Design of Modular Cell Systems for Biocatalysis with Multi-Objective Optimization

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Sergio Garcia Manogil Fernandez

May 2020

Abstract

Modular design has been the cornerstone of contemporary engineering, enabling efficient production of exchangeable parts that interact in a reproducible manner to constitute functional systems. In this thesis, we transfer engineering modular design principles to the emerging fields of synthetic biology and metabolic engineering, that have promising applications to address problems related to health, energy, security, and the environment. We focus on microbial biocatalysis which can become a renewable and lower-cost replacement of traditional chemical synthesis processes. This thesis begins with an interdisciplinary review and perspective of the concepts, methodology, and applications of modular design. Then, we develop a conceptual, mathematical, and algorithmic framework based on multi-objective optimization theory to design modular cell biocatalysts. The proposed framework is used to design modular cell systems for renewable production of diverse biofuels and biochemicals, using genome-scale metabolic models of the organisms *Escherichia coli* and *Clostridium thermocellum* to simulate metabolic phenotypes. Overall, this thesis addresses the current modular design interest in synthetic biology through development of novel design principles and quantitative tools. We anticipate this modular cell design approach will not only bring whole-cell biocatalysis closer to being an industrially competitive technology, but also provide tools to understand the natural modular architectures of metabolic networks designed by evolution for billions of years under biological constraints.

Table of Contents

Introduction **1**

1 Modular design: Concepts, methods, and applications in engineering, biology, and biotechnology **3**

1.1 Introduction 4

1.2 Modularity in engineered systems 5

1.2.1 Basic concepts 5

1.2.2 Driving forces and potential tradeoffs of modular design 7

1.2.3 Theoretical frameworks of modular design 8

1.3 Modularity in biological systems 8

1.3.1 Modularity exists across all scales of biology 9

1.3.2 Modularity explains functions of components and interactions of biological systems 10

1.3.3 Modularity is a foundational tool to control programmable cells . . . 12

1.3.4 Modularity enables evolutionary advantage and robustness 12

1.4 Modular cell engineering 13

1.4.1 Recent developments in mathematical formulation of modular cell design 14

1.4.2 Recent advances in discovery and optimization of metabolic pathways as production modules 16

1.4.3 Recent advances in the experimental implementation of modular cell design 19

1.5 Conclusions 20

2 Formulation of conceptual and mathematical framework to design modular cells **22**

2.1 Introduction . . . 23

2.2 Methods . . . 25

2.2.1 Design principles of modular cell engineering . . . 25

2.2.2 Multi-objective strain design framework for modular cell engineering . 26

2.2.3 Algorithm and implementation . . . 31

2.2.4 Analysis methods for design solutions . . . 33

2.3 Results and discussion . . . 34

2.3.1 Illustrating ModCell2 for modular cell design of a simplified network . 34

2.3.2 Comparing ModCell2 designs with first-generation MODCELL and single product designs . . . 36

2.3.3 Exploring emergent features of modular cell design using an *E. coli* genome-scale network . . . 40

2.4 Conclusion . . . 44

3 Comparison of multi-objective evolutionary algorithms to solve the modular cell design problem **46**

3.1 Introduction . . . 47

3.2 Methods . . . 49

3.2.1 Multi-objective modular cell design . . . 49

3.2.2 Optimal solutions for a multi-objective optimization problem . . . 51

3.2.3 MOEA selection . . . 51

3.2.4 Performance metrics . . . 53

3.2.5 Algorithm parameters . . . 55

3.2.6 Metabolic models . . . 55

3.2.7 Implementation . . . 56

3.3 Results and Discussion . . . 56

3.3.1 Case 1: A 3-objectives design problem . . . 56

3.3.2 Case 2: A 10-objectives design problem . . . 58

3.3.3 Case 3: Use of large population size overcomes poor MOEA performance 58
3.4 Conclusions . . . 60

4 Development of linear formulations to solve the modular cell problem and application to design a universal modular cell 63
4.1 Introduction . . . 64
4.2 Methods . . . 66
4.2.1 Modular cell design . . . 66
4.2.2 Implementation . . . 76
4.2.3 Analysis methods . . . 78
4.3 Results and discussion . . . 80
4.3.1 Performance and solution time optimization of ModCell2-MILP . . . 80
4.3.2 Design of a universal modular cell for a genome-scale metabolic model of *E. coli* . . . 82
4.3.3 Flexible metabolic flux capacity of *E. coli* core metabolism enables the design of a universal modular cell . . . 86
4.4 Conclusions . . . 94

5 Development of an updated genome-scale metabolic model of *Clostridium thermocellum* and its application for integration of multi-omics datasets and modular cell design 101
5.1 Introduction . . . 102
5.2 Results . . . 105
5.2.1 Development of an upgraded *C. thermocellum* genome-scale model named iCBI655 . . . 105
5.2.2 Comparison of iCBI655 against other genome-scale models . . . 107
5.2.3 Training of model parameters under diverse conditions . . . 109
5.2.4 Assessment of model quality and standard compliance with Memote . 109
5.2.5 Model-guided analysis of proteomics and flux datasets sheds light on redox metabolism critical for biofuel production in *C. thermocellum* . 111
5.2.6 Model-guided design of modular production strains for biofuel synthesis 116

5.3 Conclusions . . . 118
5.4 Methods . . . 121
5.4.1 Model curation . . . 121
5.4.2 Metabolic flux simulations . . . 122
5.4.3 Simulation of different growth environments . . . 123
5.4.4 Single-reaction deletion analysis to match experimentally observed phenotype . . . 124
5.4.5 Model comparison . . . 125
5.4.6 Omics integration protocol . . . 125
5.4.7 Software implementation . . . 127
5.4.8 Proteomics data collection . . . 127
5.4.9 Modular cell design . . . 129

6 Design of modular cells for large product libraries 130
6.1 Introduction . . . 131
6.2 Methods . . . 133
6.2.1 Multi-objective optimization formulation of modular cell design problem 133
6.2.2 Implementation of many-objective evolutionary algorithm with high-performance computing . . . 135
6.2.3 Computation hardware . . . 136
6.2.4 Target product identification . . . 137
6.2.5 Model configuration . . . 137
6.2.6 Design characterization . . . 137
6.2.7 Coverage performance indicator . . . 139
6.3 Results . . . 139
6.3.1 Tuning of ModCell-HPC method parameters . . . 139
6.3.2 Design of *E. coli* modular cells for large product library . . . 141
6.3.3 Design of *E. coli* modular cells for conversion of hexoses and pentoses 147
6.3.4 Compatibility towards modules unknown at the time of chassis design 149
6.4 Conclusions . . . 152

Future directions **153**

Bibliography **156**

Appendices **191**

A Supplementary Material 1 for Chapter 2 192

B Supplementary Material 2 for Chapter 2 196

B.1 Solution method: Multiobjective Evolutionary Algorithm 196

B.2 Specifying the Set of Deletion Reaction Candidates for Manipulation 203

C Supplementary Material 1 for Chapter 5 206

D Supplementary Material 1 for Chapter 6 212

Vita **217**

List of Tables

3.1 Summary of MOEAs 52

4.1 Solution time reduction by tuning the ModCell2-MILP formulation 81
4.2 Overall production module pathway stoichiometries and associated simulated secretion fluxes of the universal modular cell design 88

5.1 Comparison of mutant growth rates predicted by iAT601 and iCBI655 108
5.2 Comparison of all genome-scale models of *C. thermocellum* 108
5.3 Reaction deletions sorted by appearance frequency (counts) in the designs of the Pareto front for $\alpha = 6$, $\beta = 0$. 120

6.1 Island-MOEA parameters evaluated in ModCell-HPC. 136
6.2 Top 20 reaction deletions for design parameters $\alpha = 5$, $\beta = 1$ 143
6.3 Top 10 reactions sorted by mean unknown compatibility contribution (*ucc*) among replicates. 151
C1 Consistent reactions in the Δ*hydG*-Δ*ech* case study 206
C2 Simulated fluxes for Δ*hydG*-Δ*ech* 210
C3 Reaction deletions sorted by appeareance frequency (counts) in the designs of the Pareto front for $\alpha = 6, \beta = 0$. 211

List of Figures

1.1 Modular design in engineering 6
1.2 Hierarchical modularity across all scales of biology 10
1.3 Generalized concept of modular cell design 15
1.4 Key advances and opportunities in the design and implementation of modular cells and exchangeable production modules 17

2.1 Comparison between the conventional single-product strain design and modular cell engineering 26
2.2 Graphical representation of phenotypic spaces for different strain design objectives 28
2.3 ModCell2 workflow and analysis 35
2.4 2-D metabolic phenotypic spaces of different *sGCP* designs using the core metabolic model 37
2.5 Comparison of strain design by OptKnock and Modcell2 39
2.6 Analysis of *wGCP* designs with genome-scale model 41
2.7 Production phenotypes of proposed designs 45

3.1 Conceptual illustration of performance metrics 55
3.2 Comparison of MOEAs for a 3-objectives design problem 57
3.3 Comparison of MOEAs for a 10-objective design problem 59
3.4 Comparison of MOEAs with increased population sizes 61
3.5 Wall-clock run times 62

4.1 Principles of modular cell design 67

4.2 Biochemical and metabolic diversity of the 20 production modules 77
4.3 Effect of design parameters on solution time 83
4.4 Metabolic functions of deletion candidate reactions 85
4.5 Identification of a universal modular cell compatible with all production modules under the *wGCP* design phenotype 87
4.6 Flexible metabolic flux capacity of *E. coli* metabolism enables the universal modular cell design 92
4.7 Violin plot of sampled reaction flux distributions 95
4.8 Sampled flux distributions of the ethanol biosynthesis pathways 96

5.1 Training of iCBI655 model 110
5.2 Multi-scale data integration procedure 113
5.3 Metabolic map visualization of proteomics data 115
5.4 Proposed modular cell designs for a *C. thermocellum* platform and 12 alcohols and esters 119

6.1 Parallelization schemes for multi-objective evolutionary algorithms. 133
6.2 ModCell-HPC benchmark with 161 products 142
6.3 Module reaction usage 145
6.4 Comparison of the designs in the selected minimal cover 146
6.5 Design of modular cells for different carbon sources 148
6.6 Compatibility towards unknown products in 3 random even partitions 150
i Developments in modular cell design tools 154
ii The Vulnerable World Hypothesis 155
A1 Software architecture of ModCell2 192
A2 Biochemical properties of production modules 193
A3 Robustness analysis of designs 194
A4 Generational distance among different design parameters 195
B1 Blocked and co-set reactions in deletion candidate determination 205
D1 Solution improvement process. 212
D2 Chemical properties of the product library 213

D3 ModCell-HPC benchmarking with 20 products 214
D4 Compatibility distributions 215
D5 Bipartite graph of minimal covers 216

Introduction

Vitalism was the notion that life arises from an essential principle that cannot be explained in terms of physical and chemical phenomena. The observable behavior of living organisms (e.g., growth, reproduction) remained a mystery to science. Eduard Buchner's discovery of what would later be called enzymes during the late 19th century provided one of the final blows to vitalism, opening the door to great developments in our mechanistic understanding of living systems through the 20th century. This knowledge has recently enabled bioengineers to manipulate cellular DNA for diverse applications ranging from renewable chemical production to medical treatment. Despite recent success stories, such applied fields remain highly constrained by a lack of models with sufficient predictive and explanatory power to bridge the gap between first principles and emerging phenomena. Finding such models for biological systems might be an unprecedented challenge, and remains a remarkable opportunity for scientific discovery. In addition to this scientific challenge, our increasing capabilities to engineer biological systems need to be complemented by design principles and methods that will ensure efficient development of predictable devices. Among potential applications, biocatalysis technologies can broaden the accessibility of essential goods, such as food and medication, and reduce global warming, the two issues that can become the major drivers of civil conflict during the 21st century.[14,111,66]

Successful development of biocatalysis technologies currently relies on genetic engineering techniques in combination with abstract systems engineering concepts. Among such abstractions, modular design is key in modern engineering. Computers, vehicles, factories, and many other complex devices are often assembled from exchangeable units of self-contained functionality known as modules. By analogy with electrical engineering, synthetic

biologists have aimed to build biological parts that enable modular assembly and re-usability in different contexts.[242] However, most modularization efforts in this field have been applied qualitatively and in isolation to specific cellular components (e.g., modular metabolic pathways,[17], modular proteins,[175] modular genetic circuits[246]) rather than at the system level,[216] i.e., the cell is considered as a chassis compatible with modules that enable desired functionality. Hence, such approaches are difficult to generalize and do not explore the potential advantages of building whole-cell modular systems, such as platform strains[196] for diverse chemical production. In this thesis we advance modular design principles in synthetic biology and metabolic engineering with emphasis on biocatalysis applications. Unlike previous modularization approaches, we consider modular design at the system level and provide quantitative tools for its application. This view is developed in Chapter 1, which contains a review of modularity across the engineering, biology, and bioengineering literature.

The main contribution of this thesis is divided into three areas: i) The formulation of general modular cell design principles and associated mathematical models (Chapters 1, 2); ii) the implementation of scalable algorithms to solve the mathematical models (Chapters 3, 4, 6); and iii) the application of the resulting modular cell design tool to understand principles of natural biological modularity and to design modular biocatalyst strains based on industrially-relevant hosts and production modules (Chapters 2, 4, 5, 6). In summary, the outcome of this research is to bring modularity principles proven in conventional engineering to metabolic engineering developing more efficient and robust cellular systems; hence lowering the R&D costs that remain a major roadblock for widespread industrial application of microbial catalysis.

Chapter 1

Modular design: Concepts, methods, and applications in engineering, biology, and biotechnology

This chapter is based on the publication *Modular design: Implementing proven engineering principles in biotechnology. Garcia, S., and Trinh, C. T. Biotechnology Advances, Volume 37, Issue 7, 2019.* As first author I lead the development, implementation, and writing of this study.

Abstract

Modular design is at the foundation of contemporary engineering, enabling rapid, efficient, and reproducible construction and maintenance of complex systems across applications. Remarkably, modularity has recently been discovered as a governing principle in natural biological systems from genes to proteins to complex networks within a cell and organism communities. The convergent knowledge of natural and engineered modular systems provides a key to drive modern biotechnology to address emergent challenges associated with health, food, energy, and the environment. Here, we first present the theory and application of modular design in traditional engineering fields. We then discuss the significance and impact of modular architectures on systems biology and biotechnology. Next, we focus on the very

recent theoretical and experimental advances in modular cell engineering that seeks to enable rapid and systematic development of microbial catalysts capable of efficiently synthesizing a large space of useful chemicals. We conclude with an outlook towards theoretical and practical opportunities for a more systematic and effective application of modular engineering in biotechnology.

1.1 Introduction

Complex engineered systems such as computers, vehicles, or factories can be assembled from exchangeable units of self-contained functionality known as modules. Modular design enables efficient production, maintenance, and customization across modern engineering technologies. The inception of modular design has had a revolutionary impact on many industries. For instance, the first modular computer, named IBM System/360 and built in the 1960s, allowed re-using the same software for different application-dependent hardware, shaping information technology as we know it today[203]. Undoubtedly, modular design will continue to drive innovation in both established and emergent fields of engineering.

Among trending engineering disciplines, biotechnology is encompassing far-reaching applications driven by the recent development of enabling technologies in interdisciplinary areas of genome engineering[15], systems and synthetic biology[124]. metabolic engineering[196], and bioprocessing[54,202]. Amid many applications to address issues related to health, energy, and the environment, the chemical industry will benefit from metabolic reprogramming of microbes as cell factories to catalyze the synthesis of therapeutics, chemicals, and fuels, from renewable and sustainable feedstocks (e.g., lignocellulosic biomass, sugar cane) or waste products (e.g., waste gas from steel manufacturing, plastic waste). Even though there exists a naturally large space of molecules that can be synthesized by metabolically engineered microorganisms[150], fewer than a dozen molecules are industrially produced[196]. A major roadblock is attributed to the very laborious and costly strain engineering process partly arising from the lack of standardization and repetition of genetic manipulation tasks[137,295]. Recently, modular cell engineering has been proposed as an innovative approach to accelerate strain engineering process, harnessing a large space of molecules derived from rich and

diverse cellular metabolism and thus pushing whole-cell biocatalysis towards an industrially competitive technology[273].

In this paper, we first examine the theory and application of modular design in conventional engineering disciplines, such as mechanical, chemical, nuclear, and civil engineering, with the aim to provide perspectives and innovative methods that can be transferred to biotechnology. We next present the importance of modularity that exists in natural biological systems. Finally, we highlight the most recent theoretical and experimental developments in modular cell design for synthetic biology and metabolic engineering applications.

1.2 Modularity in engineered systems

1.2.1 Basic concepts

Based on the definition by Miller and Elgard[185], a module is "an essential and self-contained functional unit relative to the product of which it is part. The module has, relative to a system definition, standardized interfaces and interactions that allow composition of products by combination". We find this definition, out of the many available[228], to be general yet descriptive enough to illustrate the topics in this paper. Based on the above definition, modules must have a standardized interface and exchangeability in order to enable rapid and systematic assembly of components into a system with various types of modular architectures[279]. A fully modular or sectional architecture has all the components to be modules, for instance, fluid pipes or sectional couches, while an integral architecture lacks any type of modules (Figure 1.1 A). Chassis-based architectures are also common in modular design, including bus and slot. The bus modular architecture uses the same interface for all modules, e.g., universal serial bus (USB) and peripheral component interconnect (PCI) ports found in computers, while the slot modular architecture has specific interfaces for corresponding modules, e.g., tires in an automobile. The chassis-based architectures enable the use of alternative chasses that can be combined with the same modules to efficiently generate a variety of products or vice versa[120].

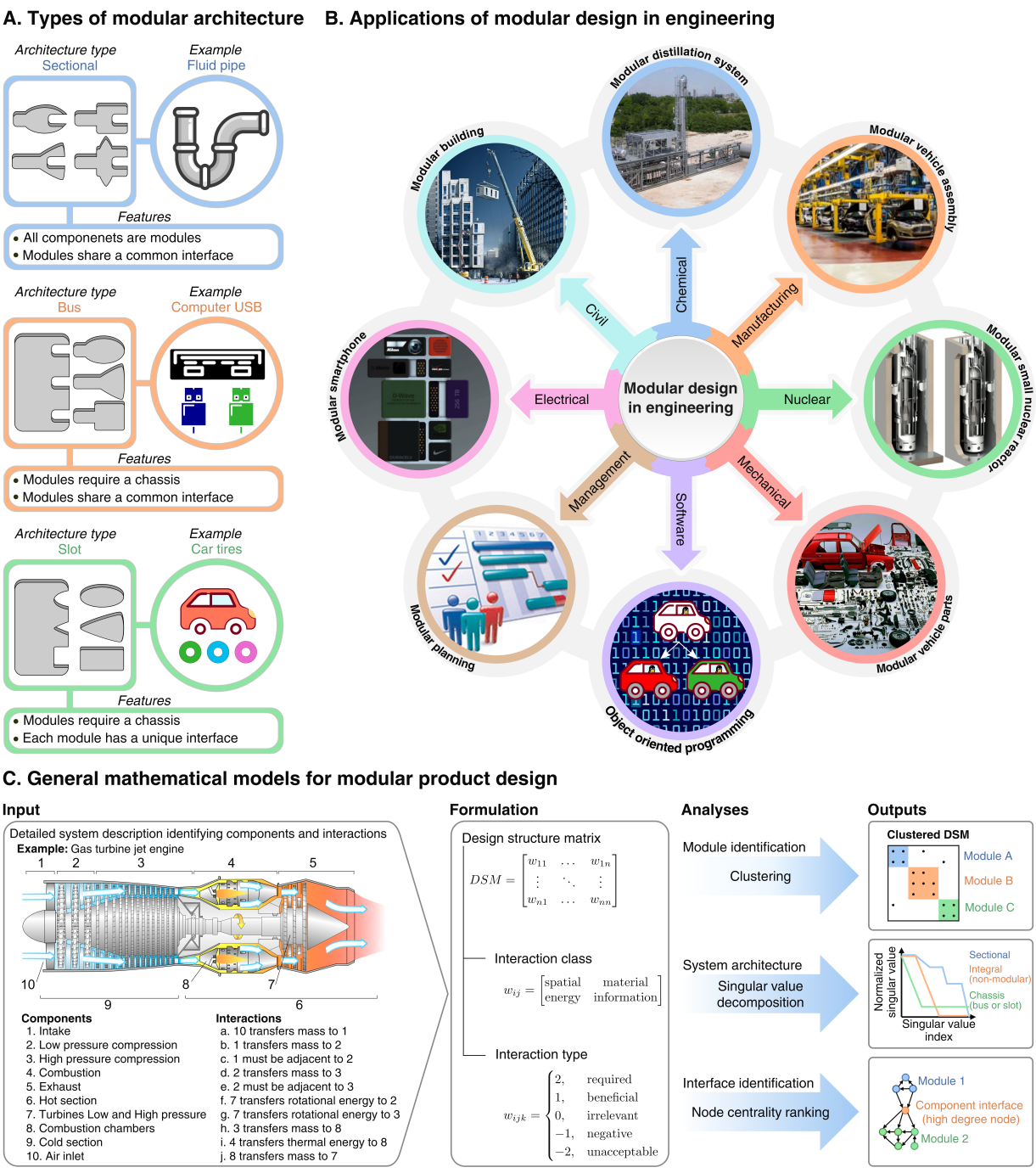


Figure 1.1: Modular design in engineering. (**A**) Common types of modular architectures. (**B**) Current applications of innovative modular design. (**C**) General mathematical framework of modular design. The input is illustrated with a gas turbine jet engine, that intakes air through the front section for heating and compression followed by air expansion to generate thrust. The interaction between system components can be formalized in a DSM model and analyzed to identify the most effective modules and interfaces.

1.2.2 Driving forces and potential tradeoffs of modular design

The driving forces for system modularization are to achieve increased efficiency and robustness, reduced complexity and cost, and better customization and maintenance options[24,185]. Modular design has been the core of many innovative technologies across engineering disciplines (Figure 1.1 B). For instance, modularized plants in chemical engineering allow faster and more cost-effective deployment, making small operations viable and hence providing economic and environmental advantages[13,134]. Likewise, modular buildings in civil engineering enable more rapid and economical construction[127]. In nuclear engineering, the use of small modularized reactors overcomes potential hazards of traditional large-scale operations, allows plant customization to energetic demand, and reduces construction and manufacturing costs[284]. The emerging area of highly automated manufacturing also implements modular production systems[290]. In addition, modular design principles have been applied with great success in abstract engineering disciplines such as software[2,82] and management engineering[33]. The decomposition of software elements into modules of defined functionality is essential to manage complexity, ensure robustness, and allow for concurrent development. Similarly, organizations and projects can be structured into modules to accomplish parallel task execution and avoid repetition.

Even though modular design is ubiquitous, it may not be desirable or feasible in every circumstance. The disadvantages and limitations of modularity tend to be field specific. When modularity is part of an innovative approach, such as modular chemical plants, a lack of experience and higher upfront costs are regarded as common drawbacks[13]. Design constraints may also limit the applicability of modularity; for example, a quantitative study[114] suggested that the portability requirements of cell phones and laptops makes them less modular than their static counterparts. In the case of buildings and chemical plants, the size of components may prevent off-site modular manufacturing if transportation is unfeasible. Thus, when choosing modular design for engineering applications, it is important to ensure that advantages outweigh disadvantages.

1.2.3 Theoretical frameworks of modular design

Modular product design is complex and field-specific but can be generally formulated using the language of graph theory. The Design Structure Matrix (DSM)[29] is a commonly used technique to model a system as a graph, where nodes represent basic components and links between nodes describe their functional interactions. Component interactions can be represented in a binary manner (i.e., whether a relationship exists or not) or in a detailed complex fashion with multiple dimensions and values to increase model accuracy. For example, with a complex interaction, a metric can be used to quantitatively assign interaction desirability between two components, i.e., a high desirability score if they require each other to work or a negative desirability score otherwise. In some cases, detailed interaction types can also be classified and integrated in the modular design; for example, a cooling system with a radiator and a fan is required to have not only a *spatial* interaction (i.e., both elements need to be in close proximity) but also a *material* interaction (i.e., the fan provides airflow across the radiator) (Figure 1.1 C)[106].

DSM can reveal properties of the system through different analyses, including singular value decomposition to capture the modular architecture type (e.g. integral, chassis-based, bus, or slot)[114], node centrality ranking to identify key interactions between modules and interfaces[247], and most importantly, clustering to identify modules (Figure 1.1 C). While many approaches exist to cluster a DSM, not all can successfully identify modules due to underlying design conflicts in product modularization; for these scenarios, integrated use of a highly descriptive DSM model to account for complex interactions and multi-objective optimization to identify Pareto optimal solutions is needed to accurately design modular systems[106].

1.3 Modularity in biological systems

In the high-throughput and quantitative era of biology, modularity has become a fundamental abstraction in understanding and redesigning biological systems that have existed and evolved for billions of years[102,286]. A variety of definitions of biological modules co-exist, arising from the multi-scale and multi-interaction nature of biological systems. From a

mathematical perspective, two general approaches are commonly used to define modules: (i) modules as clusters of highly interconnected nodes in a biological interaction network, and (ii) modules as programmable circuits that can be described with laws of mass and energy conservation and control theory. These paradigms differ in that network modules provide a holistic and simplified description, while circuit modules provide a reductionistic and detailed description. Despite their differences, both approaches seek to understand the evolutionary origin of modules, their role in fundamental biological properties such as robustness and evolvability, and their biotechnological applications.

1.3.1 Modularity exists across all scales of biology

Modularity is a ubiquitous organizing principle across all scales of biology. Within a scale, modules often interact in a hierarchical manner (Figure 1.2). At the molecular level, DNA transcription activation and rate can be controlled by a variety of modular promoter elements[69]. Additionally, certain proteins are highly modular, such as enzymes (e.g., polyketide synthase and non-ribosomal peptide synthetase[112]) with modular substrate identification elements that enable biosynthesis of a large space of secondary metabolites[132]. At the cellular level, modularity is present in all biomolecule interaction networks[187], including RNAs[251], proteins[248], and metabolites[219]. In all cases, modules are associated with specific cellular functions or pathways, which often interact in a hierarchical manner[219]. At the multi-cellular level, organs and tissues are also structured modularly. For example, in the human brain, neuron interaction networks contain modules associated with specific cognitive functions[249]. These modules are hierarchically organized into submodules to integrate and contextualize specialized functions of the brain[183]; for instance, visual perception that requires the functional integration of multiple neuron clusters in the cortical columns[207]. At the ecological scale, organism communities can be represented by networks, where nodes are species or subpopulations and links are interactions such as consumption, pollination, or competition[97]. The capability of ecological networks to avoid global failure due to small perturbations has been attributed to their modular structure both theoretically[97] and experimentally[94].

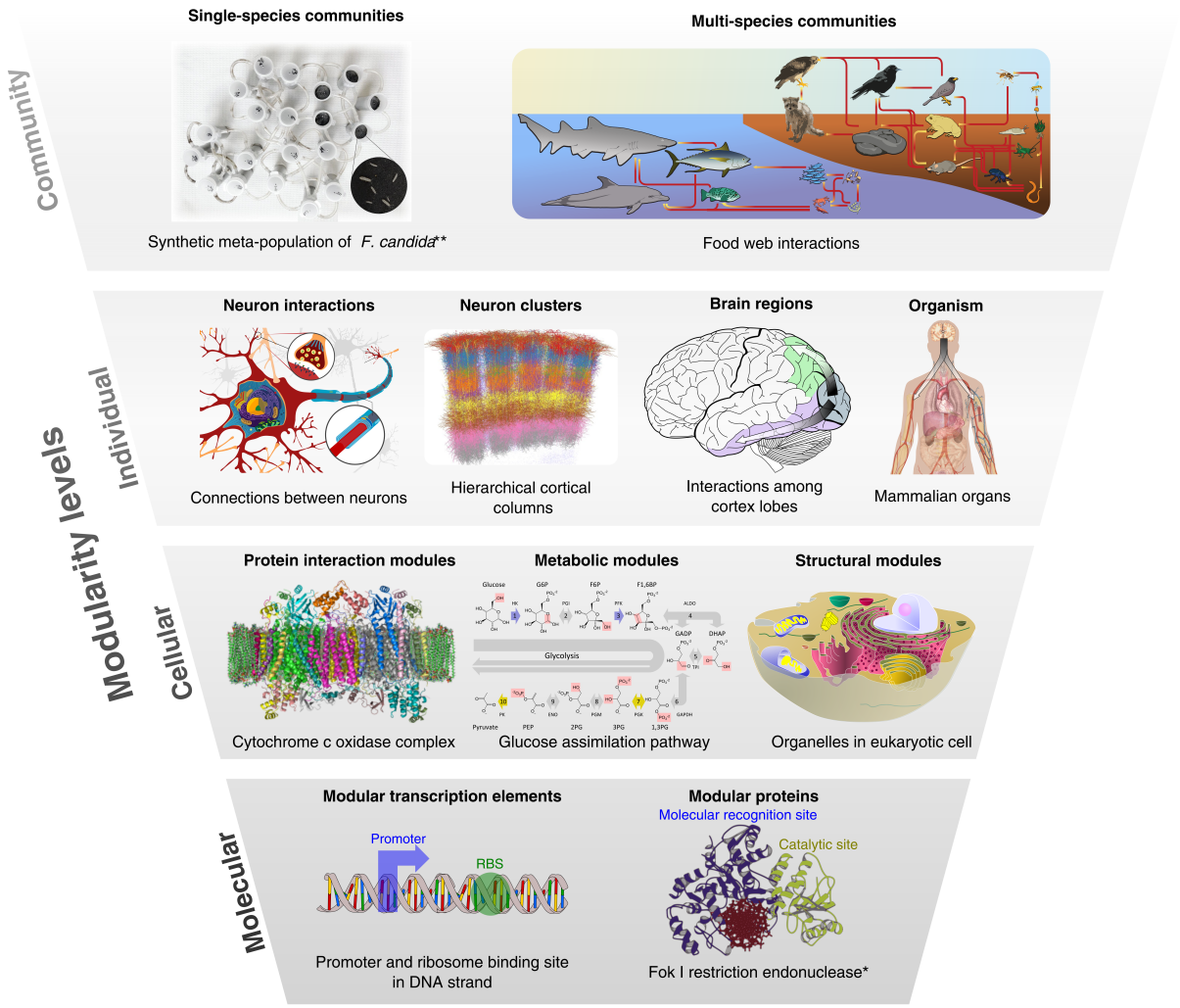


Figure 1.2: Hierarchical modularity across all scales of biology. Images marked with * and ** are adapted from Khosla and Harbury[132] and Gilarranz *et al.*[94], respectively.

1.3.2 Modularity explains functions of components and interactions of biological systems

Prior to the systems biology era, the view of modular biological systems can be traced back as early as the late 19$^{\text{th}}$ century with the initial study of what later became known as glycolysis to investigate how yeast fermentation made wine have a good taste. Throughout the first half of the 20$^{\text{th}}$ century, the complete knowledge of many major metabolic pathways

of well-defined functionality across organisms, including the EMP pathway, the Entner-Doudoroff pathway, Krebs cycle, pentose phosphate pathway, and so on, was established. These pathways were mapped to qualitatively describe the modular interconnection of functional elements within a cell, i.e., cellular metabolism that governs cell physiology. Pioneering work in the 1980s, including the comprehensive description of measurable bacterial cellular components[191] and constraint-based simulations of microbial metabolism[75], helped establish a foundation for quantitative modular analysis of cell physiology. With the explosion of high-throughput 'omics technologies in the late 90's, complex biological systems with thousands of interacting elements can now be studied holistically and quantitatively at multiple levels from genes to proteins and metabolites within the cell and microbial communities[224,167,209,294]. Graphs (or networks) can represent these systems[7] and their complex interactions, e.g., metabolites-enzymes[171,219], genes-diseases[95], protein-protein[253], or a combination of all known protein and genetic interactions[39]. Module analysis of biological networks can provide two insights: (i) identification of modules that represent transferable and self-contained functions[187], for example, a cis-regulatory element and its associated genes[200], the subunits of a protein complex[98,103], and the genes associated with a disease phenotype[95], and (ii) interactions among modules of a system. For example, by analyzing the metabolic networks of 43 organisms, Ravasz *et al.*[219] identified a hierarchical modular architecture containing hubs of highly connected metabolites and modules with specific metabolic functions. Remarkably, the identified architecture overlaps with the known primary and secondary metabolic pathways. Likewise, by analyzing metabolic networks of 63 organisms, Ma and Zeng[171] could identify a bow-tie architecture of the network containing a core of highly interconnected components linking input and output node clusters. Graph-based analysis of metabolic networks also revealed a small-world architecture (i.e., a small number of reactions between any two metabolites) that was hypothesized to confer cellular metabolism with the capability to quickly adapt to perturbations[285].

1.3.3 Modularity is a foundational tool to control programmable cells

Genetic circuits are modules that define a universal programming language of cells, such as logical gates[28,196], and can be widely found in natural biological systems. Classical examples of genetic circuits are the lac operon that enables carbon catabolite repression in bacteria and the MAPK/ERK signaling pathway in mammalian cells that controls cell division among other cellular functions. Modularity of natural signaling pathways can be harnessed for novel functions[160,215], including the production of valuable metabolites, synthesis of nanomaterials, treatment of disease, and sensors to detect hazardous molecules[28]. The laws of mass and energy conservation and principles of control theory that have been well developed in traditional engineering disciplines can be applied to enable modular design of synthetic biological systems. For instance, Del Vecchio *et al.*[63] developed a mathematical model of retroactivity that captures the impact of a downstream module on the function of an upstream module, and used this model to enhance module insulation. Even though synthetic genetic circuit modules operate correctly and reliably in an isolated environment, it remains challenging to integrate these modules into complex systems[216].

1.3.4 Modularity enables evolutionary advantage and robustness

Biological robustness is the ability of a system to maintain its function upon genetic and environmental perturbations. Both the graph- and circuit-based descriptions of modules[291] can help explain their contributions to system robustness. For example, the bow-tie architecture of biological networks[80] has been suggested[138] to enhance the biological robustness of chassis-based modularity, where the essential core processes belong to the conserved chassis and adaptable modules allow for evolution to experiment safely. This view, however, raises the question of the evolutionary origin and function of modules. It has been hypothesized that modules may arise due to natural selection or biased mutational mechanisms. Computational studies on the topic are abundant but compatible with both hypotheses[286]. Several models have suggested that when a single fitness goal is present, a modular architecture does not emerge since it does not provide a fitness advantage. However,

when multiple fitness goals are pursued, either sequentially[129] or simultaneously[47], a modular architecture is favored. Additionally, it has been demonstrated that macroscopic[244] and molecular[234] phenotypes are weighted combinations of optimal phenotypes specialized for single tasks. This view is important in the context of modular cell engineering that can be formulated as a multi-objective optimization problem[86], where each fitness goal corresponds to a module for making a desirable chemical. Thus, modular cell engineering can harness the existing features of biological modularity instead of creating entirely synthetic properties.

1.4 Modular cell engineering

Diverse and complex cellular metabolism encompasses a large space of molecules, providing a potential path towards broader industrialization of biology[53]. To realize this potential, rapid development of novel microbial biocatalysts to efficiently synthesize these molecules is critical but faces multiple challenges due to laborious and costly requirement of extensive strain optimization cycles[196,273]. Effective exploitation of modular design as seen in natural and engineered systems for biocatalyst development can offer innovative strategies to tackle these challenges.

To date, modular cell engineering has been mostly applied at the pathway level, where enzyme module expression is adjusted to increase target metabolite production. This topic has been extensively reviewed[17,118,169,298] and will not be elaborated in detail here. At the cellular level, platform strains engineered to eliminate common byproducts and increase availability of important precursors for overproduction of target molecules have been reported with some success by implementing the conventional "push-and-pull" metabolic engineering strategy[196]. More recently, a system-level modular cell design method has been developed to systematically and simultaneously design both the chassis cell and production modules for the synthesis of various target chemicals[86,272]. Unlike conventional strain optimization methods that target one product, modular cell design seeks to enable rapid and predictable creation of multiple production strains to achieve superior performance with minimal strain optimization cycle where each synthesizes a different product. Each optimal production strain is obtained by assembling a reusable modular (chassis) cell with an exchangeable production module(s)

in a plug-and-play fashion, resembling the advantages of modular design in traditional engineering disciplines. Specifically, a modular cell contains core metabolic phenotypes shared among production modules (Figure 1.3 A). The chassis interfaces with the modules through enzyme synthesis machinery and precursor metabolites (Figure 1.3 B). Modules contain auxiliary regulatory and metabolic pathways (Figure 1.3 C) that enable a desired phenotype for optimal biosynthesis of a target molecule, such as growth-coupled-to-product formation (*GCP* design) or stationary-phase product synthesis (*NGP* design) (Figure 1.3 D). These design principles are formulated to integrate state-of-the-art techniques developed in the fields of synthetic biology and metabolic engineering, including, (1) computational models that identify genetic interventions towards desirable phenotypes, (2) rapid discovery and optimization of metabolic pathways for target product synthesis (i.e., production modules), and (3) design of minimal cells and orthogonal pathways to generate a toolkit of parts that can be assembled into various functional systems.

In the following sections, we highlight the recent advancements in modular cell engineering that capture: (1) design principles and computational tools to enable construction of a modular cell and associated production modules, (2) discovery and optimization of metabolic pathways as production modules both theoretically and experimentally, and (3) experimental implementation of modular cell design principles.

1.4.1 Recent developments in mathematical formulation of modular cell design

The primer for the modular cell design principles started with the observation[269] that the optimal design of the n-butanol- and isobutanol-producing *E. coli* cells, based on constraint-based modeling[204] and conventional strain design methods[275], exhibited the same core metabolism (Figure 1.4 A-B). To systematically explore this property, the first modular cell design method, called MODCELL, was proposed and used to design an *E. coli* modular cell for alcohol and ester production[272] (Figure 1.4 E). In Chapter 2 a new formulation of the modular cell design problem, ModCell2, based on multi-objective optimization will be

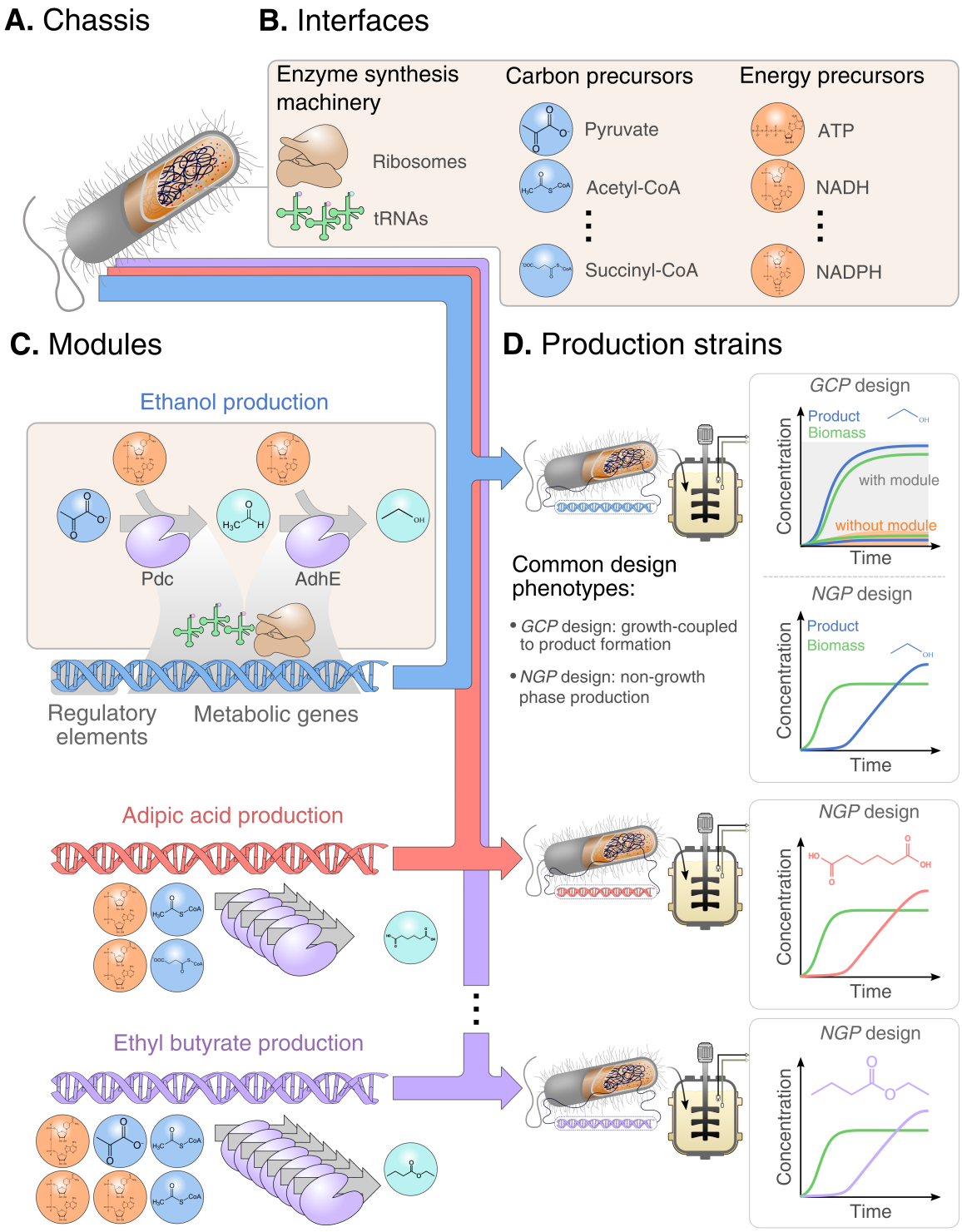


Figure 1.3: Generalized concept of modular cell design. (**A**) Modular (chassis) cell. (**B**) Interfaces. (**C**) Production modules. (**D**) Production strains. A modular cell is designed to provide the necessary precursors for biosynthesis pathway modules that are independently assembled with the modular cell to generate production strains exhibiting desirable phenotypes.

developed to design strains with minimal trade-off between compatibility, performance, and robustness (Figure 1.4 H).

While ModCell2 is to our knowledge the only tool that involves simultaneous design of chassis, modules, and interfaces, other recently developed computational tools can be applied to design these elements individually. For example, MinGenome[289] is used to design genomes of minimal cells that can serve as chassis whereas ValveFind[205] enables design of orthogonal pathways to build production modules. Topics on model-guided strain design techniques for "integral" strains, .i.e, strains optimized for production of only a single product, can be found in recent excellent reviews.[166,174,194].

