, new identifiers need to be assigned to them. In this context, the use of FAIR identifiers will help to reduce ambiguity. The FAIR principle aims to make data 'Findable', 'Accessible', 'Interoperable', and 'Reusable' [355]. A global unique identifier for each research object is the main foundation of FAIR [356]. This approach of making FAIR identifiers for metabolites has been adopted by the metabolomics community [344].

In order to improve the correctness of mapping, we need to get back to the root of the problem and answer the question about the reasons for the inconsistent mapping. Although previous studies have suggested that internal ambiguity is low and solving this will only solve part of the inconsistent problem [357], our findings in **Chapter 4** agree with earlier analysis on public databases [358] that the problem is

rooted in the internal inconsistency of each database. As shown in **Chapter 4**, the majority of identifiers link to multiple names in the same database. Multiplicity in this case increases human readability and is beneficial, as long as the alias, names, and synonyms describe the same metabolite. However, there are many cases on which alias referring to classes are incorrectly categorised as names. For instance, most of the compounds that have high ambiguity and multiplicity are general compounds such as triacylglyceride and compounds with many tautomers such as urate. These are compounds that have different side chains and/or structures in different organism and/or conditions. They are usually present in databases as spaceholders until more specific knowledge is obtained [21]. Generic names such as triacylglyceride are correct to describe the general storage lipid with no specific fatty acid chains. As I discussed in **Chapter 4**, general compounds are often used in GEMs. This does not affect the use of the model, except when predicting or simulating the production (of a specific component) of generic compounds, i.e., when 'lipids' are the main focus of the model. It is even desirable to use generic compounds in GEMs whenever a specific compound is not needed or not known since they can be universal. However, triacylglyceride should not be included as name for 1-Palmitoyl-2-palmitoleoyl-3-arachidonyl-glycerol (ChEBI:89764) where the three fatty acid chains are clearly indicated as C16:0, C16:1 and C20:0 since triacylglycerides with different fatty acid tails will have different molecular masses. In most of the databases I tested, these two compounds are linked together and the terms are considered synonyms.

Another example of the incorrect use of alias is the use of the Event and PhysicalEntity as synonyms in Reactome. These tags indicate the biological processes or pathways that metabolites participate in [359]. As indicated in **Chapter 4**, in Reactome it is possible to link H_2O to diphosphate via phys-ent-participant63109 or to pyruvate via phys-ent-participant60981. This happens because these metabolites

participate in the same reaction and hence, they share the same phys-ent-participant tags. However, in the database, these tags are classed as synonym for name of the metabolite. This creates incorrect links between these entities.

These are examples when links to wrong names will increase the probability of introducing mismatches. Since many databases integrate data from other databases, such internal errors will proliferate in the new database. Such errors were noticed to propagate from PubChem into other databases [358]. In agreement with [358], I suggest to pay attention to the removal of internal inconsistency to prevent such errors from amplifying.

Many databases have been improving their consistency. ChemSpider combined a roboticized cleansing approach and manual curation of their data by the curation team and users [360]. ChEBI has also made efforts to develop automatic tools such as a new text mining corpus to improve manual data curation [361]. In agreement with earlier studies [358, 357], our observations in **Chapter 2** also suggested that mapping without manual curation implies high risk of mismatch. Yet this is not the most efficient approach [357]. Detailed study has shown that manual curation has resolved only a small part of the inconsistency in ChemSpider [357]. As for ChEBI, our analysis in **Chapter 4** shows that this database still has inconsistencies. Given the high number of metabolites and the ambiguity of their names and structures in databases, it will take time until databases solve all their inconsistencies. A less optimistic view predicts that this task will hardly be achieved [351].

We cannot wait until public databases are fixed. Since no single database can be used as a source of standard identifiers for GEMs, we need to create a standard one for GEMs. Similar approach has been done in other communities such as the Chemical Validiation and Standardisation Platform of the Royal Society of Chemistry [351] or the MetaboLights database of the metabolomics community [344].

To prevent redundancy, BiGG can be improved to become the community standard for GEMs. We cannot yet use BiGG identifiers for all GEMs since the database coverage is not sufficient. However, BiGG is a good candidate as a standard namespace for GEMs in the future since BiGG identifiers are human-readable and the database has been specially curated for GEMs [330, 338]. Many high quality GEMs and newly published GEMs have been converting their models to this namespace [225, 362, 363, 349, 364, 365].

Beside non-systematic identifiers, structure identifiers should also be used to reduce ambiguity. These identifiers represent the chemical structures in a form usable by computers [266]. MOL, IUPAC name, SMILEs and InChI are examples of structure identifiers. Among them, InChI is considered as the most consistent source of identifiers [247, 266, 344].

In conclusion, to improve the reusability of GEMs, I propose to focus on three tasks. One task is to curate and expand the BiGG database with InChI identifiers. Since new GEMs have been published in different namespaces, without an efficient automatic mapping system, it is normal that the database has not yet covered all data in all published GEMs. BiGG has been recently updated [339]. Yet, the database only covers 108 GEMs, while there are at least 180 organisms with well-curated GEMs published until 2019 [29]. I highly suggest continuing to expand the database by introducing new published GEMs and new metabolites from these GEMs into the database. In this sense, the use of FAIR identifiers should be adopted in BiGG.

Another task is to continue curating MetaNetX. The MetaNetX has been considered as a community translation service [354, 302, 29] and has been used in community tools such as Memote [350] and RetroPath2.0 [179]. By improving upon the current standard, the MetaNetX, we can prevent the introduction of redundant systems. This database just has had a significant update by removing inconsistency and

including InChI identifiers for compounds (data retrieved on 26 Sep 2020). However, there are still inconsistencies in the database. For instance, urate from BiGG can be mapped to 19 identifiers in ChEBI via MNXM441 in MetaNetX (**Chapter 4**) (data was retrieved again on 26 Sep 2020). Many of these 19 identifiers refer to urate's tautomers instead of urate itself. Curation to eliminate incorrect links from such metabolites with high degree of ambiguity and multiplicity from MetaNetX can help to reduce the inconsistent mapping.

However, even for systematic identifiers such as those from InChI, there are still inconsistencies, they are just not as high as that in non-systematic identifiers [366]. This is due to the inconsistency and limitation in the way chemicals are described in structure files such as v2000 molfiles [367] and how these files are read by structure reading tools [366, 351]. As databases continuously integrate data from other sources and each database uses their own standardization rules to structure and store their data [351, 357], I agree with these authors that no matter how much effort has been spent and will be spent for data curation, there will always be inconsistencies that cannot be fixed.

Hence, another key task is to develop a more sophisticated mapping system that is not solely base on names, identifiers or even structure identifiers [351]. For GEMs, it should also be based on the context of the metabolites such as network topology and genes encoding associated reactions [250].

These propositions may not fully solve the inconsistent mapping problem. As many efforts have been spent on this topic, not only for metabolic databases but also other public databases such as databases for small molecules [358, 368, 357] and primary nucleotide databases [369]. The problem may be more complicated than what I suggested. We should not underestimate the influence of inconsistent namespaces on model's accuracy. A subtle mismatch can change model predictions and not all in-

correct matches can be identified in the model. As the current system is not enough to use as standard, I emphasize the need to develop standard and methods to reconcile the namespace in GEMs.

6.1.2 The lack of standards in tool development for GEMs

The development and application of GEM continues to grow rapidly and has been used in both fundamental and applied research relevant to biotechnology, microbiology and medicine. Its highly interdisciplinary nature brings together computational scientists, mathematicians, physicians and molecular biologists into the same field. Each field player brings different advantages for different tasks.

To improve GEMs and their uses, fluid and effective communication between the aforementioned classes of scientists is essential. There is often a knowledge gap between tool development and end users. Tool developers do not consider the variety background of the would be end-users. While end-users are often not aware of new tool developments [338]. As I have shown in **Chapter 5**, most of the tools to assist in gap-filling do not have workable implementation platforms and require advanced debugging skills prior to use. The end-users of these tools are often scientists who are more focused in analysing the metabolic processes rather than debugging or making computational tool. As a result, these tools end up being wasted without a user-base.

This situation of tools being difficult to use is not limited to gap-filling. GEM construction tools [341] and optimization tools [342] are also often poorly maintained, not user-friendly and often miss codes to implement the algorithm. This is a problem because unnecessary time needs to be spent in reinventing the wheel (by recoding the published algorithms) or to manually conduct a task that could be automated such as the gap-filling step in **Chapter 2**. Gap-filling in newly published GEMs in 2020 were also done either completely manually in [362–364, 370, 365] or partly

manually in combination with tools such as FASTGAPFILL [349].

Efforts to make computational tools more user friendly have been noticed [23, 371]. One of the community projects that aim to make tools available and user-friendly is the openCOBRA project (https://opencobra.github.io/). The platform is an open-source community repository for scripts and tools that have been published for metabolic models [23, 338]. In this platform all the tools come with documentation, tutorials and a forum for users to discuss, update and report bugs. The COBRA Toolbox was first developed in MATLAB [23] and later expanded to Python [371]. Nowadays, the project integrates with C, FORTRAN, Julia, Perl and Python code, as well as pre-compiled binary files [23]. However, the repository still mainly consists of tools in the COBRA Toolbox (MATLAB) and COBRApy (python) since these toolboxes are the most used in the community [338]. There are many more simulation tools that have been developed and are not in the repository. For instance, except for SMILEY, FASTGAPFILL, BoostGAPFILL and GAUGE, all other gap-filling algorithms tested in **Chapter 5** are not on the COBRA github, or the newly published tools such as FastMM [372], a toolbox to customize GEMs or GAPSPLIT [373], a sampling tool are distributed separately on individual githubs.

As more researchers have been using GEMs [318], it is urgent for the field to create and adhere to standards in publishing tools to make them accessible to everyone. To this end, beside functionality, tool development for GEMs needs to focus on the end-users and satisfy two important criteria: (i) accessibility, as tools need to be provided for public use; and (ii) user-friendliness, as tools need to be easy to install and well-documented.

In addition, GEM users are often clustered into small communities [338]. This makes it difficult for GEM users and tool developers to keep track of the introduction of new tools and advances in the field [338], because they have to integrate their so-

lutions and work on common code-bases. This would improve if all these tools were gathered into a common platform for GEMs where everyone can contribute. A good example for such a platform is the openCOBRA project. It will be a good practice for authors to deposit their codes in the same platform such as the COBRA github before it is published. A similar practice has already been implemented for sharing models before their papers are published. In the author submission guidelines in many journals from the EMBO press, Public Library of Science (PLoS), Royal Society of Chemistry (RSC), BioMed Central (BMC), ScienceDirect and FEBS Publishers model submission to BioModels have been included [374, 375, 338].

Furthermore, recoding published algorithms should be encouraged. The reward system of openCOBRA project is an excellent example. In this project, contribution on tool refinement is rewarded as the co-author in the publication on the project update [23]. This is a good example of community contribution to make tools available. However, this is not widely known, encouragement to contribute on tool refinements need to spread wider to raise awareness.

6.1.3 A guideline to evaluate the correctness of a GEM

The diversity of tools and resources for GEM construction resulted in the differences among model representations suggesting an assorted variety in model quality. There is thus a need for a concrete answer on how to evaluate the correctness of a GEM. First and foremost, GEMs are just models. General criteria that make good models can therefore be applied to GEMs, as follows:

- Traceable. GEMs are primarily based on genome-derived functional annotations and on biochemical knowledge of the target organism. Deviations between model predictions and *in-vitro* observations can result from errors and missing or outdated information in the model. Hence, GEMs

are updated continuously by adopting up-to-date experimental information on gene-protein-reaction associations and cell growth under genetic or environmental perturbations. They are also updated by solving inconsistencies such as incorrect functional genome annotations and different database identifiers for the same metabolite. As a good practice for biological models, high-quality GEMs should reveal data provenance and reason behind these updates [376].

- Truthful. Any entities, i.e. reactions and metabolites, in GEMs need to have valid evidence or strong arguments for their presence in the model [21].
- Simple. Modelling is used to simplify the complexity of reality, it is more flexible and easier to understand and implement when the model is simpler [377]. Hence GEMs should also be simple enough to only cover the necessary networks that give rise to the objective function.
- Flexible. This criterion is to grade how easy the model can be adapted for other purposes [377]. GEMs are organism specific and often built for a specific purpose but they can serve as good reference templates for constructing GEMs of other related organisms [97] or for simulating different objectives [378]. Hence, GEMs should be able to allow the expansion of the metabolic network to cover other objective function if desired.

Various tools such as Memote [350], Gsmodutils [379], and BiGG [380] have been published to assist in GEM quality assurance. Among them, Memote is considered as the community standard for evaluating the model completeness [225, 338, 363]. The tool is a framework where new features are added and curated by the community. The tool also provides version control system where data provenance is recorded. It aims to promote a standard model annotation such as to have

identifiers for metabolites and reactions in the same way as they are represented in standard databases such as BiGG, KEGG and ChEBI. It also requires the GEM to have database-independent identifiers such as InChI for metabolites [350].

Memote grading system focuses on two main tests: model annotation and model performance. In the model annotation test, Memote grades the stoichiometry consistency, mass-charge balance of the reactions and the use of standard identifiers for metabolites, reactions and genes. In the model performance test, Memote performs basic simulations to validate biomass production and energy generation. In these tests, Memote translates the model namespace based on an internal mapping via MetaNetX [354] to calculate features such as mass balance and to recognize the reactions by keywords such as 'biomass' or 'ATP'.

Memote is a good example of a community effort toward standard. The use of Memote will help to normalize the consistency among models because it encourages modellers to use standard namespaces. Memote also provide version control and provenance. This will help to keep track of model modification. In addition, Memote provides an easy to use platform to quickly check model quality. This is good in assisting paper reviewing process. Memote should be encouraged to use to couple with the publication of new GEMs. Some new published GEMs have included the test in their models [364, 225].

However, Memote depends a lot on the mapping of model namespace via MetaNetX for their tests such as to retrieve mass and charge information, to find biomass, energy and nutrient uptake reactions. Existing models that are not in the standard namespace cannot be correctly evaluated. Such examples are models for *P. putida*. iJP962 [224] is considered a good model for *P. putida* because it contains a consistent stoichiometry, correct growth rates, and complete respiratory chain that lead to a reasonable ATP production rate [381]. However, iJP962 only scores a to-

tal of 37% on Memote due to the use of custom-made identifiers for metabolites and reactions. Most of the performance tests therefore were not able to carry out. The same happen to iJN1411, another model of *P. putida* [382]. The model predicts correctly nutrient sources, growth rates, flux distribution and gene essentially [382]. The model was useful to indicate the metabolic rearrangements of *P. putida* when changing the carbon sources in the growth medium [383]. However, the lack of standardize identifiers for all genes, reactions and metabolites make it scores only 40 % on Memote. Its recent expansion version, iJN1462, which includes more links to external databases has much higher score, a 91% on Memote [225].

At the moment most of the published GEMs do not include all identifiers from databases recommended in Memote. Or even when they use standard namespaces, as shown in **Chapter 4** mapping between databases and via MetaNetX implies high risk of mismatch due to the ambiguity of names and the multiplicity of identifiers. Until a reliable mapping system is introduced, many of the tests on Memote may not be feasible or correct.

In addition, Memote still needs to improve to include more tests on performance. A good model annotation is necessary, but model performance is much more important. In the end, the purpose of a model is to correctly describe and predict a phenomenon.

For models that cannot be tested in Memote, I propose a short checklist to aid decision making in whether one should use them or not (text box 6.1.3). I have identified the minimal requirements after analysing template models for constructing a GEM for *C. oleaginous* in **Chapter 2**. I suggest to include these tests in Memote to get a more complete evaluation system.

The checklist for existing GEMs

Model format (one hour)

- ☐ Reaction formulation
- ☐ Biomass
- ☐ Reaction boundaries
- ☐ GAM and NGAM

Model performance (one day)

- ☐ Basic FBA tests (growth on known carbon sources and conditions)
- ☐ No growth in the absence of carbon, nitrogen, energy sources and/or other essential nutrient sources
- ☐ No matter production in the absence of carbon, nitrogen, energy sources and/or other essential nutrient sources
- ☐ No energy production in the absence of electron donors and/ or electron acceptors
- ☐ Reasonable ATP yield with 1 mol carbon sources

The checklist focuses on the model format and performance.
Model format

- Reaction formulation (especially for macromolecules such as lipid and protein). Natural fatty acids often have an unbranched and even-numbered

chain of 4 to 28 carbon atoms [384]. Fatty acid compositions can vary between organisms and between conditions but they often consist of a diversity of carbon lengths. Proteins are macro-molecules composed of amino acid residues. However, in some GEMs, proteins and lipids are formulated by using a generic end product (i.e. glycine → protein or oleic acid → fatty acid). This simplification is introduced when the modeller only intends to represent the presence of protein or lipid in the biomass. In this case, the overall energy and carbon used in these pathways needs to be included. However, this approach is not recommended because it is easy to introduce errors regarding the overall energy and carbon used in the whole pathway. In addition, in some models, H_2O or hydrogen are sometimes omitted from the reactions. Proton gradient is important for energy synthesis in the cell [385, 386]. Missing hydrogen or H_2O in the reaction can result in inconsistent stoichiometry [387]. Their presence is therefore important in GEMs.

- Biomass. The biomass synthesis reaction in GEMs is an artificial reaction that contains mmol components required to make up one gram of dry cell weight. In many models, the stoichiometry is, however, not correct. An assessment carried out in 2017 tested the biomass in 64 models [100]. 20 out of 64 models tested, have all the components in the biomass make up more than 20% deviation from 1g, with the biggest outliers are 0.62 gram and 1.44 gram biomass in models for *B. thetaiotaomicron* and *E. rectale*, respectively [100]. The incorrect biomass has shown to have significant impact on the quantitative simulation results [100]. The coefficients represent the mmol of each component in the biomass reaction. The biomass reaction is correct if the sum of molecular masses of each reactant in this reaction weighted by their coefficients in mmol is equal to 1g. Metabolites that represents the hydrolysis of energy for

biomass synthesis such as ATP and H2O in the biomass reaction are excluded from the calculation.

- Reaction boundaries. In GEMs, exchange reactions have special meaning. Their lower and upper bounds indicate the possible uptake and secretion of certain metabolites, respectively [21]. Currently, as long as it is consistent in one model, it is not important whether the lower bound represents uptake secretion and upper bound represents secretion or vice versa. However, to facilitate the use of automatic tools, and to fit with model formulation where if a metabolite is consumed it will have negative coefficient, and positive coefficient if it is produced, lower bounds should represent the uptake of metabolites and upper bounds represent the secretion of metabolites.