Future development of modular cell design tools will incorporate enzyme kinetics of cellular metabolism[136,199] to account for potential metabolic burdens[296] in the design of modular cells and exchangeable production modules (Figure 1.4 I). Enzyme cost analysis is also particularly useful for identifying robust strategies for adaptive laboratory evolution that prevent an unintended pathway(s) to be optimized instead of a targeted pathway(s)[65]. The lack of experimental data on enzyme catalytic efficiencies needed for these modeling approaches can be addressed through random parameter sampling[65], *omics* integration[70,131], and machine learning[104]. Recently developed models that predict *in-vivo* enzyme concentrations from the genetic sequence help bridge the gap between metabolic model predictions and experimental implementation[72,182,227].

1.4.2 Recent advances in discovery and optimization of metabolic pathways as production modules

With the arrival of quantitative 'omics era in mid 1990s, we started to gain a deeper understanding of complex biological systems and categorize them into the functional biological parts, i.e., regulatory and functional genes for retrieval through development of biological databases (e.g., Registry of Standard Biological Parts (https://parts.igem.org), KEGG[128], Biocyc[36], Brenda[231], among many others). Synergistically, the interdisciplinary areas of bioinformatics, synthetic biology, metabolic and protein engineering have also emerged, advanced, and now enabled rapid and systematic identification and assembly of

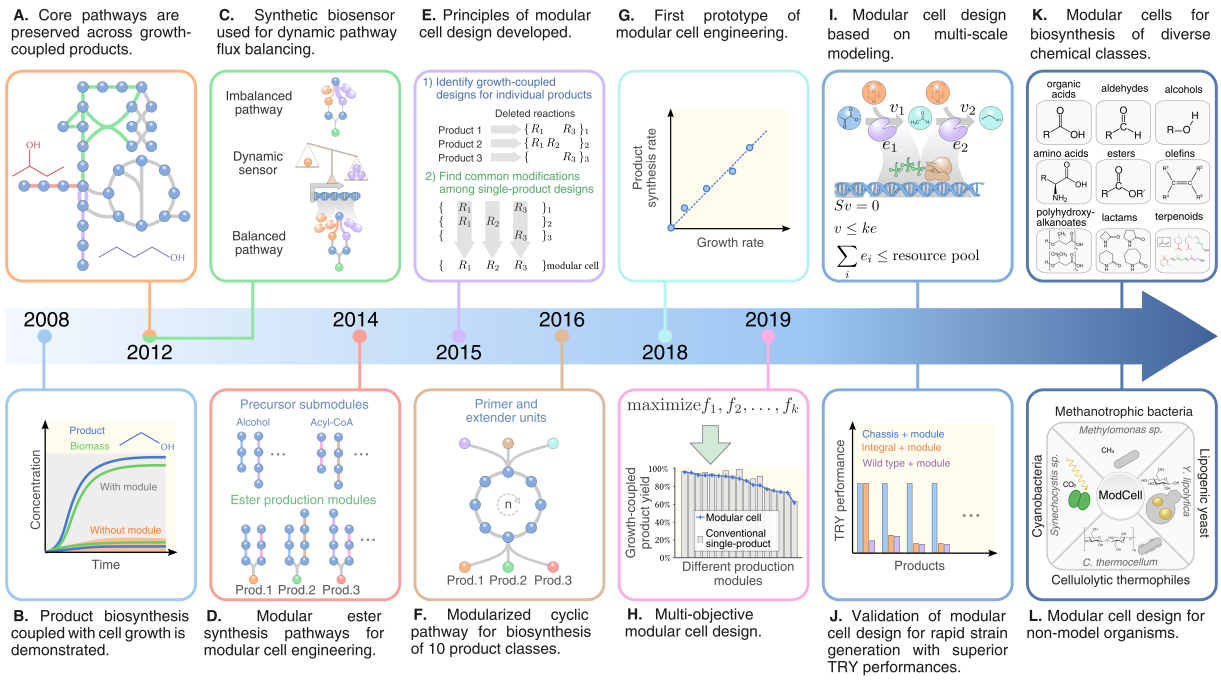


Figure 1.4: Key advances and opportunities in the design and implementation of modular cells and exchangeable production modules. (**A**) The same core metabolic pathways were revealed for butanol and isobutanol growth-coupled production based on elementary mode analysis[269]. (**B**) An *E. coli* cell with minimal metabolic functionality designed to require product (ethanol) synthesis for cell growth was experimentally validated[274]. (**C**) A dynamic sensor-regulator system was designed and implemented to balance metabolic fluxes for enhanced biosynthesis of fatty acid-derived molecules[307]. (**D**) A modular ester fermentative pathway platform derived from alcohol and acyl-CoA pathway submodules was designed and experimentally demonstrated in a chassis strain[146]. (**E**) First modular cell design method, named MODCELL, was proposed and used to design 3 modular cells for the production of a group of alcohols and derived esters[272]. (**F**). Carbon and energy efficient cyclic pathway was developed to synthesize 10 product classes from a variety of primer and extender precursors[43]. (**G**) Prototypes of modular cell engineering were demonstrated for growth-coupled to product synthesis predicted by MODCELL[272] and pathway optimization by adaptive laboratory evolution[293]. (**H**) Multi-objective optimization-based modular cell design method, named ModCell2, will be developed in Chapter 2 and used to reveal negligible trade-offs between modular and integral designs. (**I**) Next-generation of modular cell design framework is proposed to account for enzyme biosynthesis cost using metabolism and expression (ME) or kinetic models with enzyme constraints. (**J**) Experimental demonstration of rapid and systematic generation of modular production strains with many different types of production modules to achieve superior performances over the wildtype and single-product (integral) strains in terms of titer, rate, and yield (TRY). (**K**) Design of a universal modular cell compatible with a large and diverse space of production modules. (**L**) Future demonstration of modular cell design is proposed for industrially-relevant, non-model organisms with difficult-to-transfer phenotypes, such as efficient assimilation of CO_2, CH_4, or cellulose.

these biological parts into production modules to probe a large space of molecules that are only limited by one's imagination. Nowadays, development of production modules can start by using computational tools with increasing scope and accuracy[144,288], that help identify metabolic steps, associated enzymes, and genetic parts (i.e., promoters, terminators, ribosome binding sites, regulatory/sensory elements, and so on) using a combination of elementary reaction rules, yield, thermodynamic, and biophysical analyses[68,144]. Next, the genetic parts can be synthesized, assembled, and characterized for accuracy in a rapid manner to build production modules to make desirable molecules in a target host[8,20,21,41,51,93,141,158,155,240,276,278,308]. Some recent achievements, highlighting innovations in retrofitting cellular metabolism for novel biocatalysis from sustainable feedstocks, include: (i) redirection of central metabolism to make environmentally friendly, non-natural bioplastics[220], (ii) redesign of fermentative pathways to produce a large space of designer bioesters used as flavors, fragrances, biofuels, and solvents[146,224] (Figure 1.4 D), (iii) repurposing of the beta-oxidation pathway for combinatorial biosynthesis of alcohols, dicarboxylic acids, hydroxyl acids, and lactones as industrial platform chemicals[43] (Figure 1.4 F), and (iv) refactoring of polyketide and isoprenoid pathways to explore a large space of secondary metabolites as drugs[3,81,179].

While the design, construction, and characterization of production modules can be streamlined, compatibility between the modules and a target host has always posed a significant challenge mainly due to the intricate flux imbalance resulting in low product titers, rates, and yields[196]. In the case of heterologous enzyme expression, undesirable regulatory interactions at the transcriptional and metabolic levels might hinder the pathway operation in a new host. This problem can be addressed by refactoring pathways into isolated orthogonal modules that are independent of the source organism regulation and do not interfere with heterologous host regulation[81,229,255,257]. In most design scenarios, even after regulatory issues are addressed, metabolic flux imbalance is likely to occur due to the differences in expression and catalytic efficiencies of pathway enzymes. Metabolic flux balancing of engineered pathways is a combinatorial optimization problem that requires modification of gene expression elements (e.g., promoters, ribosome binding sites, etc.) and enzyme engineering to achieve the desired fluxes. Currently, this combinatorial

problem is often tackled in two ways: (i) identification of key design variables and screening and (ii) pathway selection. The search space in variable screening approaches can be reduced by pathway-level modularization, known as Multivariate Modular Metabolic Engineering[17,118,298]. Screening approaches can be effort-intensive and impractical for certain scenarios due to the extensive strain characterizations required to effectively sample the design space. Additionally, these approaches may not be able to solve poor pathway-host interactions where precursor metabolite(s) of the pathway of interest becomes the bottleneck. Alternatively, simultaneous host and pathway optimization can be accomplished by adaptive laboratory evolution, provided that a simple selectable phenotype such as growth is tightly coupled with the desirable product synthesis phenotype. For example, Wilbanks *et al.*[293] recently demonstrated a linear correlation between growth rate and product synthesis rate for computationally designed growth-coupled strains[272] (Figure 1.4 G). Pathway optimization through growth-coupled design has been applied successfully[77], and a recent computational study suggest its applicability to many different products and organisms.[282] Modular cell design is compatible with the two described module optimization tools; particularly, the design of a chassis growth-coupled to its production modules can enable rapid optimization of diverse pathways.

1.4.3 Recent advances in the experimental implementation of modular cell design

Experimental implementation of modular cell design is still at infancy of validation. Construction of modular cells can be implemented by two methods: (i) top-down approach that aims to remove undesired phenotypes from a naturally existing organism under defined environmental conditions[274,270] and (ii) bottom-up approach that seeks to create a new organism derived from a synthetic minimal cell.[113] These two methods draw the same analogy as those in the engineering modular design, classified as "product modularization" and "design with modules"[24]. In the "product modularization" approach like the "top-down" approach, the chassis, modules, and interfaces are simultaneously designed either by defining functional carriers and interfaces as part of the design (*ex ante* design) or by

clustering existing functions into modules (*ex post* design). Alternatively, in the "design with modules" approach like the "bottom-up" approach, a product is designed out of a collection of predefined compatible parts; for example, the design of a personal computer that is assembled from the existing modules, including a motherboard, a graphic card, a monitor, etc., with standard interfaces. In the modular cell design context, a minimal cell can function as chassis whereas orthogonal refactored pathways that function as modules are independently designed and combined with the chassis to build modular production strains with desirable phenotypes.

To date, all recent computational[86,272] and experimental[146,293] efforts in implementing modular cell design have been focused on the top-down approach, since it is more feasible and accessible with the current knowledge and available genetic tools. Even though the bottom-up approach is much more challenging[113], the design principles developed for top-down construction can be applied to create bottom-up minimal modular cells, which would be less prone to failure due to their simpler architecture. Regardless of which approach is chosen, the underlying genotypes essential to target product synthesis appear to be conserved, according to the recent surveys of over two decades of metabolic engineering reports[137,295]. This suggests that modular cell design can serve as a unifying platform for rapid strain engineering (Figure 1.4 J).

An anticipated challenge of modular cell design is that the chassis must provide enough precursor metabolites and enzyme synthesis machinery to support the target flux through each module. This issue may become increasingly difficult as the biochemical diversity and number of products supported by a single chassis expands (Figure 1.4 K). Recent developments in biosensors coupled with gene expression regulation tools[163,184,189,300,307] can achieve tunable control over the host metabolism to meet the requirements of each production module (Figure 1.4 C). In practice, such regulatory elements may be implemented in the host or as a part of specific modules.

1.5 Conclusions

Inspired by natural and conventional engineering modularity, bioengineers have started applying modular design principles to engineer biological systems at genetic, enzymatic, and cellular levels. Modular cell design aims to integrate all three levels for rapidly creating novel microbial biocatalysts in a plug-and-play fashion with minimal strain optimization cycles. Advancements in genome reading[96], writing[35,142], and editing[15] will provide a unique opportunity to streamline modular cell engineering that effectively harnesses a large space of molecules from cellular metabolism using single organisms or microbial consortia, especially from non-model organisms with industrially-relevant but not-easy-to-transfer traits (Figure 1.4 L)[1,34,126,170,260]. Particularly, these advancements help streamline the construction of modular cells and exchangeable production modules from both top-down and bottom-up approaches. To further advance modular cell engineering, it is important not only to optimize the interfaces between production modules and modular cell but also to account for robustness and evolvability towards desirable engineered phenotypes. By combining proven engineering methods rooted in the physical and chemical laws with system modeling frameworks (e.g., Pareto optimality theory, graph theory), we can elucidate the modular design principles in biology from natural systems to engineered ones, leading towards fundamental understanding of essential rules of life and broader industrialization of biology.

Chapter 2

Formulation of conceptual and mathematical framework to design modular cells

This chapter is based on the publication *Multiobjective strain design: A framework for modular cell engineering. Garcia, S., and Trinh, C. T. Metabolic Engineering, Volume 51, Pages 110-120, 2019.* As first author I lead the development, implementation, and writing of this study. Supplementary Files S1 and S2 are provided in Appendix A and B respectively, while Supplementary Files S3, S4, and S5 are provided as attachments.

Abstract

Diversity of cellular metabolism can be harnessed to produce a large space of molecules. However, development of optimal strains with high product titers, rates, and yields required for industrial production is laborious and expensive. To accelerate the strain engineering process, we have recently introduced a modular cell design concept that enables rapid generation of optimal production strains by systematically assembling a modular cell with an exchangeable production module(s) to produce target molecules efficiently. In this study, we formulated the modular cell design concept as a general multi-objective optimization problem with flexible design objectives derived from mass action. We developed algorithms and an

associated software package, named ModCell2 to implement the design. We demonstrated that ModCell2 can systematically identify genetic modifications to design modular cells that can couple with a variety of production modules and exhibit a minimal tradeoff among modularity, performance, and robustness. Analysis of the modular cell designs revealed both intuitive and complex metabolic architectures enabling modular production of these molecules. We envision ModCell2 provides a powerful tool to guide modular cell engineering and sheds light on modular design principles of biological systems.

2.1 Introduction

Engineering microbial cells to produce bulk and specialty chemicals from renewable and sustainable feedstocks is becoming a feasible alternative to traditional chemical methods that rely on petroleum feedstocks[196]. However, only a handful of chemicals, out of the many possible molecules offered by nature, are industrially produced by microbial conversion, mainly because the current strain engineering process is laborious and expensive for profitable biochemical production[273]. Thus, innovative technologies to enable rapid and economical strain engineering are needed to harness a large space of industrially-relevant molecules[53].

The modular organization of biological systems has been a source of inspiration for synthetic biology and metabolic engineering[216,230]. Modular pathway engineering breaks down target pathways into tractable pathway modules that can be finely tuned for optimal production of desirable chemicals[17,298]. Harnessing combinatorial pathways (e.g., fatty acid biosynthesis, reverse beta oxidation, polyketide or isoprenoid biosynthesis) is one excellent example of modular pathway engineering. These pathways contain metabolic similarity (or combinatorial characteristics) such as a group of common specific enzymes capable of catalyzing linear reaction steps[224] and/or elongation cycles[43,277,297] and hence are capable of producing a large library of unique molecules[192]. Since these molecules are derived from a common precursor metabolite(s), the optimal production strains often share common genotypes and phenotypes, and hence, the costly strain optimization process is only performed once for these molecules. Remarkably, this advantageous strain optimization

strategy can be applied even for production of molecules derived from different precursors, using the concept of modular cell (ModCell) design[272,273,293].

With the arrival of steady-state, constraint-based stoichiometric models of cellular metabolism, various computational algorithms have been developed to guide strain engineering[45,166,299]. These methods have featured the design of strains capable of growth-coupled product synthesis (GCP), enabling adaptive laboratory evolution of these designed strains to enhance product titers, rates, and yields[77,298,275,293]. Two approaches on growth-coupled production have been formulated - one based on the coexistence of maximum growth and product synthesis rates during the growth phase[30] and the other based on the obligate requirement of optimal product synthesis in any growth phase[274]. The distinction between these two types of growth coupling are also referred to weak coupling ($wGCP$) and strong coupling ($sGCP$)[139,299].

Development of most strain design algorithms has been focused on overproduction of only one target molecule. The first algorithm proposed for modular cell design compatible for overproduction of multiple target molecules is MODCELL[272], which guided several experimental studies[147,146,148,292,293]. It works by generating $sGCP$ strain designs for each target product based on elementary mode analysis[268], and then comparing the design strategies of different products to identify common genetic modifications among them. A similar approach was adapted in a subsequent work[121]. For MODCELL to find optimal solutions for multiple target products, it requires: 1) enumerating all possible designs above a predefined minimum product yield and with minimal reaction deletion sets for each production network, which might lead to a large number of solutions for each network and hence make the problem computationally intractable, and 2) the resulting designs for all products must be compared to identify common interventions, which is a computationally-hard, set-covering problem. Thus, the current enumerative approach of MODCELL might become intractable very quickly, especially for large-scale metabolic networks and potentially generate non-optimal designs, i.e., requiring more knock-outs than necessary or including fewer products than possible.

In this study, we generalized the concept of modular cell design and addressed the computational limitation of implementing it. We developed a novel computational platform

(ModCell2), based on multi-objective optimization and analysis of mass action of cellular metabolism, to guide the design of modular cells for large-scale metabolic networks. We demonstrated that ModCell2 can systematically identify genetic modifications to design modular cells that can couple with a variety of production modules and exhibit a minimal tradeoff among modularity, performance, and robustness. By analyzing these designs, we further revealed both intuitive and complex metabolic architectures enabling modularity in modular cell and production modules required for efficient biosynthesis of target molecules.

2.2 Methods

2.2.1 Design principles of modular cell engineering

In the conventional strain engineering approach, a parent strain is genetically modified to yield an optimal production strain to make only a target product. To produce each new molecule, the design-build-test cycles of strain engineering must be repeated, which is laborious and expensive (Figure 2.1). To minimize the cycles, modular cell engineering is formulated by genetically transforming a parent strain into a modular (chassis) cell that must be assembled with exchangeable modules to create optimal production strains[272]. A modular cell is designed to contain core metabolic pathways shared across designed optimal production strains. Exchangeable modules are production pathways designed to synthesize desirable chemicals. A combination of a modular cell and a production module(s) is required to balance redox, energy, and precursor metabolites for sustaining cellular metabolism during growth and/or stationary phases and exhibiting only desirable phenotypes. Practically, modular cell engineering can be applied to monocultures and polycultures, where a production module(s) can be embedded in a modular cell and activated by intracellular and/or extracellular cues such as light and/or signaling molecules.

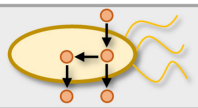
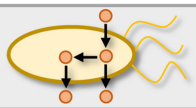
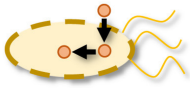
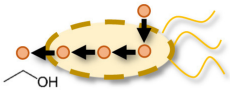
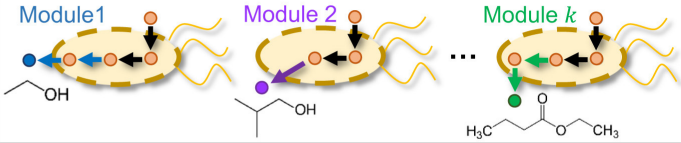
Features	Conventional strain engineering	Modular cell engineering
Parent strain		
Modular cell	Absent	 Optimized common production phenotypes
Exchangeable modules	1	k
Optimal production strains		
Design-build-test cycle	Repeated for every new product	One time for many products

Figure 2.1: Comparison between the conventional single-product strain design and modular cell engineering. In the conventional approach, each target product requires to go through the iterative optimization cycle. The modular cell engineering approach exploits common phenotypes associated with high product titers, rates, and yields; and hence, the strain optimization cycle only needs to be performed once for multiple products, which helps reduce the cost and time of strain development.

2.2.2 Multi-objective strain design framework for modular cell engineering

For modular cell engineering, we seek to design a chassis cell compatible with as many production modules as possible to achieve only desirable production phenotypes while requiring minimal genetic modifications. Since all production modules must leverage cellular resources of the modular cell (e.g. precursor metabolites, cofactors, and energy), they form competing objectives. Therefore, the framework of modular cell engineering can be formulated as a multi-objective optimization problem, named ModCell2, as described below.

$$\underset{y_j, z_{jk}}{\text{maximize}} \quad (f_1, f_2, \ldots, f_{|\mathcal{K}|})^T \quad \text{subject to} \tag{2.1}$$

$$
\begin{aligned}
f_k \in \arg\max \Bigg\{ & \sum_{j \in \mathcal{J}_k} c_{jk} v_{jk} \quad \text{subject to} && && (2.2)\\
& \sum_{j \in \mathcal{J}_k} S_{ijk} v_{jk} = 0 && \text{for all } i \in \mathcal{I}_k && (2.3)\\
& l_{jk} \le v_{jk} \le u_{jk} && \text{for all } j \in \mathcal{J}_k && (2.4)\\
& l_{jk} d_{jk} \le v_{jk} \le u_{jk} d_{jk} && \text{for all } j \in \mathcal{C} && (2.5)\\
& \text{where } d_{jk} = y_j \vee z_{jk} \Bigg\} && \text{for all } k \in \mathcal{K} &&\\
z_{jk} \le (1 - y_j) & && \text{for all } j \in \mathcal{C},\ k \in \mathcal{K} && (2.6)\\
\sum_{j \in \mathcal{C}} z_{jk} \le \beta_k & && \text{for all } k \in \mathcal{K} && (2.7)\\
\sum_{j \in \mathcal{C}} (1 - y_j) \le \alpha & && && (2.8)
\end{aligned}
$$

where i, j, and k are indices of metabolite i, reaction j, and production network k, respectively; f_k is a design objective for network k; c_{jk} represents the cellular objective for reaction j in network k associated with a design objective defined in (2.9 -2.11); v_{jk} (mmol/g DCW/h) is metabolic flux of reaction j bounded by l_{jk} and u_{jk} in network k, respectively; y_j and z_{jk} are binary design variables for deletion reaction j and module reaction j in network k, respectively; α and β_k are design parameters for deletion and module reactions, respectively; S_{ijk} is a stoichiometric coefficient of metabolite in reaction j of network k; and C (2.5) is the candidate reaction set (Supplementary File S1). The goal of the optimization problem is to simultaneously maximize all design objectives f_k.

Steady-state mass balance constraint of cellular metabolism

Quasi steady-state flux balance of cellular metabolism (2.3) is used as metabolic constraints for (2.1).[214]. A model corresponding to each modular production strain (i.e. production network k) will be derived from a parent strain (i.e. parent network) by adding necessary reactions (e.g., a production module) to produce a target molecule. A feasible flux distribution for each production network is described by mass balance (2.3) and reaction flux bounds (2.4-2.5). For a given production network, the phenotypic space can be illustrated by

the gray area that is projected onto the two-dimensional space spanned by product synthesis and growth rates (Figure 2.2).

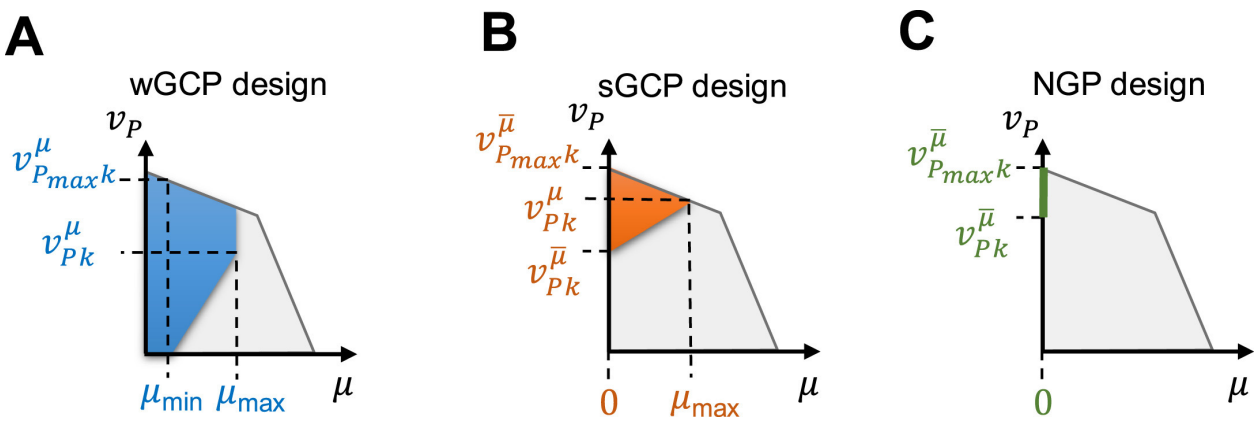


Figure 2.2: Graphical representation of phenotypic spaces for different strain design objectives including **(A)** weak growth coupling ($wGCP$), **(B)** strong growth coupling ($sGCP$), and **(C)** no-growth production (NPG). v^{μ}_{Pk} is the minimum product formation rate at the maximum growth rate for production network k, and $v^{\mu}_{P_{\mathrm{max}}k}$ is the maximum product secretion rate attainable. $v^{\overline{\mu}}_{\mathrm{Pk}}$ and $v^{\overline{\mu}}_{P_{\mathrm{max}}k}$ are the minimum and maximum product formation rates for production network k during the stationary phase, respectively.

Design variables

In our formulation for modular cell engineering, we introduced two design variables: binary reaction deletions (y_j) inherent to the modular cell and module-specific reaction insertions (z_{jk}) (2.5). These variables can be experimentally manipulated to constrain the desirable phenotypes of production strains as shown in Figure 2.2. Specifically, $y_j = 0$ if reaction j is deleted from the modular cell; otherwise, $y_j = 1$. Deleting metabolic reactions removes undesired functional states of the network and leaves those with high design objectives. Likewise, $z_{jk} = 1$ if reaction j is present in the production network k; otherwise, $z_{jk} = 0$. These module reactions are endogenous reactions removed from the parent network (2.6) , but are added back to a specific production module to enhance the compatibility of a modular cell. The maximum number of reaction deletions (α) and module-specific reaction insertions (β_k) are user-defined parameters.

Design objectives

To generalize ModCell2 design, we allow three different types of design objectives (f_k, (2.1) that determine production phenotypes for each production network. Depending on the application, a phenotype can be designed to be weak coupling (*wGCP*), strong coupling (*sGCP*), and/or non-growth production (*NGP*) (Figure 2.2). The constrained phenotypic spaces based on these design objectives are shown in color; any point within these spaces is a feasible physiological state of the cell that can be represented by a metabolic flux distribution.

The *wGCP* design seeks to achieve a high product rate at maximum growth rate (Figure 2.2A). The *wGCP* design objective, $f_k^{\mathrm{wGCP}} \in \{0, 1\}$, is calculated as follows:

$$f_k^{\mathrm{wGCP}} = \frac{v_{\mathrm{Pk}}^{\mu}}{v_{\mathrm{Pmaxk}}^{\mu}} \tag{2.9}$$

where v_{Pk}^{μ} is the minimum synthesis rate of the target product P at the maximum growth rate for production network k and v_{Pmaxk}^{μ} is the maximum synthesis rate of P (Supplementary File S1). This *wGCP* design formulation is equivalent to RobustKnock[258] or OptKnock with a tilted objective function[30,74,299]. In (2.9), f_k^{wGCP} is scaled from 0 to 1 for proper comparison among products. The *wGCP* design is appropriate for applications where growth rate is not limited by the nutrients, and the product is formed during the growth phase.

The *sGCP* design seeks to achieve a high product rate not only at optimal growth rate but also during non-growth phase (Figure 2.2B). The *sGCP* design objective,$f_k^{\mathrm{sGCP}} \in \{0, 1\}$, is calculated as follows:

$$f_k^{\mathrm{sGCP}} = \frac{v_{\mathrm{Pk}}^{\mu}}{v_{P_{\max}k}^{\mu}} \frac{v_{\mathrm{Pk}}^{\overline{\mu}}}{v_{P_{\max}k}^{\overline{\mu}}} \tag{2.10}$$

where $v_{\mathrm{Pk}}^{\overline{\mu}}$ and $v_{P_{\max}k}^{\overline{\mu}}$ are the minimum and maximum product formation rates for production network k in the stationary phase, respectively (Supplementary File S1). The *sGCP* design objective is comparable to the one implemented in MODCELL[272]. Different from *wGCP*, *sGCP* requires high product synthesis rate for any growth phase. However, the additional constraint of optimal product synthesis during the stationary phase requires more genetic manipulations or specific experimental conditions (e.g., anaerobic growth condition, supply

of intermediate metabolites). Both *wGCP* and *sGCP* designs enable fast growth selection to attain the optimum product rates by adaptive laboratory evolution[77,268].

The *NGP* design aims to maximize the minimum product rate during the non-growth phase by eliminating carbon fluxes directed to biomass synthesis (Figure 2.2C). The *NGP* design objective, $f_k^{\mathrm{NGP}} \in \{0, 1\}$, is calculated as follows:

$$f_k^{\mathrm{NGP}} = \frac{v_{\mathrm{Pk}}^{\overline{\mu}}}{v_{P_{\max}k}^{\overline{\mu}}} \tag{2.11}$$

While the *NGP* design is not suitable for growth selection, it can be derived from a *wGCP* (or *sGCP*) design by imposing additional genetic modifications. Practically, *NGP* design strains can be activated during cell culturing using a regulatory genetic circuit to toggle switch between production phases.

Design solutions

Optimal solutions for (2.1-2.8 are a Pareto set ($\mathcal{PS}$) that correspond to design variables, including reaction deletions (y_j) and module reaction insertions (z_{jk}). Each solution constitutes a design of a modular cell:

$$\mathcal{PS} = \{x \in \Omega : \nexists t \in \Omega,\, F(t) \prec F(x)\} \tag{2.12}$$

Here, $\mathbf{F}(\mathbf{t}) \prec \mathbf{F}(\mathbf{x})$ means $\mathbf{F}(\mathbf{t})$ *dominates* $\mathbf{F}(\mathbf{x})$ if and only if $f_i(\mathbf{t}) \geq f_i(\mathbf{x})$ for all i, and $\mathbf{F}(\mathbf{t})$ differs from $\mathbf{F}(\mathbf{x})$ in at least one entry. The feasible space of design variables, Ω, is defined by the problem constraints (2.2-2.8), also see Supplementary File S1). Phenotypes of modular cells will be the image of the Pareto set in the objective space, known as the Pareto front (***PF***):

$$\mathcal{PF} = \{F(x) : x \in \mathcal{PS}\} \tag{2.13}$$

For the multi-objective strain design framework, the input parameters include α, β_k, , and the production networks as input metabolic models. Each model contains a production module to produce one target chemical. The output is a Pareto set (genetic modifications) and its respective Pareto front (desirable production phenotypes). For a special case with

no trade-off among the design objectives, an optimal solution, named a utopia point, exists where each objective achieves its maximum value. The multi-objective strain design formulation presented is general and can be applied to design modular cells for any organism.

2.2.3 Algorithm and implementation

ModCell2 algorithm

To solve the multi-objective optimization problem for modular cell engineering, we used multi-objective evolutionary algorithms (MOEAs)[50]. MOEAs were selected because they can efficiently handle linear and non-linear problems and do not require preferential specification of design objectives[177]. MOEAs start by randomly generating a population of individuals (a vector of design variables), each of which is mapped to a design objective vector (i.e., a fitness vector). In ModCell2 (Supplementary File S1), the objective values of an individual are calculated by solving the linear programming problems for each production network. Next, individuals are shuffled to generate an offspring, from which the most fit individuals are kept. This process was repeated until the termination criteria was reached, for instance, either the solutions cannot be further improved or the simulation time limit is reached.

ModCell2 implementation

To streamline the modular cell design, we developed the ModCell2 software package based on three core classes (Figure S1 in Supplementary File S2). The Prodnet class parses and pre-processes production network models, and computes production phenotypes. The MCdesign class serves as an interface between the MOEA optimization method and metabolic models. Finally, the ResAnalysis class loads the Pareto set computed by MCdesign and identifies the most promising modular cell designs.

The code was written in MATLAB 2017b (The Mathworks Inc.) using the function gamultiobj() from the MATLAB Optimization Toolbox that implements the NSGA-II algorithm[61] to solve the multi-objective optimization problem. The solution and analysis methods were parallelized using the MATLAB Parallel Computing Toolbox. The linear programs to calculate metabolic fluxes were solved using the GNU Linear Programming Kit

(GLPK). The COBRA toolbox[105,233] and F2C2 0.95b[145] were also used for COBRA model preprocessing and manipulation.

Metabolic models

In our study, we used three parent models including i) a small metabolic network to illustrate the modular cell design concept[272], ii) a core metabolic network of *Escherichia coli* to compare the performance of ModCell2 with respect to the conventional single-product strain design strategy and the first-generation modular cell design platform MODCELL[272], and iii) a genome-scale metabolic network of *E. coli* (i.e., iML1515[74]) for biosynthesis of a library of endogenous and heterologous metabolites, including 4 organic acids, 6 alcohols, and 10 esters (Figures S2 in Supplementary File S2) \cite{RN197, 126, 81, 198, 83, 127, 136, 80, 110, 1045}.

Simulation protocols

Anaerobic conditions were imposed by setting oxygen exchange fluxes to be 0, and the glucose uptake rate was constrained to be at most 10 mmol/gCDW/h, as experimentally observed for *E. coli*. When using the genome-scale model iML1515 to simulate *wGCP* designs, the commonly observed fermentative products (acetate, CO2, ethanol, formate, lactate, succinate) were allowed for secretion as described elsewhere[282]. For simulation of *sGCP* and *NGP* designs, the glucose uptake rate was fixed (i.e., -10 mmol/gCDW/h); otherwise, the flux is not active during the no-growth phases, resulting in the product synthesis rate of 0 regardless of genetic manipulations. To compare ModCell2 with Optknock, we applied the OptKnock algorithm with a tilted objective function[176] to generate *wGCP* designs for each production network, using the open-source algebraic modeling language Pyomo[100]. The MILP problems were solved using CPLEX 12.8.0 with a time limit of 10,000 seconds set for each product. ModCell2 is provided as an open-source software package and is freely available for academic research. The software package and documentation can be downloaded via either `https://web.utk.edu/~ctrinh` or Github `https://github.com/TrinhLab`.

2.2.4 Analysis methods for design solutions

Compatibility

The compatibility,C, of a design is defined as the number of products that are coupled with a modular cell and has objective values above a specified cutoff value θ. As a default, we set $\theta = 0.6$ for the *wGCP* and *NGP* design objectives and $\theta = 0.36$ (0.6^2) for the *sGCP* design objective. For example, a *wGCP* design for 3 products that has the design objective values of 0.4, 0.9, and 0.6 has a compatibility of 2, given a cutoff value of $\theta \geq 0.6$.

Compatibility difference and loss

Robustness is the ability of a system to maintain its function against perturbations, and hence is very important of designing biological and engineered systems[138]. To evaluate the robustness of modular cell designs, we defined two metrics, the compatibility difference (CD) and compatibility loss ($CL \in \{0, 1\}$) as follows:

$$CD = C_{\text{initial}} - C_{\text{final}} \tag{2.14}$$

$$CL = \frac{C_{\text{initial}} - C_{\text{final}}}{C_{\text{initial}}} \tag{2.15}$$

where $C_{initial}$ and C_{final} are the compatibilities of a modular cell design before and after a single reaction deletion, respectively. The value $CD > 0$ (or $CL > 0$) means the modular gains fitness while $CD < 0$ (or $CL < 0$) means that it loses its fitness. In the analysis, we did not consider essential and blocked reactions for our single-deletion analysis; for instance, there are only 1139 potential reaction deletions in the iML1515 model.

Metabolic switch design

A metabolic switch design is a modular cell that can possess multiple production phenotypes (i.e., *wGCP*, *sGCP*, and *NPG*), activated by an environmental stimulus (e.g. metabolites, lights). The metabolic switch design is enforced to have a set of reaction (gene) deletions in one production phenotype to be a subset of the other. The metabolic switch design is beneficial for multiphase fermentation configurations that enable flexible genetic modification

and implementation. Specifically, the metabolic switch design can exhibit the *wGCP* phenotype during the growth phase and the *NPG* (or *sGCP*) phenotype during the stationary phase. The metabolic switches can be implemented using the genetic switchboard[31].

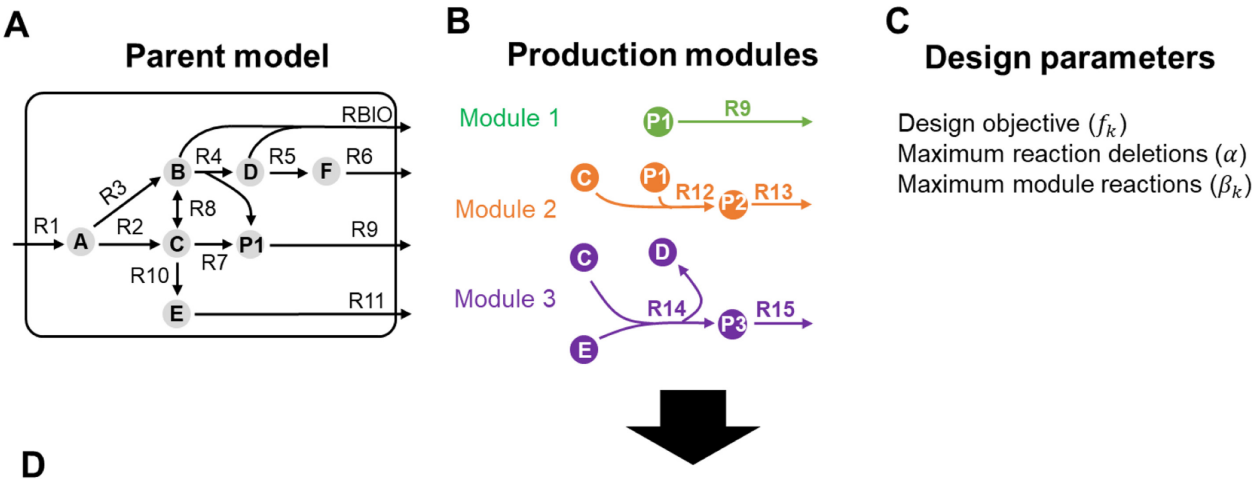
2.3 Results and discussion

2.3.1 Illustrating ModCell2 for modular cell design of a simplified network

An example parent network, adapted from[272], was used to illustrate ModCell2 (Figure 2.3A). Inputs for the multi-objective optimization problem include i) three production networks (Figure 2.3B), comprising of one endogenous production module (module 1) and two heterologous production modules (modules 2 and 3) and ii) design parameters (Figure 2.3C), containing design objective type, maximum number of deletion reactions (α), and maximum number of module reactions (β_k). The output of ModCell2 generated the Pareto set and the corresponding Pareto front for modular cell designs (Figure 2.3D). The 2-D plots of product yields versus growth rates presented the feasible phenotypic spaces of the wildtype (gray area) and the designed strain (blue area).

Using various α and β_k values, ModCell2 simulation generated three *wGCP-*α*-*β_k*-d*, four *sGCP-*α*-*β_k*-d* designs, and three *NGP-*α*-*β_k*-d* designs, where d is the design solution index (Figure 2.3D). For instance, by setting $\alpha = 3$ and $\beta_k = 0$, we found three *sGCP* designs including *sGCP-3-0-1*, *sGCP-3-0-2*, and *sGCP-3-0-3*. The first design *sGCP-3-0-1* has a compatibility of 2 with the design objective values of 0.42 and 0.97 for the products P2 and P3, respectively. In contrast, the *sGCP-3-0-2* and *sGCP-3-0-3* designs have compatibilities of only 1 with the design objectives of 0.63 for P2 and 0.45 for P1, respectively.

Based on all designs, we can clearly see the trade-offs for optimization of different products for $\beta_k = 0$. However, setting $\beta_k \geq 1$ helps increase the compatibility of a modular cell with different production modules. In addition, we found that the Pareto front collapses into a utopia point as seen in the *wGCP-1-1-1*, *sGCP-3-1-1*, and *NGP-3-1-1* designs. For instance, the modular cell, *sGCP-3-1-1*, is compatible with all three products. The three



D

Design parameters | *Pareto set* | *Pareto front*

Objective	α	β_k	d	Reaction deletions	Module reactions P1	Module reactions P2	Module reactions P3	f_k P1	f_k P2	f_k P3	Prod. yield vs growth P1	Prod. yield vs growth P2	Prod. yield vs growth P3
wGCP	1	0	1	R10				0.45	0.00	0.00			
wGCP	1	0	2	R9				0.00	0.63	0.97			
wGCP	1	1	1	R9	R9			0.45	0.63	0.97			
sGCP	3	0	1	R8, R9, R11				0.00	0.42	0.97			
sGCP	3	0	2	R6, R9, R10				0.00	0.63	0.00			
sGCP	3	0	3	R6, R8, R11				0.45	0.00	0.00			
sGCP	3	1	1	R6, R9, R11	R9		R6	0.45	0.63	0.97			
NGP	3	0	1	R6, R11				1.00	0.00	0.00			
NGP	3	0	2	R4, R9, R11				0.00	1.00	1.00			
NGP	3	1	1	R4, R9, R11	R9			1.00	1.00	1.00			

Figure 2.3: Illustration of ModCell2 workflow and analysis including (**A**) parent model, (**B**) production modules, (**C**) design parameters, and (**D**) simulation output for Pareto set and Pareto front based on design input.

corresponding optimal production strains can couple growth and product formation during the growth phase. During the stationary phase, these strains produce the products at maximum theoretical yields. In theory, a universal modular cell always exists, provided that enough reaction deletions and module reactions are used. It might be more tractable to construct such a modular cell from a synthetic minimal cell using the bottom-up approach. However, construction of a universal modular cell from a host organism (e.g., *E. coli*, *S. cerevisiae*) using the top-down approach will require a significantly large number of genetic modifications, that might be challenging.

2.3.2 Comparing ModCell2 designs with first-generation MODCELL and single product designs

ModCell2 can generate more and better designs than the first-generation modular cell design platform

To evaluate the algorithms and performance of ModCell2, we directly compared it with MODCELL[272] in two case studies, using the same core model of *E. coli* for production of five alcohols (ethanol, propanol, isopropanol, butanol, and isobutanol) and 5 derived butyrate esters (ethyl butyrate, propyl butyrate, isopropyl butyrate, butyl butyrate, and isobutyl butyrate) from glucose (Figure 2.4A).

In the first case study, we fixed the reaction module, i.e. $\beta_k = 2$ for ethanol dehydrogenase (FEM5) and ethanol export reaction (TRA1), in ModCell2 to emulate the same input as MODCELL (Supplementary File S3). The results showed that ModCell2 generated all the designs with the same *sGCP* objective values like MODCELL (Figure 2.4B, 4C, 4D) together with other alternative solutions (Supplementary File S3). Interestingly, ModCell2 only required 5 and 6 reaction deletions as opposed to 7 and 7 for the *sGCP-5-0-5* and *sGCP-5-0-6* designs, respectively. By setting the maximum reaction deletions to $\alpha \geq 6$, ModCell2 could find better design solutions with fewer deletion reaction requirement and higher objective values (Supplementary File S3).

In the second case study, we used the same model configuration but did not enforce the module reactions. By setting $\alpha = 5$ and $\beta_k = 1$, we found the *sGCP-5-1-8* design that is

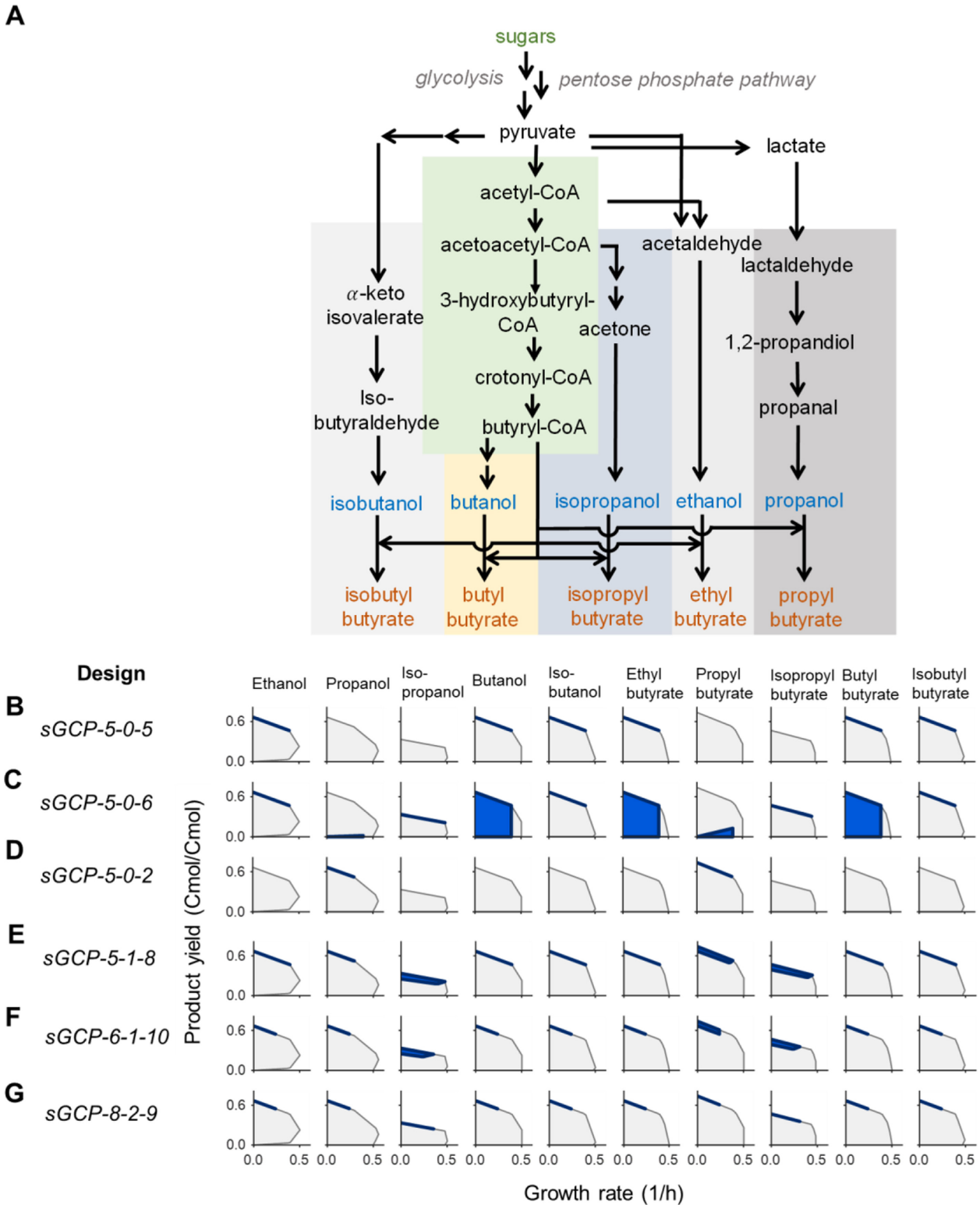


Figure 2.4: The 2-D metabolic phenotypic spaces of different *sGCP* designs using the core metabolic model. (**A**) Metabolic map, (**B**) *sGCP-5-0-5* design, (**C**) *sGCP-5-0-6* design, (**D**) *sGCP-5-0-2* design, (**E**) *sGCP-5-1-8* design, (**F**) *sGCP-6-1-10* design, and (**G**) *sGCP-8-2-9* design. For each panel, the gray and blue areas correspond to the phenotypic spaces of the wildtype and the optimal production strain, respectively.

compatible with all products and achieves the same objective values for products found in *sGCP-5-0-5*, *sGCP-5-0-6*, and *sGCP-5-0-2* (Figure 2.4E). The desirable phenotypic spaces can be further constrained for many products if α is increased from 5 to 6 (Figure 2.4F). Remarkably, by setting $\alpha = 8$ and $\beta_k = 2$, we found a utopia point design, *sGCP-8-2-9*, without any trade-off among design objectives (Figure 2.4G). This utopia point design could not be achieved with $\alpha < 8$ regardless of any β_k value.

Overall, the results demonstrate that ModCell2 can efficiently compute the Pareto front of modular cell designs. It can find better designs with fewer reaction deletion and module reaction requirements, improve design objective values, and enhance compatibility.

ModCell2 can identify designs with more compatibility than the conventional single-product designs

To evaluate if the conventional, single-product design strategy is suitable for modular cell engineering, we first used OptKnock to generate *wGCP* designs for the same 10 target molecules independently with various allowable reaction deletions ($\alpha = 2, 3, \ldots, 7$). Likewise, we employed ModCell2 to produce *wGCP* designs using the same α and various β. To directly compare OptKnock and ModCell2 solutions, we calculated the *wGCP* design objective values for all products based on each OptKnock solution (Supplementary File S3). As expected, our result showed that ModCell2 and OptKnock designs have the same highest objective values for each product (Figure 5A). However, several OptKnock solutions were always dominated by ModCell2 solutions in all parameter configurations (Figure 2.5B). With $\alpha \geq 4$, ModCell2 could identify *wGCP-α-1* designs with the maximum compatibility of 10, while the best OptKnock designs only achieved the highest compatibility of 5 (Figure 2.5C, 5D).

Overall, ModCell2 can generate modular cells compatible with the maximum number of modules and achieve high objective values. Single-product designs might not be compatible with a large number of products, and the solutions might be far from Pareto optimality.

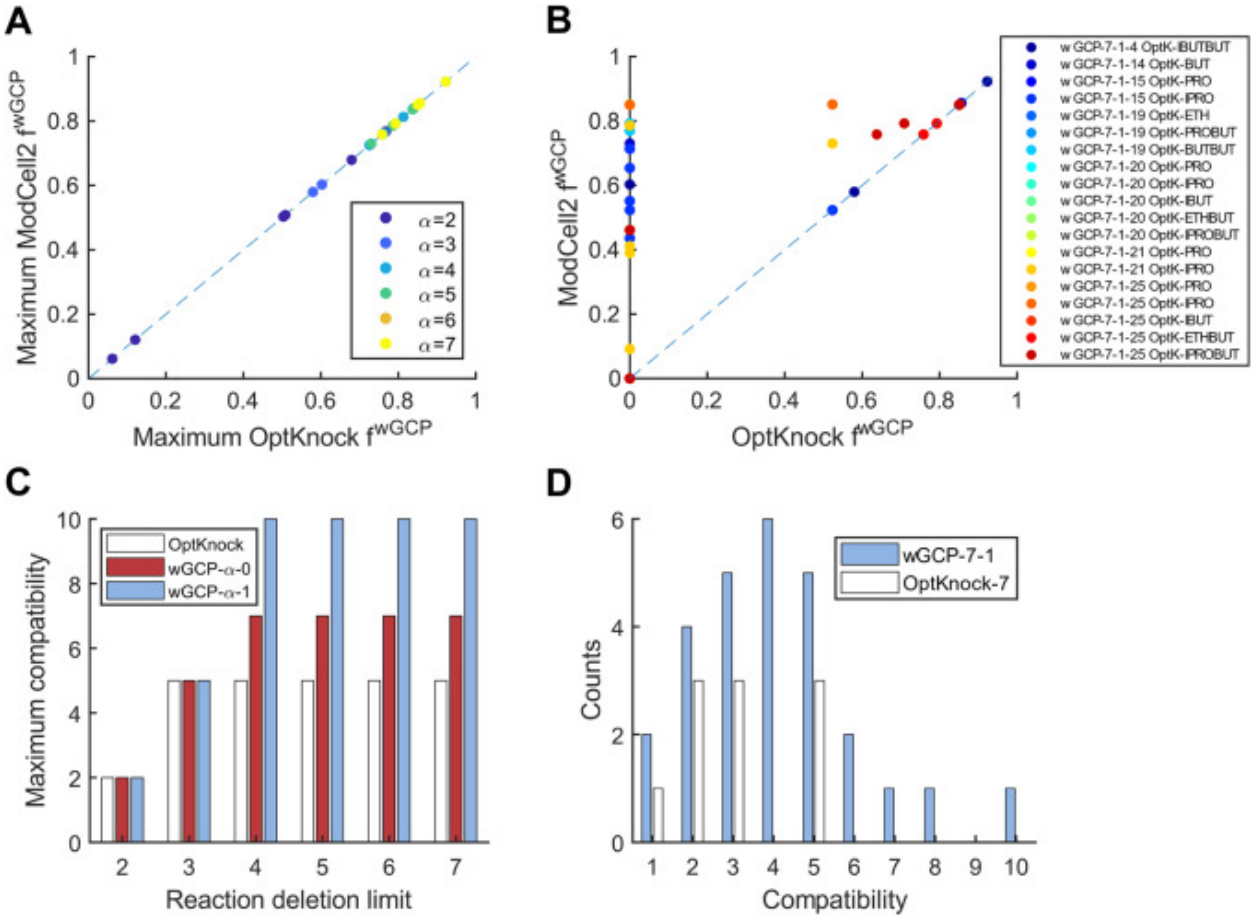


Figure 2.5: Comparison of strain design by OptKnock and Modcell2. (**A**) A correlation between the maximum objective values for each product generated by OptKnock and the equivalent values attained by ModCell2. Each point is colored based on the number of reaction deletions, with warmer colors corresponding to more reaction deletions. (**B**) A comparison between the Optknock objective vectors with at most 7 reaction deletions and the representative ModCell2 objective vector, *wGCP-7-1* which dominates them. Each color circle represents a pair of dominating *wGCP* design and dominated OptKnock solution (Supplementary File S3). (**C**) Maximum compatibility of OptKnock designs (blue), *wGCP* designs ($\beta_k = 0$, orange), wGCP designs ($\beta_k = 1$, yellow). (**D**) Compatibility distribution of Optknock ($\alpha = 7$, orange) and *wGCP-7-1* (blue).

2.3.3 Exploring emergent features of modular cell design using an *E. coli* genome-scale network

Modcell2 can design modular cells using a large-scale metabolic network.

To demonstrate that ModCell2 can be applied for a genome scale metabolic network, we tested it to generate *wGCP* designs for 20 target molecules with $\alpha = 4$ and various β_k (Supplementary File S4). The α value was chosen because with 4 deletions, OptKnock could identify single product designs with objectives above 60% of the theoretical maximum (Supplementary File S5). With $\beta_k = 0$, ModCell2 could identify modular cell designs with compatibility of 17, for example, the *wGCP-4-0-50* design featuring deletion of ACALD (acetaldehyde dehydrogenase, *adhE*), ACKr/PTAr (acetate kinase, *ack;* phosphotransacetylase, *pta*), GLYAT (glycine C-acetyltransferase, *kbl*), and LDH_D (lactate dehydrogenase, *ldhA*) (Figure 2.6A, 6D, Supplementary File S4). By analyzing all *wGCP-4-*β_k*-d* designs (257 total for $\beta_k = 0$, 1, 2, and 3), we found that the ethanol and D-lactate production modules are most compatible with all modular cell designs (Figure 2.6A, 6C, Supplementary File S4). Among reaction deletions, *LdhA* (86% of designs), *Pta* (38%), and *AdhE* (25%) are the most frequent deletion reactions (Figure 2.6B). This finding is consistent with a comprehensive survey of metabolic engineering publications[295] showing that these deleted reactions appeared in most of *E. coli* engineered strains for production of fuels and chemicals. The result supports the potential use of modular cell engineering to systematically build modular platform strains.

ModCell2 designs can capture combinatorial characteristics of production modules.

To evaluate whether ModCell2 could capture the combinatorial properties among production modules, we analyzed the Pareto front of *wGCP-4-0-d* that have a total of 58 designs. Hierarchical clustering of this Pareto front revealed certain products with similar objective values across solutions, such as ethyl esters and butyrate esters (Figure 2.6A). These products together were compatible with different modular cells and exhibited metabolic similarity

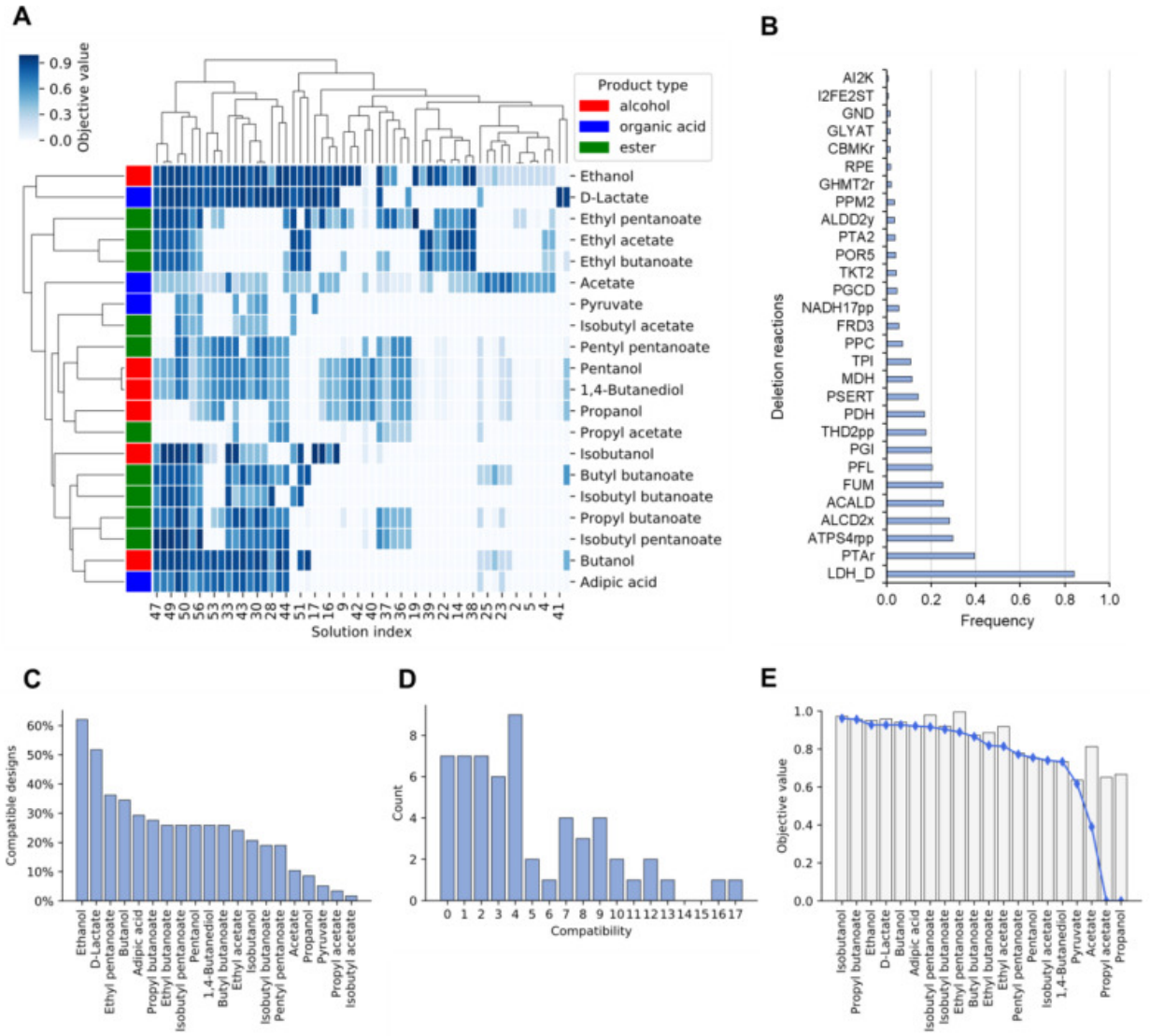


Figure 2.6: Analysis of *wGCP* designs with genome-scale model. **(A)** Pareto front of *wGCP-4-0-d*. The columns correspond to different designs labeled by their design index, *d*, where the rows correspond to different products. **(B)** Frequency of the top deletion reactions. **(C)** Product compatibility distribution across designs. **(D)** Design compatibility. **(E)** Tradeoff between modularity and performance. The bars correspond to the maximum objective values attainable for each product whereas the blue line represent the objective values of the *wGCP-4-0-48alternative* design.

in their production modules. Thus, ModCell2 could generate designs that capture the combinatorial properties useful for modular cell engineering.

ModCell2 can identify highly compatible modular cells

Analysis of compatibility shows that certain modular cells can couple with production modules that may not exhibit the combinatorial properties (Figure 2.6D). However, there exists a tradeoff between the number of feasible designs and degree of compatibility. Some modular cell designs are compatible with up to 17 out of 20 products, for instance, the most compatible design, *wGCP-4-0-48*, featuring deletions of ACALD (*adhE*), ACKr/PTAr (*ack*, *pta*), GND (phosphogluconate dehydrogenase, *gnd*) and LDH_D (*ldhA*) (Supplementary File S4). An alternative design *wGCP-4-0-48-alternative* also exists where deletion of G6PDH2r (gluose-6-phosphate dehydrogenase, *zwf*) is replaced by that of GND, the first step in the oxidative pentose phosphate pathway. The gene deletions in the design *wGCP-4-0-48-alternative* are a subset of the modular *E. coli* strain TCS095, whose modular properties have recently been validated experimentally[293].

To determine if modular cell design is a viable alternative to single-product design, we also analyzed a potential tradeoff between design performance and modularity by comparing the maximum value of each objective across all solutions in the Pareto front and the single-product design optima. If production modules exhibit competing phenotypes, a modular cell will not achieve the same performance in all modules as a single-product design strain. Analysis of the most compatible design *wGCP-4-0-48-alternative* showed that it could achieve objectives within 4% of the single-product optima in 14 products and within 10% in 3 products (Figure 2.6E). This result indicates that it is feasible to identify highly compatible modular cell designs without a significant tradeoff between performance and modularity.

Analysis of potential tradeoff between robustness and modularity can identify conserved metabolic features

To evaluate the robustness of modular cells, we analyzed the compatibility change (*CD*) of *wGCP-4-0* designs with compatibilities of 4 or greater (Figure S3 in Supplementary File 2). Remarkably, the result shows that only 7.5% of potential reaction deletions were detrimental

to the robustness of modular cells while the large remaining portion did not affect *CD* values. Out of the 85 reactions whose deletion affected compatibility, only a few appeared consistently across the designs. For instance, deletion of TPI (triose-phosphate isomerase, *tpi*) led to an average compatibility loss of 95%, inactivating most modular cell designs. Based on flux variability analysis, TPI must operate in the forward direction by converting glycerone phosphate (dhap) to glyceraldehyde-3-phosphate (g3p) to drive sufficient flux through glycolysis and hence preventing synthesis of undesired byproducts (D-lactate or 1,2-propanediol) from dhap. Likewise, deletion of carbon dioxide and water transport and exchange reactions caused compatibility loss across all designs. Pyruvate carboxylase (PPC) is an important reaction to channel carbon flux through the Krebs cycle[52], and hence, deletion of PPC reduces compatibility in most modular cell designs with an average *CL* of 43%.

While some reaction deletions are critical for modular cell robustness, others are associated with specific products. For example, deletion of PDH (pyruvate dehydrogenase complex, *lpd/aceEF*) removes compatibility in all butanol-derived designs, indicating PFL (pyruvate formate lyase, *pfl*) is not an appropriate route. To make heterologous butanol-derived molecules under anaerobic conditions, FDH (NADH-dependent formate dehydrogenase, *fdh*) is required in butanol-derived modules where enzymatic reaction pairs of PFL and FDH could substitute PDH known to be anaerobically inhibited.

Overall, analysis of tradeoff between modularity and robustness can identify not only the conserved metabolic features of modular cells but also potential bottlenecks in specific production modules.

Enabling metabolic switch among different design objectives using ModCell2

The ability to dynamically control growth and production phases can potentially enhance product titers, rates, and yields. For instance, two-phase fermentation can be employed where growth phase is optimized for biomass synthesis and stationary phase for chemical production[140]. Using ModCell2, we investigated the feasibility to design optimal strains to toggle switch desirable production phenotypes. To design a *wGCP*→*NPG* metabolic switch, we first used our reference *wGCP* design as a parent strain (Figure 2.7A) and then employed

ModCell2 to identify the most compatible *wGCP→NPG* designs. With 5 additional deletions, we could find *wGCP*→NPG designs that encompass both *wGCP* and *NPG* phenotypes, for instance, the *sup-NGP-5-0-23* design featuring deletion of PGI (glucose-6-phosphate isomerase, *pgi*), MDH (malate dehydrogenase, *mdh*), ASPT (L-aspartase, *aspT*), Tkt2 (transketolase, *tktB*), and ATPS4rpp (ATP synthase, *atp*) (Figure 2.7B). The deletion reactions in the *wGCP*→NPG designs appear in both catabolic (PGI, ATPS4tpp) and anabolic (ASP, TKT2) processes, responsible for growth disruption and direction of carbon flow to the biosynthesis of target products.

Likewise, we used ModCell2 to design a *wGCP→sGCP* metabolic switch. We identified the most compatible *wGCP→sGCP* designs with 6 additional deletions, for instance, the *sup-sGCP-6-0-39* design featuring the deletion of MGSA (Methylglyoxal synthase, *mgsA*), ALCD2x (alcohol dehydrogenase, *adhE*), PFL, MDH, FADRx (FAD reductase, *fadI*), and GLUDy ($NADP^+$dependent glutamate dehydrogenase, *gdhA*) (Figure 2.7C). Different from the *wGCP→NGP* metabolic switch, all deletions in the *wGCP→sGCP* designs are involved the elimination of biosynthesis pathways of undesirable byproducts.

While it is feasible to metabolically switch among different production phenotypes, it not only requires more reaction deletions but also reduces the product compatibility. For instance, the *wGCP*→sGCP and *wGCP*→NGP designs are only compatible with 5 products while the *wGCP* parent design have a compatibility of 17 out of 20 products with 4 deletions. The main reason is that both *wGCP→sGCP* and *wGCP→NGP* designs must eliminate all possible redundant pathways that result in biosynthesis of undesirable byproducts.

2.4 Conclusion

In this study, we developed a multi-objective strain design platform for modular cell engineering. With a new developed algorithm and computational platform, ModCell2 enables flexible design of modular cells that can couple with production modules to exhibit desirable production phenotypes. In comparison to the first-generation strain design platform, ModCell2 can handle large-scale metabolic networks and identify better solutions that require fewer genetic modifications and exhibit more product compatibility. Different from

the conventional single-product strain design, ModCell2 can find solutions that are Pareto optimal with negligible tradeoffs among modularity, performance, and robustness. We envision ModCell2 is a useful tool to implement modular cell engineering and fundamentally study modular designs in natural and synthetic biological systems.

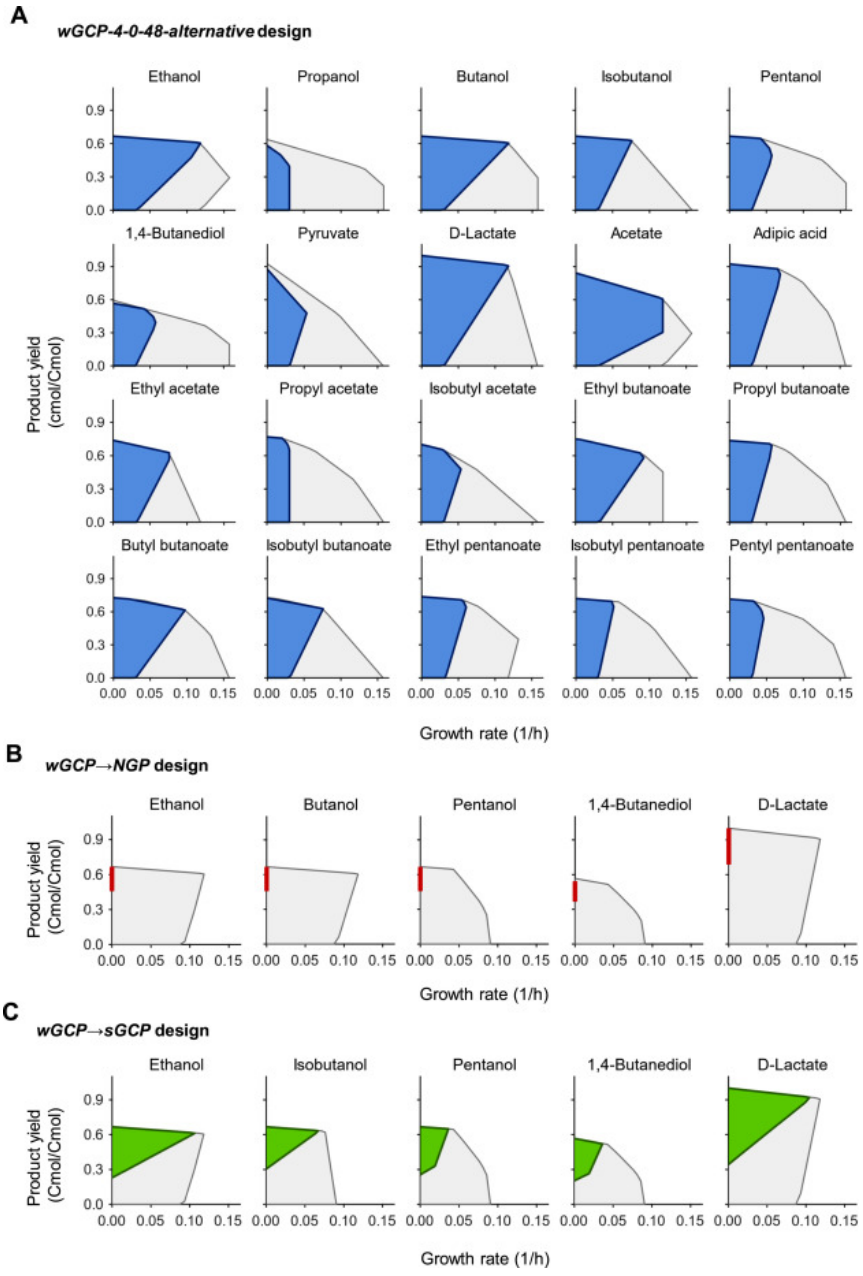


Figure 2.7: Production phenotypes of (**A**) the wild type (gray) and the representative, highly-compatible design *wGCP-4-0-48-alternative* (blue), (**B**) the *wGCP→NPG* design, *sup-NGP-5-0-23*, and (**C**) the *wGCP→sGCP* design, *sup-sGCP-6-0-39*.

Chapter 3

Comparison of multi-objective evolutionary algorithms to solve the modular cell design problem

This chapter is based on the publication *Comparison of Multi-Objective Evolutionary Algorithms to Solve the Modular Cell Design Problem for Novel Biocatalysis. Garcia, S., and Trinh, C. T. Processes, Volume 7, Issue 6, 2019.* As first author I lead the development, implementation, and writing of this study. Supplementary Material 1 is provided as an attachement.

Abstract

A large space of chemicals with broad industrial and consumer applications could be synthesized by engineered microbial biocatalysts. However, the current strain optimization process is prohibitively laborious and costly to produce one target chemical and often requires new engineering efforts to produce new molecules. To tackle this challenge, modular cell design based on a chassis strain that can be combined with different product synthesis pathway modules has been recently proposed. This approach seeks to minimize unexpected failure and avoid task repetition, leading to a more robust and faster strain engineering process. In our previous study, we mathematically formulated the modular cell design

problem based on the multi-objective optimization framework. In this study, we evaluated a library of state-of-the-art multi-objective evolutionary algorithms (MOEAs) to identify the most effective method to solve the modular cell design problem. Using the best MOEA, we found better solutions for modular cells compatible with many product synthesis modules. Furthermore, the best performing algorithm could provide better and more diverse design options that might help increase the likelihood of successful experimental implementation. We identified key parameter configurations to overcome the difficulty associated with multi-objective optimization problems with many competing design objectives. Interestingly, we found that MOEA performance with a real application problem, e.g., the modular strain design problem, does not always correlate with artificial benchmarks. Overall, MOEAs provide powerful tools to solve the modular cell design problem for novel biocatalysis.

3.1 Introduction

Multi-objective optimization is a powerful mathematical toolbox widely used in engineering disciplines to solve problems with multiple conflicting design objectives[48]. For example, in the field of chemical engineering, multi-objective optimization has been applied to balance design conflicts in the performance, material and energy requirements, and environmental sustainability of many different chemical processes[218]. In industrial biotechnology, with recent advancements in synthetic biology and metabolic engineering, microorganisms can be genetically modified to produce a large space of molecules with broad applications using renewable lignocellulosic biomass or waste products as feedstocks[273,150]. However, the current strain design process is prohibitively laborious and expensive for broad industrial application[196]. To overcome this challenge, recent studies have proposed the application of modular design principles commonly used in engineering[24] to microbial biocatalysis[269,272,86,85]. This modular cell design approach, known as ModCell, uses multi-objective optimization to account for the competing cellular objectives when cellular metabolism is (re)designed in a modular fashion to produce a diverse class of target chemicals. ModCell has been experimentally demonstrated for biosynthesis of alcohols[271,269,293] and esters[146,147,148,292,149] in *Escherichia coli*.

Despite the broad applicability of multi-objective optimization problems (MOPs) in engineering design, powerful solution algorithms remain elusive. Two approaches can be used to solve MOPs, including multi-objective evolutionary algorithms (MOEAs) and mixed integer linear programming (MILP) algorithms. Unlike MOEA, MILP can ensure that the identified MOP solutions are optimal. Nonetheless, MOEAs are widely used due to the following advantages over MILP: (i) computational scalability for large-scale networks by implementing efficient parallelization algorithms[49], (ii) compatibility with non-linear objectives and constraints, and (iii) unbiased sampling of Pareto optimal solutions without a need to pre-specify objective preference[177]. MOEAs are based on a more general type of optimization method known as evolutionary algorithms, where candidate solutions, that represent individuals of a population, are iteratively modified using heuristic rules to increase their fitness (i.e., objective function values). Recently, much attention has been placed in the development of MOEAs to solve many-objective problems (e.g., problems with 4 or more objectives) that often correspond to real-world applications, but can be very challenging to solve with conventional MOEAs[153]. For the case of ModCell problem, the popular MOEA NSGAII[180,125] was used to design a modular cell under 20 different production modules[86]. Due to a large space of molecules that can potentially be synthesized by modular cells, scalability issues are expected to occur when constructing modular cells that are designed to be compatible with tenths or hundreds of products. Furthermore, using the best solver algorithm(s) allows to explore a more diverse design space, resulting in better choices for experimental implementation.

Many MOEAs have been proposed over the past two decades since the inception of landmark algorithms such as NSGAII[61] and SPEA2[316]. New MOEAs are benchmarked against libraries of artificial problems with known solutions[315,62], and are expected to show enhanced performance for a subset of these problems in terms of scalability, identification of Pareto optimal solutions, and number of simulation generations needed to converge. This benchmarking methodology does not always reflect MOEA performance for general problems, since specialized parameter configurations or heuristics are often used and can lead to drastically different performance towards a specific problem of interest. Thus, the best MOEA for a certain application problem needs to be determined empirically. In this

study, we evaluated a library of state-of-the-art MOEAs to solve the multi-objective ModCell problem, with the focus on many-objectives methods. Several cases study of increasing difficulty were examined using common performance indicators of solution optimality and diversity, and critical algorithm parameters that determine solution quality were also investigated.

3.2 Methods

3.2.1 Multi-objective modular cell design

Modular cell design enables rapid generation of optimal production strains with desirable phenotypes from a modular (chassis) cell[86], requiring mimimal strain optimization cycles. These production strains are assembled from a modular cell and various compatible pathway modules. A modular cell is constructed by eliminating genes from a parent strain to maintain only core metabolic pathways shared across all pathway modules. Each module enables an optimized target product synthesis phenotype that leads to high yields, titers, and production rates. The different biochemical nature of each target metabolite can make the objectives compete with each other, turning the modular cell design problem into a multi-objective optimization problem known as ModCell2[86]:

$$\max_{y_j, z_{jk}} \quad (f_1, f_2, \ldots, f_{|\mathcal{K}|})^T \quad \text{s.t.} \tag{3.1}$$

$$f_k \in \arg\max \left\{ \frac{1}{f_k^{max}} \sum_{j \in \mathcal{J}_k} c_{jk} v_{jk} \quad \text{s.t.} \right. \tag{3.2}$$

$$\sum_{j \in \mathcal{J}_k} S_{ijk} v_{jk} = 0 \qquad \text{for all } i \in \mathcal{I}_k \tag{3.3}$$

$$l_{jk} \leq v_{jk} \leq u_{jk} \qquad \text{for all } j \in \mathcal{J}_k \tag{3.4}$$

$$l_{jk} d_{jk} \leq v_{jk} \leq u_{jk} d_{jk} \qquad \text{for all } j \in \mathcal{C} \tag{3.5}$$

$$\left. \text{where } d_{jk} = y_j \vee z_{jk} \right\} \qquad \text{for all } k \in \mathcal{K}$$

$$z_{jk} \leq (1 - y_j) \qquad \text{for all } j \in \mathcal{C},\ k \in \mathcal{K} \tag{3.6}$$

$$\sum_{j \in \mathcal{C}} z_{jk} \leq \beta_k \qquad \text{for all } k \in \mathcal{K} \tag{3.7}$$

$$\sum_{j \in \mathcal{C}} (1 - y_j) \leq \alpha \tag{3.8}$$

where $\mathcal{I}_k$, $\mathcal{J}_k$, and $\mathcal{K}$ are the sets of metabolites, reactions, and associated production metabolic networks (i.e., the combination of the chassis organism with a specific product synthesis pathway), respectively. The optimization problem seeks to simultaneously maximize all objectives f_k (3.1). The desirable phenotype f_k for production module k is determined based on key metabolic fluxes v_{jk} (mmol/gDCW/h) predicted by the constraint-based metabolic model (3.2-3.5)[204]. For example, the weak growth coupled to product formation ($wGCP$), a common design objective, requires a high minimum product synthesis rate at the maximum growth-rate, enabling growth selection of optimal production strains. Thus, in $wGCP$ design, the inner optimization problem seeks to maximize growth rate while calculating the minimum product synthesis rate through the linear objective function (3.2) (where c_{jk} is 1 and -0.0001 for j corresponding to the biomass and product reactions across all networks k, respectively, and 0 otherwise) subject to: (i) mass-balance constraints (3.3), where S_{ijk} represents the stoichiometric coefficient of metabolite i in reaction j of production network k, (ii) flux bound constraints (3.4) that determine reaction reversibility and available substrates, where l_{jk} and u_{jk} are lower and upper bounds respectively, and (iii) genetic manipulation constraints (3.5), i.e., deletion of a reaction j in the chassis through the binary indicator y_j, or insertion of a reaction j in a specific production network k through the binary indicator z_{jk}. The maximum product synthesis rate of each production network k, f_k^{max}, is determined by maximizing the product synthesis reaction subject to (3.3-3.4), allowing to bound f_k in $wGCP$ between 0 and 1. Only a subset of all metabolic reactions, $\mathcal{C}$, are considered as candidates for deletion, since many of the reactions in the metabolic model cannot be manipulated to enhance the target phenotype. Certain reactions can be deleted in the chassis but inserted back to specific production modules, enabling the chassis to be compatible with a broader number of modules (3.6). The numbers of module-reaction additions and reaction deletions in the chassis are constrained by the parameters β_k (3.7)

and α (3.8), respectively, to avoid unnecessary genetic manipulations that are generally time-consuming to implement and can lead to unforeseen phenotypes.

3.2.2 Optimal solutions for a multi-objective optimization problem

Optimal solutions for a MOP (3.1-3.8) are defined based on the concept of domination: A vector $a = (a_1, \ldots, a_K)^T$ *dominates* another vector $b = (b_1, \ldots, b_K)^T$, denoted as $a \prec b$ if and only if $a_i \geq b_i \; \forall i \in \{1, 2, \ldots, K\}$ and $a_i \neq b_i$ for at least one i. Letting x be the design variables (i.e., y_j and z_{jk}) and X be the feasible set determined by the problem constraints (3.2-3.8), a feasible solution $x^* \in X$ of the MOP is called a Pareto optimal solution if and only if there does not exists a vector $x' \in X$ such that $F(x') \prec F(x^*)$. The set of all Pareto optimal solutions is called Pareto set:

$$PS := \{x \in X : \nexists \, x' \in X, F(x') \prec F(x)\} \tag{3.9}$$

The projection of the Pareto set in the objective space is denoted as Pareto front:

$$PF := \{F(x) : x \in PS\} \tag{3.10}$$

3.2.3 MOEA selection

To find the best MOEAs for ModCell2, we evaluated a recent and comprehensive set of MOEAs implemented in the PlatEMO platform[265]. From over 50 algorithms available in PlatEMO, we selected 2 methods for benchmark study, including NSGAII/gamultiobj and MOEAIGDNS, and 8 methods that have been specifically developed to tackle many-objective problems with discrete variables like ModCell2, including ARMOEA, EFRRR, MaOEADDFC, SPEAR, tDEA, BiGE, NSGAIII, and SPEA2SDE (Table 3.1). It should be noted that gamultiobj is an alternative implementation of the NSGAII algorithm available in Matlab.

Table 3.1: Summary of MOEAs used in this study

Abbreviation	Name	Notes	Reference
NSGAII	Non-dominated sorting genetic algorithm 2	Highly applied MOEA	[61]
gamultiobj	Matlab implementation of NSGAII	Used in the original ModCell2 study[86]	[180]
MOEAIGDNS	Multi-objective evolutionary algorihtm based on an enhanced inverted generational distance metric	General MOEA with an implementation that works well with discrete variables	[266]
ARMOEA	Adapation to reference points multi-objective evolutionary algorithm	Many-objective EA based on MOEAIGDNS	[264]
EFRRR	Ensemble fitness ranking with ranking restriction	Many-objective EA	[306]
MaOEADDFC	Many-objective evolutionary algorithm based on directional diversity and favorable convergence	Many-objective EA	[42]
SPEAR	Strength Pareto evolutionary algorithm based on reference direction	Many-objective EA	[119]
tDEA	θ-dominance evolutionary algorithm	Many-objective EA	[305]
BiGE	Bi-goal evolution	Many-objective EA	[157]
NSGAIII	Non-dominated sorting genetic algorithm 3	Many-objective EA	[60]
SPEA2SDE	Strength Pareto evolutionary algorithm 2 with shift-based density estimation	Many-objective EA	[156]

3.2.4 Performance metrics

To evaluate the performance of different MOEAs for a given problem, each algorithm was ran for the same number of generations, and the resulting solutions, known as Pareto front approximations, are compared using functions that measure two qualities: (i) solution accuracy, i.e., to determine how similar the solution is to the true Pareto front and (ii) solution diversity, i.e., to evaluate how well distributed are the points in the solution. We selected the top 5 most used metrics according to a recent literature survey[222]. These include, in order of popularity, hypervolume (HV), generational distance (GD), epsilon indicator (ϵ), inverted generational distance (IGD), and coverage (C). Based on a recent study[235], we considered the average Hausdorff distance (Δ_p), that combines GD and IGD, and hence simplified the number of performance metrics to 4 in our study. These metrics are defined as follows:

HV: This metric measures the volume occupied by the union of the smallest hyperboxes formed by each point in the Pareto front approximation and the reference point. This Pareto front approximation corresponds to the solution of a specific MOEA (denoted as $\mathcal{PF}$) and the reference point is selected to be greater or equal to the maximum value attainable by any objective, which in our case is $\vec{1}$ (Figure 3.1a):

$$HV = \bigcup_{i \in I} \text{Volume}(\text{Box}(\mathcal{PF}_i, \vec{1})) \tag{3.11}$$

where I is the index set of $\mathcal{PF}$ points.