- GAM and NGAM. In GEMs, Growth associated maintenance (GAM) represents the energy that is spent to carry on activities related to growth such as protein synthesis [21]. Non-growth associated maintenance (NGAM) represents the energy that is spent for activities that are unrelated to growth such as flagella moving [21]. GAM and NGAM depend on the organism and the simulated conditions. In GEMs, GAM is integrated into the biomass synthesis reaction while NGAM is modelled as an ATP hydrolyis reaction and the NGAM value is assigned to the lower bound of this reaction. For instance, the lower bound of the ATP hydrolysis in *E. coli* and *P. putida* models are 8.39 [388] and 3.96 $mmol\ ATP \cdot gDCW^{-1} \cdot h^{-1}$ [389], respectively, to represent their corresponding NGAM values. However, some models fail to capture this boundary. As a result, these models predict too high grow rates. Often, energy spent for non-growth activities is low. These data are determined from experimental data. In many cases where experimental data on maintenance is

not available, data from closely related species [363] or an approximate estimation value would be assigned for NGAM [390].

Model performance

- Basic FBA test. As a validation of the model, a good GEM should describe basic phenotype tests for the organism in question. The most basic test is growth on different carbon sources.

- No growth in the absence of carbon, nitrogen, energy and /or other essential nutrient sources such as sulfur and phosphate. All living organisms require more than just water and oxygen to survive and grow [391, 392]. A good GEM should not grow when the necessary carbon, nitrogen, energy, and/or essential nutrient sources are missing. In order to test this problem, constrain all exchange reactions to allow no uptake and optimize for growth. If a (non-zero) solution to this problem exists, the suggestion is that it contain a thermodynamic inconsistency, which can be dealt with by manually inspecting the solution space and curating one or more reactions that carry fluxes.

- No matter production in the absence of carbon, nitrogen, energy, and/or other essential nutrient sources. Similar to growth, all organisms require some form of carbon and other essential nutrient sources in order to synthesize matter [392]. A good GEM should not produce matter without resources. In order to test this problem, constrain all exchange reactions to allow no uptake and optimize for each exchange reaction that represents matter secretions. If a (non-zero) solution to this problem exists, the suggestion is that it contain a thermodynamic inconsistency, which can be dealt with by manually inspecting the solution space and curating one or more reactions that carry fluxes.

- No energy production in the absence of electron donors and/or electron acceptors. To generate energy, organisms transfer electrons from an electron donor such as glucose or light, to an electron acceptor such as oxygen or nitrate [393]. A good GEM should not produce energy without these resources. In order to test this problem, constrain all exchange reactions to allow no uptake and optimize for ATP synthesis reaction. If a (non-zero) solution to this problem exists, the suggestion is that it contain a thermodynamic inconsistency, which can be dealt with by manually inspecting the solution space and curating one or more reactions that carry fluxes.

- Reasonable ATP yield with 1 mol of carbon sources. Theoretically, in respiration organisms, one molecule of glucose will yield from 30-32 molecules of ATP [394]. During fermentation, the yield of ATP per mol of glucose is 2 mol [394]. These numbers can vary according to the environment and organism. A good GEM should not produce higher or lower than these thresholds when testing for ATP production on glucose. A deviation from these values indicates infeasible thermodynamic loops that need to be curated.

6.2 The need to improve models and computational tools

6.2.1 Important features for gap-filling algorithms

Besides accessibility and user-friendliness, functionality is, obviously and by default, the most important criterion for any tool, including gap-filling algorithms. Among the 18 gap-filling algorithms that I tested in **Chapter 5**, SMILEY [313], FASTGAPFILL [320] and Meneco [317] are accessible and user-friendly. However, as shown in **Chapter 5**, they performed poorly on highly degraded networks. They address gap-filling solely based on network topology. SMILEY and Meneco only look for

the shortest path to restore growth. Longer pathways with more biological relevance maybe neglected in this approach. In addition, these algorithms search for the gap-filling reactions in a non-random manner in the database and stop when a solution is found. This approach limits the diversity of the found solutions.

A good gap-filling algorithm for constructing GEMs should allow the identification of the most biological suitable candidates. This means the suggested reaction for gap-filling should be likely to occur in the target organism. In order to find such candidates, gap-filling algorithms should not base solely on network topology. Some algorithms have already tried to rank reactions in the reference database based on functional genomics analysis. For example, likelihood-based gap filling workflows [315] and ProbAnnoWeb/ProbAnnoPy [316] base on sequence homology to predict alternative function of genes for gap-filling candidates. MIRAGE [311] uses functional genomics data and enzymes' phylogenetic profiles to calculate the probability of adding a reaction from a reference database into the model.

Gap-filling algorithms should also consider the mass conservation at steady-state assumption of GEMs. This means to assure a stoichiometric balance for the consumption and production of metabolites in the network. Hybrid Metabolic Network Completion (2017) [323] has covered this problem. BoostGAPFILL [314] has also applied flux constraints in the third mode of its action.

Furthermore, the more alternative solutions for gap-filling, the higher the probability to find a suitable candidate to restore the network. Hence, the tool should allow random search in the reference database in order to identify more alternative combinations. In addition, the solution size for gap-filling should not be constrained. Most of the gap-filling algorithms try to find the minimum number of reactions to restore growth while the shortest pathways do not necessarily mean the most biological relevant pathways. Having gap-filling solutions with different sizes increases

the chance to get the most biological suitable solution.

In conclusion, the best gap-filling algorithms should: do gap-filling beyond network topology; cover stoichiometric balance; consider extra constraint on genetic evidence; allow random search for candidate reactions from reference database and do not constrain the size of the candidate sets. Currently, there is no algorithm with all these features.

6.2.2 Improved biomass formulation

In **Chapter 2**, I demonstrated that model predictions are sensitive to changes in the coefficient of components in the biomass reaction. We need to account for uncertainty in this reaction because the biomass synthesis reaction is often used as objective function and is one of the most important elements in the model. It determines the scope of a GEM. This function is a reaction that consumes all the building blocks and energy needed to make a new cell in a fixed experimentally determined ratio to represent growth [21, 26]. During their lifetime, organisms adjust their biomass composition depending on how they interact with their environment [395]. As I demonstrated in **Chapter 2**, the C/N ratio in the medium greatly impacts the composition of *Cutaneotrichosporon oleaginous*. At high C/N ratio, lipids can make up to 80 % of the biomass, while during growth at low C/N ratio, the same lipids only take 20 % of the biomass. For such large changes, the biomass synthesis reaction needs to be flexible to reflect the fluctuation in such storage components. In GEMs, the biomass at each condition is represented by modifying the stoichiometric coefficient of each component in the biomass synthesis reaction. For instance, in *Chlamydomonas reinhardtii* GEM, iRC1080, three biomass synthesis reactions with different stoichiometric coefficients for each components were included to represent biomass in photoautotrophic, heterotrophic, and mixotrophic cultivation [396]. Growth

predictions from GEMs have shown to be sensitive to the stoichiometry coefficients of each component in the biomass synthesis reaction [397]. In cases where the perturbations have great impact on biomass composition such as the overflow of storage metabolites, i.e. lipid and starch, constant stoichiometry coefficients in biomass synthesis reaction will influence the accuracy of a simulation.

As I demonstrated in **Chapter 2**, I introduced a new way to formulate biomass reactions for each C/N ratio under nitrogen depletion conditions with uptake rates of the carbon source and nitrogen source as inputs only. This has shown more accurate simulation results for lipid formulation in such conditions. The biomass synthesis reaction was constructed based on available experimental data on lipid production at different C/N ratios. The function can be customized for other organisms given that suitable experimental data are also available. However, obtaining biomass composition for each and every change is impractical, a more feasible approach to account for the fluctuation of biomass composition over time is to introduce uncertainty in the biomass synthesis reaction.

Integrating uncertainty into the biomass synthesis reaction

Living cells consist of about 70 % water and 30 % chemicals [398]. In GEMs, the biomass synthesis reaction represents 1 gram of dry cell weight [21]. Although water is also included in this reaction, it represents the hydrolysis of ATP to generate energy that needed for synthesize new cell [21]. The biomass synthesis reaction in GEMs only represent the chemical part of the cell. The main chemical elements in any living organism are C, H, O, N, S and P [399]. They are represented in the form of four main macromolecules lipids, carbohydrates, proteins and nucleotides. A dry biomass will contain these macromolecules and micromolecules such as vitamins and minerals with a specific ratio.

Since metabolites that represent growth associated maintenance (GAM) such as ATP and H2O are not part of the 1 gram of dry cell weight, their coefficients are included in the biomass equation as the value of GAM [21]. In GEMs, this biomass is represented as follows:

$$\begin{aligned} Biomass + GAM \cdot ADP + GAM \cdot Pi \leftarrow \\ a \cdot P + b \cdot C + c \cdot L + d \cdot RNA + e \cdot DNA + \\ + f \cdot Others + GAM \cdot ATP + GAM \cdot H_2O, \end{aligned} \tag{6.1}$$

where P, C, L, and Others represent the protein, carbohydrate, lipid, and other mineral and micromolecules fraction, respectively and a, b, c, d, e, and f represent their corresponding coefficients.

The accuracy of coefficients in the biomass synthesis reactions have shown to affect the FBA solutions [400, 101, 401]. A robust analysis of metabolic pathways (RAMP) has been introduced as an alternative to standard FBA to account for uncertainty in the biomass synthesis reaction [402]. In this approach, the steady state assumption is replaced with the probability constraints that cover means and standard errors of the *in-vitro* data that are used to calculate the biomass composition. The approach allows to account for at least 0.42 % of the uncertainty. When applying on *E. coli* GEM, this approach has been shown to be significantly more consistent with experimentally determined fluxes for both aerobic and anaerobic conditions than standard FBA [402].

The stochastic model employing in RAMP describes the random events in a statistical distribution such as measurement errors [403, 404]. This is suitable for counting for the uncertainty in the measurement of biomass composition in the same condition. However, when growth condition is altered, i.e. changing C/N or P/N ratio

in the medium, changes in cellular components such as the overflow of storage components are not random and can be higher than probability uncertainty. In this case, these changes cannot be associated with probability laws [404]. To account for non-probabilistic uncertainty, each uncertain coefficient can be associated with an interval number [405]. Given that each component in the biomass has a minimum and maximum value. The biomass equation 6.2 can be written in a more general form:

$$\begin{aligned} Biomass + GAM \cdot ADP + GAM \cdot Pi \leftarrow \\ a' \cdot P + b' \cdot C + c' \cdot L + d' \cdot RNA + e' \cdot DNA + \\ + f' \cdot Others + GAM \cdot ATP + GAM \cdot H_2O, \end{aligned} \tag{6.2}$$

where a', b', c', d', d', e', and f' are coefficients defined through intervals $a' \in [a_{min}, a_{max}]$, $b' \in [b_{min}, b_{max}]$, $c' \in [c_{min}, c_{max}]$, $d' \in [d_{min}, d_{max}]$, $e' \in [e_{min}, e_{max}]$, and $f' \in [f_{min}, f_{max}]$.

The optimisation problem can be rewritten as:

$$\begin{aligned} \text{Maximize} \quad & \vec{\gamma} \cdot \vec{x} \\ \text{Subject to} \quad & S \cdot \vec{x} = \vec{o} \\ & \vec{lb} \leq \vec{x} \leq \vec{ub} \end{aligned}$$

Where S is an $m \times n$ stoichiometric matrix that contains coefficients of the metabolites in the row and the reactions they participate in in the column, $\vec{x}$ is the flux vector, $\vec{o}$ is a null vector ensuring steady-state, and $\vec{lb}$ and $\vec{ub}$ are the lower/upper bounds for each reaction. $\vec{\gamma}$ is the possibilistic variable restricted by the following set of n-row vectors:

$$F = \{\vec{c} = (c_1, c_2, \ldots, c_n) | l_i \leq c_i \leq u_i, i = 1, 2, \ldots, n\}$$

Where F is the set of objective function coefficient vectors, $\vec{c} = (c_1, c_2, \ldots c_n)$, whose $i-th$ component is in the interval $[l_i, u_i]$ and represents the possible range of γ. When the problem is posed in this way, it can be seen as a linear programming problem with interval objective function coefficients [405, 406]. This interval linear program has been applied to solve problems in other fields such as portfolio selection [407, 408], resources and environmental systems management [409], management of municipal solid waste [410], or chemical engineering problems [411]. Similar algorithms can be developed as an alternative to standard FBA for simulating GEMs. This technique can be used to find the best and the worst optimum and the coefficients that achieves them [412, 405, 413]. In addition, where *in-vitro* growth data is available it can be used to find the coefficients that yield the corresponding growth rate. This is useful to determine cellular composition when data is not available. Beside biomass synthesis reaction, the interval coefficient reactions can also be applied on other reactions with uncertain coefficients such as lipid or protein synthesis reactions to calculate their compositions.

6.3 GEMs in the DBTL cycles

6.3.1 GEMs in the design phase

I demonstrated in **Chapter 3** how GEMs can assist the pathway design phase in DBTL cycles applied to bioengineering projects. Two GEMs of *P. putida* were successfully employed to design novel production pathways for 5 chemicals. In this context, GEMs provided the known metabolic pool of the target organism for pathway design and were used to compute theoretical yields from newly predicted pathways for pathway ranking.

The aim of biosynthesis pathway design is to connect the metabolic pools of the

target organism with that of other organisms in order to introduce new ability into the target organism. This is similar to restoring the network connectivity in gap-filling algorithms that I have described in **Chapter 5**. The difference is that in gap-filling procedures, an additional curation step is required to remove heterologous reactions/pathways from the solution. Hence gap-filling algorithms can also be used to design pathways. However, as I demonstrated in **Chapter 5**, most of the gap-filling algorithms cannot be used due to the lack of workable implementations. Algorithms that have the implementations available such as SMILEY, FASTGAPFILL and Meneco are not good for filling long pathways.

To that end, gap-filling algorithms were not used in **Chapter 3**. Instead, RetroPath2.0 [179], a retrosynthesis algorithm for pathway design was employed. RetroPath2.0 was chosen for pathway design due to its ability to apply retrosynthesis rules. In this way, new reactions and thereby pathways are predicted based on possible enzyme promiscuity. This increases the possibility to identify novel heterologous pathways for less-studied or non natural chemicals or to find new options for well-studied chemicals.

At the moment, pathway design and ranking using RetroPath and GEMs is a semi-automatic task. Metabolites in predicted pathways need to be manually mapped to the GEM and reactions need to be curated to remove reactions that only seem to produce H_2O and hydrogen. A complete automated pathway design can be achieved if the GEM provides the InChI structure for metabolites that can be mapped and RetroPath produces an outcome in a similar format as GEM, i.e. a stoichiometric matrix with reactions and metabolites as entities. Pathways from RetroPath can then be added to the GEM and theoretical maximum yields can be iteratively computed. With the current computational power and an automated process, it should take only a few hours to design and rank all possible production pathways for a chemical in a

target organism.

Beside pathway design and ranking, GEMs can also be used for other tasks in the design phase. GEMs can be used for medium selection and optimization. For instance, a GEM of the green alga *Chlorella vulgaris* was used to predict minimal glucose and nitrate feeding rates. The model-driven feeding strategy improved 61% of fatty acid methyl ester production and the lutein yield by 3-fold higher [414].

In addition, GEMs can also be used to predict genetic modification to improve production performance. For example, to enhance the production of aromatic polymers in *E. coli*, GEMs predicted to remove tyrosine and aspartate aminotransferase genes from the previous modification strain. This strategy was employed and increased the production of D-phenyllactic acid to 4.35-fold higher [415]. Another example is the use of *Yarrow lipolytica* GEM to identify candidates for overexpression, knockout, and cofactor modification to increase 48 % of flux to the production of the industrial relevant dodecanedioic acid [129].

Using GEMs in the design phase helps to reduce the required manpower and experiments needed to test all the possible designs. Only the most potential *in- silico* designs will be executed experimentally [416]. Despite many limitations in the accuracy of simulations from GEMs, the predictive power of GEMs is certainly useful to narrow down the vast search space of possible outcomes.

6.3.2 GEMs in the learn phase

Many metabolic engineering processes are still based on empirical results due to the insufficient insight on the intracellular processes [417, 418]. The learn phase in the DBTL cycles aims at generating fundamental understanding of these biophenomena to improve the synthetic strain and/ or to identify bottlenecks [419, 420, 53]. In the learning processes for synthetic bioproduction, statistical tools and

machine learning are used to translate data obtained in the test phase into general knowledge to either improve product titer or expand to new products [421]. Recent studies have mentioned the use of metabolic models in the learn phase [422, 421], yet their uses are not as common as in the design phase [420, 423]. This is partly because the learn phase is currently the weakest step in the DBTL cycle and learning methods still need to be improved [420, 421]. In my view, GEMs can also be an important tool in the learning process. One of the main purposes of GEMs is to study the metabolic processes in the target organism. This is consistent with the purpose of the learn phase in the DBTL cycle in metabolic engineering.

One of the goals in the learn phase is to understand the regulation of metabolic processes by analysing data from the test phase [420]. In this context, sampling flux distributions from GEMs can be a powerful tool to discover such regulation [424]. For example, the comparison between flux changes in *Saccharomyces cerevisiae* GEM obtained from random sampling and transcriptomics data obtained from growth on four different carbon sources and in five deletion mutants reveals new transcription factors that has not been reported before [137]. In **Chapter 2**, I constructed the GEM for *C. oleaginosus* and used it to study the lipid metabolism in the fungus. The model serves as a knowledge-base to further explore the potential of *C. oleaginosus* as a cell factory for biofuel production. The model was used to study transcriptomic data obtained when the cell is cultivated in high and low C/N ratio media. It highlights that lipid production in these two conditions is not regulated at transcriptional level.

Furthermore, experimental data such as 'omics' data and parameters obtained in the previous production process can be integrated into GEMs in order to identify possible bottlenecks impacting cell performances during fermentation. For instance, GEM of CHO, the industrial cell line for biopharmaceutical products was

constrained by exometabolomics data obtained in a fed-batch culture [417]. The constrained GEM was then used to study the intracellular activities of the cell. The analysis provides insight on how toxic by-products such as ammonium accumulate in the cell line and suggests solution to reduce this accumulation, for instance by reducing asparagine in the medium [417].

In the learn phase, GEMs are good platforms to integrate learning data. Modifying these models with such data can help to identify and predict bottleneck to improve the next cycle.

6.4 GEMs as a platform to get closer to the comprehensive metabolic map

Constructing a comprehensive map of metabolism could be comparable to constructing the world map (Figure 6.4.1). The earliest maps of the world were very simple and sometimes incorrect. Still, this inspired adventurers to go and further explore new places. Currently, earth maps have such a high level of detail that almost every small alley in the world is covered. World maps nowadays also provide an interactive mode and even capture real-time data about traffic on the roads. The first map creators could never have imagined how maps have evolved today.

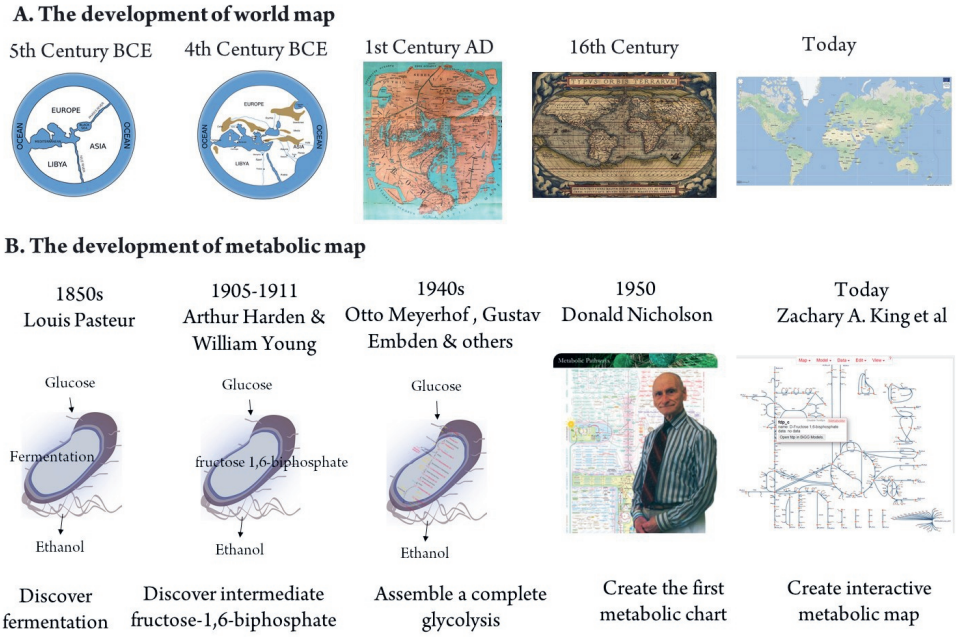


Figure 6.4.1: A comparison of the world and metabolic map developments. Pictures of world maps were obtained from https://www.amusingplanet.com/2012/11/coming-of-age-in-cartography-evolution.html on 13rd August 2020.

Similarly, metabolic maps can be developed in the same manner. Metabolism has been studied since the thirteenth century [425]. Yet only until 1940, the first metabolic pathway depicting glycolysis, an essential pathway of all organisms to convert glucose to energy [426, 427] was assembled. This was the result of a collection of findings from many experiments in almost 100 years from Gustav Embden (1874–1933), Arthur Harden (1865–1940), Karl Lohman (1898–1978), Otto Fritz Meyerhof (1841–1951), Jakob Karol Parnas (1884–1949), and Otto Heinrich Warburg (1883–1970) [426, 427]. Since then, many other important pathways have been

discovered. The first comprehensive metabolic pathway chart was hand-drawn with stencils on tracing paper by Nicholson Donald in 1955 [426]. The first printed map was then created in 1960 indicating the integration of amino acids, carbohydrates, lipids and other pathways [426]. This early map of metabolism is quite complete, including all central pathways and ATP synthesis pathways with cofactors, regulation, compartmentalization and other features.

Since then, scientists have been studying more pathways and more dynamic aspects of metabolic processes. Nowadays, the metabolic map can even allow simulations. The current map also allows to zoom out for the overview of the whole network or to zoom in for the details on each of the component. For instance, the Escher map is running on GEM foundation [428]. This map allows users to find the route between metabolites and to predict what happens when modifying the network.

GEMs can serve as an early interactive metabolic map. Deviations from GEM predictions and observations are excellent hinting tools to guide the focus of the research. This new knowledge can then be integrated into the GEM and give more details on other metabolic processes. In the quest for the comprehensive metabolic map, GEMs will serve as a platform to integrate data, simulate metabolic modifications and query information.

7

Summary

Nhung Pham

Advances in genome sequencing and high-throughput technologies have boosted the development of Synthetic biology and Systems biology. Synthetic biology aims to create and reprogram natural systems. Advances in Synthetic biology has facilitated the adoption of the Design-Build-Test-Learn cycles into metabolic engineering. The DBTL cycles are a recursive loop that aims to optimize the development of microbial factories in a more systematic and efficient manner. Systems biology aims to study living organism at system level using holistic approaches. Among different modelling tools in Systems biology, genome-scale, constraint-based metabolic modeling is the most successful approach to study the whole metabolic network. GEM is a comprehensive knowledge base that contains all metabolic reactions that known to occur in a target organism. GEMs have been used in many applications to guide metabolic engineering and contextualizing 'omics' data. The objective of this thesis is to deploy GEMs for microbial cell factories and to evaluate some of their main technical limitations.

Chapter 1 addresses reductionism and holism in life sciences, Systems biology, genome-scale constraint-based metabolic models, Synthetic biology and the design-build-test-learn cycle. **Chapter 1** provides the background for all other chapters.

In **Chapter 2** I constructed a GEM for *Cutaneotrichosporon oleaginosus* to model its lipid production under a variety of conditions. *C. oleaginosus* is a fast-growing oleaginous yeast that can grow in a wide range of low-cost carbon sources. I constructed a GEM to increase our understanding of this yeast and provide a knowledge base for further industrial use. A new modelling approach was introduced to account for changes in the biomass composition of this organism in conditions with high carbon to nitrogen (C/N) ratio in the media. This modelling approach accurately predicted high lipid accumulation using glucose, fructose, sucrose, xylose, and glycerol as sole carbon source. The model also suggests ATP-citrate lyase as a possi-

ble target to further improve lipid production.

Producing chemicals from living cells has been considering a sustainable approach towards a shift from petrochemical-based industries . The biosynthesis of many natural compounds is still limited due to the lack of efficient synthesis routes that may eventually render such a process economically viable. As a showcase of how GEMs can assist in designing pathways for chemical production in microbes, in **Chapter 3** I employed GEMs to design and evaluate pathways for cis,cis-muconic acids, anisole, aniline, 3-methylmalate, and geranic acid production in *Pseudomonas putida* in the context of the Design-Build-Test-Learn cycles. I established a general system to rank these pathways based on thermodynamic feasibility, enzyme sequence availability, and maximum theoretical yield. Among the target compounds, cis,cis-muconic acid is a well-known chemical (as an intermediate of, among other, production of nylon), with thoroughly-characterized biosynthetic pathways. Despite of this, I was able to predict 2 pathways (out of a total of 8) that had not been reported earlier. Similarly, I also predicted novel pathways for the production of anisole, aniline, 3-methylmalate, and geranic acid.

While constructing and using GEMs in **Chapters 2 and 3**,I encountered two recurring problems. The first was the use of inconsistent namespaces among GEMs. A critical step in constructing GEMs is to manually curate them by integrating information from independent (organism specific) sources to provide a comprehensive representation of what is presently known about the metabolism of the modelled organism. Combining this precious information from individual GEMs to make a consensus model of the organism is essential. Using models from different species as a foundation to construct a new model can help to avoid repeating the same time consuming manual curation step. In addition, GEMs need to be updated continuously since new knowledge is coming in short order. However, such simple tasks cannot be

done easily due to a simple reason: inconsistent namespaces. GEMs constructed for different organisms by different researchers often use different naming conventions depending on which databases were selected for model construction. While mapping between namespaces would seem the most logical solution, it involves a high risk of mismatch and may invalidate the model. I evaluated the (in)consistency of names and non-systematic identifiers used in 11 biochemical databases of biochemical reactions and the problems that arise when mapping between different namespaces and databases in **Chapter 4**. I found that such inconsistencies can be as high as 83.1%, thus emphasizing the need for strategies to deal with these issues. Currently, manual verification of the mappings appears to be the only solution to remove inconsistencies when combining models.

The second problem that has arisen relates to the efficiency of gap-filling tools. The lack of accurate functional annotations often renders GEMs incomplete, giving rise to missing reactions, the so-called 'gaps' in the network. Gap-filling becomes important during model construction not only to make a functional model but also to generate new knowledge on protein function. To assist gap-filling, many algorithms have been published. To be able to use GEMs effectively, these methods should allow the model to be as accurate as possible, preferably also in a user-friendly manner so that they become available to many researchers. However, gap-filling algorithms vastly differ in their objectives, implementation platforms, and input data requirements. These differences imply a variety in their usability and accuracy. In **Chapter 5** I conducted an extensive evaluation of these algorithms from a user's perspective. We found that most of the tools are not used due to the lack of a workable implementation. From those for which an implementation is readily available, I selected SMILEY, FASTGAPFILL and Meneco to further investigate their performances. SMILEY was the best among the three algorithms for small-scale degradation. Finally, in

Chapter 6 I discussed the three significant themes stood out across **Chapters 2, 3, 4, 5** 1) the lack of standards in namespaces, tool development, and guidelines for model evaluation; 2) the need to improve models and computational tools, for instance to account for uncertainty in the biomass synthesis reaction or to improve gap-filling algorithms, and; 3) the potential contribution of GEMs to the DBTL cycle.

In conclusion, the work presented in this thesis illustrates how the lack of standards in GEMs can hamper their usability. GEMs have great potential in the DBTL cycles. Standardization and improvement in GEM formulation are needed to maximize the use of these models.

References

[1] F. O. interview. The smartest man in the world. 1979.

[2] G. Joyce, D. Deamer, and G. Fleischaker. Origins of life: the central concepts. *Deamer, DW*, 1994.

[3] P. L. Luisi. About various definitions of life. *Origins of Life and Evolution of the Biosphere*, 28(4-6):613–622, 1998.

[4] L. Margulis and D. Sagan. *What is life?* Univ of California Press, 2000.

[5] D. Schulze-Makuch and L. N. Irwin. *Life in the Universe*. Springer, 2004.

[6] M. Vitas and A. Dobovišek. Towards a general definition of life. *Origins of Life and Evolution of Biospheres*, 49(1-2):77–88, 2019.

[7] A. Antonellis and E. D. Green. The role of aminoacyl-trna synthetases in genetic diseases. *Annu. Rev. Genomics Hum. Genet.*, 9:87–107, 2008.

[8] J. C. Smuts. *Holism and evolution*. Рипол Классик, 1926.

[9] A. Trewavas. A brief history of systems biology:"every object that biology studies is a system of systems." francois jacob (1974). *The Plant Cell*, 18(10): 2420–2430, 2006.

[10] D. Gatherer. So what do we really mean when we say that systems biology is holistic? *BMC systems biology*, 4(1):22, 2010.

[11] R. C. Looijen. *Holism and reductionism in biology and ecology: the mutual dependence of higher and lower level research programmes*, volume 23. Springer Science & Business Media, 2012.

[12] F. Mazzocchi. Complexity and the reductionism–holism debate in systems biology. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 4(5): 413–427, 2012.

[13] H. V. Westerhoff and B. O. Palsson. The evolution of molecular biology into systems biology. *Nature biotechnology*, 22(10):1249–1252, 2004.

[14] M. Mesarovic. General systems theory and biology–view of a theoretician. *General Systems Theory and Biology. Springer*, 1968.

[15] S. Kesić. Systems biology, emergence and antireductionism. *Saudi journal of biological sciences*, 23(5):584–591, 2016.

[16] R. Breitling. What is systems biology? *Front Physiol*, pages 1–9, 2010.

[17] H. Kitano. Systems biology: a brief overview. *science*, 295(5560):1662–1664, 2002.

[18] D. J. Wilkinson. Stochastic modelling for quantitative description of heterogeneous biological systems. *Nature Reviews Genetics*, 10(2):122–133, 2009.

[19] E. F. Mason and J. C. Rathmell. Cell metabolism: an essential link between cell growth and apoptosis. *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research*, 1813(4):645–654, 2011.

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

[21] I. Thiele and B. Ø. Palsson. A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nature protocols*, 5(1):93, 2010.

[22] S. Klamt and J. Stelling. Stoichiometric and constraint-based modeling. *System Modeling in Cellular Biology: From Concepts to Nuts and Bolts*, pages 73–96, 2006.

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

[24] R. Schuetz, L. Kuepfer, and U. Sauer. Systematic evaluation of objective functions for predicting intracellular fluxes in escherichia coli. *Molecular systems biology*, 3(1):119, 2007.

[25] D. Tilman. *Resource competition and community structure*. Princeton university press, 1982.

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

[27] E. Simeonidis and N. D. Price. Genome-scale modeling for metabolic engineering. *Journal of industrial microbiology & biotechnology*, 42(3):327–338, 2015.

[28] J.-C. Lachance, S. Rodrigue, and B. O. Palsson. The use of in silico genome-scale models for the rational design of minimal cells. In *Minimal Cells: Design, Construction, Biotechnological Applications*, pages 141–175. Springer, 2020.

[29] C. Gu, G. B. Kim, W. J. Kim, H. U. Kim, and S. Y. Lee. Current status and applications of genome-scale metabolic models. *Genome biology*, 20(1):121, 2019.

[30] A. Badri, K. Raman, and G. Jayaraman. Uncovering novel pathways for enhancing hyaluronan synthesis in recombinant lactococcus lactis: Genome-scale metabolic modeling and experimental validation. *Processes*, 7(6):343, 2019.

[31] O. S. Mohite, T. Weber, H. U. Kim, and S. Y. Lee. Genome-scale metabolic reconstruction of actinomycetes for antibiotics production. *Biotechnology Journal*, 14(1):1800377, 2019.

[32] J. L. Reed. Shrinking the metabolic solution space using experimental datasets. *PLoS Comput Biol*, 8(8):e1002662, 2012.

[33] S. S. C. on Emerging, S. S. C. o. H. Newly Identified Health Risks), SCCS (Scientific Committee on Consumer Safety), and E. Risks). Synthetic biology i definition. *Opinion*, 25 September 2014.

[34] A. L. Demain, E. J. Vandamme, J. Collins, and K. Buchholz. History of industrial biotechnology. *Industrial biotechnology: microorganisms*, 1:1–84, 2017.

[35] H. Benninga. *A history of lactic acid making: a chapter in the history of biotechnology*, volume 11. Springer Science & Business Media, 1990.

[36] H. G. Moon, Y.-S. Jang, C. Cho, J. Lee, R. Binkley, and S. Y. Lee. One hundred years of clostridial butanol fermentation. *FEMS microbiology letters*, 363(3), 2016.

[37] J.-L. Adrio and A. L. Demain. Recombinant organisms for production of industrial products. *Bioengineered bugs*, 1(2):116–131, 2010.

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

[39] O. Rosales-Calderon and V. Arantes. A review on commercial-scale high-value products that can be produced alongside cellulosic ethanol. *Biotechnology for biofuels*, 12(1):240, 2019.

[40] J. K. Ko and S.-M. Lee. Advances in cellulosic conversion to fuels: engineering yeasts for cellulosic bioethanol and biodiesel production. *Current opinion in biotechnology*, 50:72–80, 2018.

[41] K. R. Choi, W. D. Jang, D. Yang, J. S. Cho, D. Park, and S. Y. Lee. Systems metabolic engineering strategies: integrating systems and synthetic biology with metabolic engineering. *Trends in biotechnology*, 37(8):817–837, 2019.

[42] M. Wehrs, D. Tanjore, T. Eng, J. Lievense, T. R. Pray, and A. Mukhopadhyay. Engineering robust production microbes for large-scale cultivation. *Trends in microbiology*, 27(6):524–537, 2019.

[43] C. J. Paddon and J. D. Keasling. Semi-synthetic artemisinin: a model for the use of synthetic biology in pharmaceutical development. *Nature Reviews Microbiology*, 12(5):355–367, 2014.

[44] A. Biz, S. Proulx, Z. Xu, K. Siddartha, A. M. Indrayanti, and R. Mahadevan. Systems biology based metabolic engineering for non-natural chemicals. *Biotechnology advances*, 37(6):107379, 2019.

[45] A. S. Khalil and J. J. Collins. Synthetic biology: applications come of age. *Nature Reviews Genetics*, 11(5):367–379, 2010.

[46] D. E. Cameron, C. J. Bashor, and J. J. Collins. A brief history of synthetic biology. *Nature Reviews Microbiology*, 12(5):381–390, 2014.

[47] Y. Chen, D. Banerjee, A. Mukhopadhyay, and C. J. Petzold. Systems and synthetic biology tools for advanced bioproduction hosts. *Current Opinion in Biotechnology*, 64:101–109, 2020.

[48] P. Opgenorth, Z. Costello, T. Okada, G. Goyal, Y. Chen, J. Gin, V. Benites, M. de Raad, T. R. Northen, K. Deng, et al. Lessons from two design–build–test–learn cycles of dodecanol production in escherichia coli aided by machine learning. *ACS synthetic biology*, 8(6):1337–1351, 2019.

[49] I. S. Pretorius. Synthetic genome engineering forging new frontiers for wine yeast. *Critical reviews in biotechnology*, 37(1):112–136, 2017.

[50] B. Pouvreau, T. Vanhercke, and S. Singh. From plant metabolic engineering to plant synthetic biology: The evolution of the design/build/test/learn cycle. *Plant Science*, 273:3–12, 2018.

[51] S. C. Wheelwright and K. B. Clark. Accelerating the design-build-test cycle for effective product development. *International Marketing Review*, 1994.

[52] D. Ando and H. G. Martin. Two-scale 13 c metabolic flux analysis for metabolic engineering. In *Synthetic Metabolic Pathways*, pages 333–352. Springer, 2018.

[53] C. E. Lawson, W. R. Harcombe, R. Hatzenpichler, S. R. Lindemann, F. E. Löffler, M. A. O'Malley, H. G. Martín, B. F. Pfleger, L. Raskin, O. S. Venturelli, et al. Common principles and best practices for engineering microbiomes. *Nature Reviews Microbiology*, pages 1–17, 2019.

[54] P. Carbonell, A. J. Jervis, C. J. Robinson, C. Yan, M. Dunstan, N. Swainston, M. Vinaixa, K. A. Hollywood, A. Currin, N. J. Rattray, et al. An automated design-build-test-learn pipeline for enhanced microbial production of fine chemicals. *Communications biology*, 1(1):1–10, 2018.

[55] B. Vasconcelos, J. C. Teixeira, G. Dragone, and J. A. Teixeira. Oleaginous yeasts for sustainable lipid production—from biodiesel to surf boards, a wide range of "green" applications. *Applied microbiology and biotechnology*, pages 1–17, 2019.

[56] S. Shi and H. Zhao. Metabolic engineering of oleaginous yeasts for production of fuels and chemicals. *Frontiers in microbiology*, 8:2185, 2017.

[57] C. Ratledge. Regulation of lipid accumulation in oleaginous microorganisms, 2002.

[58] A. Beopoulos, J.-M. Nicaud, and C. Gaillardin. An overview of lipid metabolism in yeasts and its impact on biotechnological processes. *Applied microbiology and biotechnology*, 90(4):1193–1206, 2011.

[59] C. Ratledge and J. P. Wynn. The biochemistry and molecular biology of lipid accumulation in oleaginous microorganisms. *Advances in applied microbiology*, 51:1–52, 2002.

[60] J. M. Ageitos, J. A. Vallejo, P. Veiga-Crespo, and T. G. Villa. Oily yeasts as oleaginous cell factories. *Applied microbiology and biotechnology*, 90(4): 1219–1227, 2011.

[61] Z. Chi, Y. Zheng, J. Ma, and S. Chen. Oleaginous yeast cryptococcus curvatus culture with dark fermentation hydrogen production effluent as feedstock for microbial lipid production. *international journal of hydrogen energy*, 36(16): 9542–9550, 2011.

[62] A. Ykema, E. C. Verbree, M. M. Kater, and H. Smit. Optimization of lipid production in the oleaginous yeast apiotrichum curvatum in wheypermeate. *Applied microbiology and biotechnology*, 29(2-3):211–218, 1988.