GD: This metric measures the distance between the solution $\mathcal{PF}$ and the best Pareto front approximation determined by combining non-dominated points from all MOEA solutions of a specific case study, denoted $\mathcal{PF}^*$. More specifically, GD corresponds to the average Euclidean distance between each point in $\mathcal{PF}$ and the nearest point in $\mathcal{PF}^*$, denoted as $d_i = \min_{k \in K} \left(\sum_{j \in J} (\mathcal{PF}_{ij} - \mathcal{PF}^*_{kj})^2 \right)^{\frac{1}{2}}$, where I ($i \in I$), K ($k \in K$), and J ($j \in J$) correspond to the index sets of $\mathcal{PF}$ points, $\mathcal{PF}^*$ points, and problem objectives, respectively (Figure 3.1b):

$$GD = \frac{\sum_{i \in I} d_i}{|I|} \tag{3.12}$$

IGD: This metric measures the distance between $\mathcal{PF}$ and $\mathcal{PF}^*$. It is determined by the average Euclidean distance between each point in $\mathcal{PF}^*$ and the nearest point in $\mathcal{PF}$ denoted $\hat{d}_k = \min_{i \in I} \left(\sum_{j \in J} (\mathcal{PF}^*_{kj} - \mathcal{PF}_{ij})^2 \right)^{\frac{1}{2}}$ (Figure 3.1b):

$$IGD = \frac{\sum_{k \in K} \hat{d}_k}{|K|} \tag{3.13}$$

Δ_p: This metric combines *GD* and *IGD* metric and thus has superior properties[235]:

$$\Delta_p = \max(GD, IGD) \tag{3.14}$$

C: This metric determines the fraction of $\mathcal{PF}^*$ captured by the solution $\mathcal{PF}$ (Figure 3.1c):

$$C = \frac{|\mathcal{PF} \cap \mathcal{PF}^*|}{|\mathcal{PF}^*|} = \frac{|\{k \in K : \exists i \in I \text{ such that } \mathcal{PF}^*_{kj} = \mathcal{PF}_{ij} \text{ for all } j \in J\}|}{|K|} \tag{3.15}$$

ϵ: This metric is the additive epsilon indicator[317] that measures the smallest value to be added to any point in $\mathcal{PF}$ to make it non-dominated with respect to some point in $\mathcal{PF}^*$. In other words, it is the smallest value ϵ such that for any solution in $\mathcal{PF}^*$ there is at least one solution in $\mathcal{PF}$ that is not worse by a difference of ϵ (Figure 3.1d):

$$\epsilon = \inf\{\epsilon \in \mathbb{R} : \text{for all } i \in I \, \exists k \in K \text{ such that } \mathcal{PF}_{ij} + \epsilon \geq \mathcal{PF}^*_{kj} \text{ for all } j \in J\} \tag{3.16}$$

Use of these metrics can be illustrated with a two-objective design example with 4 generations of improving Pareto front approximations, where the final Pareto front is used as a reference (i.e., $\mathcal{PF}^*$) (Figure 3.1e). As the Pareto fronts contain points that dominate the previous generations, all metrics decrease monotonically with the exception of C that increases to a value of 1 when both Pareto front approximation and reference are the same (Figure 3.1f).

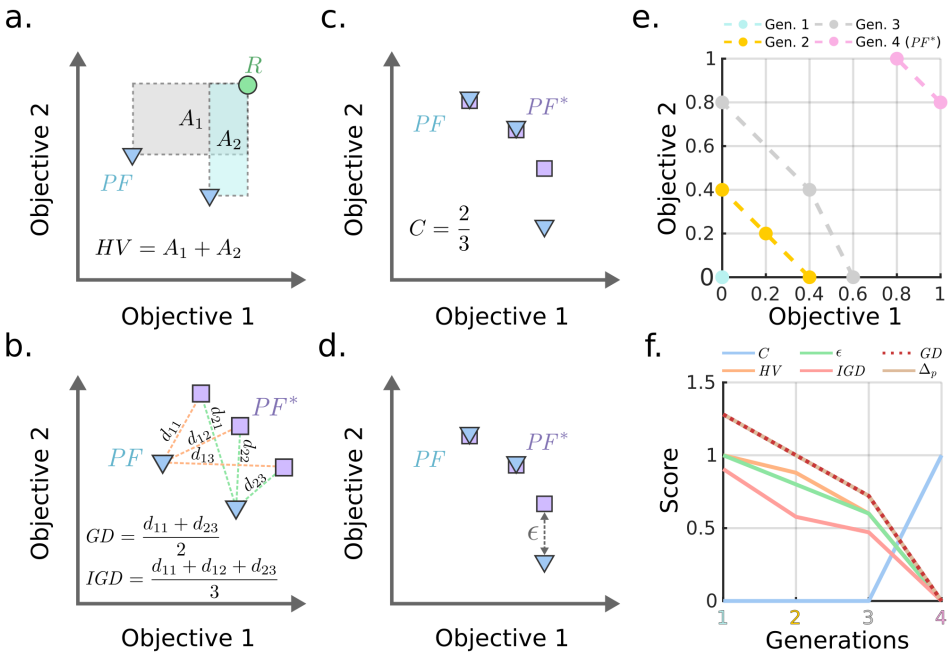


Figure 3.1: (**a-d**) Conceptual illustration of performance metrics of MOEAs for a two-objectives design problem. PF and PF^* correspond to the Pareto front approximation and the best Pareto front available, respectively. The reference point R must always dominate all solutions in PF. (**e-f**) An example of Pareto fronts with 2 dimensions and associated metrics. The 4th generation corresponds to $\mathcal{PF}^*$ used as a reference for comparison.

3.2.5 Algorithm parameters

All parameters used in the simulations of this study were left as default except the following ones. The total number of generations was set to be 200, which was sufficient to reach high quality solutions for the problems of this study. In addition, the population size was set to be 100 for all algorithms unless noted otherwise. All problems were solved in triplicates with unique random number generator seeds.

3.2.6 Metabolic models

For all simulations, we used a core *E. coli* model, downloaded from the BiGG database (`https://bigg.ucsd.edu`)[136], that captures the most important metabolic pathways[204]. The product synthesis pathways for each module correspond to native *E. coli* pathways together with well-characterized heterologus pathways for the synthesis of propanol[277],

butanol[241], isobutanol[11], and pentanol[277]. The metabolic reactions associated with these pathways are described in the software implementation (Supplementary Material 1).

3.2.7 Implementation

The simulations were performed using the ModCell2 software framework[86]. The MOEAs are implemented in the PlatEMO Matlab library[265], except *gamultiobj* which is implemented as part of the Matlab Optimization Toolbox. HV was calculated using the *hv package*[79]. All computations were executed in a computer with the Arch Linux operative system, Intel Core i7-3770 processor, and 32 GB of random-access memory. The Matlab 2018b code used to generate the results of this manuscript is available in Supplementary Material 1 and `https://github.com/trinhlab/compare-moea`.

3.3 Results and Discussion

3.3.1 Case 1: A 3-objectives design problem

We first formulated a design problem that considers an *E. coli* core model and 3 production modules based on the endogenous acetate, D-lactate, and ethanol biosynthesis pathways (Figure 3.2a). We used all MOEAs to solve for the problem by setting the following design parameters: $wGCP$ design objective, a maximum number of reaction deletions $\alpha = 3$, and no module reactions $\beta = 0$. These design parameters were sufficiently restrictive to generate conflicting objectives. A total coverage of $\mathcal{PF}^*$ ($C = 1$) was reached within 20 generations by several algorithms (Figure 3.2b, e, h, i) and by *gamultiobj* after 150 generations (Figure 3.2k), while the remaining algorithms could not attain C values above 0.8 (Figure 3.2c, d, f, g, j, l). In particular, MaOEADDFC and BiGE obtained the worst C, ϵ, and Δ_p values (Figure 3.2m). Although C, ϵ, and Δ_p values of BiGE indicated inferior performance, this algorithm had the lowest HV since it generated only one point with a high objective value (Figure 3.2o). Due to the simplicity of the problem, every algorithm except MaOEADDFC, tDEA, and BiGE converged to very similar Pareto fronts (Figure 3.2n-x), and 5 of them reached $C = 1$, indicating convergence to the reference Pareto front (Figure 3.2y).

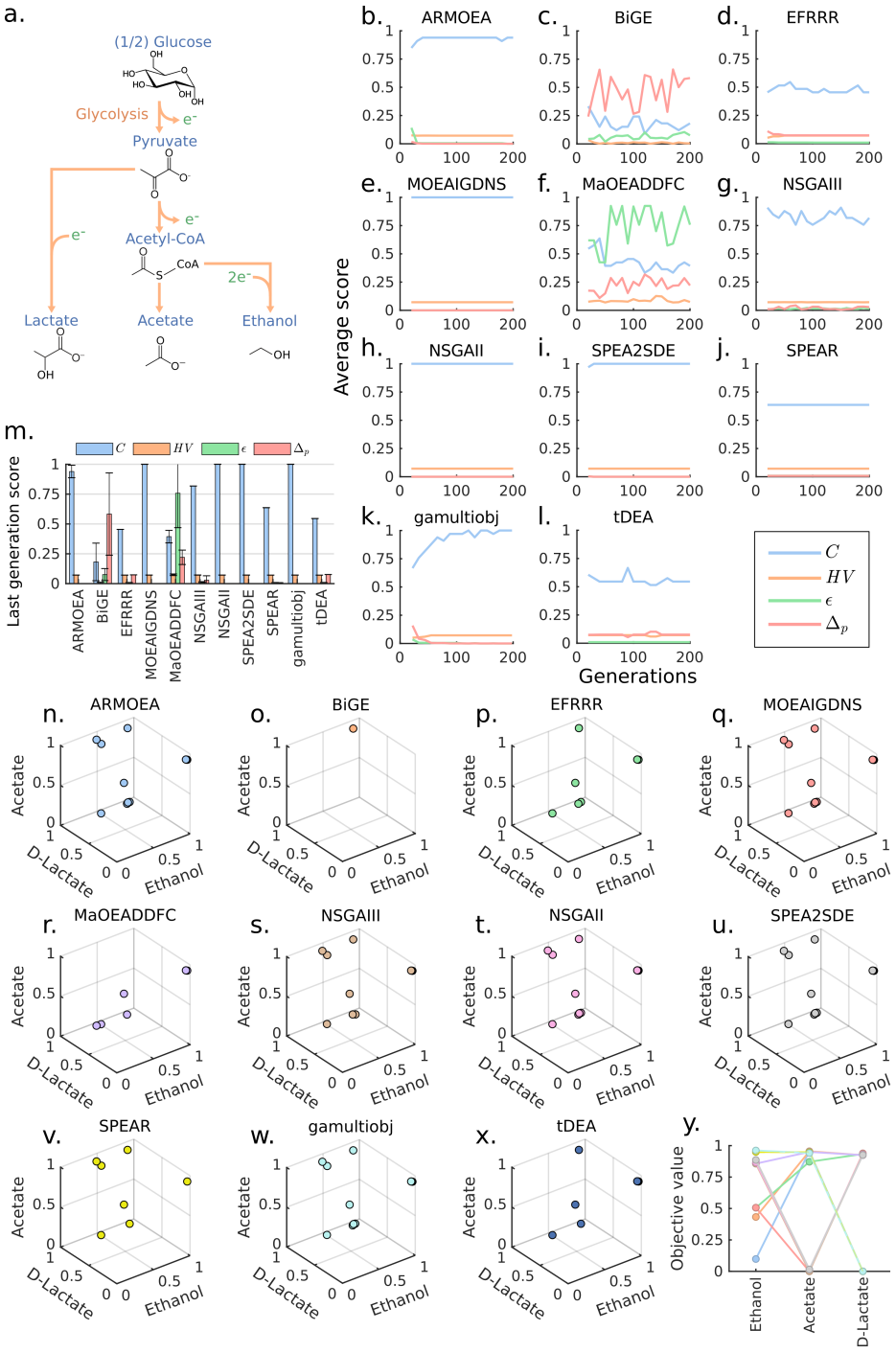


Figure 3.2: Comparison of MOEAs for a 3-objectives design problem. (**a**) The simplified metabolic pathways for conversion of glucose to the target products. Reducing equivalents are presented with e^-. (**b-l**) Generation-dependent performance metrics for various MOEAs. (**m**) Performance metrics for various MOEAs at the last generation. (**n-x**) Pareto fronts of various MOEAs at the last generation. It should be noted that only the first replicate is plotted for clear illustration. (**y**) Reference Pareto front ($\mathcal{PF}^*$). Each line represents a solution.

3.3.2 Case 2: A 10-objectives design problem

Using the same model and design parameters as in Case 1, we expanded the number of objectives to represent a more realistic scenario. These objectives correspond to 6 endogenous pathways for biosynthesis of D-lactate, acetate, ethanol, formate, pyruvate and L-glutamate and 4 heterologous pathways for biosynthesis of propanol, butanol, isobutanol, and pentanol. The additional objectives increased the difficulty of the problem, leading to more notable difference among algorithm performances (Figure 3.3a-k). The SPEA2SDE algorithm displayed consistent improvement of C as generations progressed, and quickly reached the smallest values of ϵ and Δ_p (Figure 3.3h). Other algorithms, including ARMOEA and MOEAIGDNS, also improved their ϵ values with the increasing number of generations and reached the same final values of ϵ and Δ_p as SPEA2SDE (Figure 3.3a, d). However, SPEA2SDE approached $C \cong 0.6$, which is twice the value reached by the next best-performing methods (Figure 3.3l). Remarkably, SPEA2SDE outperformed every other algorithm in all metrics, except HV. The HV metric continues to show bias towards algorithms that generated a small number of points and scored poorly in other metrics.

3.3.3 Case 3: Use of large population size overcomes poor MOEA performance

Increasing the number of objectives often leads to a combinatorial explosion of the number of feasible Pareto optimal points and consequently causes poor MOEA performance. This problem can be alleviated by using a larger population size to sample a broader volume of solution space[115]. To test this strategy for the 10-objectives design problem above, we increased the population size from 100 to 1000 individuals while all other parameters remained unchanged. The result showed that ARMOEA, MOEAIGDNS, NSGAII, SPEA2SDE (the best performer in Case 2), and *gamultiobj*, could reach C of 0.7, ϵ of 0, and Δ_p of 0 in fewer than 50 generations (Figure 3.4a, d, g, h, j). These 5 algorithms also yielded very similar final values across all metrics (Figure 3.4l). The remaining algorithms converged to considerably lower C values (Figure 3.4b, c, e, f, i, k).

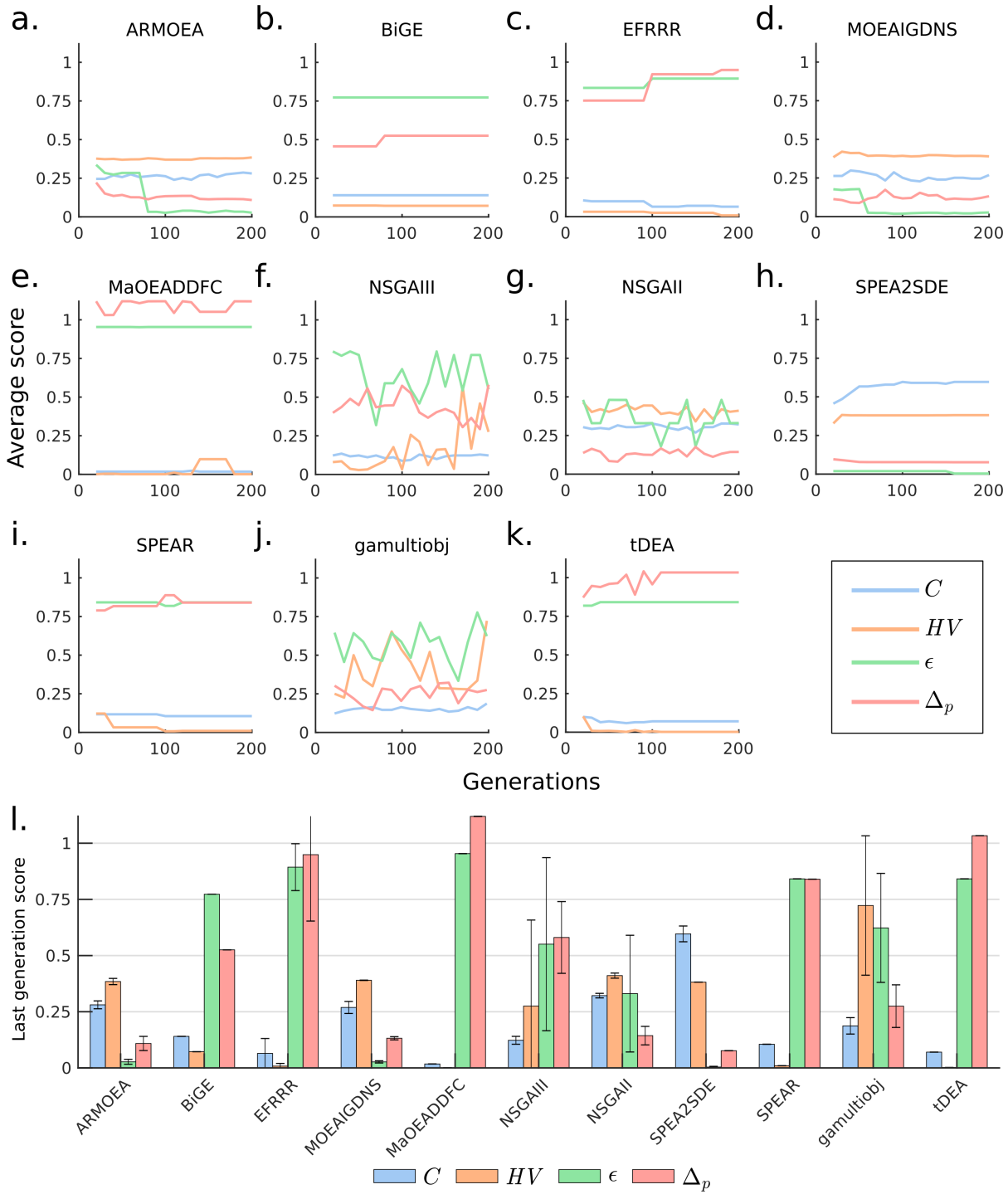


Figure 3.3: Comparison of MOEAs for a 10-objective design problem. (**a-k**) Generation-dependent performance metrics for various MOAEs. (**l**) Performance metrics for various MOEAs at the last generation.

Remarkably, NSGAII/*gamultiobj*, that is not considered a many-objective solver, performed better than more recent many-objective algorithms such as NSGAIII.

One limitation of using larger populations is an increased cost in computational time. We observed that a 10-fold increase in population sizes resulted in a 10-fold increase in the run times (Figure 3.5). Nonetheless, all metrics reached a stable value in the top performing algorithms after 50 generations (out of 200 total), suggesting that fewer generations were needed by using a larger population size. Among the best performing algorithms with large population sizes, *gamultiobj*, implemented in the Matlab Optimization Toolbox, required the shortest run time, followed by NSGAII and SPEA2SDE implemented in PlatEMO.

3.4 Conclusions

In this study, we evaluated the performance of several MOEAs to solve the modular cell design problem. SPEA2SDE, the recently developed many-objectives method, was the best performing MOEA for limited population sizes in our study. However, for sufficiently large populations, several algorithms attained the best results, including the well-established NSGAII, which performed better than more recently developed many-objectives MOEAs. We used the most popular performance metrics to compare MOEAs and found that the coverage (C) metric is the most valuable indicator. This metric can provide an intuitive quantitative meaning and tends to increase monotonically with the number of generations simulated. In contrast, hypervolume (HV) generally did not differentiate algorithm performance and was misleading in some scenarios where an algorithm generated very few solutions. Overall, these results highlight the need for empirical testing of MOEAs towards specific problems and of the population size as a more important factor in performance than the unique heuristics commonly used by different algorithms. For the application of modular cell engineering, efficient MOEAs will enable the design of modular cell(s) compatible with many product synthesis modules for large-scale metabolic networks and the identification of more diverse and better solutions that will provide more viable options for practical implementation.

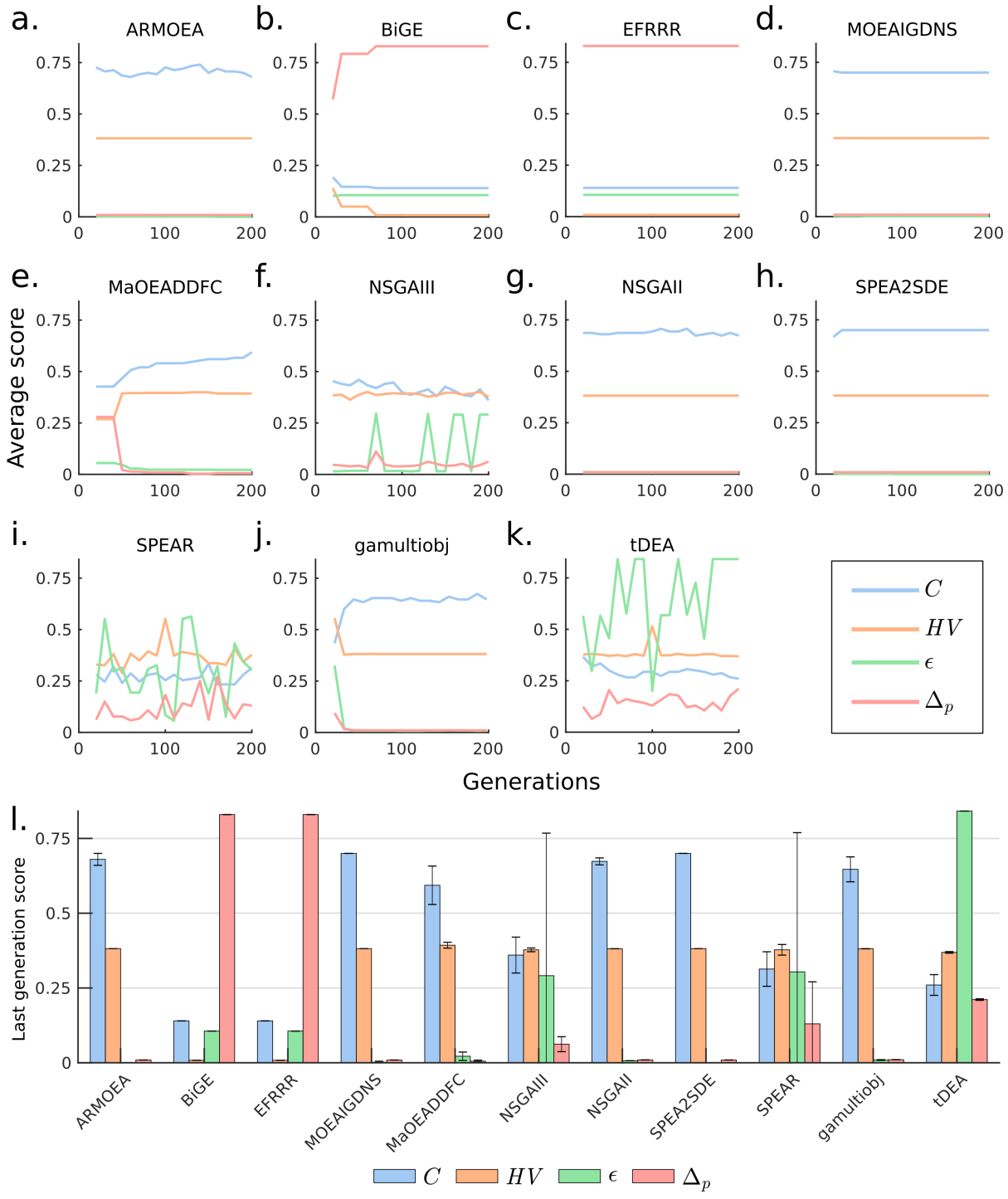


Figure 3.4: Comparison of MOEAs for a 10-objectives design problem with larger population sizes (**a-k**) Generation-dependent performance metrics for various MOAEs. (**l**) Performance metrics for various MOEAs at the last generation.

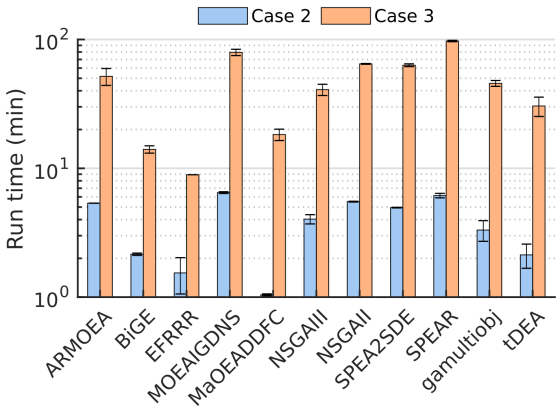


Figure 3.5: Wall-clock run times for the 10-objectives design problem with population sizes of 100 (Case 2) and 1000 (Case 3).

Chapter 4

Development of linear formulations to solve the modular cell problem and application to design a universal modular cell

This chapter is based on the publication *Harnessing natural modularity of cellular metabolism to design a modular chassis cell for a diverse class of products by using goal attainment optimization. Garcia, S., and Trinh, C. T. In review, 2020.* As first author I lead the development, implementation, and writing of this study. Supplementary Files S1 and S2 are provied as attachments.

Abstract

Modular design is key to achieve efficient and robust systems across engineering disciplines. Modular design potentially offers advantages to engineer microbial systems for biocatalysis, bioremediation, and biosensing, overcoming the slow and costly design-build-test cycles in the conventional cell engineering approach. These systems consist of a modular cell chassis compatible with modules that enable programmed functions such as overproduction of a desirable chemical. We previously proposed a multi-objective optimization framework

coupled with metabolic flux models to design modular cells and solved it using multi-objective evolutionary algorithms. However, such approach might not achieve solution optimality and hence limit design options for experimental implementation. In this study, we developed the goal attainment formulation compatible with optimization algorithms that guarantee solution optimality. We applied goal attainment to design an *Escherichia coli* modular cell capable of synthesizing all molecules in a biochemically diverse library at high yields and rates with only a few genetic manipulations. To elucidate modular organization of the designed cells, we developed a flux variance clustering (FVC) method by identifying reactions with high flux variance and clustering them to identify metabolic modules. Using FVC, we identified reaction usage patterns for different modules in the modular cell, revealing its broad pathway compatibility is enabled by the natural modularity and flexible flux capacity of endogenous core metabolism. Overall, this study not only sheds light on modularity in metabolic networks from their topology and metabolic functions but also presents a useful synthetic biology toolbox to design modular cells with broad applications.

4.1 Introduction

Microbial metabolism can be engineered to produce a large space of molecules from renewable and sustainable feedstocks[150]. Currently, only a handful of fuels and chemicals out of the many possible molecules offered by nature are industrially produced by microbial conversion, mainly because the strain engineering process is too laborious and expensive[196]. To overcome this roadblock and produce a more diverse range of molecules requires innovative technologies for rapid and economically competitive strain engineering[273,196,150]. The principles of modular design that have shown great success in traditional engineering disciplines can be adapted to construct modular cell biocatalysts in a plug-and-play fashion with minimal strain design-build-test cycles[85].

Multi-objective optimization is a powerful mathematical framework widely applied in engineering disciplines to tackle the optimal design of a complex system with multiple conflicting objectives[48,218]. This framework has recently been used to design modular systems in conventional engineering[106], and to explain the modularity of natural biological

systems that enable cellular robustness and adaptability[138,129,47,244,234]. Using multi-objective optimization, microbial metabolism can be redirected to generate modular production strains that are systematically assembled from an engineered modular cell and exchangeable production modules, where each module synthesizes a target molecule[86]. This modular cell design approach, known as ModCell2, uses the principles of mass balance and thermodynamics of biochemical reaction networks to predict metabolic fluxes upon genetic manipulations[86,83]. Based on such flux predictions, a multi-objective optimization problem is then formulated and solved with a multi-objective evolutionary algorithm (MOEA)[312,61] to yield a sample of the Pareto front (i.e., the set of optimal solutions to the problem with minimal trade-offs among objectives) that a designer can explore genetic manipulation targets for modular cell engineering.

In this study, we developed ModCell2-MILP, a ModCell2-based formulation to be compatible with mixed integer linear programming (MILP) algorithms. This framework presents a significant advancement from ModCell2 in solving the multi-objective strain design problem for modular cell engineering. Specifically, ModCell2-MILP is developed to (i) guarantee optimal solutions, (ii) completely enumerate alternative solutions of a target design, and (iii) describe practical engineering goals more directly (e.g., design of a modular cell where all production modules lead to a product yield above 50% of the theoretical maximum). By applying ModCell2-MILP to analyze the genome-scale metabolic network of *Escherichia coli*, we could identify a universal modular cell that is compatible with a diverse class of production modules. To gain a mechanistic view into the modular organization of metabolic networks, we developed a flux variance clustering (FVC) method by identifying reactions with high flux variance and clustering them to identify metabolic modules. Using FVC, we found that broad pathway compatibility of the modular cell is facilitated by its natural modularity and flexible flux capacity of endogenous core metabolism. We anticipate ModCell2-MILP and FVC can serve as powerful tools for not only elucidating natural and synthetic metabolic modularity but also rationally designing modular cells for broad biotechnological applications in biocatalysis, bioremediation, and biosensing.

4.2 Methods

4.2.1 Modular cell design

Design principles

ModCell design enables rapid assembly of production strains with desirable phenotypes from a modular (chassis) cell[272,86,85]. More specifically, a modular cell contains core metabolic pathways shared among production modules (Figure 4.1a). The chassis interfaces with the modules through enzymatic and genetic synthesis machinery and precursor metabolites (Figure 4.1b). Modules contain auxiliary regulatory and metabolic pathways (Figure 4.1c) that enable a desired phenotype for optimal biosynthesis of a target molecule, for example, *weak growth coupled to product formation* (*wGCP*), where a positive correlation between growth and product synthesis rates is enforced (Figure 4.1d)[30,139,86]. The *wGCP* phenotype is useful because it enables rapid pathway optimization by adaptive laboratory evolution[77,268] or high-throughput genetic library selection[89]. The design objective phenotypes are determined from cellular growth and product synthesis rates based on steady-state stoichiometric metabolic models[204]. A modular cell is said to be *compatible* with a module if the design objective of the resulting production strain is above a specified threshold. The different biochemical nature of production modules to synthesize target metabolites can make the design objectives compete with each other and also the cellular objectives (e.g., biomass formation) compete with the engineering objectives (e.g., product formation), turning the ModCell design problem into a multi-objective and multi-level optimization problem.

Multi-objective optimization formulation

The modular cell design problem is stated as a general multi-objective optimization problem of the form:

$$\max_{x} \quad F(x) = (f_1(x), f_2(x), \ldots, f_K(x))^\top \quad \text{s.t. } x \in X \tag{4.1}$$

where f_k is the desirable phenotype for production module k, x are the problem variables including binary design variables corresponding to genetic manipulations, and X is the set of

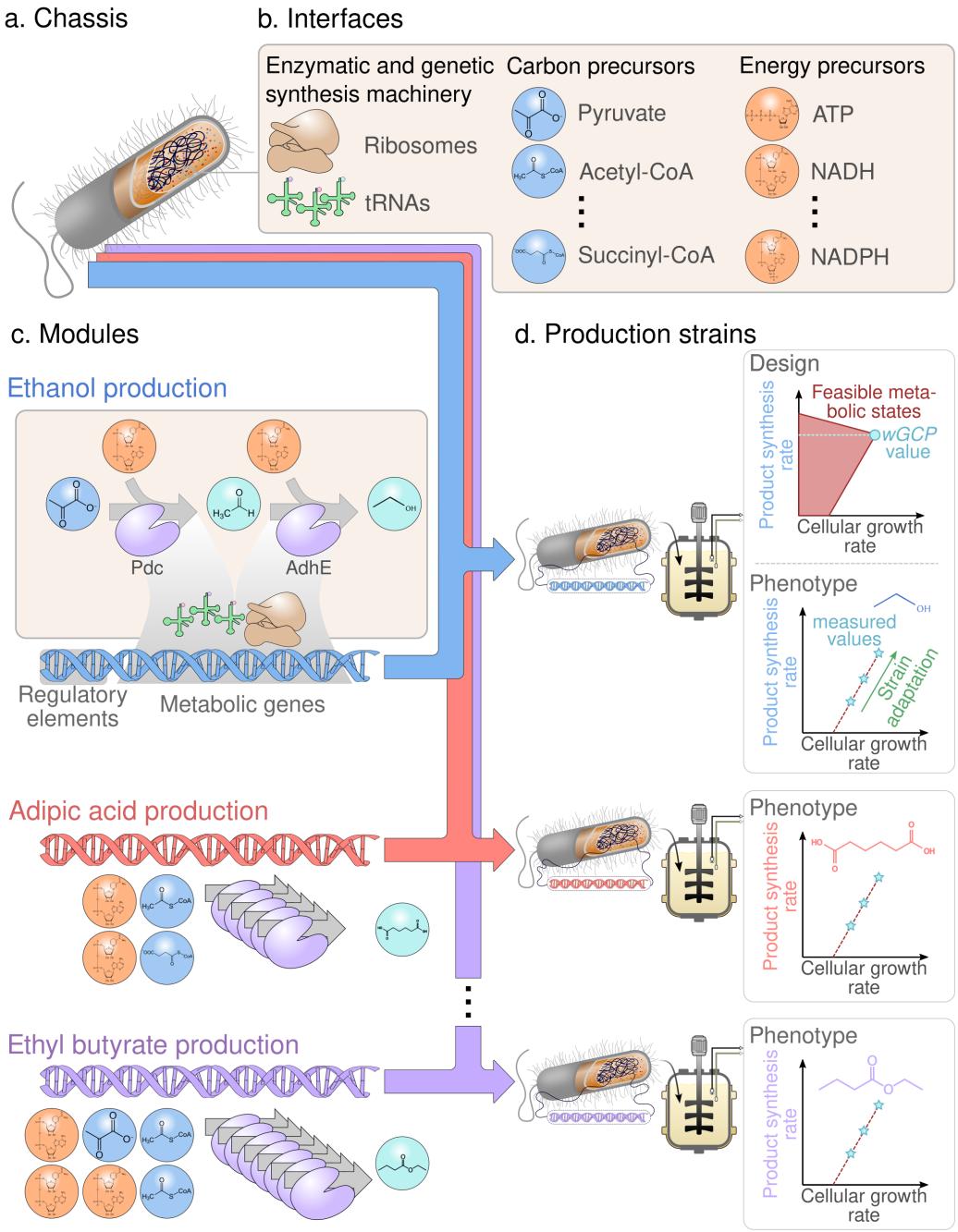


Figure 4.1: Principles of modular cell design. (a) Modular cell chassis. (b) Interfaces. (c) Production modules. (d) Production strains. A modular cell is designed to provide the necessary precursors for biosynthesis pathway modules that are independently assembled with the modular cell to generate production strains exhibiting desirable phenotypes. The *wGCP* phenotype, one of the possible design objectives, enforces the coupling between the desirable product synthesis rate and the maximum cellular growth rate.

constraints including mass balance of metabolism. Optimal solutions for the multi-objective optimization problem (4.1) are defined using the concept of domination: A vector $a = (a_1, \ldots, a_K)^\top$ *dominates* another vector $b = (b_1, \ldots, b_K)^\top$, denoted as $a \prec b$, if and only if $a_i \geq b_i \ \forall i \in \{1, 2, \ldots, K\}$ and $a_i \neq b_i$ for at least one i. A feasible solution $x^* \in X$ of the multi-objective optimization problem is called a Pareto optimal solution if and only if there does not exist a vector $x' \in X$ such that $F(x') \prec F(x^*)$. The set of all Pareto optimal solutions is called Pareto set:

$$PS := \{x \in X : \nexists\, x' \in X, F(x') \prec F(x)\} \tag{4.2}$$

The projection of the Pareto set in the objective space is denoted as Pareto front:

$$PF := \{F(x) : x \in PS\} \tag{4.3}$$

Different feasible points in PS (i.e., different genetic manipulations) which map to a single point in PF (i.e., the same phenotype) are denoted *alternative solutions*.

The design variables x in ModCell2 correspond to chassis reaction deletions, that remove undesired metabolic functions, and module reaction insertions, that allow to identify optimal module configurations without extensive prior knowledge of the product synthesis pathway. The constraint set X is comprised of two types: (i) flux simulation constraints (e.g., mass balance, reaction reversibility, and flux bound) that allow to predict fluxes in the design objectives upon genetic manipulations, and (ii) implementation constraints that involve the maximum number of reaction deletions in the chassis (denoted by α) and the maximum number of module reaction insertions per module (denoted by β). The following sections describe the problem formulation in detail using the definitions compiled in the Definitions Section.

Design objectives

Design objectives, f_k, that correspond to specific metabolic phenotypes within the space of feasible steady-state reaction fluxes, Π_{km}, of production network k (i.e., the combination of

the chassis network with the production module k) and metabolic state m, are defined as follows:

$$\Pi_{km}(e_{jk}) := \{v_{jkm} \in \mathbb{R} : \tag{4.4}$$

$$\sum_{j \in \mathcal{J}_k} S_{ijk} v_{jkm} = 0 \qquad \forall i \in \mathcal{I}_k \tag{4.5}$$

$$l_{jkm} e_{jk} \leq v_{jkm} \leq e_{jk} u_{jkm} \; \forall j \in \mathcal{J}_k \tag{4.6}$$

Here, v_{jkm} is the rate (mmol/gCDW/hr) of reaction j in production network k under metabolic state m. Constraint (4.5) enforces mass balance for all metabolites according to reaction stoichiometry given by the coefficients S_{ijk}, and constraint (4.6) imposes bounds, l_{jkm} and u_{jkm}, for the metabolic fluxes according to reaction reversibility, experimentally measured values, and specified metabolic state. The binary variable e_{jk} is used in the overall optimization problem to indicate whether reaction j in production network k is removed and thus cannot carry any flux. Two metabolic states m are considered, growth and non-growth, denoted μ and $\bar{\mu}$, respectively. These states are differentiated by their flux bounds l_{jkm} and u_{jkm}. For growth state, the lower bound of the biomass formation reaction that represents cell division, v_{Xkm}, is set to a minimum value of γ, i.e., $l_{Xk\mu} = \gamma$ ($\forall k \in \mathcal{K}$), while there is no upper limit to growth, i.e., $u_{Xk\mu} = \infty$ ($\forall k \in \mathcal{K}$). On the other hand, for the non-growth state both bounds are set to 0, i.e., $l_{Xk\bar{\mu}} = 0$ and $u_{Xk\bar{\mu}} = 0$ ($\forall k \in \mathcal{K}$).

Given the feasible metabolic flux space, Π_{km}, the following design objectives, based on the product synthesis rate reaction, v_{Pkm}, are of interest:

$$f_k^{wGCP} = \frac{v_{Pk\mu}}{v_{Pk\mu}^{max}} \in [0,1], \qquad \forall k \in \mathcal{K} \tag{4.7}$$

$$f_k^{lsGCP} = b_\mu \frac{v_{Pk\mu}}{v_{Pk\mu}^{max}} + b_{\bar{\mu}} \frac{v_{Pk\bar{\mu}}}{v_{Pk\bar{\mu}}^{max}} \in [0, b_\mu + b_{\bar{\mu}}], \quad \forall k \in \mathcal{K} \tag{4.8}$$

$$f_k^{NGP} = \frac{v_{Pk\bar{\mu}}}{v_{Pk\bar{\mu}}^{max}} \in [0,1], \qquad \forall k \in \mathcal{K} \tag{4.9}$$

The product synthesis fluxes, including $v_{Pk\mu}$, $v^{max}_{Pk\mu}$, $v_{Pk\bar{\mu}}$, and $v^{max}_{Pk\bar{\mu}}$, are computed by solving the following linear programming problems:

$$v_{Pk\mu} \in \arg\max\{v_{Xk\mu} - \epsilon\, v_{Pk\mu} : v_{k\mu} \in \Pi_{k\mu}(e_{jk})\} \tag{4.10}$$

$$v^{max}_{Pk\mu} \in \arg\max\{v_{Pk\mu} : v_{k\mu} \in \Pi_{k\mu}(e_{jk} = 1,\ \forall j \in \mathcal{J}_k)\} \tag{4.11}$$

$$v_{Pk\bar{\mu}} \in \arg\min\{v_{Pk\bar{\mu}} : v_{k\bar{\mu}} \in \Pi_{k\bar{\mu}}(e_{jk})\} \tag{4.12}$$

$$v^{max}_{Pk\bar{\mu}} \in \arg\max\{v_{Pk\bar{\mu}} : v_{k\bar{\mu}} \in \Pi_{k\bar{\mu}}(e_{jk} = 1,\ \forall j \in \mathcal{J}_k)\} \tag{4.13}$$

The maximum product synthesis fluxes (4.11) and (4.13) used for objective scaling are only calculated once by not using any deleted reactions ($e_{jk} = 1$), while the target phenotype fluxes (4.10) and (4.12) are functions of the deleted reactions e_{jk}. The design objectives, *wGCP* (4.7), *lsGCP* (4.8), and *NGP* (4.9), were previously proposed[86] and briefly described here. The weak growth coupled to product formation objective (*wGCP*) (4.7) seeks to maximize the minimum product rate at the maximum cellular growth, which is accomplished by a titled objective function[74] (4.10). The linearized strong growth coupled to product formation (*lsGCP*) (4.8) objective seeks to maximize the minimum product synthesis rate at the non-growth state $v_{Pk\bar{\mu}}$ in addition to the goal of *wGCP*. In f^{lsGCP}_k, b_μ, $b_{\bar{\mu}}$ are the weights on the growth and non-growth objectives, respectively. Finally, the non-growth production (*NGP*) (4.9) objective seeks to optimize the minimum product synthesis rate during the non-growth state.

Design constraints

All the constraints of the modular cell design problem are gathered as follows:

$$\Omega := \{f'_k \in \mathbb{R},\ y_j, z_{jk}, d_{jk}, w_k, e_{jk} \in \{0,1\} : \tag{4.14}$$

$$\sum_{j \in \mathcal{C}} (1 - y_j) \leq \alpha \tag{4.15}$$

$$\sum_{j \in \mathcal{C} - \mathcal{N}_k} z_{jk} \leq \beta_k \qquad \forall k \in \mathcal{K} \tag{4.16}$$

$$z_{jk} \leq 1 - y_j \qquad \forall j \in \mathcal{C} - \mathcal{N}_k,\ k \in \mathcal{K} \tag{4.17}$$

$$d_{jk} = y_j \vee z_{jk} \qquad \forall j \in \mathcal{C},\ k \in \mathcal{K} \tag{4.18}$$

$$f'_k = f_k w_k \qquad \forall k \in \mathcal{K} \tag{4.19}$$

$$e_{jk} = (d_{jk} \wedge w_k) \vee \neg w_k \qquad \forall j \in \mathcal{C},\ k \in \mathcal{K} \tag{4.20}$$

$$w_k \leq M^w f_k \qquad \forall k \in \mathcal{K} \tag{4.21}$$

$$v_{Pkm} \in \Psi_{km}(e_{jk}) \qquad \forall k \in \mathcal{K},\ m \in \mathcal{M}\,\} \tag{4.22}$$

Constraints (4.15)-(4.18) are formulated for practical limitations and features of the modular cell. Specifically, the two variables that represent design choices for genetic manipulations include: (i) y_j that takes a value of 0 if reaction j is deleted in the chassis (and consequently in all production networks) and 1 otherwise and (ii) z_{jk} that takes a value of 1 if reaction j is inserted in production network k. The maximum number of reaction deletions, is limited by α through constraint (4.15) while the maximum number of module reactions in each module β_k is imposed by (4.16). Constraint (4.16) excludes non-candidate reactions $\mathcal{N}_k$ (since $j \in \mathcal{C} - \mathcal{N}_k$) so that module reactions can be fixed (i.e., $z_{jk} = 1$), according to problem-specific knowledge. Constraint (4.17) ensures that only reactions deleted in the chassis can be inserted back to the modules. Constraint (4.18) indicates that reaction j is deleted in production network k if the reaction is deleted in the chassis and not added as a module reaction. The designer can gradually increase α and β_k to obtain solutions with higher performance.

Constraints (4.19)-(4.21) are introduced for modeling purposes. The indicator variable, w_k, is introduced to allow for certain production networks to be ignored from the final solution. Without w_k, the whole multi-objective problem becomes infeasible if a set of deletions renders one of the production networks infeasible (e.g., its minimum growth rate cannot be accomplished). However, in practice it is acceptable for some modules not to work with the chassis cell. If $w_k = 0$, the objective value $f'_k = 0$ (4.19) and reaction deletions do not apply to network k since $e_{jk} = 1$ (4.20); if $w_k = 1$, $f'_k = f_k$ and $e_{jk} = d_{jk}$, where f_k is any of the design objectives presented earlier (4.7)-(4.9). The use of w_k is likely to introduce symmetry (i.e., alternative integer solutions with no practical meaning) due to cases where

$f_k = 0$ for a given k while the associated production network remains feasible, allowing w_k to take a value of 0 or 1. This symmetry is removed by enforcing w_k to be 0 if $f_k = 0$ (4.21).

Finally, constraint (4.22) indicates that the fluxes featured in the design objectives, v_{Pkm}, are contained in the polytope Ψ_{km}. The space of v_{Pkm} is originally defined as an optimization problem (4.10)-(4.13), thus representing a non-linear constraint and turning the ModCell design problem into a bilevel optimization problem. These inner optimization problems are linearized, leading to Ψ_{km} as described in the following sections.

Linearization of logical expressions

The logical expressions in Ω are replaced by the following linear constraints in the final problem formulation:
$d_{jk} = y_j \vee z_{jk}$ corresponds to:

$$d_{jk} \leq y_j + z_{jk} \tag{4.23}$$

$$d_{jk} \geq y_j \tag{4.24}$$

$$d_{jk} \geq z_{jk} \tag{4.25}$$

$$0 \leq d_{jk} \leq 1 \tag{4.26}$$

$f'_k = f_k w_k$ corresponds to:

$$f'_k \leq w_k M^{obj} \tag{4.27}$$

$$f'_k \leq f_k - (1 - w_k) M^{obj} \tag{4.28}$$

$$f'_k \leq f_k \tag{4.29}$$

$$0 \leq f'_k \leq M^{obj} \tag{4.30}$$

$e_{jk} = (d_{jk} \wedge w_k) \vee \neg w_k$, given $r_{jk} = d_{jk} \wedge w_k$, corresponds to:

$$e_{jk} = r_{jk} + 1 - w_k \tag{4.31}$$

$$r_{jk} \leq w_k \tag{4.32}$$

$$r_{jk} \leq d_{jk} \tag{4.33}$$

$$r_{jk} \geq w_k + d_{jk} - 1 \tag{4.34}$$

$$0 \leq r_{jk} \leq 1 \tag{4.35}$$

Linearization of inner optimization problems

Non-linear constraints expressed as linear programming problems can be linearized using basic mathematical programming theory. Consider the following canonical linear program, with primal variables $x \in \mathbb{R}^n$ and its dual variables $u \in \mathbb{R}^m$:

$$\max \quad \{c^\top x : Ax \leq b,\ x \geq 0\} \tag{4.36}$$

$$\min \quad \{b^\top u : A^\top u \geq c,\ u \geq 0\} \tag{4.37}$$

the strong duality theorem states that the objective functions of primal (4.36) and dual (4.37) are equal at their optima, $c^\top x^* = b^\top y^*$. Thus the optimal solution to the primal problem is described by the following linear constraints:

$$x^* \in \{x \in \mathbb{R}^n : \tag{4.38}$$

$$Ax \leq b \tag{4.39}$$

$$A^\top u \geq c \tag{4.40}$$

$$c^\top x = b^\top u \tag{4.41}$$

$$x, u \geq 0\,\} \tag{4.42}$$

Using the strong duality theorem as presented by Maranas and Zomorrodi[176], the inner optimization problems (4.22) are linearized as follows:

$$\Psi_{km}(e_{jk}) := \{v_{jkm} \in \mathbb{R} : \tag{4.43}$$

$$\sum_{j \in \mathcal{J}_k} S_{ijk} v_{jkm} = 0 \qquad \forall i \in \mathcal{I}_k \tag{4.44}$$

$$l_{jkm} e_{jk} \leq v_{jkm} \leq e_{jk} u_{jkm} \qquad \forall j \in \mathcal{J}_k \tag{4.45}$$

$$\sum_{i \in \mathcal{I}_k} \lambda_{ikm} S_{ijk} - \mu^l_{jkm} + \mu^u_{jkm} = c_{jkm} \qquad \forall j \in \mathcal{J}_k \tag{4.46}$$

$$\lambda_{ikm} \in \mathbb{R} \qquad \forall i \in \mathcal{I}_k \tag{4.47}$$

$$0 \leq \mu^l_{jkm} \leq M \qquad \forall j \in \mathcal{J}_k \tag{4.48}$$

$$0 \leq \mu^u_{jkm} \leq M \qquad \forall j \in \mathcal{J}_k \tag{4.49}$$

$$\sum_{j \in \mathcal{J}_k} c_{jkm} v_{jkm} = - \sum_{j \in \mathcal{J}_k - \mathcal{C}} (l_{jkm} \mu^l_{jkm}) + \sum_{j \in \mathcal{J}_k - \mathcal{C}} (u_{jkm} \mu^u_{jkm}) - \sum_{j \in C} (l_{jkm} p^l_{jkm}) + \sum_{j \in C} (u_{jkm} p^u_{jkm}) \tag{4.50}$$

$$p^l_{jkm} \leq e_{jk} M \qquad \forall j \in \mathcal{C} \tag{4.51}$$

$$\mu^l_{jkm} - (1 - e_{jk}) M \leq p^l_{jkm} \leq \mu^l_{jkm} \qquad \forall j \in \mathcal{C} \tag{4.52}$$

$$0 \leq p^l_{jkm} \leq M \qquad \forall j \in \mathcal{C} \tag{4.53}$$

$$p^u_{jkm} \leq e_{jk} M \qquad \forall j \in \mathcal{C} \tag{4.54}$$

$$\mu^u_{jkm} - (1 - e_{jk}) M \leq p^u_{jkm} \leq \mu^u_{jkm} \qquad \forall j \in \mathcal{C} \tag{4.55}$$

$$0 \leq p^u_{jkm} \leq M \qquad \forall j \in \mathcal{C} \,\} \tag{4.56}$$

Constraints (4.44)-(4.45) correspond to the primal metabolic network problem and were introduced earlier in Π_{km}. Constraints (4.46)-(4.49) correspond to the dual problem. We use the dual variables, λ_{ikm}, for the primal mass balance constraints (4.44), together with μ^l_{jkm} and μ^u_{jkm} for the primal flux bound inequalities (4.45) involving lower and upper reaction bounds respectively. Constraints (4.47)-(4.49) emphasize the domain of the dual variables, with M being a large value above the expected value of any dual variable. Constraints (4.50)-(4.56) correspond to the strong duality equality. The left hand side of the strong duality equality (4.50) features the objectives presented in (4.10) for $m = \mu$ and (4.12) for $m = \bar{\mu}$. On the right hand side, products of binary and continuous variables appear, thus requiring linearization variables p^l_{jkm} and p^u_{jkm}. Constraints (4.51)-(4.56) ensure that $p^l_{jkm} = e_{jk} \mu^l_{jkm}$ and $p^u_{jkm} = e_{jk} \mu^u_{jkm}$.

Conversion of a multi-objective problem into a single-objective problem

The multi-objective optimization problem (4.1) is now described entirely in terms of linear constraints through Ω. However, to make the formulation compatible with MILP solver

algorithms, the objective function vector, f', must be expressed as a scalar. To accomplish this without loss of relevant information, we employed blended and goal attainment formulations[177].

Blended formulation In the blended formulation all objectives are summed as follows:

$$\max \quad \sum_{k \in \mathcal{K}} a_k \, f'_k \quad \text{s.t. } f' \in \Omega \tag{4.57}$$

where a_k is a scalar weighting factor associated with the design objective of product k. Different Pareto optimal solutions can be obtained by varying these weights. The blended formulation always provides Pareto optimal solutions as long as $a_k > 0$ ($\forall k \in K$). In practice, the product priority, a_k, can be determined by criteria such as product market value or "pathway readiness level" (i.e., certain pathways are easier to engineer than others).

Goal attainment formulation In the goal attainment problem a target value is defined for each objective:

$$\min \quad \sum_{k \in \mathcal{K}} (a_k^+ \delta_k^+ + a_k^- \delta_k^-) \tag{4.58}$$

s.t.

$$f'_k + \delta_k^+ - \delta_k^- = g_k \quad \forall k \in \mathcal{K} \tag{4.59}$$

$$\delta_k^+, \delta_k^- \geq 0 \qquad \forall k \in \mathcal{K} \tag{4.60}$$

$$f' \in \Omega \tag{4.61}$$

The problem seeks to minimize the variables δ_k^+ and δ_k^- that represent the deficiency and excess of the objective f'_k from the target value g_k, respectively. Weighting parameters a_k^+ and a_k^- correspond to different types of discrepancy to be minimized. In general, when it is important to meet the target value without exceeding it, we set $a_k^+ = a_k^- = 1$; however, when the design objective is required to be greater or equal than the target value, we set $a_k^+ = 1$ and $a_k^- = 0$, effectively converting (4.59) into $f'_k + \delta_k^+ \geq g_k$. Solutions to the goal attainment problem are not guaranteed to be Pareto optimal, even if all demands g_k are

met. To address this issue, the blended problem (4.57) can be solved where the objectives are constrained to be equal or greater than the values found by solving the goal attainment problem. In practice, the goal attainment formulation corresponds to the identification of the modular cell *compatible* with the largest number of modules. Here, a module k is said to be *compatible* if $f'_k \geq g_k$.

4.2.2 Implementation

Metabolic models

We used two parent models from which production networks were built, including: i) a core metabolic model of *E. coli*[272] to develop the ModCell2-MILP algorithm and compare with previous ModCell2 results[86], and ii) the iML1515 genome-scale metabolic model of *E. coli*[188] for biosynthesis of a library of endogenous and heterologous metabolites, including 4 organic acids, 6 alcohols, and 10 esters (Figure 4.2)[5,11,146,197,224,241,274,277,301,304]. These models were configured as in the previous ModCell2 study[86], briefly: Anaerobic conditions were imposed by setting oxygen exchange fluxes to be 0, and the glucose uptake rate was constrained to be at most 10 mmol/gCDW/h. When using the genome-scale model iML1515 to simulate *wGCP* designs, only the commonly observed fermentative products (acetate, CO_2, ethanol, formate, lactate, succinate) were allowed for secretion as described elsewhere[283].

ModCell2-MILP simulations

ModCell2-MILP was implemented using Pyomo[101], an algebraic modeling language embedded in the Python programming language. All simulations were performed on a computer with an Intel Core i7-3770 processor, 32 GB of random access memory, and the Arch Linux operative system. The implementation and scripts used to generate the results of this manuscript are available as part of the ModCell2 package via File S2 and `https://github.com/trinhlab/modcell2`.

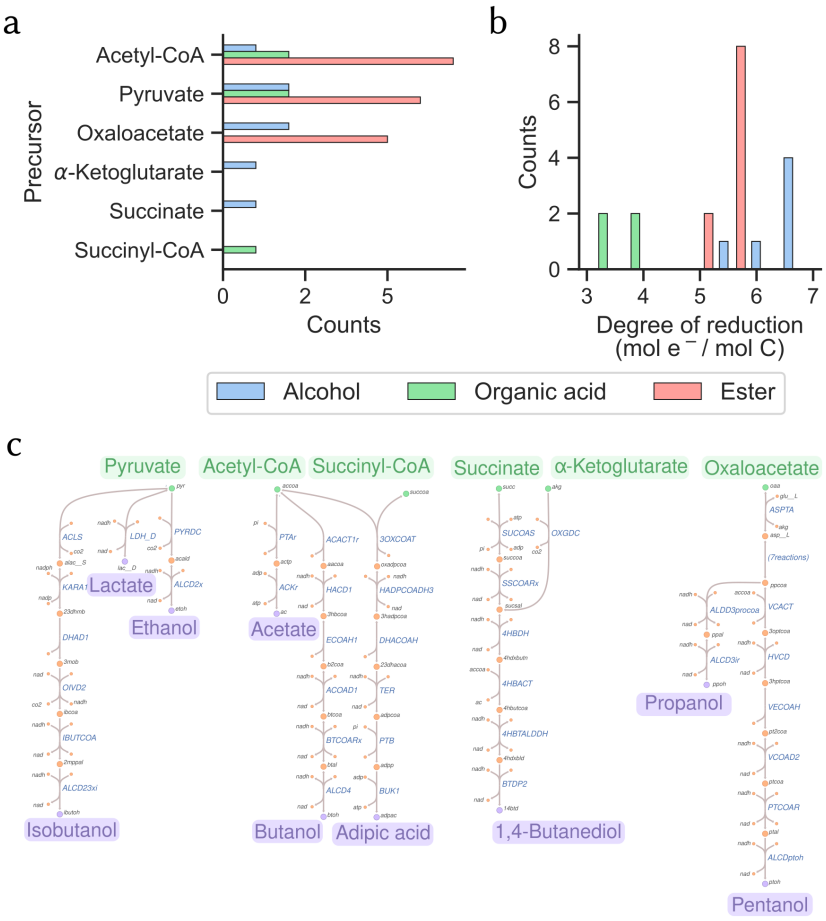


Figure 4.2: Biochemical and metabolic diversity of the 20 production modules. (a) Precursor metabolite distribution, using the 12 essential biomass precursors as a reference. (b) Degrees of reduction of the produced metabolites. (c) Production pathway metabolic modules excluding esters.

Optimization solver configuration

The Pyomo[101] implementation of ModCell2-MILP was solved with IBM Ilog Cplex 12.8.0. To avoid incorrect solutions associated with numerical issues the following Cplex parameters were changed from their default values: (i) *numerical emphasis* was set to "true", (ii) *integrality tolerance* was lowered to 10^{-7}, and (iii) the *MIP pool relative gap* was increased to 10^{-4} for enumerating alternative solutions. Alternative solutions were enumerated using the Cplex "populate" procedure.

4.2.3 Analysis methods

Reference flux distribution

The reference flux distribution, $\frac{v^*_{jk}}{|v^*_{Sk}|}$, is determined by solving the following quadratic program based on the parsimonious enzyme usage hypothesis[173,152]:

$$\min_{v_{jk}} \quad \sum_{j \in \mathcal{J}_k} v_{jk}^2 \tag{4.62}$$

s.t.

$$\sum_{j \in \mathcal{J}_k} S_{ijk} v_{jkm} = 0 \qquad \forall i \in \mathcal{I}_k \tag{4.63}$$

$$l_{jk} \leq v_{jk} \leq u_{jk} \qquad \forall j \in \mathcal{J}_k \tag{4.64}$$

$$v_{Xk} = \text{MaxDesignBio} \tag{4.65}$$

Constraint (4.63) corresponds to mass balance for the metabolic network. Constraint (4.64) corresponds to reaction bounds, including reaction deletions found in the modular cell design problem. Constraint (4.65) fixes the biomass formation rate, v_{Xk}, to the maximum reachable by the design. This value (MaxDesignBio) is obtained by maximizing v_{Xk} subject to (4.63) and (4.64). The reference flux distribution $\frac{v^*_{jk}}{|v^*_{Sk}|}$ represents the desired metabolic state of a *wGCP* designed production network. This distribution, if feasible, is unique because the convex optimization problem is formulated with a positive definite quadratic objective function (see Theorem 16.4 in Nocedal and Wright[198]).

Flux variance clustering

Flux variance clustering (FVC) seeks to identify and group reactions that exhibit high flux changes under different conditions. Reactions with high flux variance can be considered as metabolic interfaces between core metabolic processes and pathway modules. In our study, each condition corresponds to a different product synthesis phenotype. FVC implementation entails three steps. First, flux distributions are simulated for each condition and standard deviations for each reaction are calculated. Then, a standard deviation threshold is set to select the reactions with highest flux changes across conditions. Finally, these selected

reactions are clustered to identify patterns that repeat under specific conditions. The filtering threshold is set to capture the top reactions with most change while maintaining a sufficiently small list that is biologically meaningful. In our study, we chose an *ad-hoc* value that captures well-known reactions in *E. coli* central metabolism. To cluster the selected reactions, only flux magnitudes, not directionality, were considered in our study. Further, each reaction flux was normalized by the maximum value of that same reaction across all production networks. Clustering was performed using the method *clustermap()* with default clustering-related parameters from the Python library Seaborn 0.9.

Flux sampling

To determine an ensemble of flux distributions for a production network, we used the ACHR algorithm[130] in the COBRA toolbox[105]. Constraints for flux sampling simulation include the reaction deletions and module reactions found in the ModCell design problem solution, a fixed substrate uptake rate of -10 mmol glucose/gCDW/hr, and a minimum product synthesis flux of 50% of its maximum value.

Metabolic map drawing

Drawings of metabolic map were performed using the Escher[135] tool (`https://escher.github.io`) that produces *svg* files. Coloring, highlighting candidate reactions, and other systematic adjustments of metabolic maps were done with the Python-based *lxml* module. Additional editing for visual enhancement was done with the Inkscape software.

4.3 Results and discussion

4.3.1 Performance and solution time optimization of ModCell2-MILP

ModCell2-MILP can not only reproduce the results of the original ModCell2 formulation but also find more alternative solutions

To evaluate ModCell2-MILP, we compared its performance with the previously developed ModCell2 platform[86] that solves the optimization problem with multi-objective evolutionary algorithms (MOEAs). As a basis of comparison, we used the same *E. coli* core metabolic model, maximum number of deletion reactions α, and maximum number of module-specific reactions β_k for both ModCell2 and ModCell2-MILP. Due to fundamental differences in problem formulations for MOEA and MILP, we used the *lsGCP* design objective for ModCell2-MILP with multiple weighting factors, a_k, specifically selected to reproduce previous results, in the blended formulation and the *sGCP* design objective for ModCell2 (File S1). The results showed that ModCell2-MILP could generate the same Pareto optimal designs like ModCell2. In addition, ModCell2-MILP enumerated a larger number of alternative solutions than ModCell2. For example, the design named *sGCP-5-0-6* generated by ModCell2 had 3 alternative solutions while ModCell2-MILP found 8 alternative solutions. By increasing α to 8 and β to 2, we could identify a utopia design (i.e., one solution with the maximum value for all objectives) with 192 alternative solutions, expanding the possibilities for experimental implementation.

Tuning MILP formulations significantly improves solution times

We considered three techniques that can improve solution times of ModCell2-MILP, including:

(i) *Fixing the network feasibility indicator* w_k. If all modules are expected to be compatible with a final ModCell design (i.e., $f_k > 0, \forall k \in \mathcal{K}$), w_k is set to be 1 for all $k \in \mathcal{K}$ to avoid computational efforts in finding non-optimal feasible solutions.

(ii) *Flux bound tightening.* Constraints of the form $e_{jkm}l_{jkm} \leq v_{jkm} \leq e_{jkm}u_{jkm}$ are known to result in weak linear relaxations, i.e., feasible values of v_{jkm} are far from their bounds l_{jkm} and u_{jkm}. To tighten the formulation by making continuous relaxations closer to the feasible integer solution, smaller values of u_{jkm} and l_{jkm} are determined by solving a series of linear programs that maximize and minimize each flux v_{jkm} in the parent production networks $\Pi_{km}(e_{jk} = 1, \forall j \in \mathcal{J}_k)$.

(iii) *Benders decomposition.* ModCell2-MILP has a separable structure compatible with Benders decomposition[90,76] that creates a master problem, using binary variables and associated constraints (4.15)-(4.21), and sub-problems for each production network $\Psi_{km}(e_{jk})$ with fixed binary variables. This decomposition implementation is automatically done by Cplex 12.8.

We evaluated these three techniques for tuning MILP formulations and used the core *E. coli* model[86] for the benchmark study. The results showed that flux bound tightening, fixed w_k, and Benders decomposition could reduce the solution time to find solutions by 50%, 80%, and 95%, respectively (Table 4.1). By combining these techniques, the solution time was shortened by 96% from 63.3 s to 2.8 s. In subsequent studies, we used these three tuning techniques to solve the ModCell design problem unless otherwise noted.

Table 4.1: Solution time reduction by tuning the ModCell2-MILP formulation. Fixed network indicator means $w_k = 1\ , \forall k \in \mathcal{K}$. The simulations were performed in triplicates.

Feasibility indicator w_k fixed	Benders decomposition	Bounds tightened	Solution time (s)
No	No	No	63.3 ± 16.9
No	No	Yes	32.5 ± 10.2
No	Yes	No	3.6 ± 0.1
No	Yes	Yes	3.4 ± 0.4
Yes	No	No	13.8 ± 2.7
Yes	No	Yes	11.9 ± 1.7
Yes	Yes	No	2.7 ± 0.3
Yes	Yes	Yes	2.8 ± 0.1

Choice of design parameters affect solution time

In designing a modular cell with ModCell2-MILP, the designer needs to specify the formulation type (i.e., blended or goal attainment formulation), the target phenotype (e.g., *wGCP, lsGCP*, and *NGP*), and the limits of deletion reactions (α) and module-specific reactions (β_k). We evaluated the impact of these parameters on solution time using the *E. coli* core model (Figure 4.3). Regardless of the formulation type, increasing α and β led to harder problems and hence required more solution time due to the exponentially increasing number of feasible solutions as expected. The goal attainment formulation took longer time to solve for the *lsGCP* and *NGP* design objectives, but about the same time for the *wGCP* design objective. Interestingly, the overall difficulty of *wGCP* is higher than that of *lsGCP* in both the blended and goal attainment formulations, despite *lsGCP* having approximately twice the number of constraints. Furthermore, the *NGP* design objective could be solved most quickly, likely due to the narrower design space associated with the no-growth associated production of target metabolites.

4.3.2 Design of a universal modular cell for a genome-scale metabolic model of *E. coli*

Reduction of the candidate reaction deletion set enables ModCell2-MILP to find modular cell designs for a large-scale metabolic network

Finding genetic modifications towards a desired phenotype using mathematical optimization for large-scale metabolic networks is a computationally expensive task, due to the combinatorial search space spanned by a large number of reaction deletion candidates in the network[74,281]. Preprocessing of metabolic networks to reduce reaction candidates is not only critical but also practical for experimental implementation. The set of reaction candidates in the iML1515 *E. coli* model[188] was reduced from 2,712 to 276 by ModCell2[86]. Using this model and the *wGCP* objective, an *E. coli* modular cell was then identified to be compatible with 17 out of 20 products with requirement of only 4 reaction deletions[86]. Since MOEA implemented in ModCell2 does not guarantee optimality, here we aimed to evaluate the

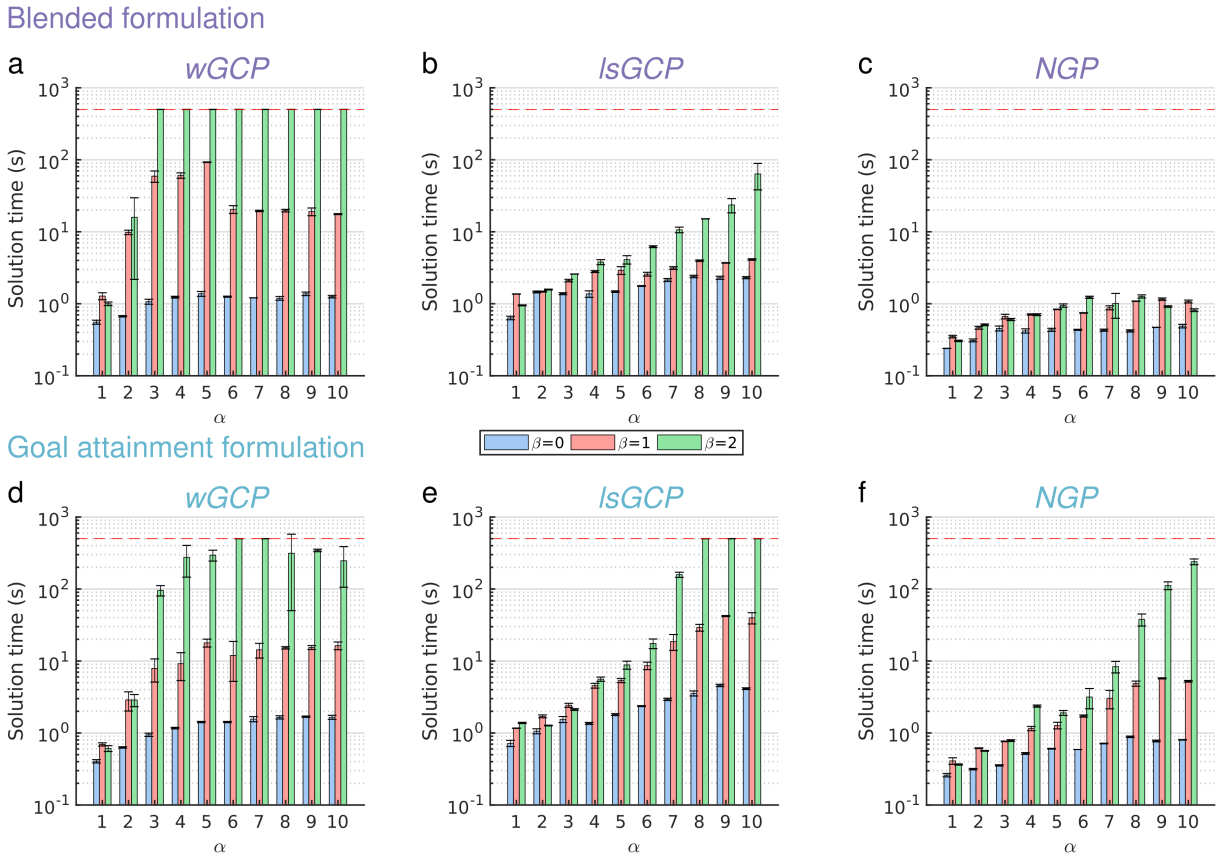


Figure 4.3: Effect of design parameters on solution time. We examined the target design objective (i.e., *wGCP*, *lsGCP*, and *NGP*) and the limits of deletion reactions α and endogenous module-specific reactions β_k, on computation time for solving the ModCell2-MILP problem with the blended (a-c) and goal attainment (d-f) formulations. A time limit of 500 seconds indicated by a red dashed line was used in all cases, but only reached by certain *wGCP* and *lsGCP* cases with $\beta \geq 2$. The simulations were performed in duplicates.

capability of ModCell2-MILP for handling a large-scale metabolic network and identifying the Pareto optimality and potential alternative solutions.

We applied ModCell2-MILP to analyze the same iML1515 model with a set of 20 products using the same design parameters (i.e., α and β_k) and the blended formulation with all objective weights $a_k = 1$. The simulation shows that ModCell2-MILP could not solve the ModCell design problem to optimality over 2 days of run time, likely due to the large number of candidate deletion reactions still present in the genome-scale model. Currently the best MILP solution algorithms do not scale well with parallel computing. In order to obtain solutions within an acceptable time, the set of candidate reactions must be further reduced. Since only a small subset of all metabolic reactions in genome-scale models tend to be deleted by strain design algorithms[74,137,86], we used a pool of *wGCP* designs with $\alpha = 4, 5, 6$ and $\beta = 0, 1$ reported with ModCell2[86] to identify relevant deletion candidates. From a set of 601 designs found by ModCell2, only 33 out of 276 reaction deletion candidates were used at least once. Hence, these 33 reactions were used to create a new, computationally-tractable set of reaction candidates. This new set contains reactions mostly from the well-characterized central metabolic pathways (Figure 4.4a) while the original set includes reactions in peripheral pathways that lead to biomass synthesis. Interestingly, within these 33 reaction candidates, only a few are used in most designs (Figure 4.4b), highlighting the importance of their removal in growth-coupled production phenotypes. Reactions with high deletion frequencies mainly occur in high-flux central metabolic pathways (Figure 4.4c), closely associated with cellular energetics and carbon precursors that interface with the production modules (Figure 4.4d).

Using the reduced reaction deletion candidate set, ModCell2-MILP could find an optimal solution in $\sim$ 30 min and enumerated all optimal solutions in $\sim$ 8 hours. All the optimal solutions found by ModCell2-MILP in this case were in agreement with those previously found by ModCell2[86]. It should be noted that a reduced deletion candidate set can be identified for any metabolic model and target products by using previous designs identified with either ModCell2 or single-phenotype strain design algorithms[166]. This heuristic for reaction deletion candidate selection might affect design optimality to the extent that it

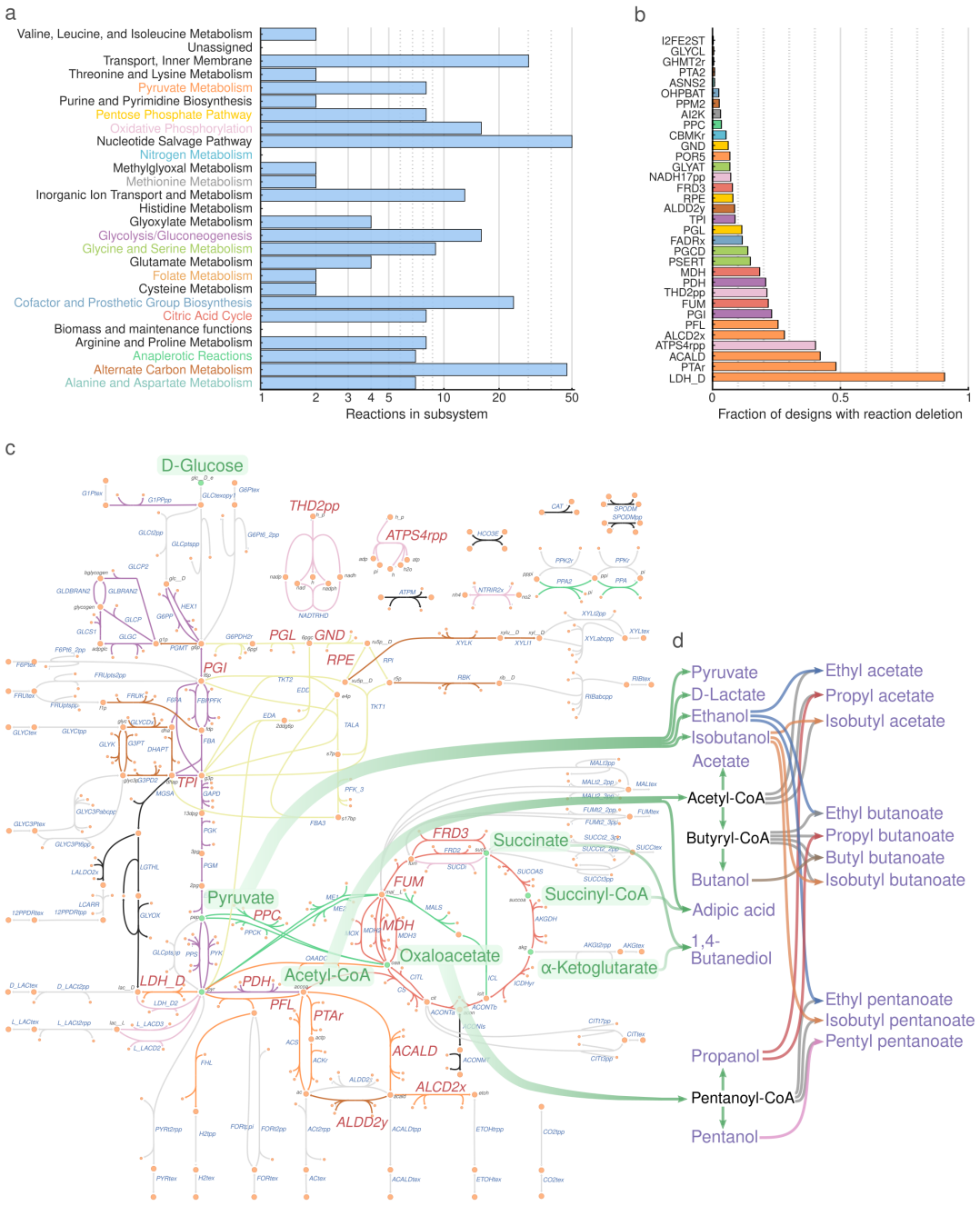


Figure 4.4: Metabolic functions of deletion candidate reactions. (a) Subsystem distribution for the original set of 276 candidate reactions in the iML1515 model. Those subsystems that contain a reaction used in at least one design are colored. (b) Deletion frequency for the reduced set of 33 candidate reactions. The analysis is based on a pool of 601 *wGCP* designs from different α and β parameters whose Pareto fronts were previously determined with ModCell2.[86] Bar colors indicate membership of these reactions to the subsystems. (c) Metabolic map of core metabolism. Reactions are colored according to subsytem labels indicated in (a), reactions colored in light gray do not appear in any of the subsytems of (a), and reactions that are candidates for deletion, listed in (b), are labeled in red. (d) Link between major precursors and target products.

omits relevant reactions. However, using a sufficiently informative pool of designs from other strain design algorithms helps minimize the chances of missing relevant candidates.

ModCell2-MILP can identify a universal modular cell compatible with all exchangeable production modules

Based on the computationally tractable reaction deletion candidate set, we next evaluated whether the goal programming formulation could help identify a universal ModCell design that is compatible with all modules. By screening for increasing α and β_k, we found that a design with $\alpha = 6$ and $\beta_k = 1$ could overcome the performance trade-offs between modules and hence constitute a universal modular cell that is compatible with all production networks considered (Figure 4.5a). Once coupled with a module, each resulting production strain displays the engineered phenotype with the defined minimum objective goal of 0.5 (i.e., 50% of the theoretical maximum product yield attained at the maximum growth rate). Remarkably, most products surpassed this minimum goal with yields above 90% of the theoretical maximum values (Figure 4.5b). All production networks displayed a feasible metabolic space where an increase in product synthesis rate is needed to attain faster growth rates (Figure 4.5c). This designed phenotype is useful for optimal pathway selection using adaptive laboratory evolution[77,268] and/or pathway libraries[89].

4.3.3 Flexible metabolic flux capacity of *E. coli* core metabolism enables the design of a universal modular cell

Interface reactions responsible for metabolic flexibility of a universal modular cell are identified by comparing flux distributions of production networks

The designed universal modular cell (Figure 4.5) can theoretically adapt to the conflicting metabolic requirements of all production modules (Table 4.2). To gain further insight into this unique metabolic capability of the modular cell, we analyzed the simulated *reference flux distributions* of each production network using the *flux variance clustering* (FVC) analysis (Materials and Methods). Reactions with the highest flux changes across the production networks are likely critical for the proper operation of the universal modular cell and might

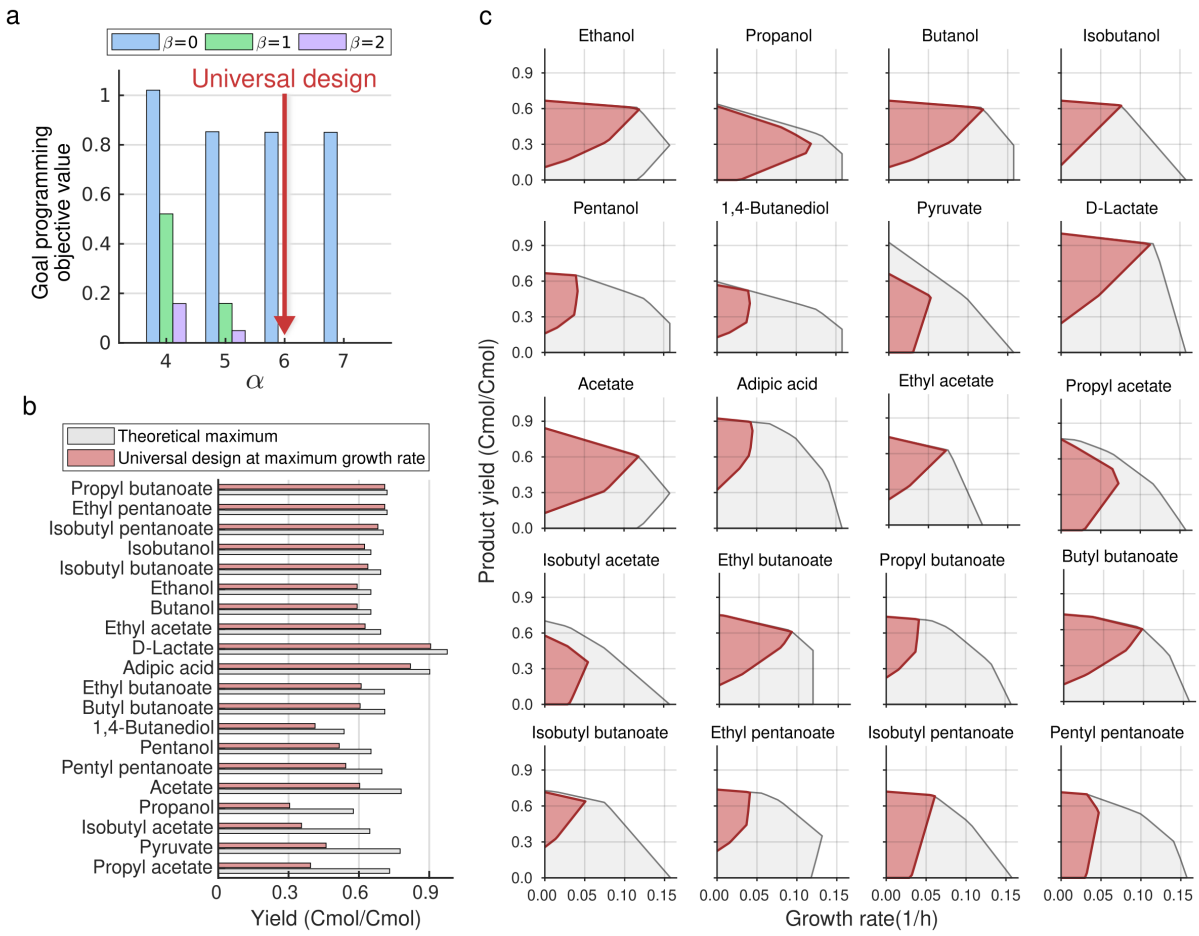


Figure 4.5: Identification of a universal modular cell compatible with all production modules under the *wGCP* design phenotype. (a) Goal programming solutions with increasing α and β values. The goal programming objective value (4.58) in the y-axis measures the difference between the performance of production strains and the target goal, i.e., $\sum_{k\in\{k\in\mathcal{K}:f'_k<g_k\}}(f'_k - g_k)$ where the target goal is set to be $g_k = 0.5$. The parameters $\alpha = 6$ and $\beta = 1$ are sufficient to identify a universal modular cell design meeting the required goal for all production networks. (b) Comparison between the yield performances of the designed modular production strains and maximum theoretical values. (c) The feasible flux spaces for the wild-type (gray) and designed modular production strains (crimson). Based on the *wGCP* design phenotype, to increase growth rate, each mutant must increase product synthesis rate. The genetic manipulations of this universal modular cell design are indicated in the metabolic map of Figure 4.6c.

present potential bottlenecks. Such reactions were identified by filtering their reference flux standard deviation calculated across production networks with an *ad hoc* threshold of 0.2 (mol/substrate mol). Over 90% of the 535 active reactions, each of which carries a non-zero flux in at least one production network, had standard deviation values below the threshold, indicating highly conserved metabolic core pathways among production networks. Only 9.5% of the active reactions presented a standard deviation magnitude above the threshold (Figure 4.6a).

Table 4.2: Overall production module pathway stoichiometries and associated simulated secretion fluxes of the universal modular cell design. DoR is the degree of reduction of the final product (mol e^- / mol C). Metabolite secretion profiles are determined from the simulated reference flux distributions (mol C / mol C) of the universal modular cell design. Flux abbreviations: r_p, product; r_{ac}, acetate; r_{co_2}, CO_2; r_{for}, formate; r_{succ}, succinate. Note that the negative CO_2 fluxes in pyruvate and acetate production networks indicate overall CO_2 uptake enabled by phosphoenolpyruvate carboxylase (PPC).

Overall reaction	DoR	r_p	r_{ac}	r_{co_2}	r_{for}	r_{succ}
pyr + nadh → **ethanol** | accoa + 2 nadh → **ethanol** (native)	7.0	0.58	0.01	0.27	0.04	-
oaa + glu + 2 atp + 2 nadph + nadh → akg + **propanol**	6.7	0.31	0.36	0.07	0.18	-
2 accoa + 4 nadh → **butanol**	6.5	0.59	0.01	0.28	0.04	-
2 pyr + nadph + nadh → **isobutanol**	6.5	0.62	-	0.31	-	-
oaa + glu + accoa + 3 nadh + 2 atp + 2 nadph → akg + **pentanol**	6.4	0.50	0.21	0.24	0.03	-
succ + akg + atp + 4 nadh + accoa → ac + **1,4-butanediol**	5.5	0.46	0.33	0.17	-	-
→ **pyruvate**	3.0	0.46	-	-0.16	-	0.66
pyr + nadh → **D-lactate**	3.7	0.91	-	-	-	-
accoa → atp + **acetate**	3.5	0.60	0.60	-0.30	0.61	-
accoa + succoa + 2 nadh → atp + **adipic acid**	4.0	0.82	0.05	0.04	0.06	-
accoa + pyr + nadh → **ethyl acetate**	5.0	0.63	-	-	0.32	-
accoa + oaa + glu + 2 atp + 2 nadph + nadh → akg + **propyl acetate**	5.2	0.41	0.30	-	0.24	-
accoa + 2 pyr + nadph + nadh → **isobutyl acetate**	5.3	0.36	-	0.02	0.06	0.52
2 accoa + 3 nadh + pyr → **ethyl butanoate**	5.3	0.61	-	0.09	0.23	-
2 accoa + 3 nadh + oaa + glu + 2 atp + 2 nadph → akg + **propyl butanoate**	5.4	0.68	0.03	0.23	0.04	-
4 accoa + 6 nadh → **butyl butanoate**	5.5	0.61	-	0.14	0.18	-
2 accoa + 3 nadh + 2 pyr + nadph → **isobutyl butanoate**	5.5	0.64	-	0.16	0.16	-
oaa + glu + accoa + 2 nadh + 2 atp + 2 nadph + pyr → akg + **ethyl pentanoate**	5.4	0.68	0.03	0.23	0.04	-
oaa + glu + accoa + 2 nadh + 2 atp + 3 nadph + 2 pyr → akg + **isobutyl pentanoate**	5.6	0.67	0.01	0.25	0.03	-
2 oaa + 2 glu + 2 accoa + 4 nadh + 4 atp + 4 nadph → 2 akg + **pentyl pentanoate**	5.6	0.53	0.22	0.20	0.02	-

In our case study of designing a universal modular cell compatible with all 20 production modules, unbiased hierarchical clustering (Figure 4.6b) revealed the presence of four interface reaction types in the core metabolism of *E. coli* that are activated to fit specific production modules (Figure 4.6c). In the context of chassis metabolism, an endogenous module

corresponds to a reaction or group of highly coupled reactions that become active to accomplish a certain metabolic function. The interface reaction classification can be understood in terms of location (i.e., proximity in the metabolic network) and three metabolic functions. The first function is the direction of carbon towards general precursor metabolites including (i) pyruvate and acetyl-CoA captured by acetyl-CoA-associated modules and (ii) oxaloacetate, succinate, succinyl-CoA, and α-ketoglutarate captured by TCA-associated modules. The second function is the direction of carbon from the precursor metabolites towards secretable molecules, captured by the upstream and TCA-associated modules. The third function is the use of ATP- and NADP(H)-dependent pathways required to maintain homeostasis, captured by the acetyl-CoA-associated and energetic modules. While these functions are conceptually separable, their biochemical manifestation overlaps, i.e., specific metabolic reactions or pathways can simultaneously fulfill several functions.