[63] W. Zhou, Z. Gong, L. Zhang, Y. Liu, J. Yan, and M. Zhao. Feasibility of lipid production from waste paper by the oleaginous yeast cryptococcus curvatus. *BioResources*, 12(3):5249–5263, 2017.

[64] N. Annamalai, N. Sivakumar, and P. Oleskowicz-Popiel. Enhanced production of microbial lipids from waste office paper by the oleaginous yeast cryptococcus curvatus. *Fuel*, 217:420–426, 2018.

[65] F. Bracharz, T. Beukhout, N. Mehlmer, and T. Brück. Opportunities and challenges in the development of cutaneotrichosporon oleaginosus atcc 20509 as a new cell factory for custom tailored microbial oils. *Microbial cell factories*, 16(1):178, 2017.

[66] A. Yaguchi, D. Rives, and M. Blenner. New kids on the block: emerging oleaginous yeast of biotechnological importance. *AIMS Microbiol*, 3:227–47, 2017.

[67] P. A. Meesters and G. Eggink. Isolation and characterization of a δ-9 fatty acid desaturase gene from the oleaginous yeast cryptococcus curvatus cbs 570. *Yeast*, 12(8):723–730, 1996.

[68] Y. Li, Z. K. Zhao, and F. Bai. High-density cultivation of oleaginous yeast rhodosporidium toruloides y4 in fed-batch culture. *Enzyme and microbial technology*, 41(3):312–317, 2007.

[69] X. Meng, J. Yang, X. Xu, L. Zhang, Q. Nie, and M. Xian. Biodiesel production from oleaginous microorganisms. *Renewable energy*, 34(1):1–5, 2009.

[70] S. Wu, X. Zhao, H. Shen, Q. Wang, and Z. K. Zhao. Microbial lipid production by rhodosporidium toruloides under sulfate-limited conditions. *Bioresource technology*, 102(2):1803–1807, 2011.

[71] J. Liu, X. Huang, R. Chen, M. Yuan, and J. Liu. Efficient bioconversion of high-content volatile fatty acids into microbial lipids by cryptococcus curvatus atcc 20509. *Bioresource technology*, 239:394–401, 2017.

[72] Y. Liang, Y. Cui, J. Trushenski, and J. W. Blackburn. Converting crude glycerol derived from yellow grease to lipids through yeast fermentation. *Bioresource technology*, 101(19):7581–7586, 2010.

[73] K. Qiao, T. M. Wasylenko, K. Zhou, P. Xu, and G. Stephanopoulos. Lipid production in yarrowia lipolytica is maximized by engineering cytosolic redox metabolism. *Nature biotechnology*, 35(2):173, 2017.

[74] S. Papanikolaou and G. Aggelis. Lipid production by yarrowia lipolytica growing on industrial glycerol in a single-stage continuous culture. *Bioresource technology*, 82(1):43–49, 2002.

[75] Q. Fei, M. O'Brien, R. Nelson, X. Chen, A. Lowell, and N. Dowe. Enhanced lipid production by rhodosporidium toruloides using different fed-batch feeding strategies with lignocellulosic hydrolysate as the sole carbon source. *Biotechnology for biofuels*, 9(1):130, 2016.

[76] V. W. Johnson, M. Singh, V. S. Saini, D. K. Adhikari, V. Sista, and N. K. Yadav. Utilization of molasses for the production of fat by an oleaginous yeast, rhodotorula glutinis iip-30. *Journal of industrial microbiology*, 14(1): 1–4, 1995.

[77] A. Anschau, M. C. Xavier, S. Hernalsteens, and T. T. Franco. Effect of feeding strategies on lipid production by lipomyces starkeyi. *Bioresource technology*, 157:214–222, 2014.

[78] M. Hassan, P. J. Blanc, L.-M. Granger, A. Pareilleux, and G. Goma. Lipid production by an unsaturated fatty acid auxotroph of the oleaginous yeast apiotrichum curvatum grown in single-stage continuous culture. *Applied microbiology and biotechnology*, 40(4):483–488, 1993.

[79] X.-Z. Liu, Q.-M. Wang, M. Göker, M. Groenewald, A. Kachalkin, H. T. Lumbsch, A. Millanes, M. Wedin, A. Yurkov, T. Boekhout, et al. Towards an integrated phylogenetic classification of the tremellomycetes. *Studies in Mycology*, 81:85–147, 2015.

[80] J. W. Fell, T. Boekhout, A. Fonseca, G. Scorzetti, and A. Statzell-Tallman. Biodiversity and systematics of basidiomycetous yeasts as determined by large-subunit rdna d1/d2 domain sequence analysis. *International journal of systematic and evolutionary microbiology*, 50(3):1351–1371, 2000.

[81] P. Gujjari, S.-O. Suh, K. Coumes, and J. J. Zhou. Characterization of oleaginous yeasts revealed two novel species: Trichosporon cacaoliposimilis sp. nov. and trichosporon oleaginosus sp. nov. *Mycologia*, 103(5):1110–1118, 2011.

[82] C. T. Evans and C. Ratledge. A comparison of the oleaginous yeast, candida curvata, grown on different carbon sources in continuous and batch culture. *Lipids*, 18(9):623–629, 1983.

[83] C. T. Evans and C. Ratledge. Induction of xylulose-5-phosphate phosphoketolase in a variety of yeasts grown on d-xylose: the key to efficient xylose metabolism. *Archives of microbiology*, 139(1):48–52, 1984.

[84] X.-F. Huang, Y.-H. Wang, Y. Shen, K.-M. Peng, L.-J. Lu, and J. Liu. Using nonionic surfactant as an accelerator to increase extracellular lipid production by oleaginous yeast cryptococcus curvatus mucl 29819. *Bioresource technology*, 274:272–280, 2019.

[85] D. Awad, F. Bohnen, N. Mehlmer, and T. Brueck. Multi-factorial-guided media optimization for enhanced biomass and lipid formation by the oleaginous yeast cutaneotrichosporon oleaginosus. *Frontiers in bioengineering and biotechnology*, 7:54, 2019.

[86] C. Görner, V. Redai, F. Bracharz, P. Schrepfer, D. Garbe, and T. Brück. Genetic engineering and production of modified fatty acids by the non-conventional oleaginous yeast trichosporon oleaginosus atcc 20509. *Green Chemistry*, 18(7):2037–2046, 2016.

[87] J. L. Reed, T. D. Vo, C. H. Schilling, and B. O. Palsson. An expanded genome-scale model of escherichia coli k-12 (i jr904 gsm/gpr). *Genome biology*, 4(9): R54, 2003.

[88] D. Close and J. Ojumu. Draft genome sequence of the oleaginous yeast *Cryptococcus curvatus* atcc 20509. *Genome Announc.*, 4(6):e01235–16, 2016.

[89] K. J. Hoff, S. Lange, A. Lomsadze, M. Borodovsky, and M. Stanke. Braker1: unsupervised rna-seq-based genome annotation with genemark-et and augustus. *Bioinformatics*, 32(5):767–769, 2015.

[90] O. Tehlivets, K. Scheuringer, and S. D. Kohlwein. Fatty acid synthesis and elongation in yeast. *Biochimica et Biophysica Acta (BBA)-Molecular and Cell Biology of Lipids*, 1771(3):255–270, 2007.

[91] L. Klug and G. Daum. Yeast lipid metabolism at a glance. *FEMS yeast research*, 14(3):369–388, 2014.

[92] S. Fakas. Lipid biosynthesis in yeasts: A comparison of the lipid biosynthetic pathway between the model nonoleaginous yeast saccharomyces cerevisiae and the model oleaginous yeast yarrowia lipolytica. *Engineering in Life Sciences*, 17(3):292–302, 2017.

[93] L. Garba, M. Shukuri Mo, S. Nurbaya Os, and R. Noor Zalih. Review on fatty acid desaturases and their roles in temperature acclimatisation. *Journal of Applied Sciences*, 17:282–295, 2017.

[94] H. Rismani-Yazdi, B. Z. Haznedaroglu, K. Bibby, and J. Peccia. Transcriptome sequencing and annotation of the microalgae dunaliella tertiolecta: pathway description and gene discovery for production of next-generation biofuels. *BMC genomics*, 12(1):148, 2011.

[95] C. Capusoni, V. Rodighiero, D. Cucchetti, S. Galafassi, D. Bianchi, G. Franzosi, and C. Compagno. Characterization of lipid accumulation and lipidome analysis in the oleaginous yeasts rhodosporidium azoricum and trichosporon oleaginosus. *Bioresource technology*, 238:281–289, 2017.

[96] G. M. Carman and M. C. Kersting. Phospholipid synthesis in yeast: regulation by phosphorylation. *Biochemistry and cell biology*, 82(1):62–70, 2004.

[97] N. Loira, T. Dulermo, J.-M. Nicaud, and D. J. Sherman. A genome-scale metabolic model of the lipid-accumulating yeast yarrowia lipolytica. *BMC systems biology*, 6(1):35, 2012.

[98] I. Nookaew, M. C. Jewett, A. Meechai, C. Thammarongtham, K. Laoteng, S. Cheevadhanarak, J. Nielsen, and S. Bhumiratana. The genome-scale metabolic model iin800 of saccharomyces cerevisiae and its validation: a scaffold to query lipid metabolism. *BMC Systems Biology*, 2(1):71, 2008.

[99] M. A. van den Berg, P. de Jong-Gubbels, C. J. Kortland, J. P. van Dijken, J. T. Pronk, and H. Y. Steensma. The two acetyl-coenzyme a synthetases of saccharomyces cerevisiae differ with respect to kinetic properties and transcriptional regulation. *Journal of Biological Chemistry*, 271(46):28953–28959, 1996.

[100] S. H. Chan, J. Cai, L. Wang, M. N. Simons-Senftle, and C. D. Maranas. Standardizing biomass reactions and ensuring complete mass balance in genome-scale metabolic models. *Bioinformatics*, 33(22):3603–3609, 2017.

[101] D. Dikicioglu, B. Kırdar, and S. G. Oliver. Biomass composition: the "elephant in the room" of metabolic modelling. *Metabolomics*, 11(6):1690–1701, 2015.

[102] S. S. Tchakouteu, A. Chatzifragkou, O. Kalantzi, A. A. Koutinas, G. Aggelis, and S. Papanikolaou. Oleaginous yeast cryptococcus curvatus exhibits interplay between biosynthesis of intracellular sugars and lipids. *European Journal of Lipid Science and Technology*, 117(5):657–672, 2015.

[103] P. Meeuwse. *Production of fungal lipids: kinetic modeling and process design*. 2011.

[104] Y. Yang and M. Sha. A beginner's guide to bioprocess modes–batch, fed-batch, and continuous fermentation. Technical report, Eppendorf Application Note, 2019.

[105] P. Meesters, G. Huijberts, and G. Eggink. High-cell-density cultivation of the lipid accumulating yeast cryptococcus curvatus using glycerol as a carbon source. *Applied microbiology and biotechnology*, 45(5):575–579, 1996.

[106] X. Yu, Y. Zheng, X. Xiong, and S. Chen. Co-utilization of glucose, xylose and cellobiose by the oleaginous yeast cryptococcus curvatus. *Biomass and Bioenergy*, 71:340–349, 2014.

[107] A. Ykema, E. Verbree, H. Van Verseveld, and H. Smit. Mathematical modelling of lipid production by oleaginous yeasts in continuous cultures. *Antonie Van Leeuwenhoek*, 52(6):491–506, 1986.

[108] P. A. BOTHAM and C. Ratledge. A biochemical explanation for lipid accumulation in candida 107 and other oleaginous micro-organisms. *Microbiology*, 114(2):361–375, 1979.

[109] C. T. EVANS, A. H. SCRAGG, and C. RATLEDGE. Regulation of citrate efflux from mitochondria oleaginous and non-oleaginous yeasts by adenine nucleotides. *European journal of biochemistry*, 132(3):609–615, 1983.

[110] H. Zhang, L. Zhang, H. Chen, Y. Q. Chen, W. Chen, Y. Song, and C. Ratledge. Enhanced lipid accumulation in the yeast yarrowia lipolytica by overexpression of atp: citrate lyase from mus musculus. *Journal of biotechnology*, 192:78–84, 2014.

[111] E. J. Kerkhoven, K. R. Pomraning, S. E. Baker, and J. Nielsen. Regulation of amino-acid metabolism controls flux to lipid accumulation in yarrowia lipolytica. *NPJ systems biology and applications*, 2:16005, 2016.

[112] R. Huerlimann, E. J. Steinig, H. Loxton, K. R. Zenger, D. R. Jerry, and K. Heimann. The effect of nitrogen limitation on acetyl-coa carboxylase expression and fatty acid content in chromera velia and isochrysis aff. galbana (tiso). *Gene*, 543(2):204–211, 2014.

[113] R. Kourist, F. Bracharz, J. Lorenzen, O. N. Kracht, M. Chovatia, C. Daum, S. Deshpande, A. Lipzen, M. Nolan, R. A. Ohm, et al. Genomics and transcriptomics analyses of the oil-accumulating basidiomycete yeast trichosporon oleaginosus: insights into substrate utilization and alternative evolutionary trajectories of fungal mating systems. *MBio*, 6(4):e00918–15, 2015.

[114] D. A. Fell and S. Thomas. Physiological control of metabolic flux: the requirement for multisite modulation. *Biochemical Journal*, 311(1):35–39, 1995.

[115] A. Pfitzner, C. Kubicek, and M. Röhr. Presence and regulation of atp: citrate lyase from the citric acid producing fungus aspergillus niger. *Archives of microbiology*, 147(1):88–91, 1987.

[116] I. A. Potapova, M. R. El-Maghrabi, S. V. Doronin, and W. B. Benjamin. Phosphorylation of recombinant human atp: citrate lyase by camp-dependent protein kinase abolishes homotropic allosteric regulation of the enzyme by citrate and increases the enzyme activity. allosteric activation of atp: citrate lyase by phosphorylated sugars. *Biochemistry*, 39(5):1169–1179, 2000.

[117] N. J. Moon, E. Hammond, and B. A. Glatz. Conversion of cheese whey and whey permeate to oil and single-cell protein1. *Journal of Dairy Science*, 61 (11):1537–1547, 1978.

[118] A. Dobin, C. A. Davis, F. Schlesinger, J. Drenkow, C. Zaleski, S. Jha, P. Batut, M. Chaisson, and T. R. Gingeras. Star: ultrafast universal rna-seq aligner. *Bioinformatics*, 29(1):15–21, 2013.

[119] M. I. Love, W. Huber, and S. Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15(12):550, 2014.

[120] D. Kim, B. Langmead, and S. L. Salzberg. Hisat: a fast spliced aligner with low memory requirements. *Nature methods*, 12(4):357, 2015.

[121] J. J. Koehorst, J. C. van Dam, E. Saccenti, V. A. Martins dos Santos, M. Suarez-Diez, and P. J. Schaap. Sapp: functional genome annotation and analysis through a semantic framework using fair principles. *Bioinformatics*, 34(8): 1401–1403, 2017.

[122] J. C. van Dam, J. J. Koehorst, J. O. Vik, V. A. M. dos Santos, P. J. Schaap, and M. Suarez-Diez. The empusa code generator and its application to gbol, an extendable ontology for genome annotation. *Scientific data*, 6(1):1–9, 2019.

[123] E. M. Zdobnov and R. Apweiler. Interproscan–an integration platform for the signature-recognition methods in interpro. *Bioinformatics*, 17(9):847–848, 2001.

[124] N.-N. Nguyen, S. Srihari, H. W. Leong, and K.-F. Chong. Enzdp: Improved enzyme annotation for metabolic network reconstruction based on domain composition profiles. *Journal of bioinformatics and computational biology*, 13 (05):1543003, 2015.

[125] MATLAB. *version R2015b*. The MathWorks Inc., Natick, Massachusetts, 2015.

[126] GLPK. (gnu linear programming kit), 2009. URL https://www.gnu.org/software/glpk/.

[127] P. Pan and Q. Hua. Reconstruction and in silico analysis of metabolic network for an oleaginous yeast, yarrowia lipolytica. *PLoS One*, 7(12):e51535, 2012.

[128] M. Kavšček, G. Bhutada, T. Madl, and K. Natter. Optimization of lipid production with a genome-scale model of yarrowia lipolytica. *BMC systems biology*, 9(1):72, 2015.

[129] P. Mishra, N.-R. Lee, M. Lakshmanan, M. Kim, B.-G. Kim, and D.-Y. Lee. Genome-scale model-driven strain design for dicarboxylic acid production in yarrowia lipolytica. *BMC systems biology*, 12(2):9–20, 2018.

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

[131] B. Teusink, J. Passarge, C. A. Reijenga, E. Esgalhado, C. C. Van der Weijden, M. Schepper, M. C. Walsh, B. M. Bakker, K. Van Dam, H. V. Westerhoff, et al. Can yeast glycolysis be understood in terms of in vitro kinetics of the constituent enzymes? testing biochemistry. *European Journal of Biochemistry*, 267(17):5313–5329, 2000.

[132] N. Raimundo, B. E. Baysal, and G. S. Shadel. Revisiting the tca cycle: signaling to tumor formation. *Trends in molecular medicine*, 17(11):641–649, 2011.

[133] C. J. Fritzemeier, D. Hartleb, B. Szappanos, B. Papp, and M. J. Lercher. Erroneous energy-generating cycles in published genome scale metabolic networks: Identification and removal. *PLoS computational biology*, 13(4): e1005494, 2017.

[134] G. M. Cooper and R. E. Hausman. *The cell: A molecular approach. 4th edition*. 2004.

[135] J. Schellenberger, R. Que, R. M. Fleming, I. Thiele, J. D. Orth, A. M. Feist, D. C. Zielinski, A. Bordbar, N. E. Lewis, S. Rahmanian, et al. Quantitative prediction of cellular metabolism with constraint-based models: the cobra toolbox v2. 0. *Nature protocols*, 6(9):1290, 2011.

[136] J. Schellenberger and B. Ø. Palsson. Use of randomized sampling for analysis of metabolic networks. *Journal of biological chemistry*, 284(9):5457–5461, 2009.

[137] S. Bordel, R. Agren, and J. Nielsen. Sampling the solution space in genome-scale metabolic networks reveals transcriptional regulation in key enzymes. *PLoS Comput Biol*, 6(7):e1000859, 2010.

[138] J. C. Philp, R. J. Ritchie, and J. E. Allan. Biobased chemicals: the convergence of green chemistry with industrial biotechnology. *Trends in biotechnology*, 31 (4):219–222, 2013.

[139] A. J. Straathof. Transformation of biomass into commodity chemicals using enzymes or cells. *Chemical reviews*, 114(3):1871–1908, 2014.

[140] V. Chubukov, A. Mukhopadhyay, C. J. Petzold, J. D. Keasling, and H. G. Martín. Synthetic and systems biology for microbial production of commodity chemicals. *NPJ systems biology and applications*, 2(1):1–11, 2016.

[141] L. Rohlin, M.-K. Oh, and J. C. Liao. Microbial pathway engineering for industrial processes: evolution, combinatorial biosynthesis and rational design. *Current opinion in microbiology*, 4(3):330–335, 2001.