Interface reactions enable the exchangable production modules to properly operate with the universal modular cell. These interface reactions might become potential metabolic bottlenecks in practice if they cannot satisfy the predicted fluxes, and thus might be critical engineering targets when the associated production modules are used.

Acetyl-CoA-associated interface reactions. This interface type contains pyruvate formate lyase (PFL) and pyruvate dehydrogenase enzyme complex (PDH) reactions that convert pyruvate to acetyl-CoA. Intuitively, products derived from pyruvate, such as isobutanol, require a low flux through PFL and PDH while those derived from acetyl-CoA require a high flux. Remarkably, the redox states of production strains determine the ratios of PFL to PDH fluxes. For example, the ethanol production network has a relatively high flux through PDH and a low flux through PFL; however, for ethyl acetate that has a lower degree of reduction than ethanol (Table 4.2), PFL with formate secretion is prioritized over PDH with NADH generation. Note that our model did not include the regulatory restriction that PDH is inhibited in *E. coli* anaerobically because the function of PDH is equivalent with the coupling of PFL and heterologous NADH-dependent formate dehydrogenase (FDH) demonstrated experimentally for increased butanol[241,195] and pentanol[277] production.

Upstream interface reactions. This module type is formed by reactions located directly upstream of a secretable metabolite, often associated with the target production module, and thus provides the necessary precursor metabolite(s). Such reactions are commonly over-expressed in practice, e.g., the ECOAH1-HACD1-ACACT1r interface reaction group (comprising of 3-hydroxyacyl-CoA dehydratase, 3-hydroxyacyl-CoA dehydrogenase, and acetyl-CoA acetyl transferase) responsible for generating butyryl-CoA and the ACLS-DHAD1-KARA1 interface reaction group (comprising of acetolactate synthase, dihydroxy-acid dehydratase, and keto-acid reductoisomease) responsible for generating isobutyryl-CoA. These interface reactions can also become active to form byproducts in certain production networks, e.g., the PTAr-ACKr-ACT2rpp-ACtex interface reaction group (comprising of phosphate acetyl transferase, acetate kinase, and cytosolic and periplasmic acetate transport) that not only carries the highest flux in the acetate production network but also becomes active in the propanol-associated modules.

TCA-associated interface reactions This interface type has the same function as the upstream interface reactions but it is localized in the TCA (Krebs) cycle. Several products, including adipic acid, 1,4-butanediol, propanol, pentanol, and their associated esters, are derived from the TCA intermediates and interface with the universal modular cell via the TCA-derived interface reactions. The SUCOAS-MMM-MMCD interface reaction group (comprising of succinyl-CoA synthetase, Methylmalonyl-CoA mutase, methylmalonyl-CoA decarboxylase) must be activated to convert succinate into succinyl-CoA and then propanoyl-CoA. Remarkably, two routes are present to synthesize fumarate from oxaloacetate, including the conventional MDH-FUM interface reaction group (comprising of malate dehydrogenase and fumarase) that consumes NADH and the cyclic ASPTA-GLUDY-ASPT interface reaction group (comprising of aspartate transaminase, glutamate dehydrogenase, and L-aspartase) that consumes NADPH. These NADH/NADPH cofactors are not interchangeable due to the deletion of the transhydrogenase THD2pp in the universal modular cell, so the isobutyl pentanoate and pentyl pentanoate modules, that are derived from the ASPTA-GLUDY-ASPT interface reaction group, also have a high NADPH requirement. Some production networks, such as pyruvate and isobutyl acetate that are not based

on the TCA-derived interface reactions, secrete succinate instead of ethanol and/or lactate to balance redox by using the PPC-MDH-FUM-SUCCtex interface reactions (comprising of phosphoenolpyruvate carboxylase, malate dehydrogenase, fumarase, and succinate transport).

Energetic interface reactions This interface type primarily involves NAD(P)-dependent transhydrogenase (THD2pp) and ATP synthase (ATPS4rpp). Other reactions that allow coupling of phosphate- and electron-transfer cofactors are also included. The reactions in this module help buffer the diverse electron and ATP requirements of production networks. THD2pp is deleted in the chassis but used as a module reaction in the isobutanol and acetate production networks. In the case of isobutanol production, transhydrogenase expression has been demonstrated to increase the synthesis of NADPH and thus isobutanol[243]. Acetate has the smallest degree of reduction after pyruvate, which results in redox imbalance that is compensated via formate secretion. In conjunction with these mechanisms, ATPsynthase works in the reverse direction by hydrolyzing excess ATP. Other production networks also use ATPS4rpp to eliminate excess ATP as observed, for example, in the ethyl acetate production network. This strategy is consistent with ATP wasting approaches recently demonstrated[99].

Comparison between simulated and measured intracellular fluxes reveals flexible metabolic flux capacity of *E. coli* to accommodate the required wide flux ranges

Flux analysis of the production networks suggests that the core metabolic reactions (Figure 4.6b) require a wide range of fluxes when coupled with different production modules. To successfully implement this modular design in practice, we need to evaluate whether the metabolism of *E. coli* has the inherent metabolic flux capacity to accommodate these required fluxes. We compared the simulated reference flux distributions with a recent collection of 45 measured metabolic fluxes[131] that are collected from multiple studies across various conditions (e.g., growth under aerobic and anaerobic conditions, use of glucose or acetate or pyruvate as a carbon source) and genotypes (e.g., wild-type *E. coli* and mutants with single gene deletions)[117,123,310,311]. Note that while the experimental data set provides a flux distribution baseline for wild-type and relatively small deviations for single gene deletion

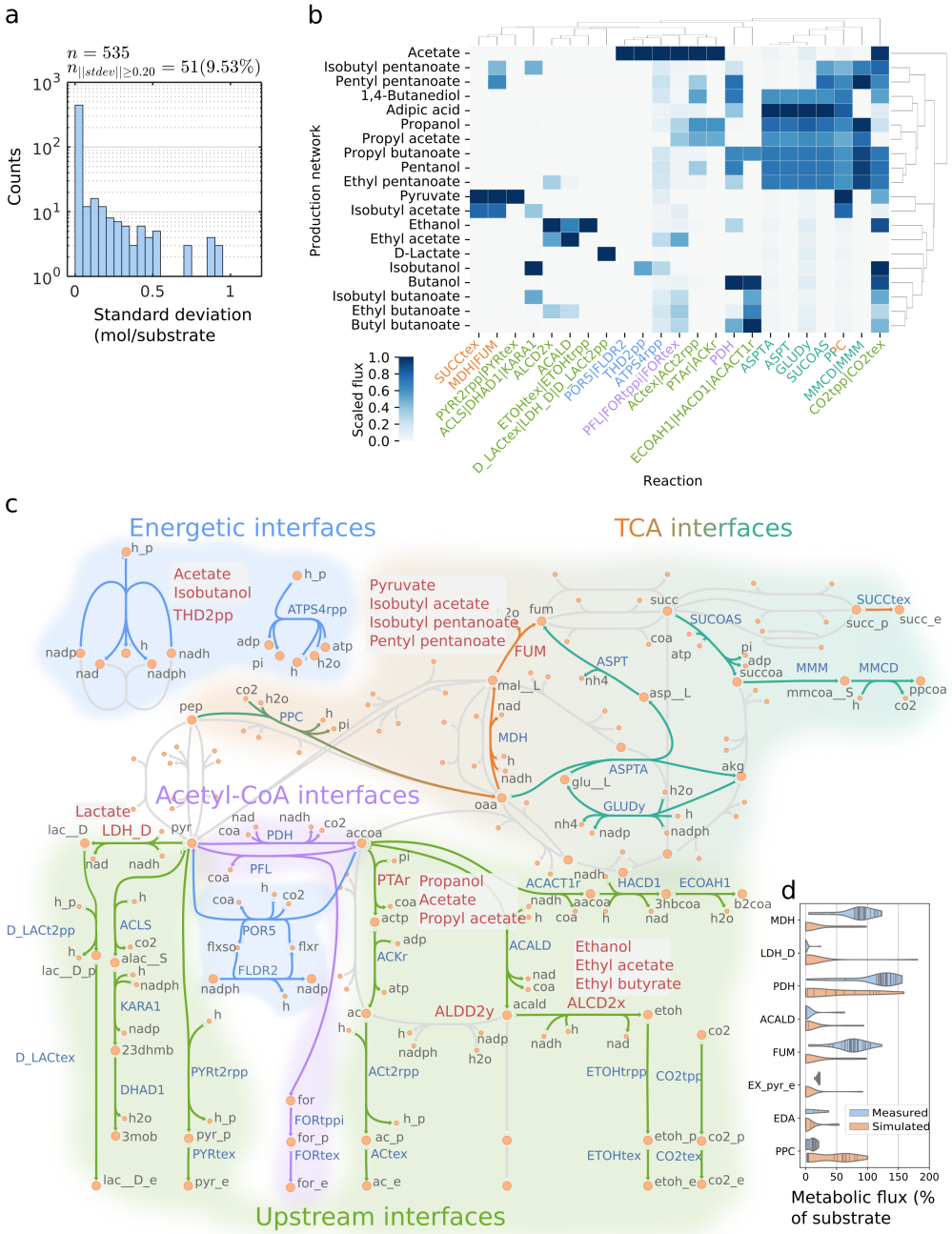


Figure 4.6: Flexible metabolic flux capacity of *E. coli* metabolism enables the universal modular cell design. (a) Standard deviation of each reaction flux across production networks. (b) Scaled fluxes of the 51 reactions with standard deviation magnitude above 0.2, excluding proton, water transport, and exchange reactions. A scaled flux for a reaction is determined by the reference flux distribution value divided by the maximum value of that reaction across all production networks. Hence, a scaled flux of 0 indicates a given reaction does not carry any flux, and a scaled flux of 1 indicates that this reaction carries the highest flux across production networks. (c) Interface reactions of the universal modular cell. The reactions colored in red are deleted in the chassis, but are used as module reactions in the production networks shown in the adjacent gray boxes. (d) Comparison between simulated and measured fluxes. The solid lines within the "violins" correspond to samples.

mutants, we anticipate that highly engineered strains with more gene deletions are likely to exhibit wider flux distributions.

Within the reactions present in the 23 groups that constitute interface reactions (Figure 4.6b), 8 reactions also appeared in the experimental dataset (Figure 4.6d). Remarkably, a highly consistent overlap of flux ranges was observed between the simulated and measured fluxes for malate dehydrogenase (MDH), pyruvate dehydrogenase (PDH), acetaldehyde dehydrogenase (ACALD), fumarase (FUM), and 2-dehydro-3-deoxy-phosphogluconate aldolase (EDA). For the cases of D-lactate dehydrogenase (LDH_D), and pyruvate secretion (EX_pyr_e) that are directly coupled with the biosynthesis of lactate and pyruvate, respectively, we observed the maximum simulated fluxes surpass the measured values, suggesting that further engineering of wild-type and single-gene deletion *E. coli* is needed to attain the requried fluxes. Indeed, previous studies[314,37] have been able to redirect metabolic fluxes in *E. coli* for yields of lactate and pyruvate above 75% of the theoretical maximum values by simultaneous elimination of competing fermentative pathways for biosynthesis of acetate ($\Delta ackA$), formate ($\Delta pflB$), and ethanol ($\Delta adhE$). The only remaining discrepancy between the simulated and measured fluxes is PPC. Studies, not included in the comparison data set, have reported up to 50% more PPC flux observed under aerobic conditions[210,245], which is still considerably below several of the simulated fluxes. This result suggests that PPC can be a potential metabolic bottleneck in certain production modules. One possible solution is to include in the affected production modules the heterologous PPC from *Actinobacillus succinogenes* which has been successfully over-expressed in *E. coli* for increased succinate production[133]. Additionally, bacterial PPC activity can be increased by elevating the acetyl-CoA pool[161].

Random sampling of metabolic fluxes confirms the narrow operation range of interface reactions

The reference flux distributions analyzed so far represent the ideal metabolic states for each production strain. However, other metabolic states might also exist. To address this uncertainty, we performed randomized flux sampling[130,105] for each production network under the constraint that product synthesis rate must be above 50% of the maximum

value. The results show that the metabolic flux distributions (S.M. 1 - Figure S2) for most reactions involved in the interface reactions are very narrow, except for the two alternative pathways of ethanol biosynthesis, i.e., the endogenous PDH-ACALD-ALCD2x route comprising of pyruvate dehydrogenase, acetaldehyde dehydrogenase, and alcohol dehydrogenase (Figure 4.8a) and the heterologous PYRDC-ALCD2x route comprising of pyruvate decarboxylase and alcohol dehydrogenase. These two pathways can be used interchangeably in the ethanol production network, where there is a linear correlation between PYRDC and PDH fluxes (Figure 4.8b). Notably, the sampled fluxes cannot indicate preferential use of either the ethanol synthesis route because the model does not take into account of kinetic (k_{cat}, k_{M}) and regulatory constraints. In summary, even though interface reactions must have flexible metabolic flux capacities to enable a universal modular cell to be compatible with various exchangeable production modules, they must also operate within in a narrow flux range when interfacing with a specific production module.

4.4 Conclusions

In this study, we formulated multi-objective modular strain design as blended and goal attainment optimization problems. These problems can be solved by MILP algorithms that guarantee Pareto optimal solutions, exhaustively search the space of alternative solutions, and specify design requirements such as module prioritization or universal compatibility. This multi-objective strain design approach can be extended with additional design variables (e.g., up- and down-regulation[212] or reaction insertion from database[213]) or alternative flux prediction models[44,65] to expand its applications, including use of exchangeable metabolic modules for bioremediation and biosensing. In terms of biological significance, the ModCell2-MILP and FVC methods developed could identify a universal modular cell that harnesses the inherent modularity and flexibility of native *E. coli* metabolism to properly interface with a variety of biochemically diverse pathways. This universal design was predicted to display a growth-coupled-to- product-formation phenotype for all pathways, enabling its use as a platform for pathway optimization through high-throughput library selection and/or adaptation. The feasibility of this universal design strategy is found to be consistent with

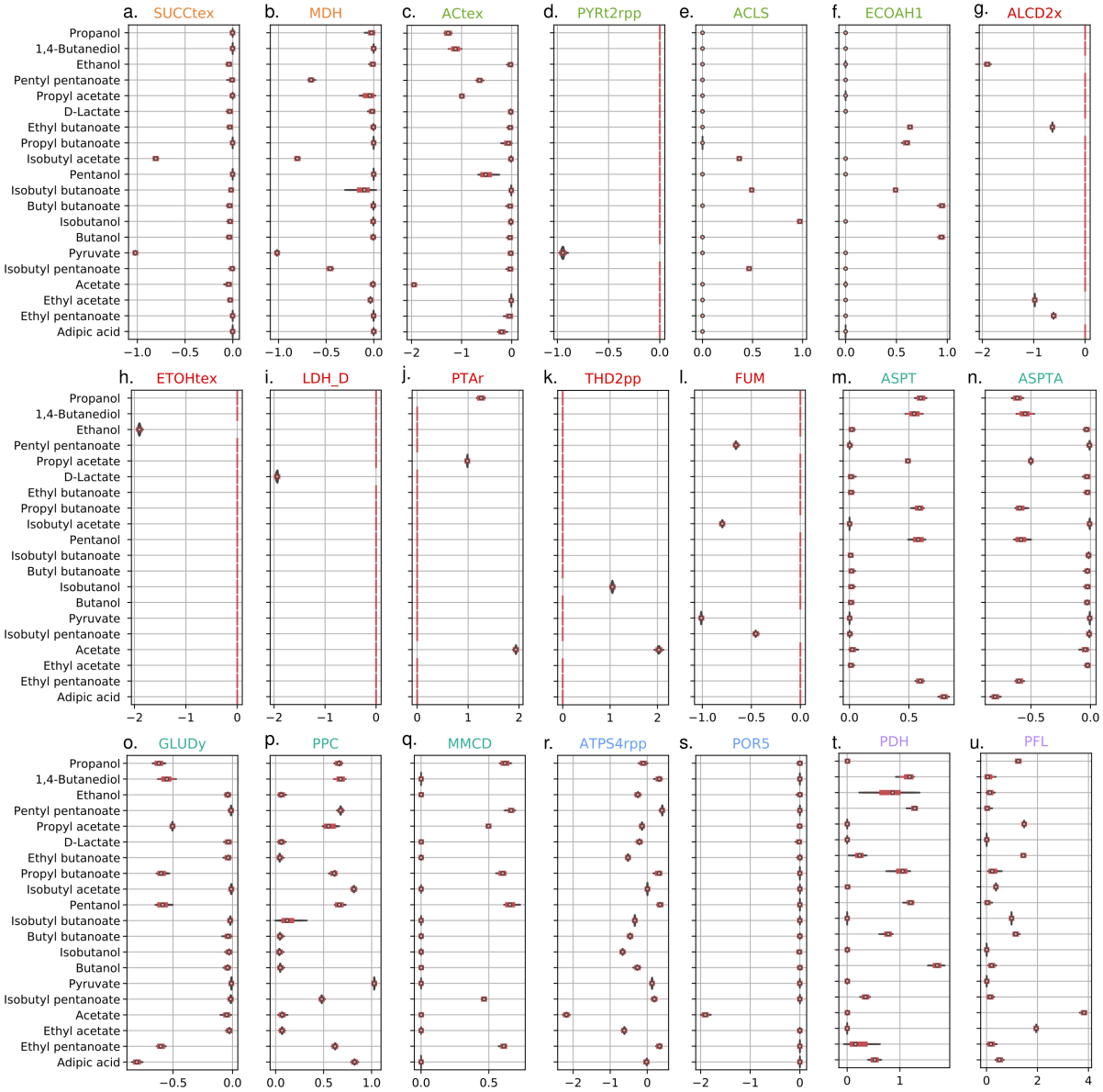


Figure 4.7: Violin plot of sampled reaction flux distributions. Reaction colors are consistent with Figure 4.6. The flux of SUCOAS could not be sampled since this reaction is involved in a thermodynamically infeasible cycle.

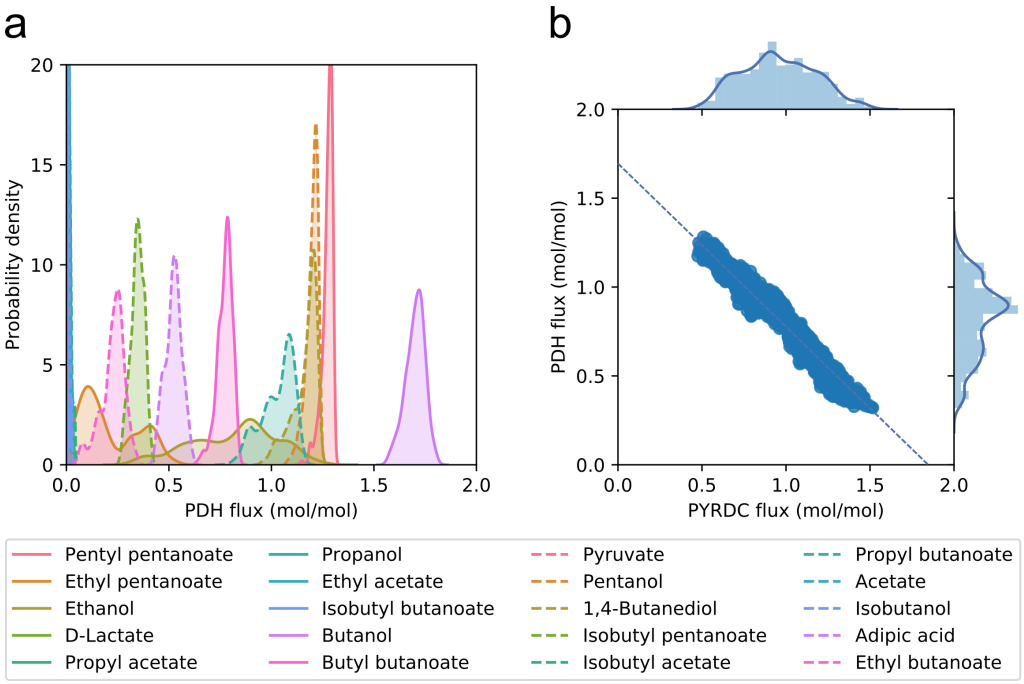


Figure 4.8: Sampled flux distributions of the ethanol biosynthesis pathways. (a) Probability density function of sampled fluxes for pyruvate dehydrogenase (PDH) across all production networks. Note that ethanol and ethyl pentanoate have the widest operation range. (b) Sampled metabolic fluxes of pyruvate decarboxylase (PYRDC) and PDH in the ethanol production network.

experimental evidence of isolated metabolic engineering strategies towards target products and measured intracellular flux ranges. Furthermore, analysis of the metabolic fluxes in this universal design revealed clusters of reactions in central metabolism, named interface reactions, that become activated to interact with specific pathways, providing a mechanistic view into the modularity of metabolic pathways. In this study, the universal design developed was limited to a library of 20 molecules in *E. coli* because one primary aim was to compare the MILP solution with the MOEA solution previously presented. Future studies will use the developed methods to design and understand modularity for different organisms and a larger library of product synthesis modules.

Supporting information

File S1 Spreadsheet with modular cell designs for *E. coli* core model.

File S2 Computer programs used to generate the results of this study.

Definitions

Sets

$\mathcal{I}_k$ Metabolites in production network k.

$\mathcal{J}_k$ Reactions in production network k.

$\mathcal{K}$ Production networks that are derived from a combination of the parent metabolic network with the metabolic pathways associated with production modules. The parent metabolic network is the network of the host strain that is genetically manipulated to build a modular cell chassis.

$\mathcal{M}$ Metabolic states that correspond to the growth phase, denoted μ, and the non-growth or stationary phase, denoted $\bar{\mu}$.

$\mathcal{C}$ Candidate deletion reaction set. The removal of these reactions are applied to all production networks, $\mathcal{C} \subseteq \mathcal{J}^{\text{parent}} \subseteq \mathcal{J}_k,\ \forall k \in \mathcal{K}$.

$\mathcal{N}_k$ Non-targeted deletion reaction set in production network k. This set arises from the use of fixed module reactions z_{jk} in certain production networks.

Binary variables

y_j Reaction deletion indicator that takes a value of 0 if reaction j is deleted in the chassis and 1 otherwise.

z_{jk} Module reaction indicator that takes a value of 1 if reaction j is added back as module reaction in production network k and 0 otherwise.

d_{jk} Reaction activity indicator that takes a value of 0 if reaction j in production network k might not carry a flux and 0 otherwise, thus $d_{jk} = y_j \vee z_{jk}$. This variable is declared

as a continuous and linear constraints enforce the OR relation and thus makes the variable binary.

w_k Production network feasibility indicator that takes a value of 0 if reaction deletions are ignored and the objective value is set to 0 for production network k, and a value of 1 otherwise.

e_{jk} Reaction activity indicator adjusted to w_k that takes the value of d_{jk} if $w_k = 1$ and a value of 1 if $w_k = 0$, thus $e_{jk} = (d_{jk} \wedge w_k) \vee \neg w_k$.

r_{jk} Linearization variable, $r_{jk} = d_{jk} \vee w_k$.

Continuous variables

v_{jkm} Flux (mmol/gCDW/hr) of reaction j from network k at metabolic state m.

v_{Pkm} Flux (mmol/gCDW/hr) of product synthesis reaction from network k at metabolic state m.

v_{Xkm} Flux (mmol/gCDW/hr) of biomass synthesis reaction from network k at metabolic state m.

f_k General objective function for production network k that can be represented by f_k^{wGCP}, f_k^{lsGCP}, or f_k^{NGP}.

f'_k Objective function adjusted by w_k such that $f'_k = f_k$ if $w_k = 1$ and $f'_k = 0$ otherwise.

δ_k^+ Amount required by the objective value f'_k to attain the target goal g_k, i.e.. $\delta_k^+ = g_k - f_k$ if $f'_k < g_k$.

δ_k^- Amount that the objective value f'_k surpasses the target goal g_k, i.e., $\delta_k^- = f'_k - g_k$ if $f'_k > g_k$.

λ_{ikm} Dual variable associated with mass balance constraint of metabolite i from production network k at growth state m.

μ_{jkm}^l Dual variable associated with the lower bound of reaction j from production network k at growth state m.

μ_{jkm}^u Dual variable associated with the upper bound of reaction j from production network k at growth state m.

p^l_{jkm} Linearization variable, $p^l_{jkm} = e_{jk}\mu^l_{jkm}$.

p^u_{jkm} Linearization variable, $p^u_{jkm} = e_{jk}\mu^u_{jkm}$.

Parameters

S_{ijk} Stoichiometric coefficient of metabolite i in reaction j of production network k.

l_{jkm} Lower bound for reaction j of production network k at metabolic state m.

u_{jkm} Upper bound for reaction j of production network k at metabolic state m.

γ Minimum biomass synthesis rate required for growth states. Note that in this study a conservative value of 20% of the maximum predicted growth rate of the wild-type strain was used to generate all results.

α Maximum number of deleted reactions in the modular cell chassis.

β_k Maximum number of module reactions in production network k.

ϵ Small scalar used for tilting the biomass objective function, leading to the minimum product rate available at the maximum growth rate. Note that in our study $\epsilon = 0.0001$ was used to generate all results.

b_μ, $b_{\bar{\mu}}$ Weights on the growth and non-growth objectives of f_k^{lsGCP}, respectively. Note that in our study $b_\mu = 1$ and $b_{\bar{\mu}} = 10$ were used to generate all results.

a_k Weighting factor applied to the objective function for production network k in the blended formulation. Note that in our study $a_k = 1,\ \forall k \in \mathcal{K}$ was used unless otherwise noted.

g_k Target value for objective f'_k in the goal programming formulation.

a_k^+ Weighting factor applied to δ_k^+ which emphasizes the importance of objective value f'_k to avoid falling below the target value g_k. Note that in our study $a_k^+ = 1,\ \forall k \in \mathcal{K}$ was used in all cases.

a_k^- Weighting factor applied to δ_k^- which emphasizes the importance of the objective f'_k to avoid surpassing the target value g_k. Note that in our study $a_k^- = 1,\ \forall k \in \mathcal{K}$ was chosen everywhere except to determine the universal modular cell design, where $a_k^- = 0,\ \forall k \in \mathcal{K}$ was used.

M^w Determines the minimum value of f_k that allows w_k to not be 0. A value of 10, corresponding to $f_k \geq 0.01$ for $w_k \neq 0$, was used in all cases.

M^{obj} Upper bound for each objective value. Note that in our study a value of 20 was set for all cases.

M Upper bound for dual variables. Note that in our study a value of 100 was set for all cases.

Chapter 5

Development of an updated genome-scale metabolic model of *Clostridium thermocellum* and its application for integration of multi-omics datasets and modular cell design

This chapter is based on the publication *Development of an updated genome-scale metabolic model of Clostridium thermocellum and its application for integration of multi-omics datasets. Garcia, S., Thompson, R.A., Giannone, R. J., Dash, S., Maranas , C. D., and Trinh, C. T. In review, 2020.* As first author I lead the development, implementation, and writing of this study. Supplementary Material 1 is provided in Appendix C, while Supplementary Files 2, 3, and 4 are provided as attachments.

Abstract

Solving environmental and social challenges such as climate change requires a shift from our current non-renewable manufacturing model to a sustainable bioeconomy. To lower carbon emissions in the production of fuels and chemicals, plant biomass feedstocks can replace petroleum using microorganisms as catalysts. The anaerobic thermophile *Clostridium thermocellum* is a promising bacterium for bioconversion due to its capability to efficiently degrade untreated lignocellulosic biomass. However, the complex metabolism of *C. thermocellum* is not fully understood, hindering metabolic engineering to achieve high titers, rates, and yields of targeted molecules. In this study, we developed an updated genome-scale metabolic model of *C. thermocellum* that accounts for recent metabolic findings, has improved prediction accuracy, and is standard-conformant to ensure easy reproducibility. We illustrated two applications of the developed model. We first formulated a multi-omics integration protocol and used it to understand redox metabolism and potential bottlenecks in biofuel (e.g., ethanol) production in *C. thermocellum*. Second, we used the metabolic model to design modular cells for efficient production of alcohols and esters with broad applications as flavors, fragrances, solvents, and fuels. The proposed designs not only feature intuitive push-and-pull metabolic engineering strategies, but also novel manipulations around important central metabolic branch-points. We anticipate the developed genome-scale metabolic model will provide a useful tool for system analysis of *C. thermocellum* metabolism to fundamentally understand its physiology and guide metabolic engineering strategies to rapidly generate modular production strains for effective biosynthesis of biofuels and biochemicals from lignocellulosic biomass.

5.1 Introduction

Global oil reserves will be soon depleted,[239] and climate change could become a major driver of civil conflict.[111] These challenges to security and the environment need to be addressed by replacing our current non-renewable production of energy and materials for a renewable and carbon neutral approach.[217] The gram-positive, thermophilic, cellulolytic, strict anaerobe

C. thermocellum is capable of efficient degradation of lignocellulosic biomass to produce biofuels and biomaterial precursors, making this organism an ideal candidate for consolidated bioprocessing (CBP), where saccharification and fermentation take place in a single step.[202] However, its complex and poorly understood metabolism remains the main roadblock to achieve industrially competitive titers, rates, and yields of biofuels such as ethanol[263] and isobutanol.[162]

For the past decade, significant efforts have been dedicated to characterize and manipulate *C. thermocellum*'s central metabolism, due to increasing interest in developing this organism as a CBP manufacturing platform for biofuels production.[4] *C. thermocellum* possesses atypical central metabolism, characterized by the important roles of pyrophosphate and ferredoxin,[313] which makes redirection of both carbon and electron flows for biofuel production challenging to achieve. Specifically, *C. thermocellum* has various reactions to regulate intracellular concentration levels of NADH, NADPH, and reduced ferredoxin. These cofactors are used as electron donors with high specificity throughout metabolism. To maintain redox balance, *C. thermocellum* also possesses several hydrogenases to oxidize these reduced cofactors to molecular hydrogen that is secreted by the cell. Removal of these hydrogenases through deletion of *ech* (encoding the ferredoxin-dependent hydrogenase, ECH) and *hydG* (associated with the bifurcating hydrogenase, BIF, and bidirectional hydrogenase, H2ASE_syn) was successfully applied to increase ethanol yield by electron rerouting.[19] Thompson *et al.*[261] characterized the $\Delta hydG\Delta ech$ strain in depth by flux analysis of its core metabolism, concluding that the major driver for ethanol production was redox rather than carbon balancing. In particular, the conversion of reduced ferredoxin to NAD(P)H is likely the most rate limiting step. In a subsequent study, Lo *et al.*[164] over-expressed *rnf* (encoding the ferredoxin-NAD oxidoreductase, RNF) in the $\Delta hydG\Delta ech$ strain that is expected to enhance NADH supply, but did not achieve improved ethanol yield.

In an attempt to redirect carbon and electron flows for enhanced ethanol production, Deng *et al.*[64] manipulated the pyruvate node and malate shunt of *C. thermocellum*. By converting phosphoenolpyruvate (*pep*) to oxaloacetate (*oaa*) and then to pyruvate (*pyr*), this shunt can interchange one mole of NADPH with one mol of NADH generated from glycolysis. Interestingly, the authors noted that replacement of the malate shunt by alternative pathways

not linked to NADPH increased ethanol production and carbon recovery but reduced amino acid formation, confirming the role of the malate shunt as an NADPH source in *C. thermocellum*.

Sulfur metabolism also plays a key role in redox metabolism of *C. thermocellum* and has been investigated for its role in ethanol production. Sulfate, a component of *C. thermocellum* media, serves as an electron acceptor, which is capable of oxidizing sulfate to sulfite and then sulfide. Thompson *et al.*[261] demonstrated that the strain $\Delta hydG\Delta ech\Delta pfl$, which cannot grow in conventional medium due to its inability to secrete hydrogen or formate, recovered growth by sulfate supplementation to the culture medium. More recently, Biswas *et al.*[18] reported an increase in final sulfide concentration and over-expression of the associated sulfate uptake and reduction pathway in the $\Delta hydG$ strain, but did not observe a significant difference in final sulfide concentration in $\Delta hydG\Delta ech$. Remarkably, neither of the strains consumed cysteine from the medium, unlike the wild-type. Sulfide can be converted to cysteine by CYSS (Cysteine synthase) or homocysteine and then methionine by SHSL2 and METS (succinyl-homoserine succinate-lyase and methionine synthase), but the connection between the cessation of cysteine uptake and sulfate metabolism remains unclear.

Overall, the complex interactions of *C. thermocellum* metabolic pathways remain challenging to understand and engineer with conventional methods, and hence require a quantitative systems biology approach to decipher. To this end, several genome-scale metabolic models (GSMs) of *C. thermocellum* have been developed. The first GSM, named iSR432, was constructed for the strain ATCC27405 and applied to identify gene deletion strategies for high ethanol yield.[223] This model was then further curated into iCth446.[55] More recently, Thompson *et al.* developed the iAT601 genome-scale model[260] for the strain DSM1313, which is genetically tractable.[9] The iAT601 model was used to identify genetic manipulations for high ethanol, isobutanol, and hydrogen production,[260] and to understand growth cessation prior to substrate depletion observed under high-substrate loading fermentations that simulate industrial conditions.[262] In addition to these core and genome-scale steady-state metabolic models, a kinetic model of central metabolism, k-ctherm118, was recently developed and used to elucidate the mechanisms of nitrogen limitation and ethanol stress.[55] Due to the biotechnological relevance of the *Clostridium* genus, GSMs

have also been developed for other species,[57] including *C. acetubutylicum*,[149,236,226,181,287,56,303] *C. beijerinckii*,[186] *C. butyricum*,[238] *C. cellulolyticum*,[226] and *C. ljungdahlii*.[190]

In this study, we developed an updated genome-scale metabolic model of *C. thermocellum*, named iCBI655, with more comprehensive and precise metabolic coverage, enhanced prediction accuracy, and extensive documentation. This model is a human-curated database that coherently represents all the available genetic, genomic, and metabolic knowledge of *C. thermocellum* from both experimental literature and bioinformatic predictions. Furthermore, the model can be applied not only to enable metabolic flux simulation but also to provide a framework to contextualize disparate datasets at the system level. As a demonstration for the model application, we first developed a quantitative multi-omics integration protocol and used it to to fundamentally study redox metabolism and potential redox bottlenecks critical for production of biofuels (e.g., ethanol) in *C. thermocellum*. Furthermore, we used the model, in combination with the previously developed ModCell tool,[86,87] to design modular (chassis) cells[85] for alcohol and ester production.

5.2 Results

5.2.1 Development of an upgraded *C. thermocellum* genome-scale model named iCBI655

The iCBI655 model was developed using the published iAT601 model[260] as a starting point. The model improvements include updated metabolic pathways, new annotation, and new extensive documentation. A detailed account of these changes can be found in the Supplementary Material 3. Here, we highlight the most relevant modifications.

Modeling updates

To facilitate model usage and reduce human error, the identifiers of reactions and metabolites were converted from KEGG into BiGG human-readable form.[136] Additionally, reaction and metabolite identifiers were linked to the modelSEED database[107] that enables analysis through the KBase web interface.[10] The gene identifiers and functional descriptions were

updated to the most current annotation (NCBI Reference Sequence: NC_017304.1). Metabolite formulas and charges from the modelSEED database[107] were included in the model and reactions were systematically corrected for charge and mass balance by the addition of protons and water.

Metabolic updates

The automated construction process used in the previous model introduced several inconsistencies that were corrected in the current model. We removed reactions that were blocked and non-gene-associated, apparently introduced during automated gap-filling. Two notable examples are (i) the blocked selenate pathway which lacks experimental evidence (e.g., selenoproteins have not been found in *C. thermocellum*), and (ii) blocked reactions involving molecular oxygen (e.g., oxidation of Fe^{2+} to Fe^{3+}) that are not possible in strict anaerobes like *C. thermocellum*. Furthermore, tRNA cycling reactions were unblocked by including tRNA into the biomass reaction.[221] Metabolite isomers were examined and consolidated under the same metabolite identifier when possible, leading to the removal of duplicated reactions and the elimination of gaps. Transport and exchange reactions were updated to reflect the export of amino acids and uptake of pyruvate as observed during fermentation experiments.[110]

In terms of specific reactions, oxaloaceate decarboxylase was eliminated from the model in accordance with recent findings.[201] The stoichiometries of pentose-phospate reactions, including sedoheptulose 1,7-bisphosphate D-glyceraldehyde-3-phosphate-lyase (FBA3) and sedoheptulose 1,7-bisphosphate ppi-dependent phosphofructokinase (PFK3_ppi), were corrected (according to experimental evidence[225]) from the previous model by ensuring mass balance and avoiding lumping multiple steps into one reaction. Transaldolase (TALA) was removed from the model due to lack of annotation for this gene in *C. thermocellum* and to also ensure PFK essentiality as observed experimentally (personal communication from Prof. Lynd's lab at Dartmouth College).

Several modifications were also performed in key bioenergetic reactions. The reactions catalyzed by membrane-bound enzymes, including inorganic diphosphatase (PPA)[313] and membrane-bound ferredoxin-dependent hydrogenase (ECH)[32], were corrected to capture

proton translocation. Furthermore, hydrogenase reactions were updated to ensure ferredoxin association for all cases and remove those reactions that do not involve ferredoxin and only use NAD(P)H as a cofactor based on our recent understanding of *C. thermocellum* metabolism.[18] Gene-protein-reaction associations were updated to represent experimental knowledge. For instance, the hydrogenases BIF (CLO11313_RS09060-09070) and H2ASE (CLO1313_RS12830, CLO1313_RS02840) require the maturase Hyd (CLO1313_RS07925, CLO1313_RS11095, CLO1313_RS12830) to be functional, and the maturase itself requires all of its subunits to operate, which enables accurate representations of *hydG* deletion genotypes.[19]

Two hypothetical reaction modifications were introduced to ensure consistency with reported phenotypes. First, to enable growth without the need for succinate secretion, as observed in experimental data (Supplementary Material 3), the reaction homoserine-O-trans-acetylase (HSERTA) was added to enable methionine biosynthesis (essential for growth) without succinate formation. Although this reaction is not currently known to be associated with any gene in *C. thermocellum*, it is known to be present in other *Clostridium* GSMs.[190] Next, the reaction deoxyribose-phosphate aldolase (DRPA) was removed to ensure correct lethality prediction of the $\Delta hydG\Delta ech\Delta pfl$ mutant strain as well as the correct prediction of growth recovery in this mutant by addition of external electron sinks such as sulfate or ketoisovalerate (Table 5.1). The correct prediction of $\Delta hydG\Delta ech\Delta pfl$-associated phenotypes is critical to successfully use the model for computational strain design.[166,194,176,289,86,85,83]

5.2.2 Comparison of iCBI655 against other genome-scale models

We compared iCBI655 with the previous GSMs of *C. thermocellum* and the highly-curated GSM iML1515 of the extensively studied bacterium *Escherichia coli* (Table 5.2). The increased number of genes in iCBI655 with respect to iAT601 cover a variety of functions, including hydrogenase chaperons, cellulosome and cellulase, ATP synthase, and transporters. Remarkably, iCBI655 has a smaller percentage of blocked reactions than iAT601, indicating higher biochemical consistency. The number of metabolites in iCBI655 was smaller than those in iAT601 due mainly to the removal of metabolites that did not appear in any reaction,

Table 5.1: Comparison of mutant growth rates predicted by iAT601 and iCBI655. Experimental values are taken form Thompson *et al.*[261]; for some mutants whose growth recovery, not growth rate, was reported, they are presented with "+".

Gene deletions	Medium	Fraction of W.T. growth rate (%)		
		iAT601	*iCBI655*	Experiment
hydg	MTC	100	100	73
hydg-ech	MTC	85	85	67
hydg-pta-ack	MTC	100	100	48
hydG-ech-pfl	MTC	58	0	0
hydG-ech-pfl	MTC + fumarate	377	726	0
hydG-ech-pfl	MTC + sulfate	58	65	+
hydG-ech-pfl	MTC + ketoisovalerate	97	101	+

duplicated metabolites (e.g., certain isomers), and blocked pathways added automatically during gap-filling but that lack any gene association. *C. thermocellum* DSM1313 has 2911 protein coding genes, although not all are metabolic, of which iCBI655 covers 22%, while *E. coli* MG1655 has 4240 genes of which iML1515 covers 35%. Overall, iCBI655 increases the coverage of the metabolic functionality of *C. thermocellum* but remains far from the highly studied *E. coli*.

Table 5.2: Comparison of all genome-scale models of *C. thermocellum* and the latest *E. coli* genome-scale model.

	iSR432	iCth446	iAT601	iCBI665	iML1515
Strain	ATCC27405	ATCC27405	DSM1313	DSM1313	MG1655
Genes	432	446	601	665	1515
Metabolites	583	599	903	795	1877
Reactions	632	660	872	854	2712
Blocked reactions	39.2%	32.1%	40.8%	35.1%	9.8%
Reference	[223]	[55]	[260]	This study	[188]

5.2.3 Training of model parameters under diverse conditions

Growth and non-growth associated maintenance (GAM and NGAM) are parameters that capture the consumption of ATP towards cell division and homeostasis, respectively. These are known to be condition-specific; however, genome-scale models do not include a mechanistic description that allows to determine these ATP consumption rates as part of the simulation. Instead, GAM is incorporated into the biomass pseudo-reaction and NGAM has its own pseudo-reaction that hydrolyzes ATP at a rate tuned by constraint parameters.