[142] P. Carbonell. *Metabolic Pathway Design: A Practical Guide*. Springer Nature, 2019.

[143] X. Sun, X. Shen, R. Jain, Y. Lin, J. Wang, J. Sun, J. Wang, Y. Yan, and Q. Yuan. Synthesis of chemicals by metabolic engineering of microbes. *Chemical society reviews*, 44(11):3760–3785, 2015.

[144] C. J. Petzold, L. J. G. Chan, M. Nhan, and P. D. Adams. Analytics for metabolic engineering. *Frontiers in bioengineering and biotechnology*, 3:135, 2015.

[145] P. S. Freemont. Synthetic biology industry: data-driven design is creating new opportunities in biotechnology. *Emerging Topics in Life Sciences*, 3(5): 651–657, 2019.

[146] R. Kelwick, J. T. MacDonald, A. J. Webb, and P. Freemont. Developments in the tools and methodologies of synthetic biology. *Frontiers in bioengineering and biotechnology*, 2:60, 2014.

[147] E. Pitkänen, P. Jouhten, and J. Rousu. Inferring branching pathways in genome-scale metabolic networks. *BMC systems biology*, 3(1):103, 2009.

[148] A. Chowdhury and C. D. Maranas. Designing overall stoichiometric conversions and intervening metabolic reactions. *Scientific reports*, 5:16009, 2015.

[149] N. Hadadi and V. Hatzimanikatis. Design of computational retrobiosynthesis tools for the design of de novo synthetic pathways. *Current opinion in chemical biology*, 28:99–104, 2015.

[150] X. Sun, Y. Lin, Q. Huang, Q. Yuan, and Y. Yan. A novel muconic acid biosynthesis approach by shunting tryptophan biosynthesis via anthranilate. *Appl. Environ. Microbiol.*, 79(13):4024–4030, 2013.

[151] Y. Lin, X. Sun, Q. Yuan, and Y. Yan. Extending shikimate pathway for the production of muconic acid and its precursor salicylic acid in escherichia coli. *Metabolic engineering*, 23:62–69, 2014.

[152] X. Sun, Y. Lin, Q. Yuan, and Y. Yan. Biological production of muconic acid via a prokaryotic 2, 3-dihydroxybenzoic acid decarboxylase. *ChemSusChem*, 7(9):2478–2481, 2014.

[153] B. Thompson, S. Pugh, M. Machas, and D. R. Nielsen. Muconic acid production via alternative pathways and a synthetic "metabolic funnel". *ACS synthetic biology*, 7(2):565–575, 2017.

[154] Y. Yan and L. Yuheng. Microbial production of muconic acid and salicylic acid, Jan. 17 2017. US Patent 9,546,387.

[155] Anisole. URL https://pubchem.ncbi.nlm.nih.gov/compound/Anisole.

[156] T. Higa, T. Fujiyama, and P. J. Scheuer. Halogenated phenol and indole constituents of acorn worms. *Comparative Biochemistry and Physiology Part B: Comparative Biochemistry*, 65(3):525–530, 1980.

[157] G. W. Gribble. The natural production of organobromine compounds. *Environmental Science and Pollution Research*, 7(1):37–49, 2000.

[158] C. Flodin and F. Whitfield. Biosynthesis of bromophenols in marine algae. *Water science and technology*, 40(6):53–58, 1999.

[159] W. Vetter and D. Janussen. Halogenated natural products in five species of antarctic sponges: compounds with pop-like properties? *Environmental science & technology*, 39(11):3889–3895, 2005.

[160] M. Renavd, P. Chantal, and S. Kaliaguine. Anisole production by alkylation of phenol over zsm5. *The Canadian Journal of Chemical Engineering*, 64(5): 787–791, 1986.

[161] Aniline. URL https://pubchem.ncbi.nlm.nih.gov/compound/Aniline#section=Overview.

[162] W. Johnston. The discovery of aniline and the origin of the term "aniline dye". *Biotechnic & Histochemistry*, 83(2):83–87, 2008.

[163] R. T. Driessen, P. Kamphuis, L. Mathijssen, R. Zhang, L. G. van der Ham, H. van den Berg, and A. J. Zeeuw. Industrial process design for the production of aniline by direct amination. *Chemical Engineering & Technology*, 40 (5):838–846, 2017.

[164] W. Lu, J. E. Ness, W. Xie, X. Zhang, F. Liu, J. Cai, J. Minshull, and R. A. Gross. Biosynthesis of monomers for plastics from renewable oils. In *Biobased Monomers, Polymers, and Materials*, pages 77–90. ACS Publications, 2012.

[165] Y. Cao and X. Zhang. Production of long-chain hydroxy fatty acids by microbial conversion. *Applied microbiology and biotechnology*, 97(8):3323–3331, 2013.

[166] C. Liu, F. Liu, J. Cai, W. Xie, T. E. Long, S. R. Turner, A. Lyons, and R. A. Gross. Polymers from fatty acids: Poly (ω-hydroxyl tetradecanoic acid) synthesis and physico-mechanical studies. *Biomacromolecules*, 12(9):3291–3298, 2011.

[167] Y. Cao, T. Cheng, G. Zhao, W. Niu, J. Guo, M. Xian, and H. Liu. Metabolic engineering of escherichia coli for the production of hydroxy fatty acids from glucose. *BMC biotechnology*, 16(1):26, 2016.

[168] T. Yang, G. Stoopen, N. Yalpani, J. Vervoort, R. de Vos, A. Voster, F. W. Verstappen, H. J. Bouwmeester, and M. A. Jongsma. Metabolic engineering of geranic acid in maize to achieve fungal resistance is compromised by novel glycosylation patterns. *Metabolic engineering*, 13(4):414–425, 2011.

[169] J. Mi, D. Becher, P. Lubuta, S. Dany, K. Tusch, H. Schewe, M. Buchhaupt, and J. Schrader. De novo production of the monoterpenoid geranic acid by metabolically engineered pseudomonas putida. *Microbial cell factories*, 13(1):170, 2014.

[170] A. Hartmann, A. Vila-Santa, N. Kallscheuer, M. Vogt, A. Julien-Laferrière, M.-F. Sagot, J. Marienhagen, and S. Vinga. Optpipe-a pipeline for optimizing metabolic engineering targets. *BMC systems biology*, 11(1):1–9, 2017.

[171] P. Schneider and S. Klamt. Characterizing and ranking computed metabolic engineering strategies. *Bioinformatics*, 35(17):3063–3072, 2019.

[172] A. D. Comer, M. R. Long, J. L. Reed, and B. F. Pfleger. Flux balance analysis indicates that methane is the lowest cost feedstock for microbial cell factories. *Metabolic engineering communications*, 5:26–33, 2017.

[173] P. Carbonell, J. Wong, N. Swainston, E. Takano, N. J. Turner, N. S. Scrutton, D. B. Kell, R. Breitling, and J.-L. Faulon. Selenzyme: Enzyme selection tool for pathway design. *Bioinformatics*, 34(12):2153–2154, 2018.

[174] S. Yu, M. R. Plan, G. Winter, and J. O. Krömer. Metabolic engineering of pseudomonas putida kt2440 for the production of para-hydroxy benzoic acid. *Frontiers in Bioengineering and Biotechnology*, 4:90, 2016.

[175] L. M. Blank, G. Ionidis, B. E. Ebert, B. Bühler, and A. Schmid. Metabolic response of pseudomonas putida during redox biocatalysis in the presence of a second octanol phase. *The FEBS journal*, 275(20):5173–5190, 2008.

[176] K. Lang, J. Zierow, K. Buehler, and A. Schmid. Metabolic engineering of pseudomonas sp. strain vlb120 as platform biocatalyst for the production of isobutyric acid and other secondary metabolites. *Microbial cell factories*, 13 (1):2, 2014.

[177] P. I. Nikel, E. Martínez-García, and V. De Lorenzo. Biotechnological domestication of pseudomonads using synthetic biology. *Nature Reviews Microbiology*, 12(5):368–379, 2014.

[178] L. F. Kampers, R. J. Volkers, and V. A. Martins dos Santos. Pseudomonas putida kt 2440 is hv 1 certified, not gras. *Microbial biotechnology*, 12(5):845–848, 2019.

[179] B. Delépine, T. Duigou, P. Carbonell, and J.-L. Faulon. Retropath2. 0: A retrosynthesis workflow for metabolic engineers. *Metabolic engineering*, 45: 158–170, 2018.

[180] K. M. Draths and J. W. Frost. Environmentally compatible synthesis of adipic acid from d-glucose. *Journal of the American Chemical Society*, 116(1):399–400, 1994.

[181] D. R. Vardon, M. A. Franden, C. W. Johnson, E. M. Karp, M. T. Guarnieri, J. G. Linger, M. J. Salm, T. J. Strathmann, and G. T. Beckham. Adipic acid production from lignin. *Energy & Environmental Science*, 8(2):617–628, 2015.

[182] C.-M. Wu, C.-C. Wu, C.-C. Su, S.-N. Lee, Y.-A. Lee, and J.-Y. Wu. Microbial synthesis of cis, cis-muconic acid from benzoate by sphingobacterium sp. mutants. *Biochemical engineering journal*, 29(1-2):35–40, 2006.

[183] P. Pharkya. Microorganisms for the production of aniline, Apr. 28 2011. US Patent App. 12/910,671.

[184] B. Lupa, D. Lyon, M. D. Gibbs, R. A. Reeves, and J. Wiegel. Distribution of genes encoding the microbial non-oxidative reversible hydroxyarylic acid decarboxylases/phenol carboxylases. *Genomics*, 86(3):342–351, 2005.

[185] V. Subramanian and C. Vaidyanathan. Anthranilate hydroxylase from aspergillus niger: new type of nadph-linked nonheme iron monooxygenase. *Journal of bacteriology*, 160(2):651–655, 1984.

[186] J. B. Powlowski, S. Dagley, V. Massey, and D. Ballou. Properties of anthranilate hydroxylase (deaminating), a flavoprotein from trichosporon cutaneum. *Journal of Biological Chemistry*, 262(1):69–74, 1987.

[187] H. J. Pel, J. H. De Winde, D. B. Archer, P. S. Dyer, G. Hofmann, P. J. Schaap, G. Turner, R. P. De Vries, R. Albang, K. Albermann, et al. Genome sequencing and analysis of the versatile cell factory aspergillus niger cbs 513.88. *Nature biotechnology*, 25(2):221–231, 2007.

[188] F. Fukumori and C. P. Saint. Nucleotide sequences and regulational analysis of genes involved in conversion of aniline to catechol in pseudomonas putida ucc22 (ptdn1). *Journal of bacteriology*, 179(2):399–408, 1997.

[189] K. Kirimura, H. Gunji, R. Wakayama, T. Hattori, and Y. Ishii. Enzymatic kolbe–schmitt reaction to form salicylic acid from phenol: enzymatic characterization and gene identification of a novel enzyme, trichosporon moniliiforme salicylic acid decarboxylase. *Biochemical and biophysical research communications*, 394(2):279–284, 2010.

[190] A. J. Harrison, M. Yu, T. Gårdenborg, M. Middleditch, R. J. Ramsay, E. N. Baker, and J. S. Lott. The structure of mbti from mycobacterium tuberculosis, the first enzyme in the biosynthesis of the siderophore mycobactin, reveals it to be a salicylate synthase. *Journal of bacteriology*, 188(17):6081–6091, 2006.

[191] F. Chen, D. Tholl, J. C. D'Auria, A. Farooq, E. Pichersky, and J. Gershenzon. Biosynthesis and emission of terpenoid volatiles from arabidopsis flowers. *The Plant Cell*, 15(2):481–494, 2003.

[192] D. Brodkorb, M. Gottschall, R. Marmulla, F. Lüddeke, and J. Harder. Linalool dehydratase-isomerase, a bifunctional enzyme in the anaerobic degradation of monoterpenes. *Journal of biological chemistry*, 285(40): 30436–30442, 2010.

[193] F. Lüddeke, A. Wülfing, M. Timke, F. Germer, J. Weber, A. Dikfidan, T. Rahnfeld, D. Linder, A. Meyerdierks, and J. Harder. Geraniol and geranial dehydrogenases induced in anaerobic monoterpene degradation by castellaniella defragrans. *Appl. Environ. Microbiol.*, 78(7):2128–2136, 2012.

[194] Y. Iijima, D. R. Gang, E. Fridman, E. Lewinsohn, and E. Pichersky. Characterization of geraniol synthase from the peltate glands of sweet basil. *Plant physiology*, 134(1):370–379, 2004.

[195] S. Iwamori, T. Oikawa, K. Ishiwata, and N. Makiguchi. Cloning and expression of the erwinia herbicola tyrosine phenol-lyase gene in escherichia coli. *Biotechnology and applied biochemistry*, 16(1):77–85, 1992.

[196] J. Axelrod and J. Daly. Phenol-o-methyltransferase. *Biochimica et Biophysica Acta (BBA)-Enzymology*, 159(3):472–478, 1968.

[197] T. Suzuki, N. Akiyama, A. Yoshida, T. Tomita, K. Lassak, M. F. Haurat, T. Okada, K. Takahashi, S.-V. Albers, T. Kuzuyama, et al. Biochemical characterization of archaeal homocitrate synthase from sulfolobus acidocaldarius. *Febs Letters*, 594(1):126–134, 2020.

[198] R. M. Drevland, A. Waheed, and D. E. Graham. Enzymology and evolution of the pyruvate pathway to 2-oxobutyrate in methanocaldococcus jannaschii. *Journal of bacteriology*, 189(12):4391–4400, 2007.

[199] O. Zelder, B. Beatrix, and W. Buckel. Cloning, sequencing and expression in escherichia coli of the gene encoding component s of the coenzyme b12-dependent glutamate mutase from clostridium cochlearium. *FEMS microbiology letters*, 118(1-2):15–21, 1994.

[200] M. Khomyakova, Ö. Bükmez, L. K. Thomas, T. J. Erb, and I. A. Berg. A methylaspartate cycle in haloarchaea. *Science*, 331(6015):334–337, 2011.

[201] P. C. Maxwell. Process for the production of muconic acid, May 13 1986. US Patent 4,588,688.

[202] C. W. Johnson, D. Salvachúa, P. Khanna, H. Smith, D. J. Peterson, and G. T. Beckham. Enhancing muconic acid production from glucose and lignin-derived aromatic compounds via increased protocatechuate decarboxylase activity. *Metabolic engineering communications*, 3:111–119, 2016.

[203] P. K. Arora. Bacterial degradation of monocyclic aromatic amines. *Frontiers in microbiology*, 6:820, 2015.

[204] S. Murakami, Y. Nakanishi, N. Kodama, S. Takenaka, R. Shinke, and K. AOKI. Purification, characterization, and gene analysis of catechol 2, 3-dioxygenase from the aniline-assimilating bacterium pseudomonas species aw-2. *Bioscience, biotechnology, and biochemistry*, 62(4):747–752, 1998.

[205] S. G. Bang, W. J. Choi, C. Y. Choi, and M. H. Cho. Production ofcis, cis-muconic acid from benzoic acidvia microbial transformation. *Biotechnology and Bioprocess Engineering*, 1(1):36–40, 1996.

[206] P. Pazmino and R. M. Weinshilboum. Human erythrocyte phenol o-methyltransferase: radiochemical microassay and biochemical properties. *Clinica Chimica Acta*, 89(2):317–329, 1978.

[207] D. R. Gang, N. Lavid, C. Zubieta, F. Chen, T. Beuerle, E. Lewinsohn, J. P. Noel, and E. Pichersky. Characterization of phenylpropene o-methyltransferases from sweet basil: facile change of substrate specificity and

convergent evolution within a plant o-methyltransferase family. *The Plant Cell*, 14(2):505–519, 2002.

[208] N. Pham, R. G. van Heck, J. C. van Dam, P. J. Schaap, E. Saccenti, and M. Suarez-Diez. Consistency, inconsistency, and ambiguity of metabolite names in biochemical databases used for genome-scale metabolic modelling. *Metabolites*, 9(2):28, 2019.

[209] P. Carbonell, A.-G. Planson, D. Fichera, and J.-L. Faulon. A retrosynthetic biology approach to metabolic pathway design for therapeutic production. *BMC systems biology*, 5(1):122, 2011.

[210] A. Harder, B. I. Escher, and R. P. Schwarzenbach. Applicability and limitation of qsars for the toxicity of electrophilic chemicals. *Environmental science & technology*, 37(21):4955–4961, 2003.

[211] A.-G. Planson, P. Carbonell, E. Paillard, N. Pollet, and J.-L. Faulon. Compound toxicity screening and structure–activity relationship modeling in escherichia coli. *Biotechnology and Bioengineering*, 109(3):846–850, 2012.

[212] J. L. Ramos, E. Duque, M.-T. Gallegos, P. Godoy, M. I. Ramos-González, A. Rojas, W. Terán, and A. Segura. Mechanisms of solvent tolerance in gram-negative bacteria. *Annual Reviews in Microbiology*, 56(1):743–768, 2002.

[213] H. J. Heipieper, G. Neumann, S. Cornelissen, and F. Meinhardt. Solvent-tolerant bacteria for biotransformations in two-phase fermentation systems. *Applied microbiology and biotechnology*, 74(5):961–973, 2007.

[214] P. I. Nikel and V. de Lorenzo. Pseudomonas putida as a functional chassis for industrial biocatalysis: from native biochemistry to trans-metabolism. *Metabolic engineering*, 50:142–155, 2018.

[215] N. J. Wierckx, H. Ballerstedt, J. A. de Bont, and J. Wery. Engineering of solvent-tolerant pseudomonas putida s12 for bioproduction of phenol from glucose. *Applied and environmental microbiology*, 71(12):8221–8227, 2005.

[216] J.-P. Meijnen, S. Verhoef, A. A. Briedjlal, J. H. de Winde, and H. J. Ruijssenaars. Improved p-hydroxybenzoate production by engineered pseudomonas putida s12 by using a mixed-substrate feeding strategy. *Applied microbiology and biotechnology*, 90(3):885–893, 2011.

[217] I. Faizal, K. Dozen, C. S. Hong, A. Kuroda, N. Takiguchi, H. Ohtake, K. Takeda, H. Tsunekawa, and J. Kato. Isolation and characterization of solvent-tolerant pseudomonas putida strain t-57, and its application to biotransformation of toluene to cresol in a two-phase (organic-aqueous) system. *Journal of Industrial Microbiology and Biotechnology*, 32(11-12):542–547, 2005.

[218] R. S. Bose, S. Dey, S. Saha, C. K. Ghosh, and M. G. Chaudhuri. Enhanced removal of dissolved aniline from water under combined system of nano zerovalent iron and pseudomonas putida. *Sustainable Water Resources Management*, 2(2):143–159, 2016.

[219] E. S. Papadopoulou, C. Perruchon, S. Vasileiadis, C. Rousidou, G. Tanou, M. Samiotaki, A. Molassiotis, and D. G. Karpouzas. Metabolic and evolutionary insights in the transformation of diphenylamine by a pseudomonas putida strain unravelled by genomic, proteomic, and transcription analysis. *Frontiers in microbiology*, 9:676, 2018.