To increase model prediction accuracy for various conditions, we trained GAM and NGAM parameters of iCBI655 using an extensive dataset of 28 extracellular fluxes (Supplementary Material 3) measured during the growth phase under different reactor configurations, carbon sources, and gene deletion mutants. This approach is based on the method used to train the iML1655 *E. coli* model.[188] Remarkably, we observed highly linear trends under three different conditions, including chemostat reactor with cellobiose as a carbon source, chemostat reactor with cellulose as a carbon source, and batch reactor with either cellobiose or cellulose as carbon sources (Figure 5.1 a). This model training has led to improved growth rate prediction of iCBI655 as compared to iAT601 that has previously been trained with only a smaller dataset (Figure 5.1 b). Specifically, the iAT601 training dataset was limited to batch conditions; hence, the inaccurate predictions of iAT601 were observed for chemostat conditions (Figure 5.1 c).

5.2.4 Assessment of model quality and standard compliance with Memote

The field of metabolic network modeling suffers from a lack of standard enforcement and quality control metrics that limit model reproducibility and applicability. To address this issue, Lieven *et al.* recently developed the Memote framework that systematically tests for standards and best practices in GSMs.[159] We applied Memote to both the iCBI655 and *E. coli* iML1515 models for comparison (Figure 5.1 d). This analysis produces five independent scores that assess model quality. The *consistency score* measures basic biochemical requirements, such as mass and charge balance of metabolic reactions, and it

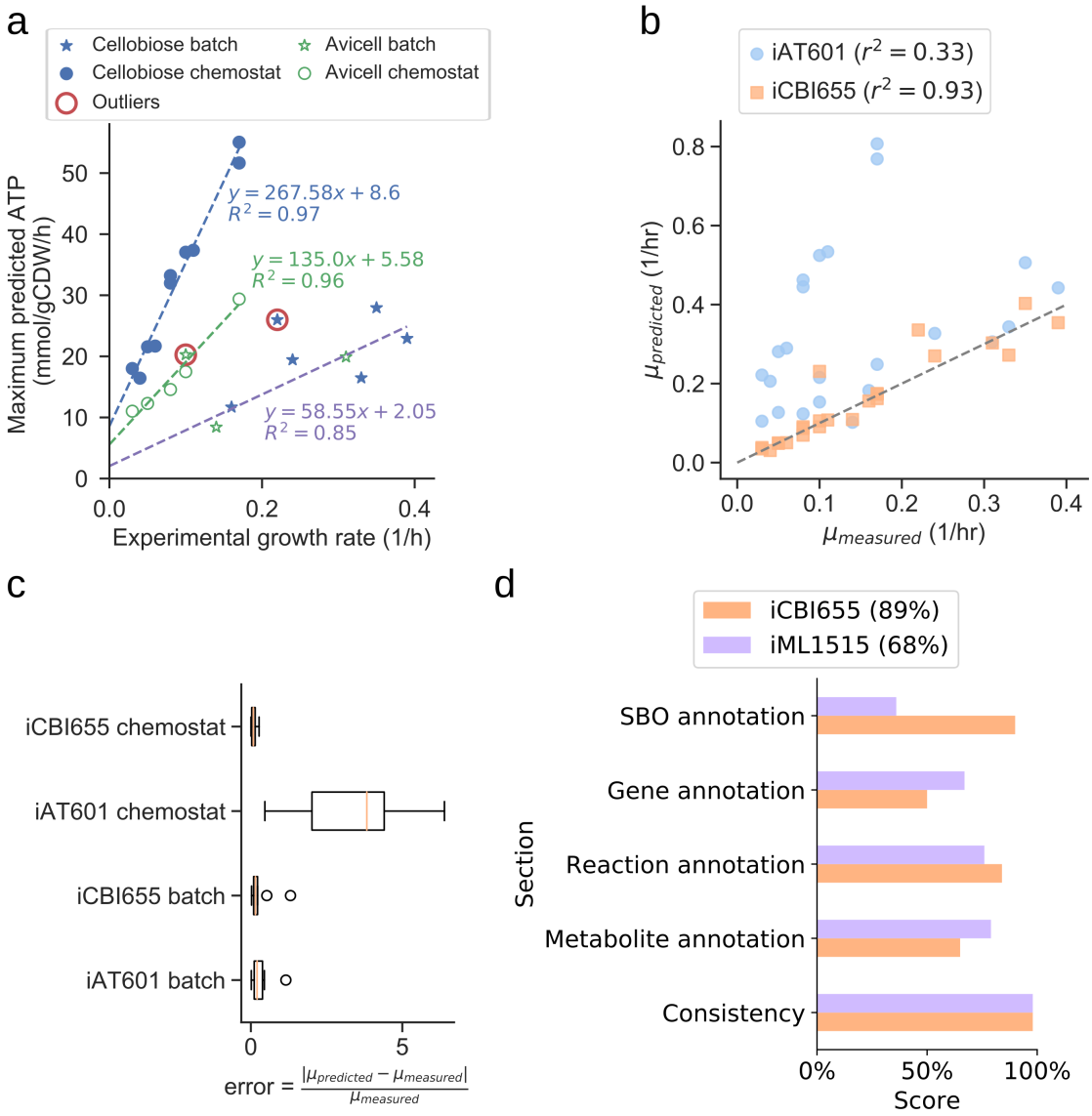


Figure 5.1: (**a**) Training of GAM and NGAM parameters. Discrete points correspond to experimental data. The slope of the linear regression function corresponds to the GAM, while the intercept corresponds to the NGAM. The data points circled as outliers were not included in any of the linear regression calculations. (**b**) Comparison of growth prediction error between iCBI655 and iAT601. The measured substrate uptake and product secretion fluxes form each dataset Supplementary Material 3 were used to constraint the model and maximum growth rate was calculated. r^2 corresponds to the Pearson correlation coefficient. (**c**) Error in growth predictions under batch and chemostat conditions. Predicted and measured growth rates correspond to the values included in **b**. (**d**) Scores provided by the quality control tool Memote[159] for iCBI655 and iML1515 (overall score indicated in the legend).

was near 100% for both models. Additionally, the different annotation scores quantify how many elements in the model contain relevant metadata. More specifically, the *systems biology ontology (SBO) annotation* indicates if an object in the model refers to a metabolite, reaction, or gene, while the respective *annotation scores* of these elements correspond to properties (e.g., name, chemical formula, etc.) and identifiers linking them to relevant databases (e.g., KEGG[128] or modelSEED[107]). The *overall score* is computed as a weighted average of all the individual scores with additional emphasis on the *consistency score*. In summary, the high scores obtained by iCBI655 indicate the quality of the model and ensure its applicability for future studies.

5.2.5 Model-guided analysis of proteomics and flux datasets sheds light on redox metabolism critical for biofuel production in *C. thermocellum*

For the first application of the genome-scale metabolic model, we aimed to understand the complex redox metabolism and potential redox bottlenecks critical for enhanced biofuel production in *C. thermocellum*. We used the model as a scaffold to analyze proteomics and metabolic flux data collected for the *C. thermocellum* wild-type and $\Delta hydG\Delta ech$ strains. The $\Delta hydG\Delta ech$ mutant was engineered to redirect electron flow from hydrogen to ethanol by removal of primary hydrogenases.[19,261] Previous studies of $\Delta hydG\Delta ech$ based on analysis of secretion fluxes[261] or omics[18] suggested the presence of redox bottlenecks in this mutant but did not identify which specific pathways and cofactors (i.e., NADH versus NADPH) are responsible. We aim to solve this problem through integrated and quantitative analysis of omics and fluxes at the genome scale.

Development of fold change-based omics integration protocol

To perform the analysis, we formulated an omics integration protocol anchored in the quantification of fold change (FC) between case and control samples (Figure 5.2a). In this approach, we first compared FCs between simulated intracellular fluxes and measured omics

data. Next, we identified *consistent reactions* with FCs of the same sign and different from zero in both measured proteomics and simulated fluxes for further analysis (Section 5.4.6).

To start the FC-based omics integration, we obtained measured FCs by mapping the measured proteomics data to 510 out of the 856 reactions in the model through the gene-protein-reaction (GPR) associations (Figure 5.2b). Then, we identified 70 consistent reactions by comparing measured FCs with two types of simulated FCs: i) parsimonious flux balance analysis (pFBA) that determines the most efficient flux distribution (assuming all enzymes are roughly as efficient) and ii) flux variability analysis (FVA) that identifies the feasible flux range of each reaction.

The Pearson correlation coefficients between simulated and measured FCs for the consistent reactions were 0.26 and 0.09 for pFBA and FVA, respectively (Figure 5.2c). In general, the FVA reaction flux ranges remained mostly unchanged, suggesting that pFBA is a better representation of actual metabolic fluxes as previously observed.[173] The top consistent reactions with the highest proteomics FCs (Supplementary Material 1 - Table S1) belong primarily to the central metabolism of *C. thermocellum* (Figure 5.3). Interestingly, discrepancies in magnitude between flux and protein FCs for consistent reactions could be used to identify bottlenecks. For example, for a given enzyme a small increase in flux combined with a large increase in translation is an indicator of low catalytic efficiency. Similar comparisons between simulated flux and omics has previously been used to identify regulatory mechanisms.[25] Overall, the identification and analysis of consistent reactions is an effective approach to gain certainty on the activity changes of metabolic pathways between conditions.

FC-based omics integration reveals redirection of electron flow for NADPH supply in Δ*hydG*Δ*ech* strain

Our analysis reveals consistent indications of increased NADPH biosynthesis in the Δ*hydG*Δ*ech* mutant with respect to the wild-type across three major metabolic areas: i) an increased protein level of FRNDPR2r (also known as NFN) that converts one mol of reduced ferredoxin (*fdxr_42*) and one mole of NADH into two moles of NADPH (Figure 5.3a), ii) an increased protein level of all three malate shunt enzymes and a decreased protein

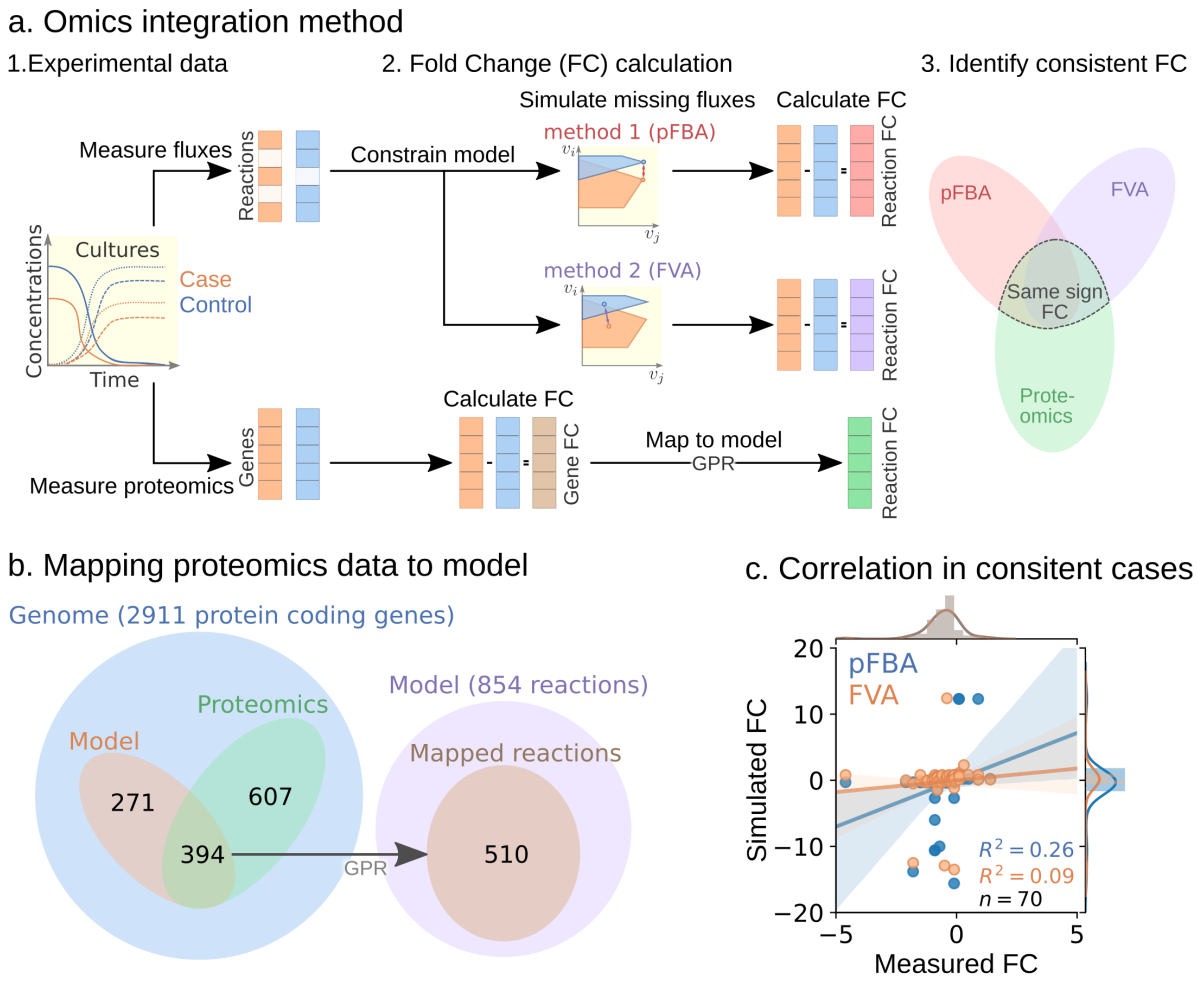


Figure 5.2: (**a.**) Multi-scale data integration and analysis procedure. (**b.**) Mapping of proteomics data for the $\Delta hydG$-Δech case study to model reactions. (**c.**) Correlation between measured fold changes and simulated fold changes (pFBA in blue and FVA in orange) for all 70 consistent reactions of the $\Delta hydG$-Δech case study.

level of the alternative route PPDK (Figure 5.3b); and iii) a decreased protein level of sulfur transporter and of HSOR that oxidizes sulfite into sulfide consuming NADPH (Figure 5.3c). These observations are consistent with the failure of *rnf* over-expression to enhance ethanol production,[164] since RNF produces NADH but the key cofactor bottleneck seems to be NADPH. Furthermore, a direct look at the proteomics data revealed that RNF subunits (Clo1313_0061-Clo1313_0066) had a statistically significant decrease in protein levels of the mutant (Supplementary Material 3). The preference of $\Delta hydG\Delta ech$ towards NADPH could be due to the cofactor specificity of the remaining redox balancing pathways (e.g., isobutanol), thermodynamics and protein cost constraints, or a combination of both. While the contribution of thermodynamic constraints is beyond the scope of this study, a recent analysis[58] of the ethanol production pathway in *C. thermocellum* highlighted the importance of engineering strategies to increase NADPH, for instance, introduction of NADPH-linked GAPDH that converts glyceraldehyde-3-phosphate to 3-phospho-D-glyceroyl phosphate in glycolysis or overexpression of NADPH-FNOR that transfers electrons from reduced ferredoxin to NADPH. The contribution of alternative redox balancing pathway towards the increased NADPH biosynthesis will be examined next.

Analysis of simulated fluxes reveals the role of NADPH in redox balancing

The analysis based on consistent reactions strongly indicates that NADPH production is important in the mutant to achieve redox balance. However, the pathways oxidizing NADPH remain unknown since not all reactions in the model could be mapped to proteomics measurements and carbon recovery was lower in the mutant strain.[261] To identify these pathways, we examined the simulated fluxes of all reactions (instead of only consistent reactions) that differed in value between wild-type and mutant, and limited this search to exchange reactions and reactions that involve NADPH (Supplementary Material 1 - Table S2). These simulated fluxes predicted an increase in the isobutanol pathway, including keto-acid reductoisomerase (KARA1) that consumes NADPH and isobutanol secretion (EX_ibutoh_e). The isobutanol pathway can consume NADPH through several enzymes[162] and has increased flux during overflow metabolism at high-substrate loading.[110,262] The model also predicted a decrease in valine secretion (EX_val__L_e), since the isobutanol

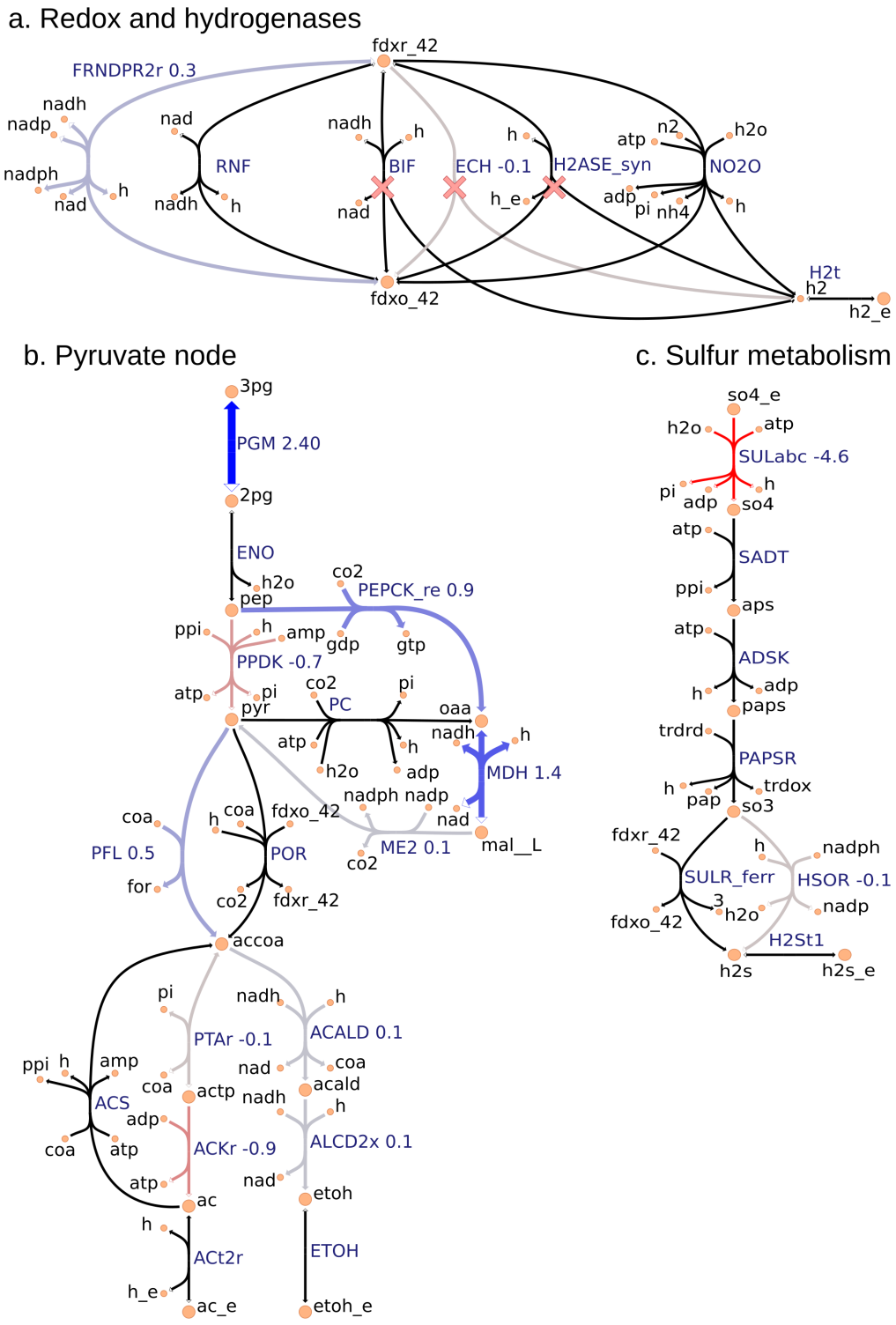


Figure 5.3: Metabolic map visualization using the iCBI Escher map. Values next to reaction labels correspond to proteomics fold change between $\Delta hydG$-Δech and wild-type strains only for the 70 consistent reactions identified by the omics analysis (Section 5.2.5). The labels of extracellular metabolites are appended with "_e". Reactions marked with a red cross are deleted in $\Delta hydG$-Δech.

pathway competes with the valine pathway after KARA1. Remarkably, this prediction is consistent with the lower valine secretion measured in $\Delta hydG\Delta ech$.[18] A certain amount of NADPH is likely oxidized by the mutated alcohol-dehydrogenase enzyme observed after short adaptation in $\Delta hydG$ that is compatible with both NADH and NADPH.[19] However, this feature is not captured by the model since in general gene knock-outs are simulated by blocking the associated reactions. Overall this analysis indicates that $\Delta hydG\Delta ech$ likely increases isobutanol secretion to alleviate redox imbalance.

Taken altogether, model-guided data analysis illustrates the power of the model as contextualization tool and provides new insights into the redox bottlenecks present in *C. thermocellum* that are critical in the production of reduced molecules. The integration of omics and fluxes led to the resolution of NADPH as the key cofactor in redox imbalance of $\Delta hydG\Delta ech$, and it identified specific pathways that undergo major changes in protein levels, providing interesting target reactions for further engineering. Generally, the developed FC-based omics integration protocol can be applied to different omics data types due to its simplicity. The method does not require one to formulate or assume a quantitative relationship between omics measurements and simulated fluxes. Furthermore, fold change in biomolecule concentrations implemented in the method is currently much easier to measure in a quantitatively reliable manner for many molecules than case-specific absolute concentrations.

5.2.6 Model-guided design of modular production strains for biofuel synthesis

Another common application of genome-scale models is strain design.[166,194,176,289,86,85,83] We used the iCBI655 model combined with the ModCell tool[87] to design *C. thermocellum* modular production strains for efficient biosynthesis of alcohols and esters. Briefly, with ModCell we aim to design a modular (chassis) cell that can be rapidly combined with different pathway modules in a plug-and-play fashion to obtain modular production strains exhibiting target phenotypes with minimal strain optimization cycles. In this study, the target phenotype for modular production strains is growth-coupled to product synthesis

($wGCP$), that corresponds to the minimum product synthesis rate at the maximum growth rate. The design variables to attain the target phenotypes involve genetic manipulations of two types: i) reaction deletions, constrained by the parameter α, that corresponds to gene knock-outs; and ii) module reactions, constrained by the parameter β, that corresponds to reactions deleted in the modular cell but added back to specific modules to enhance the compatibility of a modular cell. Once these two parameters are specified, the solution to the problem is a set of Pareto optimal designs named Pareto front. In a Pareto optimal design, the performance (i.e., objective value) of a given module can only be increased at the expense of lowering the performance of another module. To characterize the practicality of each design, we say a modular cell is compatible with certain modules if the design objective is above a specific threshold (e.g., 0.5 in this study). Hence, the *compatibility* of a design corresponds to the number of compatible modules.

To design *C. thermocellum* modular cells, we first evaluated a range of design parameters α and β with an increasing number of genetic manipulations (Figure 5.4 a). As expected, increasing the number of deletions leads to more compatible designs, at the expense of more complexity in the implementation. We selected an intermediate point of $\alpha = 6$, $\beta = 0$ for further analysis. This Pareto front is composed of 12 designs that can be clustered into two groups (Figure 5.4 b). The first group (e.g., designs 8, 3, and 9) are compatible with all products except butanol and its derived esters, whereas the second group (e.g., designs 2, 12, 1, and 10) have high objective values for butanol and its derived esters.

To understand the characteristics of each group, we can inspect the deletions of each design (Figure 5.4 c). Designs 8, 3, and 9 all have in common H2ASE_syn, GLUDy, PPDK, and FRNDPR2r deletion, while the last two deletions never appear in design 2, 12, 1, or 10. The majority of deletion targets are central metabolic reactions (Table 5.3). These include common targets such as deletion of hydrogenases that appear in the cluster of designs 10, 12, 4, 11, 2, and 7 with the $\Delta hydG\Delta ech$ genotype discussed earlier or removal of reactions that form fermentative byproducts such as ALCD2x and ACALD (ethanol), PFL (formate), LDH_L (lactate). Interestingly, ACKr or PTA (acetate) does not appear in this list, likely because acetate production can serve as a regulatory valve for redox metabolism, especially in a modular cell that must be compatible with products of diverse degrees of reduction.

More interestingly, we also found important branch-point deletion reactions[250] in central metabolism that have not yet been explored for strain design. Most prominently, these reactions include GLUDy, PEPCK_re, and PPDK, which appear in 50%, 33%, and 25% of the designs, respectively (Table 5.3). Both PEPCK_re and PPDK present two alternative routes that influence the ratio of NADPH to NADH, which is relevant to control metabolic fluxes though the specific dependencies of certain enzymes towards each redox cofactor. Since GLUDy consumes NADPH and is a key reaction in amino-acid metabolism, this enzyme and related ones (e.g., GLUSy) are interesting targets for practical implementation. We speculate the two product groups emerge likely because the butanol pathway strictly depends on NADH due to the reactions ACOAD1z (acyl-CoA dehydrogenase) and HACD1 (3-hydroxyacyl-CoA dehydrogenase), while the ethanol, propanol, and isobutanol pathways are more flexible in their use of NADH or NADPH. The designs 8, 3, and 9 perform poorly with butanol, and are also the only ones containing PPDK deletion. This deletion forces pep to pyruvate flux through the malate shunt that converts NADH to NADPH. Engineering of the cofactor specificities of the butanol pathway could be used to build one modular cell compatible with all products under consideration.

Two representative designs from the groups mentioned earlier are 3 and 12. Their feasible growth and production phenotypes reveal a tight coupling between product formation and growth rate (Figure 5.4 d). This phenotype enables pathway optimization through adaptive laboratory evolution, as previously done for ethanol,[263] overcoming one of the main challenges of *C. thermocellum* engineering that is optimization of enzyme expression levels. Hence, the proposed modular cells can also serve as platforms for pathway selection and optimization. In summary, this analysis demonstrates the potential of the model to identify non-intuitive metabolic engineering strategies that can be key to build effective modular platform strains for the production of biofuels and biochemicals in *C. thermocellum*.

5.3 Conclusions

In this study, we developed a genome-scale metabolic model of the biotechnologically relevant organism *C. thermocellum*. Model development followed standards and best practices to

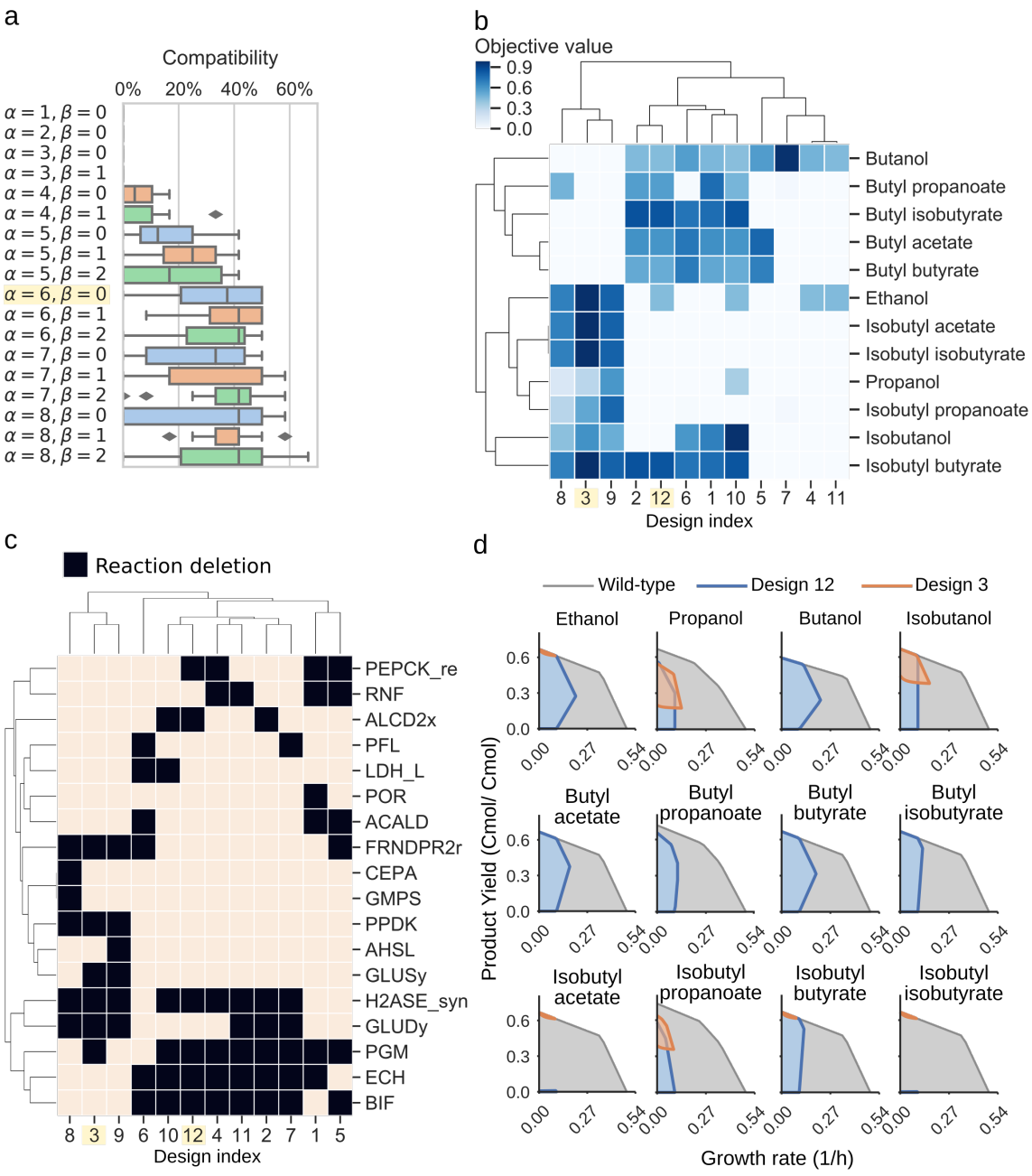


Figure 5.4: Proposed modular cell designs for a *C. thermocellum* platform and 12 alcohols and esters. (**a**) Parameter scan, and compatibility distribution of the designs in each resulting Pareto front. (**b**) Pareto front for parameters $\alpha = 6, \beta = 0$. (**c**) Pareto set for parameters $\alpha = 6, \beta = 0$. Reaction names and formulas are included in Supplementary Material 1 - Table S3. (**d**) Feasible phenotypic spaces according to reaction stoichiometry for selected designs.

Table 5.3: Reaction deletions sorted by appearance frequency (counts) in the designs of the Pareto front for $\alpha = 6$, $\beta = 0$.

ID	Name	Formula	Counts (%)
PGM	Phosphoglycerate mutase	2pg_c ↔ 3pg_c	75
H2ASE_syn	Bidirectional [NiFe] Hydrogenase (Fe-H2)	h2_c + nadp_c ↔ h_c + nadph_c	75
ECH	(FeFe)-hydrogenase, ferredoxin dependent, membrane-bound	2.0 fdxr_42_c + 3.0 h_c ↔ 2.0 fdxo_42_c + h2_c + h_e	66.7
BIF	Bifurcating Hydrogenase	2.0 fdxr_42_c + 3.0 h_c + nadh_c ↔ 2.0 fdxo_42_c + 2.0 h2_c + nad_c	66.7
GLUDy	Glutamate dehydrogenase (NADP)	glu__L_c + h2o_c + nadp_c ↔ akg_c + h_c + nadph_c + nh4_c	50
FRNDPR2r	Ferredoxin: nadp reductase (NFN)	2.0 fdxr_42_c + h_c + nadh_c + 2.0 nadp_c ↔ 2.0 fdxo_42_c + nad_c + 2.0 nadph_c	41.7
RNF	Ferredoxin:NAD oxidoreductase (membrane bound)	2.0 fdxr_42_c + 2.0 h_c + nad_c ↔ 2.0 fdxo_42_c + h_e + nadh_c	33.3
PEPCK_re	Phosphoenolpyruvate carboxykinase (GTP)	co2_c + gdp_c + pep_c → gtp_c + oaa_c	33.3
ALCD2x	Alcohol dehydrogenase (ethanol)	acald_c + h_c + nadh_c → etoh_c + nad_c	25
ACALD	Acetaldehyde dehydrogenase (acetylating)	accoa_c + h_c + nadh_c → acald_c + coa_c + nad_c	25
PPDK	Pyruvate, phosphate dikinase	amp_c + 2.0 h_c + pep_c + ppi_c → atp_c + pi_c + pyr_c	25
GLUSy	Glutamate synthase (NADPH)	akg_c + gln__L_c + h_c + nadph_c → 2.0 glu__L_c + nadp_c	16.7
PFL	Pyruvate formate lyase	coa_c + pyr_c → accoa_c + for_c	16.7
LDH_L	L-lactate dehydrogenase	h_c + nadh_c + pyr_c → lac__L_c + nad_c	16.7
POR	Pyruvate-ferredoxin oxidoreductase	coa_c + 2.0 fdxo_42_c + pyr_c → accoa_c + co2_c + 2.0 fdxr_42_c + h_c	8.3
CEPA	Cellobiose phosphorylase	cellb_c + pi_c → g1p_c + glc__D_c	8.3
GMPS	GMP synthase	atp_c + nh4_c + xmp_c → amp_c + gmp_c + 3.0 h_c + ppi_c	8.3
AHSL	O-Acetyl-L-homoserine succinate-lyase	achms_c + cys__L_c ↔ ac_c + cyst_L_c + h_c	8.3

ensure reproducibility and accessibility. We demonstrated the enhanced predictions of the model for diverse fermentation conditions and gene lethality. Genome-scale models have a broad range of applications in systems biology, including metabolic engineering, physiological discovery, phenotype interpretation, and studies of evolutionary processes.[73,204] To illustrate the model applications, we chose to tackle the challenge of disparate data integration and interpretation at the systems level. We developed a fold-change-based omics integration method for this purpose, and used it to identify routes in central metabolism that were selected to increase NADPH generation in the $\Delta hydG\Delta ech$ strain. This analysis revealed the importance of NADPH cofactor over its alternatives and provided new engineering targets for enhanced biosynthesis of reduced products in *C. thermocellum*. We also illustrated the use of the model to design *C. thermocellum* modular cells, using the ModCell tool.[85] The proposed designs cover C2 through C4 alcohols and their derived esters, which are key target molecules for renewable production with *C. thermocellum*.[211] The proposed designs feature a combination of previously-explored and novel strategies to couple target metabolite production with cellular growth. Like the well-developed genome-scale models[188,168] of the important organisms *Escherichia coli* and *Saccharomyces cerevisiae* broadly used for strain engineering both in academia[22] and industry[302], we anticipate the iCBI655 genome-scale model will also provide a versatile tool for systems metabolic engineering of *C. thermocellum*.

5.4 Methods

5.4.1 Model curation

The genome scale model iCBI655 was constructed from iAT601[260] by following the standard GSM development protocol.[259] Reaction and metabolite identifiers were mapped from KEGG to BiGG using the BiGG API.[136] Metabolite charges were obtained from modelSEED when available, and otherwise calculated using the Chemaxon pKa plugin[252] for a pH of 7.2.[259] The biomass objective function was consolidated into one pseudo-reaction avoiding the use of intermediate pseudo-metabolites present in iAT601. Reactions were assigned with a confidence level based on a standard genome-scale model annotations.[259]

5.4.2 Metabolic flux simulations

Constraint-based metabolic network modeling[204] is based on the feasible flux space, Ω_k, defined by network stoichiometry and flux bounds that represent thermodynamic constraints and measured values:

$$\Omega_k := \{v_{jk} \in \mathbb{R} : \sum_{j \in \mathcal{J}} S_{ij} v_{jk} = 0 \ \ \forall i \in \mathcal{I} \tag{5.1}$$

$$l_{jk} \leq v_{jk} \leq u_{jk} \ \forall j \in \mathcal{J}\} \tag{5.2}$$

Here $\mathcal{I}$ and $\mathcal{J}$ are the sets of metabolites and reactions in the model, respectively, and v_j is the metabolic flux (mmol/gCDW/h) through reaction j. Constraint (5.1) enforces mass balance for all metabolites in the network, where S_{ij} represents the stoichiometric coefficient of metabolite i in reaction j. Constraint (5.2) enforces lower and upper bounds l_j and u_j, respectively, for each reaction j in the network.

In different simulation conditions, k, S_{ij} remains fixed given the structure of the network for all $i, j \in \mathcal{I}, \mathcal{J}$. However, certain bounds u_{jk} and l_{jk} are modified to represent specific metabolic constraints. For example, to apply measured reaction fluxes such as in the case of GAM and NGAM calculation or the omics integration protocol (Section 5.4.6), l_{jk} and u_{jk} are specified using the experimentally measured average (μ_{jk}) and standard deviation (σ_{jk}), which for normally distributed samples with 3 replicates produces an interval with a confidence level above 90% (5.3-5.4). Similarly, to represent a certain gene deletion mutant k, the bounds are set to be $u_{jk} = l_{jk} = 0$ for the associated reaction j.

$$l_{jk} = \mu_{jk} - \sigma_{jk} \ \forall j \in \text{Measured}_k \tag{5.3}$$

$$u_{jk} = \mu_{jk} + \sigma_{jk} \ \forall j \in \text{Measured}_k \tag{5.4}$$

The feasible flux space Ω_k can be explored in different ways;[275,204] for instance, an optimization objective is often defined to identify specific flux distributions $v_{jk}^{\text{sim}} \forall j \in \mathcal{J}$:

$$v_{jk}^{\text{sim}} \in \arg\max \left\{ \sum_{j \in \mathcal{J}} c_j v_{jk} : v_{jk} \in \Omega_k \right\} \forall j \in \mathcal{J} \tag{5.5}$$

Here c_j is the coefficient of reaction j in the linear objective function, which is changed according to the simulation context. For example, to train GAM and NGAM (Figure 5.1a) the objective was set to maximize flux through the ATP hydrolysis reaction, i.e., $c_j = 1$ for j corresponding to ATP hydrolysis reaction, and 0 otherwise. To evaluate growth prediction accuracy (Figure 5.1b,c), the objective was set to maximize growth, i.e., $c_j = 1$ for j corresponding to growth pseudo-reaction and 0 otherwise.

5.4.3 Simulation of different growth environments

The model is configured to generally represent different medium and reactor conditions by modifying three features. The first feature involves model boundaries specifying which metabolites may enter the intracellular environment (i.e., present in the growth medium) or may exit the extracellular environment (i.e., secreted by *C. thermocellum*). This feature can be adjusted through u_j and l_j for exchange reactions. In our simulations, only essential metabolites required for *in silico* growth may be consumed and only commonly observed metabolites may be produced, unless otherwise noted. The second feature involves biomass objective function. iCBI655 contains 3 possible biomass reactions: i) BIOMASS_CELLOBIOSE used for growth in cellobiose with cellulosome constituting 2% of cell dry weight (CDW)[309], ii) BIOMASS_CELLULOSE used for growth on cellulose with cellulosome constituting 20% of CDW[309], and iii) BIOMASS_NO_CELLULOSOME, a biomass function that does not consider cellulosome production and only used as a control. The combination of cellulosome and protein fractions accounts for 52.85% of the CDW in all cases.[223,261] Cellobiose conditions were used in all simulations unless otherwise noted. The third feature involves GAM/NGAM. Three sets of these parameters are considered

including batch, chemostat-cellulose, and chemostat-cellobiose, based on fitting the model to experimental data. Batch conditions were used in all simulations unless otherwise noted.

For growth on cellulose, the experimentally measured glucose equivalent uptake was represented in the model through the following pseudo-reactions: *3 glceq_e* → *cell3_e*; *4 glceq_e* → *cell4_e*; *5 glceq_e* → *cell5_e*; *6 glceq_e* → *cell6_e*. Here *cell3_e*, *cell4_e*, *cell5_e*, and *cell6_e* are cellodextrin polymers with 3, 4, 5, and 6 glucose monomers, respectively. These polymers can be imported inside the cell through the oligo-cellulose transport ABC system. The model is free to use any cellodextrin length, although higher lengths have higher ATP yield.[309,260]

5.4.4 Single-reaction deletion analysis to match experimentally observed phenotype

A core model of *C. thermocellum*[261] correctly predicted the experimentally observed lethality of Δ*hydG*Δ*ech*Δ*pfl*; however, the iAT601 genome-scale model built by extension of this core model failed, suggesting that the genome-scale model has alternative active pathways leading to the false growth prediction *in silico*. To resolve this false positive prediction in iCBI655, we calculated the maximum growth rates for all possible additional single reaction deletions in the Δ*hydG*Δ*ech*Δ*pfl* mutant. This analysis resulted in three possible additional reaction deletions that are predicted to be lethal (i.e., maximum growth rate prediction below 20% of the simulated wild-type value[204]), including the removal of i) glycine secretion (EX_gly_e), ii) 5,10-methylenetetrahydrofolate oxidoreductase (MTHFC), and (iii) deoxyribose-phosphate aldolase (DRPA). For the first removal, addition of sulfate or ketoisovalerate in the growth medium of Δ*hydG*Δ*ech*Δ*pfl* fails to predict growth recovery as observed experimentally,[261] making this option invalid. Likewise, the second removal is invalid because it makes PFL an essential reaction in the wild-type strain; however, experimental evidence demonstrates that Δ*pfl* mutant is able to grow.[206] The last option was chosen since it correctly predicts growth recovery of Δ*hydG*Δ*ech*Δ*pfl* by sulfate or ketoisovalerate addition in the growth medium, and does not make PFL essential in the wild-type strain.

5.4.5 Model comparison

The *C. thermocellum* and *E. coli* models were obtained from their respective publications in SBML format. Blocked reactions are calculated by allowing all exchange reactions to have an unconstrained flux (i.e., $lb_j = -1000$, $ub_j = 1000\ \forall j \in \mathit{Exchange}$). This enables the most general scenario which produces the smallest number of blocked reactions in each model. Additional details can be found in Supplementary Material 3.

5.4.6 Omics integration protocol

The omics integration protocol developed in this study consists of three steps: i) simulation of fold changes, ii) mapping of measured gene fold changes to reactions, and iii) comparison of measured and simulated fold changes.