[220] D.-K. Ro, E. M. Paradise, M. Ouellet, K. J. Fisher, K. L. Newman, J. M. Ndungu, K. A. Ho, R. A. Eachus, T. S. Ham, J. Kirby, et al. Production of the antimalarial drug precursor artemisinic acid in engineered yeast. *Nature*, 440(7086):940–943, 2006.

[221] J. M. DeJong, Y. Liu, A. P. Bollon, R. M. Long, S. Jennewein, D. Williams, and R. B. Croteau. Genetic engineering of taxol biosynthetic genes in saccharomyces cerevisiae. *Biotechnology and bioengineering*, 93(2):212–224, 2006.

[222] H. Minami, J.-S. Kim, N. Ikezawa, T. Takemura, T. Katayama, H. Kumagai, and F. Sato. Microbial production of plant benzylisoquinoline alkaloids. *Proceedings of the National Academy of Sciences*, 105(21):7393–7398, 2008.

[223] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel, and B. Wiswedel. Knime-the konstanz information miner: version 2.0 and beyond. *AcM SIGKDD explorations Newsletter*, 11(1):26–31, 2009.

[224] M. A. Oberhardt, J. Puchałka, V. A. M. Dos Santos, and J. A. Papin. Reconciliation of genome-scale metabolic reconstructions for comparative systems analysis. *PLoS computational biology*, 7(3):e1001116, 2011.

[225] J. Nogales, J. Mueller, S. Gudmundsson, F. J. Canalejo, E. Duque, J. Monk, A. M. Feist, J. L. Ramos, W. Niu, and B. O. Palsson. High-quality genome-scale metabolic modelling of pseudomonas putida highlights its broad metabolic capabilities. *Environmental microbiology*, 22(1):255–269, 2020.

[226] S. R. Heller, A. McNaught, I. Pletnev, S. Stein, and D. Tchekhovskoi. Inchi, the iupac international chemical identifier. *Journal of cheminformatics*, 7(1): 23, 2015.

[227] J. Schellenberger, J. O. Park, T. M. Conrad, and B. Ø. Palsson. Bigg: a biochemical genetic and genomic knowledgebase of large scale metabolic reconstructions. *BMC bioinformatics*, 11(1):213, 2010.

[228] S. Moretti, O. Martin, T. Van Du Tran, A. Bridge, A. Morgat, and M. Pagni. Metanetx/mnxref–reconciliation of metabolites and biochemical reactions to bring together genome-scale metabolic networks. *Nucleic acids research*, 44(D1):D523–D526, 2015.

[229] D. Vrandečić and M. Krötzsch. Wikidata: a free collaborative knowledge base. 2014.

[230] G. Wohlgemuth, P. K. Haldiya, E. Willighagen, T. Kind, and O. Fiehn. The chemical translation service—a web-based tool to improve standardization of metabolomic reports. *Bioinformatics*, 26(20):2647–2648, 2010.

[231] A. Flamholz, E. Noor, A. Bar-Even, and R. Milo. equilibrator—the biochemical thermodynamics calculator. *Nucleic acids research*, 40(D1):D770–D775, 2011.

[232] E. Noor, A. Bar-Even, A. Flamholz, E. Reznik, W. Liebermeister, and R. Milo. Pathway thermodynamics highlights kinetic obstacles in central metabolism. *PLoS Comput Biol*, 10(2):e1003483, 2014.

[233] U. Consortium. Uniprot: a hub for protein information. *Nucleic acids research*, 43(D1):D204–D212, 2014.

[234] M. A. Oberhardt, B. O. Palsson, and J. A. Papin. Applications of genome-scale metabolic reconstructions. *Molecular Systems Biology*, 5:320, 2009. ISSN 1744-4292. doi: 10.1038/msb.2009.77.

[235] K. R. Patil, M. Åkesson, and J. Nielsen. Use of genome-scale microbial models for metabolic engineering. *Current opinion in biotechnology*, 15(1):64–69, 2004.

[236] C. Zhang and Q. Hua. Applications of Genome-Scale Metabolic Models in Biotechnology and Systems Medicine. *Frontiers in Physiology*, 6:413, 2015. ISSN 1664-042X. doi: 10.3389/fphys.2015.00413.

[237] A. Contreras, M. Ribbeck, G. D. Gutiérrez, P. M. Cañon, S. N. Mendoza, and E. Agosin. Mapping the physiological response of oenococcus oeni to ethanol stress using an extended genome-scale metabolic model. *Frontiers in microbiology*, 9:291, 2018.

[238] S. Gudmundsson, L. Agudo, and J. Nogales. Applications of genome-scale metabolic models of microalgae and cyanobacteria in biotechnology. In *Microalgae-Based Biofuels and Bioproducts*, pages 93–111. Elsevier, 2018.

[239] D. A. Cuevas, J. Edirisinghe, C. S. Henry, R. Overbeek, T. G. O'Connell, and R. A. Edwards. From dna to fba: how to build your own genome-scale metabolic model. *Frontiers in microbiology*, 7:907, 2016.

[240] M. DeJongh, K. Formsma, P. Boillot, J. Gould, M. Rycenga, and A. Best. Toward the automated generation of genome-scale metabolic networks in the seed. *BMC bioinformatics*, 8(1):139, 2007.

[241] P. D. Karp, S. Paley, and P. Romero. The pathway tools software. *Bioinformatics*, 18(suppl_1):S225–S232, 2002.

[242] R. Agren, L. Liu, S. Shoaie, W. Vongsangnak, I. Nookaew, and J. Nielsen. The raven toolbox and its use for generating a genome-scale metabolic model for penicillium chrysogenum. *PLoS computational biology*, 9(3):e1002980, 2013.

[243] J. P. Faria, M. Rocha, I. Rocha, and C. S. Henry. Methods for automated genome-scale metabolic model reconstruction. *Biochemical Society Transactions*, 46(4):931–936, 2018. ISSN 1470-8752. doi: 10.1042/BST20170246.

[244] P. D. Karp, M. Riley, S. M. Paley, and A. Pellegrini-Toole. The metacyc database. *Nucleic acids research*, 30(1):59–61, 2002.

[245] M. Kanehisa. The kegg database. In *'In Silico' Simulation of Biological Processes: Novartis Foundation Symposium 247*, volume 247, pages 91–103. Wiley Online Library, 2002.

[246] A. Ravikrishnan and K. Raman. Critical assessment of genome-scale metabolic networks: the need for a unified standard. *Briefings in bioinformatics*, 16(6):1057–1068, 2015.

[247] S. Heller, A. McNaught, S. Stein, D. Tchekhovskoi, and I. Pletnev. Inchi-the worldwide chemical structure identifier standard. *Journal of cheminformatics*, 5(1):7, 2013.

[248] C. Lieven, M. E. Beber, B. G. Olivier, F. T. Bergmann, M. Ataman, P. Babaei, J. A. Bartell, L. M. Blank, S. Chauhan, K. Correia, et al. Memote: A community-driven effort towards a standardized genome-scale metabolic model test suite. *BioRxiv*, page 350991, 2018.

[249] M. J. Herrgård, N. Swainston, P. Dobson, W. B. Dunn, K. Y. Arga, M. Arvas, N. Blüthgen, S. Borger, R. Costenoble, M. Heinemann, et al. A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology. *Nature biotechnology*, 26(10):1155–1160, 2008.

[250] R. G. van Heck, M. Ganter, V. A. M. dos Santos, and J. Stelling. Efficient reconstruction of predictive consensus metabolic network models. *PLoS Comput Biol*, 12(8):e1005085, 2016.

[251] J. L. Reed. Genome-scale metabolic modeling and its application to microbial communities. In *The Chemistry of Microbiomes: Proceedings of a Seminar Series*. National Academies Press, 2017.

[252] S. Magnúsdóttir, A. Heinken, L. Kutt, D. A. Ravcheev, E. Bauer, A. Noronha, K. Greenhalgh, C. Jäger, J. Baginska, P. Wilmes, R. M. T. Fleming, and I. Thiele. Generation of genome-scale metabolic reconstructions for 773 members of the human gut microbiota. *Nature Biotechnology*, 35(1):81–89, jan 2017. ISSN 1546-1696. doi: 10.1038/nbt.3703.

[253] D. Machado, S. Andrejev, M. Tramontano, and K. R. Patil. Fast automated reconstruction of genome-scale metabolic models for microbial species and communities. *Nucleic Acids Research*, 46(15):7542–7553, sep 2018. ISSN 0305-1048. doi: 10.1093/nar/gky537. URL https://academic.oup.com/nar/article/46/15/7542/5042022.

[254] M. Mednis and A. Vigants. Automatic comparison of metabolites names: impact of criteria thresholds. *Biosystems and Information Technology*, 2:1–5, May 2013. doi: 10.11592/bit.130501.

[255] X. Qi, Z. M. Ozsoyoglu, and G. Ozsoyoglu. Matching metabolites and reactions in different metabolic networks. *Methods*, 69(3):282–297, 2014.

[256] S. Moretti, O. Martin, T. Van Du Tran, A. Bridge, A. Morgat, and M. Pagni. Metanetx/mnxref–reconciliation of metabolites and biochemical reactions to bring together genome-scale metabolic networks. *Nucleic acids research*, page gkv1117, 2015.

[257] A. Kumar, P. F. Suthers, and C. D. Maranas. Metrxn: a knowledgebase of metabolites and reactions spanning metabolic models and databases. *BMC bioinformatics*, 13(1):1, 2012.

[258] T. Bernard, A. Bridge, A. Morgat, S. Moretti, I. Xenarios, and M. Pagni. Reconciliation of metabolites and biochemical reactions for metabolic networks. *Briefings in bioinformatics*, page bbs058, 2012.

[259] H. S. Haraldsdóttir, I. Thiele, and R. M. Fleming. Comparative evaluation of open source software for mapping between metabolite identifiers in metabolic network reconstructions: application to recon 2. *Journal of cheminformatics*, 6(1):1, 2014.

[260] A. J. Williams, S. Ekins, and V. Tkachenko. Towards a gold standard: regarding quality in public domain chemistry databases and approaches to improving the situation. *Drug discovery today*, 17(13-14):685–701, 2012.

[261] H. Redestig, M. Kusano, A. Fukushima, F. Matsuda, K. Saito, and M. Arita. Consolidating metabolite identifiers to enable contextual and multi-platform metabolomics data analysis. *BMC bioinformatics*, 11(1):214, 2010.

[262] S. A. Akhondi, S. Muresan, A. J. Williams, and J. A. Kors. Ambiguity of non-systematic chemical identifiers within and between small-molecule databases. *Journal of cheminformatics*, 7(1):1, 2015.

[263] A. A. Labena, Y.-Z. Gao, C. Dong, H.-l. Hua, and F.-B. Guo. Metabolic pathway databases and model repositories. *Quantitative Biology*, 6(1):30–39, March 2018. ISSN 2095-4697. doi: 10.1007/s40484-017-0108-3. URL https://doi.org/10.1007/s40484-017-0108-3.

[264] M. Latendresse. Efficiently gap-filling reaction networks. *BMC Bioinformatics*, 15(1):225, jun 2014. ISSN 1471-2105. doi: 10.1186/1471-2105-15-225. URL https://doi.org/10.1186/1471-2105-15-225.

[265] N. Christian, P. May, S. Kempa, T. Handorf, and O. Ebenhöh. An integrative approach towards completing genome-scale metabolic networks.

Molecular BioSystems, 5(12):1889–1903, nov 2009. ISSN 1742-2051. doi: 10.1039/B915913B. URL https://pubs.rsc.org/en/content/articlelanding/2009/mb/b915913b.

[266] S. A. Akhondi, J. A. Kors, and S. Muresan. Consistency of systematic chemical identifiers within and between small-molecule databases. *Journal of cheminformatics*, 4(1):1, 2012.

[267] A. Ravikrishnan and K. Raman. Critical assessment of genome-scale metabolic networks: the need for a unified standard. *Briefings in Bioinformatics*, 16(6):1057–1068, November 2015. ISSN 1477-4054. doi: 10.1093/bib/bbv003.

[268] W. Gottstein, B. G. Olivier, F. J. Bruggeman, and B. Teusink. Constraint-based stoichiometric modelling from single organisms to microbial communities. *Journal of The Royal Society Interface*, 13(124):20160627, nov 2016. ISSN 1742-5689, 1742-5662. doi: 10.1098/rsif.2016.0627. URL http://rsif.royalsocietypublishing.org/content/13/124/20160627.

[269] K. C. H. van der Ark, R. G. A. van Heck, V. A. P. Martins Dos Santos, C. Belzer, and W. M. de Vos. More than just a gut feeling: constraint-based genome-scale metabolic models for predicting functions of human intestinal microbes. *Microbiome*, 5(1):78, jul 2017. ISSN 2049-2618. doi: 10.1186/s40168-017-0299-x. URL https://doi.org/10.1186/s40168-017-0299-x.

[270] I. Thiele and B. Ø. Palsson. A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nature protocols*, 5(1):93–121, 2010.

[271] D. Young, T. Martin, R. Venkatapathy, and P. Harten. Are the chemical structures in your qsar correct? *QSAR & combinatorial science*, 27(11-12):1337–1345, 2008.

[272] M. L. Neal, M. König, D. Nickerson, G. Mısırlı, R. Kalbasi, A. Dräger, K. Atalag, V. Chelliah, M. Cooling, D. L. Cook, et al. Harmonizing semantic annotations for computational models in biology. *BioRxiv*, page 246470, 2018.

[273] A. A. Labena, Y.-Z. Gao, C. Dong, H.-l. Hua, and F.-B. Guo. Metabolic pathway databases and model repositories. *Quantitative Biology*, pages 1–10, 2018.

[274] Z. A. King, J. Lu, A. Dräger, P. Miller, S. Federowicz, J. A. Lerman, A. Ebrahim, B. O. Palsson, and N. E. Lewis. Bigg models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic acids research*, 44(D1):D515–D522, 2015.

[275] S. Devoid, R. Overbeek, M. DeJongh, V. Vonstein, A. A. Best, and C. Henry. Automated genome annotation and metabolic model reconstruction in the seed and model seed. In *Systems Metabolic Engineering*, pages 17–45. Springer, 2013.

[276] K. Degtyarenko, P. De Matos, M. Ennis, J. Hastings, M. Zbinden, A. McNaught, R. Alcántara, M. Darsow, M. Guedj, and M. Ashburner. Chebi: a database and ontology for chemical entities of biological interest. *Nucleic acids research*, 36(suppl_1):D344–D350, 2007.

[277] J. Wicker, T. Lorsbach, M. Gütlein, E. Schmid, D. Latino, S. Kramer, and K. Fenner. envipath–the environmental contaminant biotransformation pathway resource. *Nucleic acids research*, 44(D1):D502–D508, 2015.

[278] D. S. Wishart, D. Tzur, C. Knox, R. Eisner, A. C. Guo, N. Young, D. Cheng, K. Jewell, D. Arndt, S. Sawhney, et al. Hmdb: the human metabolome database. *Nucleic acids research*, 35(suppl_1):D521–D526, 2007.

[279] M. Sud, E. Fahy, D. Cotter, A. Brown, E. A. Dennis, C. K. Glass, A. H. Merrill Jr, R. C. Murphy, C. R. Raetz, D. W. Russell, et al. Lmsd: Lipid maps structure database. *Nucleic acids research*, 35(suppl_1):D527–D532, 2006.

[280] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. Gopinath, G. Wu, L. Matthews, et al. Reactome: a knowledgebase of biological pathways. *Nucleic acids research*, 33(suppl_1): D428–D432, 2005.

[281] U. Wittig, R. Kania, M. Golebiewski, M. Rey, L. Shi, L. Jong, E. Algaa, A. Weidemann, H. Sauer-Danzwith, S. Mir, et al. Sabio-rk—database for biochemical reaction kinetics. *Nucleic acids research*, 40(D1):D790–D796, 2011.

[282] L. Aimo, R. Liechti, N. Hyka-Nouspikel, A. Niknejad, A. Gleizes, L. Götz, D. Kuznetsov, F. P. David, F. G. van der Goot, H. Riezman, et al. The swisslipids knowledgebase for lipid biology. *Bioinformatics*, 31(17):2860–2866, 2015.

[283] D. Shepelin, A. S. L. Hansen, R. Lennen, H. Luo, and M. J. Herrgård. Selecting the best: evolutionary engineering of chemical production in microbes. *Genes*, 9(5):249, 2018.

[284] D. Liu, A. Hoynes-O'Connor, and F. Zhang. Bridging the gap between systems biology and synthetic biology. *Frontiers in microbiology*, 4:211, 2013.

[285] L. P. Wackett. Engineering microbes to produce biofuels. *Current opinion in biotechnology*, 22(3):388–393, 2011.

[286] J. M. Monk, C. J. Lloyd, E. Brunk, N. Mih, A. Sastry, Z. King, R. Takeuchi, W. Nomura, Z. Zhang, H. Mori, et al. i ml1515, a knowledgebase that computes escherichia coli traits. *Nature biotechnology*, 35(10):904–908, 2017.

[287] H. W. Aung, S. A. Henry, and L. P. Walker. Revising the representation of fatty acid, glycerolipid, and glycerophospholipid metabolism in the consensus model of yeast metabolism. *Industrial biotechnology*, 9(4):215–228, 2013.

[288] M. Spagnuolo, A. Yaguchi, and M. Blenner. Oleaginous yeast for biofuel and oleochemical production. *Current opinion in biotechnology*, 57:73–81, 2019.

[289] A. Dornau, J. F. Robson, G. H. Thomas, and S. J. McQueen-Mason. Robust microorganisms for biofuel and chemical production from municipal solid waste. *Microbial Cell Factories*, 19(1):1–18, 2020.

[290] N. K. Arora and H. Panosyan. Extremophiles: applications and roles in environmental sustainability, 2019.

[291] J. J. Koehorst, J. C. van Dam, E. Saccenti, V. A. Martins dos Santos, M. Suarez-Diez, and P. J. Schaap. Sapp: functional genome annotation and analysis through a semantic framework using fair principles. *Bioinformatics*, 34(8): 1401–1403, 2018.

[292] T. Seemann. Prokka: rapid prokaryotic genome annotation. *Bioinformatics*, 30(14):2068–2069, 2014.

[293] Y. Tanizawa, T. Fujisawa, and Y. Nakamura. Dfast: a flexible prokaryotic genome annotation pipeline for faster genome publication. *Bioinformatics*, 34(6):1037–1039, 2018.

[294] P. J. Turnbaugh, V. K. Ridaura, J. J. Faith, F. E. Rey, R. Knight, and J. I. Gordon. The effect of diet on the human gut microbiome: a metagenomic analysis in humanized gnotobiotic mice. *Science translational medicine*, 1(6):6ra14–6ra14, 2009.

[295] S. Louca, S. M. Jacques, A. P. Pires, J. S. Leal, D. S. Srivastava, L. W. Parfrey, V. F. Farjalla, and M. Doebeli. High taxonomic variability despite stable functional structure across microbial communities. *Nature ecology & evolution*, 1 (1):1–12, 2016.

[296] J. D. Orth and B. Ø. Palsson. Systematizing the generation of missing metabolic knowledge. *Biotechnology and bioengineering*, 107(3):403–412, 2010.

[297] K. C. H. van der Ark, S. Aalvink, M. Suarez-Diez, P. J. Schaap, W. M. de Vos, and C. Belzer. Model-driven design of a minimal medium for *Akkermansia muciniphila* confirms mucus adaptation. *Microbial Biotechnology*, Jan. 2018. ISSN 17517915. doi: 10.1111/1751-7915.13033. URL http://doi.wiley.com/10.1111/1751-7915.13033.

[298] S. Pan and J. L. Reed. Advances in gap-filling genome-scale metabolic models and model-driven experiments lead to novel metabolic discoveries. *Current opinion in biotechnology*, 51:103–108, 2018.

[299] C. S. Henry, M. DeJongh, A. A. Best, P. M. Frybarger, B. Linsay, and R. L. Stevens. High-throughput generation, optimization and analysis of genome-scale metabolic models. *Nature biotechnology*, 28(9):977, 2010.

[300] E. Pitkänen, P. Jouhten, J. Hou, M. F. Syed, P. Blomberg, J. Kludas, M. Oja, L. Holm, M. Penttilä, J. Rousu, et al. Comparative genome-scale reconstruction of gapless metabolic networks for present and ancestral species. *PLoS computational biology*, 10(2), 2014.

[301] P. D. Karp, M. Latendresse, S. M. Paley, M. Krummenacker, Q. D. Ong, R. Billington, A. Kothari, D. Weaver, T. Lee, P. Subhraveti, et al. Pathway tools version 19.0 update: software for pathway/genome informatics and systems biology. *Briefings in bioinformatics*, 17(5):877–890, 2016.

[302] H. Wang, S. Marcišauskas, B. J. Sánchez, I. Domenzain, D. Hermansson, R. Agren, J. Nielsen, and E. J. Kerkhoven. Raven 2.0: A versatile toolbox for metabolic network reconstruction and a case study on streptomyces coelicolor. *PLoS computational biology*, 14(10):e1006541, 2018.

[303] D. Machado, S. Andrejev, M. Tramontano, and K. R. Patil. Fast automated reconstruction of genome-scale metabolic models for microbial species and communities. *Nucleic acids research*, 46(15):7542–7553, 2018.

[304] M. Aite, M. Chevallier, C. Frioux, C. Trottier, J. Got, M. P. Cortés, S. N. Mendoza, G. Carrier, O. Dameron, N. Guillaudeux, et al. Traceability, reproducibility and wiki-exploration for "à-la-carte" reconstructions of genome-scale metabolic models. *PLoS computational biology*, 14(5):e1006146, 2018.

[305] M. Latendresse and P. D. Karp. Evaluation of reaction gap-filling accuracy by randomization. *BMC bioinformatics*, 19(1):53, 2018.

[306] M. Latendresse, M. Krummenacker, M. Trupp, and P. D. Karp. Construction and completion of flux balance models from pathway databases. *Bioinformatics*, 28(3):388–396, 2012.

[307] D. S. Weaver, I. M. Keseler, A. Mackie, I. T. Paulsen, and P. D. Karp. A genome-scale metabolic flux model of escherichia coli k–12 derived from the ecocyc database. *BMC systems biology*, 8(1):79, 2014.

[308] V. S. Kumar, M. S. Dasika, and C. D. Maranas. Optimization based automated curation of metabolic reconstructions. *BMC bioinformatics*, 8(1):212, 2007.

[309] I. Thiele, N. Vlassis, and R. M. Fleming. fastgapfill: efficient gap filling in metabolic networks. *Bioinformatics*, 30(17):2529–2531, 2014.

[310] N. Christian, P. May, S. Kempa, T. Handorf, and O. Ebenhöh. An integrative approach towards completing genome-scale metabolic networks. *Molecular BioSystems*, 5(12):1889–1903, 2009.

[311] E. Vitkin and T. Shlomi. Mirage: a functional genomics-based approach for metabolic network model reconstruction and its application to cyanobacteria networks. *Genome biology*, 13(11):R111, 2012.

[312] V. S. Kumar and C. D. Maranas. Growmatch: an automated method for reconciling in silico/in vivo growth predictions. *PLoS computational biology*, 5 (3):e1000308, 2009.

[313] J. L. Reed, T. R. Patel, K. H. Chen, A. R. Joyce, M. K. Applebee, C. D. Herring, O. T. Bui, E. M. Knight, S. S. Fong, and B. O. Palsson. Systems approach to refining genome annotation. *Proceedings of the National Academy of Sciences*, 103(46):17480–17484, 2006.

[314] T. Oyetunde, M. Zhang, Y. Chen, Y. Tang, and C. Lo. Boostgapfill: improving the fidelity of metabolic network reconstructions through integrated constraint and pattern-based methods. *Bioinformatics*, 33(4):608–611, 2016.

[315] M. N. Benedict, M. B. Mundy, C. S. Henry, N. Chia, and N. D. Price. Likelihood-based gene annotations for gap filling and quality assessment in genome-scale metabolic models. *PLoS computational biology*, 10(10): e1003882, 2014.

[316] B. King, T. Farrah, M. A. Richards, M. Mundy, E. Simeonidis, and N. D. Price. Probannoweb and probannopy: probabilistic annotation and gap-filling of metabolic reconstructions. *Bioinformatics*, 34(9):1594–1596, 2017.

[317] S. Prigent, C. Frioux, S. M. Dittami, S. Thiele, A. Larhlimi, G. Collet, F. Gutknecht, J. Got, D. Eveillard, J. Bourdon, et al. Meneco, a topology-based gap-filling tool applicable to degraded genome-wide metabolic networks. *PLoS computational biology*, 13(1):e1005276, 2017.

[318] A. Zupanic, H. C. Bernstein, and I. Heiland. Systems biology: current status and challenges. *Cellular and Molecular Life Sciences*, pages 1–2, 2020.

[319] Gapfillling. URL https://cobrapy.readthedocs.io/en/latest/gapfilling.html.

[320] N. Vlassis, M. P. Pacheco, and T. Sauter. Fast reconstruction of compact context-specific metabolic network models. *PLoS computational biology*, 10 (1):e1003424, 2014.

[321] A. P. Arkin, R. W. Cottingham, C. S. Henry, N. L. Harris, R. L. Stevens, S. Maslov, P. Dehal, D. Ware, F. Perez, S. Canon, et al. Kbase: the united states department of energy systems biology knowledgebase. *Nature biotechnology*, 36(7):566, 2018.

[322] L. Liu, Z. Zhang, T. Sheng, and M. Chen. Def: an automated dead-end filling approach based on quasi-endosymbiosis. *Bioinformatics*, 33(3):405–413, 2017.

[323] C. Frioux, T. Schaub, S. Schellhorn, A. Siegel, and P. Wanko. Hybrid metabolic network completion. In *International Conference on Logic Programming and Nonmonotonic Reasoning*, pages 308–321. Springer, 2017.

[324] W. L. Schroeder and R. Saha. Optfill: A tool for infeasible cycle-free gapfilling of stoichiometric metabolic models. *iScience*, 23(1):100783, 2020.

[325] M. J. Herrgård, S. S. Fong, and B. Ø. Palsson. Identification of genome-scale metabolic network models using experimentally measured flux profiles. *PLoS computational biology*, 2(7):e72, 2006.

[326] D. Hartleb, F. Jarre, and M. J. Lercher. Improved metabolic models for e. coli and mycoplasma genitalium from globalfit, an algorithm that simultaneously matches growth and non-growth data sets. *PLoS computational biology*, 12 (8):e1005036, 2016.

[327] G. Gelius-Dietrich, A. A. Desouki, C. J. Fritzemeier, and M. J. Lercher. Sybil–efficient constraint-based modelling in r. *BMC systems biology*, 7(1):125, 2013.

[328] M. B. Biggs and J. A. Papin. Metabolic network-guided binning of metagenomic sequence fragments. *Bioinformatics*, 32(6):867–874, 2016.

[329] Z. Hosseini and S.-A. Marashi. Discovering missing reactions of metabolic networks by using gene co-expression data. *Scientific reports*, 7:41774, 2017.

[330] M. B. Biggs and J. A. Papin. Managing uncertainty in metabolic network structure and improving predictions using ensemblefba. *PLoS computational biology*, 13(3):e1005413, 2017.

[331] N. Martyushenko and E. Almaas. Errortracer: an algorithm for identifying the origins of inconsistencies in genome-scale metabolic models. *Bioinformatics*, 2019.

[332] S. Pan, K. Nikolakakis, P. A. Adamczyk, M. Pan, E. G. Ruby, and J. L. Reed. Model-enabled gene search (megs) allows fast and direct discovery of enzymatic and transport gene functions in the marine bacterium vibrio fischeri. *Journal of Biological Chemistry*, 292(24):10250–10261, 2017.

[333] M. Ponce-de León, F. Montero, and J. Peretó. Solving gap metabolites and blocked reactions in genome-scale models: application to the metabolic network of blattabacterium cuenoti. *BMC systems biology*, 7(1):114, 2013.

[334] J. M. Dale, L. Popescu, and P. D. Karp. Machine learning methods for metabolic pathway prediction. *BMC bioinformatics*, 11(1):15, 2010.

[335] P. Rana, C. Berry, P. Ghosh, and S. S. Fong. Recent advances on constraint-based models by integrating machine learning. *Current Opinion in Biotechnology*, 64:85–91, 2020.

[336] G. L. Medlock and J. A. Papin. Guiding the refinement of biochemical knowledgebases with ensembles of metabolic networks and machine learning. *Cell Systems*, 2020.

[337] M. Ferris. Matlab and gams: Interfacing optimization and visualization software. Technical report, 1999.

[338] M. A. Carey, A. Dräger, M. E. Beber, J. A. Papin, and J. T. Yurkovich. Community standards to facilitate development and address challenges in metabolic modeling. *Molecular Systems Biology*, 16(8):e9235, 2020.

[339] C. J. Norsigian, N. Pusarla, J. L. McConn, J. T. Yurkovich, A. Dräger, B. O. Palsson, and Z. King. Bigg models 2020: multi-strain genome-scale models and expansion across the phylogenetic tree. *Nucleic acids research*, 48(D1): D402–D406, 2020.

[340] R. Caspi, R. Billington, L. Ferrer, H. Foerster, C. A. Fulcher, I. M. Keseler, A. Kothari, M. Krummenacker, M. Latendresse, L. A. Mueller, et al. The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic acids research*, 44(D1):D471–D480, 2015.

[341] S. N. Mendoza, B. G. Olivier, D. Molenaar, and B. Teusink. A systematic assessment of current genome-scale metabolic reconstruction tools. *Genome biology*, 20(1):1–20, 2019.

[342] P. Maia, M. Rocha, and I. Rocha. In silico constraint-based strain optimization methods: the quest for optimal cell factories. *Microbiology and Molecular Biology Reviews*, 80(1):45–67, 2016.

[343] A. F. Villaverde, D. Henriques, K. Smallbone, S. Bongard, J. Schmid, D. Cicin-Sain, A. Crombach, J. Saez-Rodriguez, K. Mauch, E. Balsa-Canto, et al. Biopredyn-bench: a suite of benchmark problems for dynamic modelling in systems biology. *BMC systems biology*, 9(1):8, 2015.

[344] M. Van Rijswijk, C. Beirnaert, C. Caron, M. Cascante, V. Dominguez, W. B. Dunn, T. M. Ebbels, F. Giacomoni, A. Gonzalez-Beltran, T. Hankemeier, et al. The future of metabolomics in elixir. *F1000Research*, 6, 2017.

[345] M. Kumar, B. Ji, P. Babaei, P. Das, D. Lappa, G. Ramakrishnan, T. E. Fox, R. Haque, W. A. Petri, F. Bäckhed, et al. Gut microbiota dysbiosis is associated with malnutrition and reduced plasma amino acid levels: lessons from genome-scale metabolic modeling. *Metabolic engineering*, 49:128–142, 2018.

[346] K. S. Ang, M. Lakshmanan, N.-R. Lee, and D.-Y. Lee. Metabolic modeling of microbial community interactions for health, environmental and biotechnological applications. *Current Genomics*, 19(8):712–722, 2018.

[347] M. Xenos. Usability perspective in software quality. In *Usability Engineering Workshop, The 8th Panhellenic Conference on Informatics with International Participation, Southern Cyprus*. Citeseer, 2001.

[348] Y. Peng, G. Wang, G. Kou, and Y. Shi. An empirical study of classification algorithm evaluation for financial risk prediction. *Applied Soft Computing*, 11 (2):2906–2915, 2011.

[349] H. Nouri, H. Fouladiha, H. Moghimi, and S.-A. Marashi. A reconciliation of genome-scale metabolic network model of zymomonas mobilis zm4. *Scientific Reports*, 10(1):1–11, 2020.

[350] C. Lieven, M. E. Beber, B. G. Olivier, F. T. Bergmann, M. Ataman, P. Babaei, J. A. Bartell, L. M. Blank, S. Chauhan, K. Correia, et al. Memote for standardized genome-scale metabolic model testing. *Nature biotechnology*, 38(3): 272–276, 2020.

[351] A. Hersey, J. Chambers, L. Bellis, A. P. Bento, A. Gaulton, and J. P. Overington. Chemical databases: curation or integration by user-defined equivalence? *Drug Discovery Today: Technologies*, 14:17–24, 2015.

[352] H. Hefzi, K. S. Ang, M. Hanscho, A. Bordbar, D. Ruckerbauer, M. Lakshmanan, C. A. Orellana, D. Baycin-Hizal, Y. Huang, D. Ley, et al. A consensus genome-scale reconstruction of chinese hamster ovary cell metabolism. *Cell Systems*, 3(5):434–443, 2016.

[353] A. Richelle, A. W. Chiang, C.-C. Kuo, and N. E. Lewis. Increasing consensus of context-specific metabolic models by integrating data-inferred cell functions. *PLoS computational biology*, 15(4):e1006867, 2019.

[354] S. Moretti, O. Martin, T. Van Du Tran, A. Bridge, A. Morgat, and M. Pagni. Metanetx/mnxref–reconciliation of metabolites and biochemical reactions to bring together genome-scale metabolic networks. *Nucleic acids research*, 44(D1):D523–D526, 2016.

[355] M. Boeckhout, G. A. Zielhuis, and A. L. Bredenoord. The fair guiding principles for data stewardship: fair enough? *European journal of human genetics*, 26(7):931, 2018.

[356] N. Juty, S. M. Wimalaratne, S. Soiland-Reyes, J. Kunze, C. A. Goble, and T. Clark. Unique, persistent, resolvable: Identifiers as the foundation of fair. *Data Intelligence*, 2(1-2):30–39, 2020.

[357] S. A. Akhondi, S. Muresan, A. J. Williams, and J. A. Kors. Ambiguity of non-systematic chemical identifiers within and between small-molecule databases. *Journal of cheminformatics*, 7(1):54, 2015.

[358] A. J. Williams. Public chemical compound databases. *Current Opinion in Drug Discovery and Development*, 11(3):393, 2008.

[359] A. Fabregat, F. Korninger, G. Viteri, K. Sidiropoulos, P. Marin-Garcia, P. Ping, G. Wu, L. Stein, P. D'Eustachio, and H. Hermjakob. Reactome graph

database: Efficient access to complex pathway data. *PLoS computational biology*, 14(1):e1005968, 2018.

[360] A. J. Williams. Chemspider: a platform for crowdsourced collaboration to curate data derived from public compound databases. *Collaborative computational technologies for biomedical research*, pages 363–386, 2011.

[361] M. Shardlow, N. Nguyen, G. Owen, C. O'Donovan, A. Leach, J. McNaught, S. Turner, and S. Ananiadou. A new corpus to support text mining for the curation of metabolites in the chebi database. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 280–285, 2018.

[362] D. Botero, J. Monk, M. J. Rodríguez Cubillos, A. Rodríguez Cubillos, M. Restrepo, V. Bernal-Galeano, A. Reyes, A. González Barrios, B. Ø. Palsson, S. Restrepo, et al. Genome-scale metabolic model of xanthomonas phaseoli pv. manihotis: an approach to elucidate pathogenicity at the metabolic level. *Frontiers in Genetics*, 11:837, 2020.

[363] A. Ahmad, A. Tiwari, and S. Srivastava. A genome-scale metabolic model of thalassiosira pseudonana ccmp 1335 for a systems-level understanding of its metabolism and biotechnological potential. *Microorganisms*, 8(9):1396, 2020.

[364] C. S. Jensen, C. J. Norsigian, X. Fang, X. C. Nielsen, J. J. Christensen, B. O. Palsson, and J. M. Monk. Reconstruction and validation of a genome-scale metabolic model of streptococcus oralis (icj415), a human commensal and opportunistic pathogen. *Frontiers in Genetics*, 11:116, 2020.

[365] S. Ofaim, R. Zarecki, S. Porob, D. Gat, T. Lahav, Y. Kashi, R. Aly, H. Eizenberg, Z. Ronen, and S. Freilich. Genome-scale reconstruction of paenarthrobacter aurescens tc1 metabolic model towards the study of atrazine bioremediation. *Scientific reports*, 10(1):1–11, 2020.

[366] J. Galgonek and J. Vondrášek. On inchi and evaluating the quality of cross-reference links. *Journal of Cheminformatics*, 6(1):15, 2014.

[367] A. Dalby, J. G. Nourse, W. D. Hounshell, A. K. Gushurst, D. L. Grier, B. A. Leland, and J. Laufer. Description of several chemical structure file formats used by computer programs developed at molecular design limited. *Journal of chemical information and computer sciences*, 32(3):244–255, 1992.

[368] S. A. Akhondi, J. A. Kors, and S. Muresan. Consistency of systematic chemical identifiers within and between small-molecule databases. *Journal of cheminformatics*, 4(1):35, 2012.

[369] Q. Chen, J. Zobel, and K. Verspoor. Duplicates, redundancies and inconsistencies in the primary nucleotide databases: a descriptive study. *Database*, 2017, 2017.

[370] X. Huang and Y.-H. Lin. Reconstruction and analysis of a three-compartment genome-scale metabolic model for pseudomonas fluorescens. *Biotechnology and applied biochemistry*, 67(1):133–139, 2020.

[371] J. Jeon and H. U. Kim. Setup of a scientific computing environment for computational biology: Simulation of a genome-scale metabolic model of escherichia coli as an example. *Journal of Microbiology*, 58(3):227–234, 2020.

[372] G.-H. Li, S. Dai, F. Han, W. Li, J. Huang, and W. Xiao. Fastmm: an efficient toolbox for personalized constraint-based metabolic modeling. *BMC bioinformatics*, 21(1):1–7, 2020.

[373] T. C. Keaty and P. A. Jensen. Gapsplit: Efficient random sampling for non-convex constraint-based models. *Bioinformatics*, 36(8):2623–2625, 2020.

[374] R. S. Malik-Sheriff, M. Glont, T. V. Nguyen, K. Tiwari, M. G. Roberts, A. Xavier, M. T. Vu, J. Men, M. Maire, S. Kananathan, et al. Biomodels—15 years of sharing computational models in life science. *Nucleic Acids Research*, 48(D1):D407–D415, 2020.

[375] V. Chelliah, N. Juty, I. Ajmera, R. Ali, M. Dumousseau, M. Glont, M. Hucka, G. Jalowicki, S. Keating, V. Knight-Schrijver, et al. Biomodels: ten-year anniversary. *Nucleic acids research*, 43(D1):D542–D548, 2015.

[376] V. L. Porubsky, A. P. Goldberg, A. K. Rampadarath, D. P. Nickerson, J. R. Karr, and H. M. Sauro. Best practices for making reproducible biochemical models. *Cell Systems*, 11(2):109–120, 2020.

[377] D. L. Moody and G. G. Shanks. What makes a good data model? evaluating the quality of entity relationship models. In *International Conference on Conceptual Modeling*, pages 94–111. Springer, 1994.

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

[379] J. Gilbert, N. Pearcy, R. Norman, T. Millat, K. Winzer, J. King, C. Hodgman, N. Minton, and J. Twycross. Gsmodutils: a python based framework for test-driven genome scale metabolic model development. *Bioinformatics*, 35(18): 3397–3403, 2019.

[380] Z. A. King, J. Lu, A. Dräger, P. Miller, S. Federowicz, J. A. Lerman, A. Ebrahim, B. O. Palsson, and N. E. Lewis. Bigg models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic acids research*, 44(D1):D515–D522, 2016.

[381] Q. Yuan, T. Huang, P. Li, T. Hao, F. Li, H. Ma, Z. Wang, X. Zhao, T. Chen, and I. Goryanin. Pathway-consensus approach to metabolic network reconstruction for pseudomonas putida kt2440 by systematic comparison of published models. *PloS one*, 12(1):e0169437, 2017.

[382] J. Nogales, S. Gudmundsson, E. Duque, J. L. Ramos, and B. O. Palsson. Expanding the computable reactome in pseudomonas putida reveals metabolic cycles providing robustness. *BioRxiv*, page 139121, 2017.

[383] L. Molina, R. L. Rosa, J. Nogales, and F. Rojo. Pseudomonas putida kt2440 metabolism undergoes sequential modifications during exponential growth in a complete medium as compounds are gradually consumed. *Environmental microbiology*, 21(7):2375–2390, 2019.

[384] G. Moss, P. Smith, and D. Tavernier. Glossary of class names of organic compounds and reactivity intermediates based on structure (iupac recommendations 1995). *Pure and applied chemistry*, 67(8-9):1307–1375, 1995.

[385] J. J. Tomashek and W. S. Brusilow. Stoichiometry of energy coupling by proton-translocating atpases: a history of variability. *Journal of bioenergetics and biomembranes*, 32(5):493–500, 2000.

[386] P. Turina, J. Petersen, and P. Gräber. Thermodynamics of proton transport coupled atp synthesis. *Biochimica Et Biophysica Acta (BBA)-Bioenergetics*, 1857(6):653–664, 2016.

[387] A. Gevorgyan, M. G. Poolman, and D. A. Fell. Detection of stoichiometric inconsistencies in biomolecular models. *Bioinformatics*, 24(19):2245–2251, 2008.

[388] A. M. Feist, C. S. Henry, J. L. Reed, M. Krummenacker, A. R. Joyce, P. D. Karp, L. J. Broadbelt, V. Hatzimanikatis, and B. Ø. Palsson. A genome-scale metabolic reconstruction for escherichia coli k-12 mg1655 that accounts for 1260 orfs and thermodynamic information. *Molecular systems biology*, 3(1): 121, 2007.

[389] J. B. van Duuren, J. Puchałka, A. E. Mars, R. Bücker, G. Eggink, C. Wittmann, and V. A. M. dos Santos. Reconciling in vivo and in silico key biological parameters of pseudomonas putida kt2440 during growth on glucose under carbon-limited condition. *BMC biotechnology*, 13(1):1–13, 2013.

[390] A. M. Feist, J. C. Scholten, B. Ø. Palsson, F. J. Brockman, and T. Ideker. Modeling methanogenesis with a genome-scale metabolic reconstruction of methanosarcina barkeri. *Molecular systems biology*, 2(1):2006–0004, 2006.

[391] H. W. Doelle. *Bacterial metabolism*. Academic Press, 2014.

[392] R. S. Tanner. Cultivation of bacteria and fungi. In *Manual of Environmental Microbiology, Third Edition*, pages 69–78. American Society of Microbiology, 2007.

[393] K. Rabaey and W. Verstraete. Microbial fuel cells: novel biotechnology for energy generation. *TRENDS in Biotechnology*, 23(6):291–298, 2005.

[394] T. Pfeiffer, S. Schuster, and S. Bonhoeffer. Cooperation and competition in the evolution of atp-producing pathways. *Science*, 292(5516):504–507, 2001.

[395] A. N. Brooks, S. Turkarslan, K. D. Beer, F. Yin Lo, and N. S. Baliga. Adaptation of cells to new environments. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 3(5):544–561, 2011.

[396] R. L. Chang, L. Ghamsari, A. Manichaikul, E. F. Hom, S. Balaji, W. Fu, Y. Shen, T. Hao, B. Ø. Palsson, K. Salehi-Ashtiani, et al. Metabolic network reconstruction of chlamydomonas offers insight into light-driven algal metabolism. *Molecular systems biology*, 7(1):518, 2011.

[397] L. Koduru, Y. Kim, J. Bang, M. Lakshmanan, N. S. Han, and D.-Y. Lee. Genome-scale modeling and transcriptome analysis of leuconostoc mesenteroides unravel the redox governed metabolic states in obligate heterofermentative lactic acid bacteria. *Scientific reports*, 7(1):1–15, 2017.

[398] C. M. O'Connor and J. U. Adams. *Essentials of Cell Biology*. Cambridge, MA: NPG Education, 2010.

[399] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. The chemical components of a cell. In *Molecular Biology of the Cell. 4th edition*. Garland Science, 2002.

[400] H. Yuan, C. Cheung, P. A. Hilbers, and N. A. van Riel. Flux balance analysis of plant metabolism: the effect of biomass composition and model structure on model predictions. *Frontiers in plant science*, 7:537, 2016.

[401] J. Pramanik and J. Keasling. Effect of escherichia coli biomass composition on central metabolic fluxes predicted by a stoichiometric model. *Biotechnology and bioengineering*, 60(2):230–238, 1998.

[402] M. MacGillivray, A. Ko, E. Gruber, M. Sawyer, E. Almaas, and A. Holder. Robust analysis of fluxes in genome-scale metabolic pathways. *Scientific reports*, 7(1):1–20, 2017.

[403] A. N. Kolmogorov and A. T. Bharucha-Reid. *Foundations of the theory of probability: Second English Edition*. Courier Dover Publications, 2018.

[404] F. Riedewald. *Comparison of deterministic, stochastic and fuzzy logic uncertainty modelling for capacity extension projects of DI/WFI pharmaceutical plant utilities with variable/dynamic demand*. PhD thesis, University College Cork, 2011.

[405] J. Chinneck and K. Ramadan. Linear programming with interval coefficients. *Journal of the operational research society*, 51(2):209–220, 2000.

[406] M. Inuiguchi and M. Sakawa. Minimax regret solution to linear programming problems with an interval objective function. *European Journal of Operational Research*, 86(3):526–536, 1995.

[407] M. Hladık. Tolerances in portfolio selection via interval linear programming. *A- A*, 1:2, 2008.

[408] K. K. Lai, S. Wang, J. Xu, S. Zhu, and Y. Fang. A class of linear interval programming problems and its application to portfolio selection. *IEEE Transactions on Fuzzy Systems*, 10(6):698–704, 2002.

[409] G. Cheng, G. Huang, and C. Dong. Convex contractive interval linear programming for resources and environmental systems management. *Stochastic Environmental Research and Risk Assessment*, 31(1):205–224, 2017.

[410] G. Huang, B. W. Baetz, and G. G. Patry. A grey linear programming approach for municipal solid waste management planning under uncertainty. *Civil Engineering Systems*, 9(4):319–335, 1992.

[411] P. D. Robers and A. Ben-Israel. Interval programming. new approach to linear programming with applications to chemical engineering problems. *Industrial & Engineering Chemistry Process Design and Development*, 8(4):496–501, 1969.

[412] H. M. Nehi, H. Ashayerinasab, and M. Allahdadi. Solving methods for interval linear programming problem: a review and an improved method. *Operational Research*, pages 1–25, 2018.

[413] A. Sengupta, T. K. Pal, and D. Chakraborty. Interpretation of inequality constraints involving interval coefficients and a solution to interval linear programming. *Fuzzy Sets and systems*, 119(1):129–138, 2001.

[414] C.-T. Li, J. Yelsky, Y. Chen, C. Zuñiga, R. Eng, L. Jiang, A. Shapiro, K.-W. Huang, K. Zengler, and M. J. Betenbaugh. Utilizing genome-scale models to optimize nutrient supply for sustained algal growth and lipid productivity. *NPJ systems biology and applications*, 5(1):1–11, 2019.

[415] J. E. Yang, S. J. Park, W. J. Kim, H. J. Kim, B. J. Kim, H. Lee, J. Shin, and S. Y. Lee. One-step fermentative production of aromatic polyesters from glucose by metabolically engineered escherichia coli strains. *Nature communications*, 9(1):1–10, 2018.

[416] R. Ledesma-Amaro, P. I. Nikel, and F. Ceroni. Synthetic biology-guided metabolic engineering. *Frontiers in Bioengineering and Biotechnology*, 8, 2020.

[417] C. Calmels, A. McCann, L. Malphettes, and M. R. Andersen. Application of a curated genome-scale metabolic model of cho dg44 to an industrial fed-batch process. *Metabolic engineering*, 51:9–19, 2019.

[418] R. García-Granados, J. A. Lerma-Escalera, and J. R. Morones-Ramírez. Metabolic engineering and synthetic biology: synergies, future, and challenges. *Frontiers in bioengineering and biotechnology*, 7:36, 2019.

[419] T. Jin, J. Lian, and L. R. Jarboe. Ethanol: a model biorenewable fuel. *Discovery of design strategies for enabling pyrolytic sugars tolerance and utilization by Escherichia coli*, 1001:8, 2016.

[420] J. Nielsen and J. D. Keasling. Engineering cellular metabolism. *Cell*, 164(6): 1185–1197, 2016.

[421] C. J. Vavricka, T. Hasunuma, and A. Kondo. Dynamic metabolomics for engineering biology: accelerating learning cycles for bioproduction. *Trends in biotechnology*, 38(1):68–82, 2020.

[422] L. R. Jarboe. Improving the success and impact of the metabolic engineering design, build, test, learn cycle by addressing proteins of unknown function. *Current opinion in biotechnology*, 53:93–98, 2018.

[423] X. Wang, Q. He, Y. Yang, J. Wang, K. Haning, Y. Hu, B. Wu, M. He, Y. Zhang, J. Bao, et al. Advances and prospects in metabolic engineering of zymomonas mobilis. *Metabolic engineering*, 50:57–73, 2018.

[424] H. A. Herrmann, B. C. Dyson, L. Vass, G. N. Johnson, and J.-M. Schwartz. Flux sampling is a powerful tool to study metabolism under changing environmental conditions. *NPJ systems biology and applications*, 5(1):1–8, 2019.

[425] A. R. Fernie and E. Pichersky. Focus issue on metabolism: metabolites, metabolites everywhere, 2015.

[426] D. Nicholson. A lifetime of metabolism. *Cell Mol Life Sci*, 63(1):1–5, 2005.

[427] J. Peretó. *Embden-Meyerhof-Parnas Pathway*, pages 485–485. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-11274-4. doi: 10.1007/978-3-642-11274-4_503. URL https://doi.org/10.1007/978-3-642-11274-4_503.

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

Acknowledgements

The PhD journey is a long and winding road which no one could finish alone. There were many people whose presence certainly brought something into this journey. This section is especially for those people.

My supervisors, **Vitor, Peter** and **Maria**, thanks for your support and tolerance in these four years. **Vitor** thank you very much for giving me the freedom to explore science and encouraging me in the EPP GA meetings. **Peter** your advice and honest feedback have shaped my research skills. **Maria** I cannot thank you enough for helping me to finish this PhD. You were always patient and empathetic to my problems throughout the whole journey. I often told you I would never finish the PhD but now it actually happens. I finished the book in four years. This is partly thanks to your belief and encouragement that boosted my motivation to continue. We are very lucky to have you in SSB.

Niru, my dear colleague and friend, thanks for everything that you have done to help me finish this PhD. You were always there when I needed you. You are the most patient friend that I ever had. You encourage and remind me to do things. You cheer me up. You feed me every now and then. You and **Rajaram** are always my reliable sources for advice.

Erika, my dear colleague, friend, and paranymph. We got closer only after you became my officemate. Still, we had so much fun together. You are a very dedicated friend, not only to me but to everyone that you like. You will help any of your friends when they need you. I admire the energy that you have for so many people.

Rik, my dear colleague, friend, and paranymph. I do not remember when and how we became friends. The topics at work became more interesting to discuss with

you. I appreciate that I get to know you and through you I get to know **Tjasa** and **Sasja**. I enjoyed every time we cooked together. Our common love for Asian cuisine helps with my homesickness a lot. Thanks for feeding me so many times, listening to my problems so many times, being next to me so many times. you made my days in Wageningen more meaningful. Thanks **Rik**, **Tjasa** and **Sasja**, my Wageningen family.

Ruben and **Dorett**, thanks for everything you did for me. I will never forget how the two of you helped me to stand up again after many things. You will always be a nice part of my life. I hope you will always be healthy, happy, and successful.

Maarten, **Bastian**, and **Benoit**, we only met and were colleagues shortly before you finished your PhDs and abandoned us. Still, you did make my days in SSB more exciting. It was always nice to hang out with you. **Maarten** all the meetings that we had were very 'fun'. **Bastian** talking to you is always nice. **Benoit** you were always a playful 'boy' in SSB, you made SSB livelier. Whatever you do next **Maarten**, **Bastian** and **Benoit**, I wish you lots of luck and success.

Bart, our zero digital-footprint computer scientist. I cannot find any picture of you online (I was looking for them to use in the presentation). Thanks a lot for all the nice talks that we had. You are a fun and nice colleague that everybody would want to work with.

Jasper the data geek in the group, thanks for all the nice discussion that we had. It is always fun to be your colleague. We had so many fun memories like the time six of us took a long bill in the Taste with **Erika**, **Niels** and ...I was too drunk to remember who else were there.

Niels, another SSB betrayer, you abandoned us and went for your Bitcoin company. It was nice to be your colleague and got to know Bitcoin, solar panels, and investment plans during our lunch breaks. I wish you success with your new career.

Melanie another awesome piece of 6032. We had so much fun together in the office that even annoy the supervisors. They planned to separate us but never succeeded. You are such an active person with all the things you do besides your job such as organizing the Popronde in Wageningen and the monthly Quiet is the new loud in the central library. I really enjoy these events, they make Wageningen livelier. Keep up with the good job **Melanie** so that people can have things to do in Wageningen.

Anna, we became close after organizing the PhD symposium together. It was a

very fun activity. I am happy that we got to do it together. You introduced me to GoalTrainers. Every week, we trained outside no matter if it was raining, freezing, or dark. I 'enjoyed' a lot all the exercises we did. I always felt death and reborn every time we finished them. Thanks for being in SSB and giving me these nice memories **Anna**.

Willemijn, the most efficient Secretary of SSB. You always respond swiftly to my emails. Thanks for helping SSB arranging so many things and making our lives easier with administrative works.

Stamati, **Emma**, **Linde**, **Anna Deneer**, **Nong**, **Marta**, **Sanjee**, **Sabine**, **Sara B.**, **Sara M.**, **Wasin**, **Christos**, **Enrique**, **Lyon**, **Maria M.**, **Luis**, **Cristina**, **Rob**, **Tom** and **Edo** the old and current members of SSB. Thanks for making SSB more enjoyable. I wish you all success with your PhDs and your careers.

Living far away from home is extremely difficult. Luckily, we have a small Vietnamese community in Wageningen. I would like to thank all my Vietnamese friends. I cannot list all of you out here but if you were with me when we gathered to make Vietnamese food, to celebrate Vietnamese holidays, or to speak Vietnamese with me, thank you for bringing home closer to me.

My family although being far, will always be there and encourage me in their special ways. I am grateful to still have you all with me.

And many other people who have had certain impacts on my PhD, thanks for making this part of my life memorable.

List of publications

Pham, N., Reijnders M., Suarez-Diez, M., Nijsse B., Springer J., Eggrink G., Schaap, P.J., Martins Dos Santos V. (2020). Genome-scale metabolic modelling underscores the potential of *Cutaneotrichosporon oleaginosus* ATCC 20509 as a cell factory for biofuel production. Manuscript accepted for publication in Biotechnology in Biofuels

Pham, N., van Heck, R. G., van Dam, J. C., Schaap, P. J., Saccenti, E., Suarez-Diez, M. (2019). Consistency, Inconsistency, and Ambiguity of Metabolite Names in Biochemical Databases Used for Genome-Scale Metabolic Modelling. Metabolites, 9(2), 28

Overview of completed training activities

Discipline specific activities	Organised by	Year
GA EPP meeting	EPP consortium	2017
Fairdom workshop	EPP consortium	2017
Quantitative and Predictive modelling	BioSB/WUR	2017
Big data in the life sciences	VLAG	2017
DD_DeCaF workshop	DSM	2017
Bioprocess Design	BiotechDelft/ VLAG	2017
GA EPP meeting	EPP consortium	2018
Managing and integrating life science	BioSB	2018
BioSB conference	BioSB	2018
COBRA conference	The International Metabolic Engineering Society and the Institute for Systems Biology	2018
BioSB conference 2019	BioSB	2019
Symposium "The Brave New World of Smart Data & Semantics in the Life Sciences"	WUR	2019
General courses		
workshop Carousel	WGS	2017
Introduction to R	VLAG	2017
VLAG PhD week	VLAG	2017
Competence assessment	WGS	2017
SW5		2017
Brain training	WGS	2017
Brain friendly working and writing	WGS	2018
Career orientation	WGS	2019
Supervising BSc MSc students	WGS	2018-2019
Optionals		
Preparation the research proposal	WUR-SSB	2017
Organizing the PhD symposium 'Science with Impact"	WUR	2019
Group meeting	WUR-SSB	2017-2019
Seminars	WUR-SSB	2017-2020

Colophon

The research described in this thesis was financially supported by the Eruopean Union Horizon2020 projects EmPowerPutida (Project reference: 635536).

Cover designed by Nhung Pham.
Thesis layout by Nhung Pham.
Printed by: ProefschrftMaken.

01000001 01110011 00100000 01101100 01101111 01101110 01100111 00100000 01100001 01110011 00100000 01111001 01101111 01110101 00100000 01100100 01101111 00100000 01101110 01101111 01110100 00100000 01100111 01101001 01110110 01100101 00100000 01110101 01110000 00101100 00100000 01111001 01101111 01110101 00100000 01110111 01101001 01101100 01101100 00100000 01100111 01100101 01110100 00100000 01110100 01101000 01100101 01110010 01100101