Calculation of simulated fold changes

To simulate metabolic fluxes, lower and upper bounds (5.2) are constrained according to experimental data as described in Section 5.4.2. Then, for the pFBA method, a quadratic optimization problem (5.6) is solved leading to a unique flux distribution $v_{jk}^{\mathrm{pFBA}}\ \forall j \in \mathcal{J}$.

$$v_{jk}^{\mathrm{pFBA}} \in \arg\min \left\{ \sum_{j \in \mathcal{J}} v_{jk}^2 : v_{jk} \in \Omega_k \right\} \forall j \in \mathcal{J} \tag{5.6}$$

For the FVA method, a sequence of linear programming problems is solved where each flux is minimized (5.7) and maximized (5.8):

$$v_{jk}^{\min} \in \arg\min \{ v_{jk} : v_{jk} \in \Omega_k \}\ \forall j \in \mathcal{J} \tag{5.7}$$

$$v_{jk}^{\max} \in \arg\max \{ v_{jk} : v_{jk} \in \Omega_k \}\ \forall j \in \mathcal{J} \tag{5.8}$$

Note that for computation we applied the loop-less FVA method,[232,38] as implemented in cobrapy,[71] that introduces additional constraints in Ω_k to remove thermodynamically infeasible cycles from all feasible flux distributions.

FVA produces a flux range $[v_{jk}^{\min}, v_{jk}^{\max}]$ for each reaction $j \in \mathcal{J}$. To compare between states k (e.g., wild-type and mutant), we define the *FVA center*, a scalar variable that indicates a change in this range (5.9).

$$v_{jk}^{\text{FVA}} = \frac{v_{jk}^{max} + v_{jk}^{min}}{2} \tag{5.9}$$

The FVA center, v_{jk}^{FVA}, does not attempt to quantify the fraction of overlap between ranges nor to identify what type of shift might have occurred from all possible permutations, but simply provide an indicator of whether there is an upward shift (center increase) or downward (center decrease) between two conditions k. Unlike v_{jk}^{pFBA}, v_{jk}^{FVA} does not necessarily represent a feasible flux distribution of Ω_k.

Finally, to determine the fold change for either pFBA or FVA simulated fluxes, the conventional procedure for fold change calculation in omics data is emulated. First, values are floored to avoid very large (or infinite) fold changes in cases with very small magnitude change. Given the minimum flux flooring value $\epsilon = 0.0001$ a flooring function (5.10) is defined.

$$\mathit{floor}(v_j) = \begin{cases} v_j + \epsilon & \text{if } 0 < v_j < \epsilon \\ v_j - \epsilon & \text{if } 0 > v_j > -\epsilon \\ v_j & \text{otherwise} \end{cases} \tag{5.10}$$

Then, the fluxes are normalized to the substrate uptake rate $v_{\text{uptake},k}$ and fold change is calculated in $\log_2$ space (5.11).

$$FC_j^{\text{sim}}(v_{j,\text{mut}}, v_{j,\text{wt}}) = \log_2\left[\mathit{floor}\left(\frac{v_{j,\text{mut}}}{|v_{\text{uptake,mut}}|}\right)\right] - \log_2\left[\mathit{floor}\left(\frac{v_{j,\text{wt}}}{|v_{\text{uptake,wt}}|}\right)\right] \tag{5.11}$$

Calculation of measured fold changes

Fold change between case and control samples, FC_l, is calculated in $\log_2$ space for each gene $l \in \mathcal{L}$, where $\mathcal{L}$ is the set of genes in the model. These gene fold changes can be mapped to metabolic reaction fold changes using the gene-protein reaction associations (GPR), given

$\mathcal{G}_j$ as the set of genes with $FC_l \neq 0$ in the GPR of reaction j:

$$FC_j^{\text{meas}} = \frac{1}{|\mathcal{G}_j|} \sum_{l \in \mathcal{G}_j} FC_l \tag{5.12}$$

Identification of consistent fold changes

A reaction j is said to have a consistent fold change if the measured fold change has the same sign of at least one of the simulated fold changes, more formally:

$$\begin{aligned} \mathcal{M} := \Big\{ j \in \mathcal{J} : \Big(\Big[(FC_j^{\text{sim,pFBA}} < 0) \vee (FC_j^{\text{sim,FVA}} < 0) \Big] \wedge (FC_j^{\text{meas}} < 0) \Big) \\ \vee \Big(\Big[(FC_j^{\text{sim,pFBA}} > 0) \vee (FC_j^{\text{sim,FVA}} > 0) \Big] \wedge (FC_j^{\text{meas}} > 0) \Big) \Big\} \end{aligned} \tag{5.13}$$

where $\mathcal{M} \subseteq \mathcal{J}$ is the set of consistent reactions which is considered for further analysis and the simulated fold changes are re-defined for brevity (5.14-5.15).

$$FC_j^{\text{sim,pFBA}} := FC_j^{\text{sim}}(v_{j,\text{mut}}^{\text{pFBA}}, v_{j,\text{wt}}^{\text{pFBA}}) \tag{5.14}$$

$$FC_j^{\text{sim,FVA}} := FC_j^{\text{sim}}(v_{j,\text{mut}}^{\text{FVA}}, v_{j,\text{wt}}^{\text{FVA}}) \tag{5.15}$$

5.4.7 Software implementation

Model development was performed using Python and Jupyter notebooks with open-source Python libraries including cobrapy.[70] The sequence of upgrades and improvements can be seen in the Git version control records. The repository is available online through Github (`https://github.com/trinhlab/ctherm-gem`) and in Supplementary Material 3.

5.4.8 Proteomics data collection

C. thermocellum wild-type and Δ*hydG*Δ*ech* strains were cultured in batch reactors and metabolic fluxes were calculated as previously described.[261] For proteomics measurements, the wild-type and mutant strains were cultured in MNM and MTC media,[143] respectively. While both wild-type and mutant were originally cultured in MTC,[261] the wild-type had to be cultured separately in MNM medium due to insufficient volume for proteomics sampling in

the MTC culture. MTC has higher nitrogen and trace mineral concentrations, but previous studies have shown no effect on growth rates.[143] During the mid-exponential growth phase 10 mL samples were collected, centrifuged, and the resulting pellet was stored at −20 °C. Cell pellets were then prepared for LC–MS/MS-based proteomic analysis. Briefly, proteins extracted via SDS, boiling, and sonic disruption were precipitated with trichloroacetic acid.[92] The precipitated protein was then resolubilized in urea and treated with dithiothreitol and iodoacetamide to reduce and block disulfide bonds prior to digestion with sequencing-grade trypsin (Sigma-Aldrich). Following two-rounds of proteolysis, tryptic peptides were salted, acidified, and filtered through a 10 kDa MWCO spin column (Vivaspin 2; GE Healthcare) and quantified by BCA assay (Pierce).

For each LC–MS/MS run, 25 µg of peptides were loaded via pressure cell onto a biphasic MudPIT column for online 2D HPLC separation and concurrent analysis via nanospray MS/MS using a LTQ-Orbitrap XL mass spectrometer (Thermo Scientific) operating in data-dependent acquisition (one full scan at 15 k resolution followed by 10 MS/MS scans in the LTQ, all one µscan; monoisotopic precursor selection; rejection of analytes with an undecipherable charge; dynamic exclusion = 30 s).[91]

Eleven salt cuts (25 mM, 30 mM, 35 mM, 40 mM, 45 mM, 50 mM, 65 mM, 80 mM, 100 mM, 175 mM and 500 mM ammonium acetate) were performed per sample run with each followed by 120 min organic gradient to separate peptides.

Resultant peptide fragmentation spectra (MS/MS) were searched against the *C. thermocellum* DSM1313 proteome database concatenated with common contaminants and reversed sequences to control false-discovery rates using MyriMatch v.2.1.[254] Peptide spectrum matches (PSM) were filtered by IDPicker v.3[172] to achieve a peptide-level FDR of <1 % per sample run and assigned matched-ion intensities (MIT) based on observed peptide fragment peaks. PSM MITs were summed on a per-peptide basis and those uniquely mapping to their respective proteins were imported into InfernoRDN.[256] Peptide intensities were log2-transformed, normalized across replicates by LOESS, standardized by median absolute deviation, and median centered across all samples. Peptide abundance data were then assembled to proteins via RRollup and further filtered to maintain at least two values

in at least one replicate set. Protein abundances were then used for the modeling efforts describe herein.

All raw and database-searched LC-MS/MS data pertaining to this study have been deposited into the MassIVE proteomic data repository and have been assigned the following accession numbers: MSV000084488 (MassIVE) and PXD015973 (ProteomeXchange). Data files are available upon publication (`ftp://massive.ucsd.edu/MSV000084488/`).

5.4.9 Modular cell design

The iCBI655 model in its batch reaction and cellobiose carbon source configuration (Supplementary Material 2) was used as basis for the strain design. The alcohol pathways were curated from recent literature,[110,162,165] where adapted Adh can use either NADH or NADPH to form the target alcohol.[19] The esters pathway require the inclusion of a condensation reaction between alcohol and acyl-CoA that are already present in the alcohol pathways. This reaction can be performed by cloramphenicol acetyl transferase (CAT) as recently demonstrated.[237] All the software involved in the generation of these designs is available online at `https://github.com/trinhlab/modcell-hpc` and `https://github.com/trinhlab/modcell-hpc-study`.

Supplementary Materials

1. Supplementary tables.
2. iCBI655 model in various formats for cellobiose growth condition and map of central metabolic pathways in Escher format.
3. Flux dataset used to train the iCBI655 model and proteomics dataset for the wild-type and $\Delta hydG\Delta ech$ strains.
4. Software used to develop, configure, and analyze iCBI655.

Chapter 6

Design of modular cells for large product libraries

This chapter is based on the publication *Computational design of modular cells for large libraries of exchangeable product synthesis modules. Garcia, S. and Trinh, C. T. In preparation, 2020.* As first author I lead the development, implementation, and writing of this study. Supplementary Material 1 is provided in Appendix D, while Supplementary Materials 2 and 3 are provided as attachments.

Abstract

Microbial metabolism can be harnessed to produce a large library of useful chemicals from renewable resources such as plant biomass. However, it is laborious and expensive to create microbial biocatalysts to produce each new product. To tackle this challenge, we have recently developed modular cell (ModCell) design principle that enables rapid generation of production strains by assembling a modular (chassis) cell with exchangeable production modules to achieve target molecule overproduction. Previous computational ModCell design methods are limited to analyze small libraries of around 20 products. In this study, we developed a new computational method, named ModCell-HPC, capable of designing modular cells for large libraries with hundredths of products with highly parallel multi-objective evolutionary algorithm. We demonstrated ModCell-HPC to design *Escherichia coli* modular

cells towards a library of 161 endogenous production modules. From these simulations, we identified *E. coli* modular cells with few genetic manipulations that can produce dozens of molecules in a growth-coupled manner under different carbons sources. These designs revealed key genetic manipulations at the chassis and module levels to accomplish versatile modular cells. Furthermore, we used ModCell-HPC to identify design features that allow an existing modular cell to be re-purposed towards production of new molecules. Overall, ModCell-HPC is a useful tool towards more efficient and generalizable design of modular cells to help reduce research and development cost in biocatalysis.

6.1 Introduction

Modular design has gained recent interest as an effective approach to understand and redesign cellular systems.[85] In the fields of metabolic engineering and synthetic biology, various modularization strategies[17,272,86,83,84] have been proposed to address the slow and expensive design-build-test cycles of developing microbial catalysts for renewable chemical synthesis.[196] A promising system-level modularization[216] approach is ModCell,[86] that aims to design a modular (chassis) cell compatible with exchangeable production modules that enable metabolite overproduction. ModCell could be an effective tool to efficiently produce the vast number of molecules offered by nature,[273,150] but it remains unexplored for large product libraries.

Previous efforts in computational modular cell design are limited to analyze small libraries of around 20 products.[86,84] However, the design of modular cells for larger product libraries is both of practical and theoretical interest. Theoretically, using large libraries can lead to more general modular cell design rules, which might help to explain the naturally existing modularity of metabolic networks.[85] Practically, such modular cells could be implemented with genetic engineering techniques that enable rapid pathway generation, such as combinatorial ester pathways,[146] and where the modular cell could serve as a versatile platform for pathway selection and optimization using adaptive laboratory evolution.[293]

Modular cell design was formulated as a multi-objective optimization problem (MOP), named ModCell2, where each target phenotype activated by a module is an independent

objective.[86] ModCell2 was solved with multi-objective evolutionary algorithms (MOEAs) that used a master-slave parallelization scheme, where the objective functions are evaluated in parallel by slave processes, but every other step in the algorithm is performed serially (Figure 6.1 a).[86,83] This approach contains many serial steps, and hence limits the scalability of the algorithm with the number processes according to Ahmdal's law.[109] In particular, large population sizes, an effective strategy to deal with many objectives,[83,115] can dramatically slow down serial algorithm operations such as non-dominated sorting in NSGA-II,[61] one of the best performing MOEA to solve ModCell2.[83] Furthermore, increasing the product library size for ModCell leads to many-objective multi-objective optimization problems, which are notoriously difficult to solve.[116,154] Therefore, the master-slave approach used in ModCell2 is not suitable to analyze large problems that contain hundredths of exchangeable production modules. A new parallelization approach that uses high-performance computing (HPC) more effectively is needed to advance ModCell.

In recent years, multiple approaches to harness HPC have been developed to solve single-objective evolutionary algorithms (EA).[6] In particular, the island-parallelization approach has been proposed, where multiple instances of the EA are run independently but communicate with each other to enhance overall convergence towards optimal solutions (Figure 6.1 b). This addresses the serial bottlenecks of the master-slave approach by separating the algorithm into highly independent processes that directly map to the computing hardware. While this approach has not been throughly examined in MOEA, there are a few successful applications to specific design problems.[178,122,88]

In this study, we developed ModCell-HPC, a highly parallel MOEA that uses the island parallelization approach to solve modular cell design problems with hundredths of objectives. We demonstrated ModCell-HPC to design *Escherichia coli* modular cells with a large production module library of metabolically and biochemically diverse endogenous compounds. Analysis of these designs revealed key genetic manipulations both at the chassis and module levels required for highly compatible modular cells. Furthermore, we designed modular cells for conversion of various hexoses and pentoses, since these sugars are the main components of biomass feedstocks.[27] Finally, we used ModCell-HPC to identify the features of a modular cell that make it compatible towards new production modules.

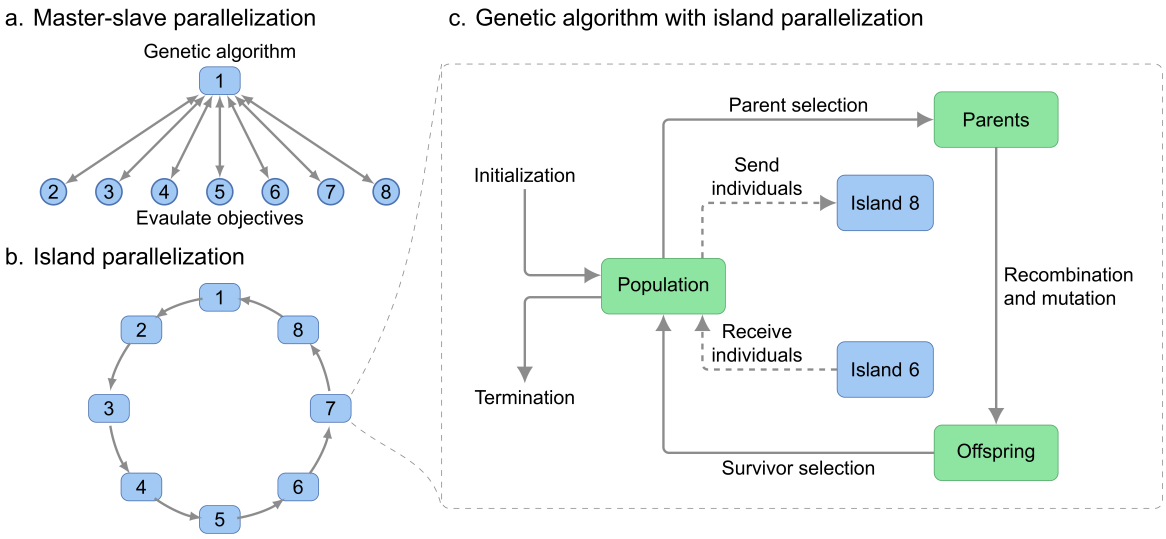


Figure 6.1: Parallelization schemes for multi-objective evolutionary algorithms. (a) Master-slave approach used in the original ModCell2 implementation. (b) Island parallelization following ring topology implemented in ModCell-HPC. (c) Key steps in evolutionary algorithm.

6.2 Methods

6.2.1 Multi-objective optimization formulation of modular cell design problem

The modular (chassis) cell is built in a top-down manner by removing metabolic functions from a parent strain, and then inserting exchangeable modules into the chassis to create production strains that optimally display the target phenotypes. Due to the conflicting biochemical and metabolic requirements of different product synthesis pathways, the modular cell design problem was formulated as the following MOP known as ModCell2:[86]

$$\max_{y_j, z_{jk}} \quad (f_1, f_2, \ldots, f_{|\mathcal{K}|})^T \quad \text{s.t.} \tag{6.1}$$

$$f_k \in \arg\max \left\{ \frac{1}{f_k^{max}} \sum_{j \in \mathcal{J}_k} c_{jk} v_{jk} \quad \text{s.t.} \right. \tag{6.2}$$

$$\sum_{j \in \mathcal{J}_k} S_{ijk} v_{jk} = 0 \qquad \text{for all } i \in \mathcal{I}_k \tag{6.3}$$

$$l_{jk} \leq v_{jk} \leq u_{jk} \qquad \text{for all } j \in \mathcal{J}_k \tag{6.4}$$

$$l_{jk} d_{jk} \leq v_{jk} \leq u_{jk} d_{jk} \qquad \text{for all } j \in \mathcal{C} \tag{6.5}$$

$$\left.\begin{array}{l} d_{jk} = y_j \vee z_{jk} \end{array}\right\} \qquad \text{for all } k \in \mathcal{K} \tag{6.6}$$

$$z_{jk} \leq (1 - y_j) \qquad \text{for all } j \in \mathcal{C},\ k \in \mathcal{K} \tag{6.7}$$

$$\sum_{j \in \mathcal{C}} z_{jk} \leq \beta \qquad \text{for all } k \in \mathcal{K} \tag{6.8}$$

$$\sum_{j \in \mathcal{C}} (1 - y_j) \leq \alpha \tag{6.9}$$

This MOP simultaneously maximizes all objectives f_k (6.1), where k belongs to the set of production networks $\mathcal{K}$. Each production network represents the combination of the chassis with a specific production module, and it is simulated through a stoichiometric model[204] (6.2-6.6) with a set of metabolites $\mathcal{I}_k$ and a set of reactions $\mathcal{J}_k$. The stoichiometric model predicts metabolic fluxes according to the following constraints: (i) mass-balance (6.3), where S_{ijk} represents the stoichiometric coefficient of metabolite i in reaction j of production network k, (ii) flux bounds (6.4) that determine reaction reversibility and available substrates, where l_{jk} and u_{jk} are lower and upper bounds respectively, and (iii) genetic manipulation (6.5), i.e., deletion of a reaction j in the chassis through the binary indicator y_j, or insertion of a reaction j in a specific production network k through the binary indicator z_{jk}. Only a subset of all metabolic reactions, $\mathcal{C}$, are considered as candidates for deletion, since many of the reactions in the metabolic model cannot be manipulated to enhance the target phenotype.

The desirable phenotype f_k for production module k is determined based on key metabolic fluxes v_{jk} (mmol/gDCW/h) predicted by the model (6.2-6.5). For this study, we selected the weak growth coupled to product formation ($wGCP$) design objective that requires a high minimum product synthesis rate at the maximum growth-rate, enabling growth selection of optimal production strains. Hence, in $wGCP$ design, the inner optimization problem seeks to maximize growth rate while calculating the minimum product synthesis rate through the linear objective function (6.2). Here c_{jk} is 1 and -0.0001 for j corresponding to the biomass and product reactions across all networks k, respectively, and 0 otherwise. In general, the definition of f_k need not be linear and other design phenotypes can be defined.[86]

Finally, design constraints (6.7-6.9) define the limitations of the design variables representing genetic manipulations, y_j and z_{jk}. As part of modular cell design, reactions can be removed from the chassis but inserted back to specific production modules, enabling the chassis to be compatible with a broader number of modules (6.7). The total module reaction additions and reaction deletions in the chassis are limited by parameters β (6.8) and α (6.9), respectively.

To define the solutions of ModCell2 (6.1-6.9), consider the following general multi-objective optimization problem with design variables x from a set $\mathcal{X}$ and objective functions $f_i(x)$:

$$\max_x \quad F(x) = (f_1(x), f_2(x), \ldots)^T \; \forall x \in \mathcal{X}$$

The solution of such optimization problem is denoted as Pareto set:

$$\mathcal{PS} := \{x \in \mathcal{X} : \nexists\, x' \in \mathcal{X}, F(x') \prec F(x)\}$$

Here $F(x') \prec F(x)$ indicates that the objective vector $F(x')$ *dominates* $F(x)$, defined as $f_i(x') \geq f_i(x)$ for all objectives i, and $f_i(x') \neq f_i(x)$ for at least one i. Hence, the Pareto set contains all non-dominated solutions to the optimization problem, i.e., when comparing any two non-dominated solutions, the value of a certain objective must be diminished in order to increase the value of a different objective. The projection of the Pareto set on the objective space is denoted Pareto front:

$$\mathcal{PF} := \{F(x) : x \in \mathcal{PS}\}$$

6.2.2 Implementation of many-objective evolutionary algorithm with high-performance computing

To overcome the issues of the master-slave approach (Figure 6.1 a) used in ModCell2,[86] we implemented an island parallelization scheme,[6] where each computing process is an instance of the MOEA (Figure 6.1 b). These instances exchange individuals (i.e., potential solutions) in a process called migration, hence enhancing overall convergence towards optimal solutions

(Figure 6.1 c). The migration operation can be performed in different modes, depending on which individuals from the local population are exchanged, and also how often such exchanges happen. These options are captured by the migration type and migration interval parameters, respectively (Table 6.1). To enhance performance and scalability, the migration process was implemented asynchronously, i.e., the population within each island can continue to evolve without needing to wait for sent individuals to arrive at their destination island or for incoming individuals to be received.

To improve the quality of the MOEA solutions, we implemented two post-processing steps specific to ModCell (Figure D1): First, we eliminate *futile module reactions*. These module reactions once removed do not diminish the objective value of the associated production network. Second, we coalesce multiple designs with the same deletions but different module reactions. This combination helps obtain a superior solution.

The software implementation of the proposed island-MOEA, denoted *ModCell-HPC*, is written in the C programming language and available in Supplementary Material 3 and `https://github.com/TrinhLab/modcell-hpc`.

Table 6.1: Island-MOEA parameters evaluated in ModCell-HPC.

Name	Description
Population size	Number of individuals per island.
Migration type	"ReplaceBottom": After non-dominated sorting of the Pareto front[61] (survivor selection), top individuals are sent and bottom individuals replaced. "Random": Random individuals are sent and replaced.
Migration interval	Number of generations between migration events.
Run time	Wall-clock time for which the MOEA runs. It will determine the total number of generations.
Cores	Each island is a computing core at the hardware level.

6.2.3 Computation hardware

We conducted all ModCell-HPC computations in *beacon* nodes from the Advanced Computing Facility at the Joint Institute for Computational Science, The University of Tennessee

and Oak Ridge National Laboratory. Each node contains a 16 core Intel Xeon E5-2670 central processing unit (CPU) and 256 GB of random access memory (RAM). The results were analyzed in a desktop computer with an Intel Core i7-3770 CPU and 32 GB of RAM.

6.2.4 Target product identification

The target products are endogenous *E. coli* metabolites that meet the following requirements: i) their maximum theoretical yields are above 0.1 (mol product/mol of substrate); ii) they are organic; and iii) they could be produced anaerobically in a growth coupled manner with a yield above 50%, a property determined in a previous study.[283] If a given metabolite meets all these conditions but appears in multiple compartments, only one location is choosen. This resulted in 161 target metabolites. Metabolites that did not have a secretion mechanism originally present in the model required an exchange pseudo-reaction that represents metabolite secretion to the growth medium or intracellular accumulation at steady-state. The products in the selected library have diverse molecular weights and are overall highly reduced (Figure D2).

6.2.5 Model configuration

We used the iML1515 *E. coli* model[188] for all simulations. To configure the model, glucose uptake was set to 15 (mmol/gCDW/h); the default ATP maintenance value in iML1515 was used; 20% of the maximum anaerobic growth rate was used as the minimum growth rate, corresponding to 0.0532 (1/h); and only commonly observed fermentation products were allowed for secretion. This model configuration is equivalent to the previous modular cell design studies[86] except for the higher glucose uptake rate. This rate was increased to match the study of Kamp and Klamt[283] which was partially used here to identify target products.

6.2.6 Design characterization

Compatible modules and compatibility

An important qualitative feature of a designed modular (chassis) cell is module compatibility. The chassis is *compatible* with a module if the performance of the resulting production

strain is above a defined threshold. In this study, we used the *wGCP* design objective that corresponds to the minimum product yield at the maximum growth rate,[86] and selected a threshold of 0.5 to establish compatibility. Under these conditions, we expect a module compatible with the chassis can lead to a product yield above 50% of the theoretical maximum during the growth phase. The *compatibility* of a modular cell corresponds to the number of modules that are compatible with it.

Minimal covers

A *minimal cover* is the smallest group of modular cells needed to ensure all potentially compatible products in a library are compatible with at least one of the modular cells. To identify minimal (set) covers computationally, we use the classical integer programming formulation:

$$\min_{x_h \in \{0,1\}} \sum_{h \in \mathcal{H}} (\gamma_h x_h) \tag{6.10}$$

subject to:

$$\sum_{h \in \mathcal{H}} a_{hk} x_h \geq 1 \qquad \forall\, k \in \mathcal{K}' \tag{6.11}$$

This optimization problem minimizes the number of designs in the set cover, where $\mathcal{H}$ is the set of strain designs, h, produced by ModCell-HPC (6.10). The binary indicator variable x_h takes a value of 1 if design h is selected as part of the set cover and 0 otherwise. Certain designs can be prioritized (e.g., they contain preferable genetic manipulations) using the weighting parameter γ_h. However, we set $\gamma_h = 1$ in all our simulations. All compatible products k must be included in at least one of the selected designs (6.11). The parameter a_{hk} takes a value of 1 if product k is compatible with design h and 0 otherwise. There must exist at least one $h \in \mathcal{H}$ for which $a_{hk} = 1$ to ensure a feasible solution exists; therefore $\mathcal{K}'$ is the subset of products compatible in at least one design of $\mathcal{H}$.

To enumerate all minimal covers, we iteratively solved the minimal cover problem (6.10-6.11) with the addition, in each iteration, of an integer cut inequality (6.12) that removes a

previously found solution $\mathcal{S}$.

$$\sum_{h \in \mathcal{S}} x_h \leq |\mathcal{S}| - 1 \tag{6.12}$$

6.2.7 Coverage performance indicator

Algorithm performance is tested against several parameter configurations, each producing a Pareto front approximation ($\mathcal{PF}$). All resulting Pareto fronts are gathered into a reference Pareto front ($\mathcal{PF}^*$). Coverage, C, is defined as the fraction of solutions in $\mathcal{PF}^*$ captured by a given approximation $\mathcal{PF}$:

$$C = \frac{|\mathcal{PF} \cap \mathcal{PF}^*|}{|\mathcal{PF}^*|} \tag{6.13}$$

In our analysis, we only use unique non-dominated points in both $\mathcal{PF}$ and $\mathcal{PF}^*$ to avoid many alternative solutions from biasing the coverage indicator.

6.3 Results

6.3.1 Tuning of ModCell-HPC method parameters

A known challenge of heuristic optimization approaches is their reliance on parameter tuning for rapid convergence towards optimal solutions. To identify sensible default parameters for ModCell-HPC, we first scanned parameter combinations with a previous[86] 20 objectives problem that is fast to solve, then focused on the most relevant parameters for a large-scale problem with 161 objectives corresponding to the current product library. In both cases, we used two performance metrics to identify the best algorithm parameters: i) *Coverage*, that indicates the fraction of Pareto optimal solutions identified by a given parameter configuration (Section 6.2.7). ii) *minimal cover size*, i.e., the smallest number of modular cells needed to ensure all compatible products in the library that are compatible in at least one (Section 6.2.6). Coverage is a general and unbiased quantitative measure which is preferred over other similar metrics based on a previous study,[83] while minimal cover size is based on practical goals.

In our initial benchmark study with the 20 objectives problem, we screened different total run times, migration interval, migration types, and population sizes (Table 6.1) for

best achieving modular cell designs. The design parameters were set to $\alpha = 6$ and $\beta = 1$, which are sufficient to find highly compatible designs.[84] For 1 hour run time, we observed the smallest population size (100) reached more generations (Figure D3 e,f) and hence achieved better results in both metrics (Figure D3 a,b). However, for a 2 hour run time, both population sizes of 100 and 500 attained similar cover sizes (Figure D3 g), indicating that a minimum of approximately 150 generations (Figure D3 e,f,k,l) is necessary for convergence of this problem irrespectively of population size. Taken together, the different performance between 100 and 500 population sizes in relation to run time indicates that under limited run times an optimal population size could be found to attain sufficient generations for convergence. The migration interval only appeared detrimental at the highest value of 50 with the smallest population size of 100 at 1 hour (Figure D3 a,b,g,h); otherwise this parameter was considered secondary, and hence an intermediate value of 25 was selected for further simulations. Similarly, migration policy also appeared to be a secondary parameter; nonetheless, the "ReplaceBottom" migration policy was selected for further simulations since it is better or equal to the "Random" policy in all cases (Figure D3 c,d,i,j).

For the large-scale benchmark with 161 products, we investigated the importance of run time, population size, and the number of computational cores (Table 6.1). For this benchmark, the design parameters were set to $\alpha = 10$ and $\beta = 2$ to enable successful designs without a large number of genetic modifications that can lead to unrealistic model predictions and implementation requirements. We evaluated 5 and 10 hour run times. For 5 hours run time, a population size of 200 was better in all metrics (Figure 6.2 a,b,c,e,f,g) and reached 50-100 generations (Figure 6.2 d). For a 10 hours run time, the population sizes of 200 and 300 had equivalent performance (Figure 6.2 e-g), despite the population size of 200 reaching approximately 50 generations more than the 300 population size. The population size of 100 underperformed at both run-times (Figure 6.2 a,b,e,f). Taken together, this large-scale benchmark study indicates that after a given number of generations, larger population sizes are comparable as long as they are above a minimum size. Hence, a population size of 200 is the minimum required for proper convergence and should be used under limited run times. Increasing the number of cores leads to more solutions (Figure 6.2 c,g), due to a larger meta-population (the total population of all islands). However, additional cores do not necessary

find better solutions in terms of minimal cover size and individual product compatibility (Figure 6.2 b,f). These indicators plateaued at around 48 cores in both cases so this value was used for further simulations. Alternative communication topologies among islands[108] may provide better scaling with cores but were not explored here.

In summary, the benchmark performed here aims to provide a general guideline to use the ModCell-HPC. Furthermore, this parameter meta-optimization procedure can be repeated to fine-tune the algorithm to specific problem features (e.g., number of objectives) and computational resources available (e.g., run time and computing cores).

6.3.2 Design of *E. coli* modular cells for large product library

A small number of genetic manipulations are sufficient for highly compatible modular cell After tuning ModCell-HPC, we used it to design *E. coli* modular cells for our library of 161 products. First, we scanned a broad range of design parameter combinations (α-β: 5-1, 10-2, 20-4, and 40-8) to identify the required genetic manipulations for highly compatible designs (Figure D4 a). Increasing the number of genetic manipulations led to an average increase in design compatibility. However, the maximum compatibility remained around 50% of the library (80 products) for all cases. This result indicates that highly compatible platforms can be built with a small number of genetic manipulations. We selected the designs with $\alpha = 5$, $\beta = 1$ (Supplementary Material 2) for further analysis, since designs with few genetic manipulations are more accurately simulated and also better to implement in practice.

A few reaction deletions in central metabolism targeting byproducts and branch-points are key to build modular cells We sorted reaction deletions according to how often they appear across designs (Table 6.2). The top 7 reactions are used $\geq$10% of the designs and belong to central metabolism, indicating their importance to accomplish growth-coupled-to-product-formation phenotypes. Overall, the role of these deletions can be classified into two functions: i) to eliminate major byproducts and ii) to alter key branch-points in metabolism that influence the pools of precursor metabolites, including carbon, redox, and energy precursors. The first type of manipulations is generally intuitive and often

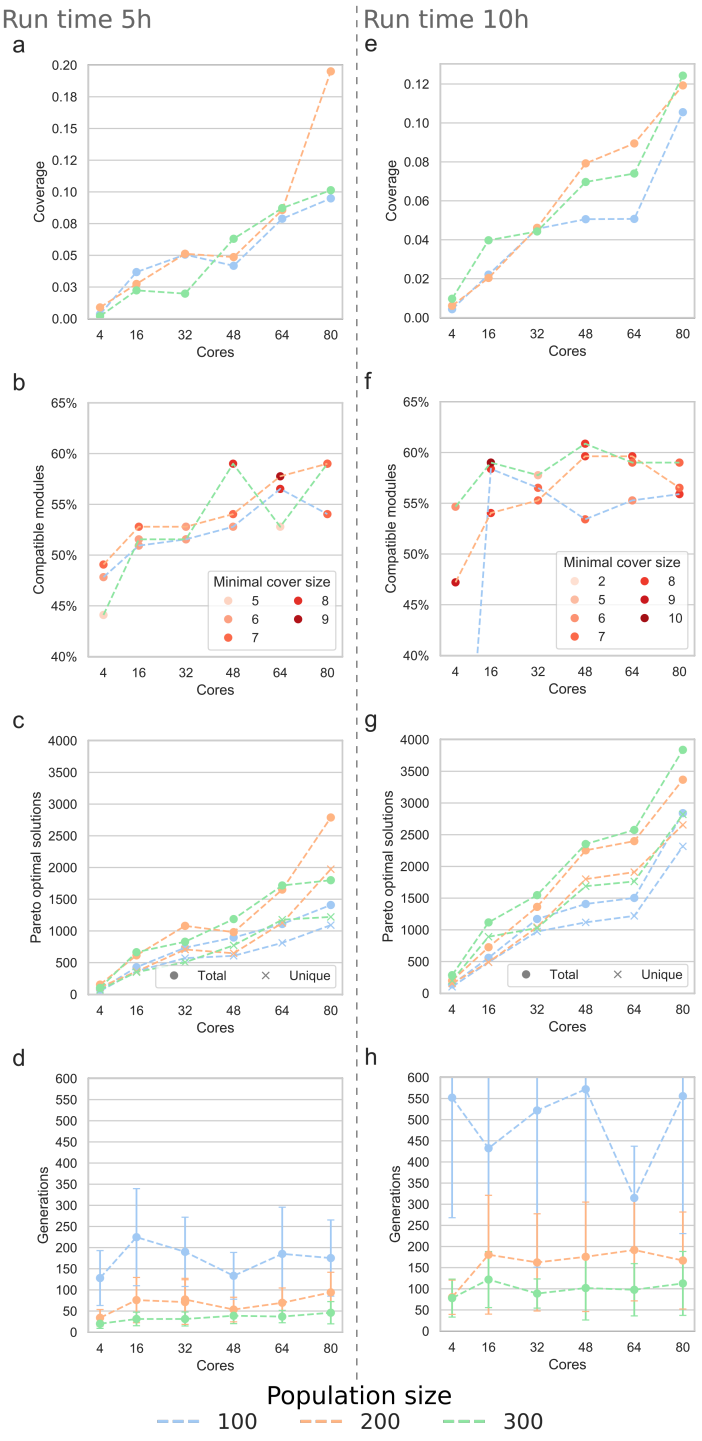


Figure 6.2: ModCell-HPC benchmark with 161 products. (a) and (e) Coverage is the fraction of Pareto optimal designs captured by a Pareto front approximation (Section 6.2.7). (b) and (f) Compatible modules indicates the products that appear in at least one design with a design objective above the compatibility threshold, while minimal cover size is the smallest number of designs needed to capture all compatible products (Section 6.2.6). (c) and (g) Total and unique number of solutions in the Pareto front approximations. (d) and (h) Total generations.

used in metabolic engineering efforts.[295] The second type of manipulations are not commonly identified unless metabolic model simulations are used,[267,280,40] even though the importance of targeting metabolic branch-points was noted early.[250] An example of this second type observed in our designs is TPI deletion, that activates the methylglyoxal bypass,[78] reducing the overall ATP yield resulting from glucose conversion into pyruvate. Lower ATP yield limits biomass formation hence redirecting carbon flow towards products of interest. While such strategies are not common, TPI deletion predicted by model simulations was successfully used for enhanced 3-hydroxypropionic acid production,[267] and ATP waisting has recently been proposed to enhance production of certain molecules.[23] Another example of branch-point manipulation is PPC deletion, that has been shown to lower flux from lower glycolysis towards the TCA cycle,[59,210] resulting in lower succinate production, and an increased pool of *pep*, pyruvate and acetyl-CoA. Additionally, PPC deletion to increase the *nadph* pool for production of flavonoids was predicted by model simulation and experimentally validated.[40] In summary, design of highly compatible modular cells requires not only major byproduct removal, but also manipulation of key branch points in central metabolism.

Table 6.2: Top 20 reaction deletions for design parameters $\alpha = 5$, $\beta = 1$ with 162 designs. Counts indicate the percentage of designs where the deletion is used. All reaction and metabolite abbreviations used in this study correspond to BiGG identifiers (`http://bigg.ucsd.edu`).

ID	Name	Formula	Counts (%)
ALCD2x	Alcohol dehydrogenase (ethanol)	etoh_c + nad_c ↔ acald_c + h_c + nadh_c	57.4
TPI	Triose-phosphate isomerase	dhap_c ↔ g3p_c	45.1
ACALD	Acetaldehyde dehydrogenase (acetylating)	acald_c + coa_c + nad_c ↔ accoa_c + h_c + nadh_c	40.7
FLDR2	Flavodoxin reductase (NADPH)	2.0 flxso_c + nadph_c → 2.0 flxr_c + h_c + nadp_c	24.1
PPC	Phosphoenolpyruvate carboxylase	co2_c + h2o_c + pep_c → h_c + oaa_c + pi_c	21.6
TKT2	Transketolase	e4p_c + xu5p__D_c ↔ f6p_c + g3p_c	15.4
LDH_D	D-lactate dehydrogenase	lac__D_c + nad_c ↔ h_c + nadh_c + pyr_c	13
G3PD2	Glycerol-3-phosphate dehydrogenase (NADP)	glyc3p_c + nadp_c ↔ dhap_c + h_c + nadph_c	7.4
POR5	Pyruvate synthase	coa_c + 2.0 flxso_c + pyr_c ↔ accoa_c + co2_c + 2.0 flxr_c + h_c	7.4
ACKr	Acetate kinase	ac_c + atp_c ↔ actp_c + adp_c	6.8
THD2pp	NAD(P) transhydrogenase (periplasm)	2.0 h_p + nadh_c + nadp_c → 2.0 h_c + nad_c + nadph_c	6.2
GLUDy	Glutamate dehydrogenase (NADP)	glu__L_c + h2o_c + nadp_c ↔ akg_c + h_c + nadph_c + nh4_c	5.6
ASPT	L-aspartase	asp__L_c → fum_c + nh4_c	5.6
ASNS2	Asparagine synthetase	asp__L_c + atp_c + nh4_c → amp_c + asn__L_c + h_c + ppi_c	4.9
CBMKr	Carbamate kinase	atp_c + co2_c + nh4_c ↔ adp_c + cbp_c + 2.0 h_c	4.3
RNDR4	Ribonucleoside-diphosphate reductase (UDP)	trdrd_c + udp_c → dudp_c + h2o_c + trdox_c	3.7
RPE	Ribulose 5-phosphate 3-epimerase	ru5p__D_c ↔ xu5p__D_c	3.1
SERD_L	L-serine deaminase	ser__L_c → nh4_c + pyr_c	3.1
LCARS	Lacaldehyde reductase (S-propane-1,2-diol forming)	h_c + lald__L_c + nadh_c ↔ 12ppd__S_c + nad_c	2.5
FUM	Fumarase	fum_c + h2o_c ↔ mal__L_c	2.5

Module reaction usage reveals pathway interfaces and unbiased module definition The modular cell optimization formulation (Section 6.2.1) not only identifies genetic manipulations in the modular cell, but also in the production modules. Module reactions correspond to reactions deleted in the chassis but inserted back in specific production modules to enable compatibility. We examined the module reactions used by all designs (Figure 6.3). As expected, ALCD2x, ACKr, and LDH_D, are used by ethanol, acetate, and lactate production networks, respectively. More notably, we observe not highly products such as acetate, use ACALD; similarly 3-methyl-2-oxobutanotae and 2,3-dihydroxy-3-methylbutanoate, that are naturally precursors of valine and artificially of isobutanol[11,12], use FUM and MDH. These module reactions likely play a role in both the synthesis of relevant TCA precursors and the secretion of succinate as an electron sink. Interestingly, fatty acids tend to use TPI, which as mentioned earlier, its deletion activates the methylglyoxal bypass lowering the overall ATP yield. The first step in fatty acid biosynthesis, acetyl-CoA carboxylase, requires one ATP per mol of malonyl-CoA, explaining the usage of TPI as a module reaction for this family of products. Overall, module reactions enhance the compatibility of a modular cell, leading to more efficient strategies and revealing potential metabolic flux bottlenecks that are not always directly upstream of the target product.

Three modular cells is the smallest set needed to cover all compatible products We next aimed to identify the smallest set of modular cells that includes all compatible products in the library (Section 6.2.6). For the Pareto set of designs $\alpha = 5$, $\beta = 1$, we enumerated a total of 12 minimal covers of size 3. These covers are spanned by combinations of 8 unique designs (Figure D5). We selected cover k that contains designs 82, 121, and 124, which use few deletions and have similar genetic manipulations among them. All designs in this cover have in common the deletion of ALCD2x and LDH_D, disabling production of ethanol and lactate, the major reduced products of anaerobic growth in *E. coli*. Designs 121 and 124 have 57 compatible products in common, while design 121 is uniquely compatible with ethanol, formate, and 2,3-dihydroxymethylbutanoate, and design 124 is uniquely compatible succinate (Figure 6.4 a). These two designs only differ in that design 121 uses FUM deletion while design 124 uses MDH deletion (Figure 6.4 b). Design 82 is the

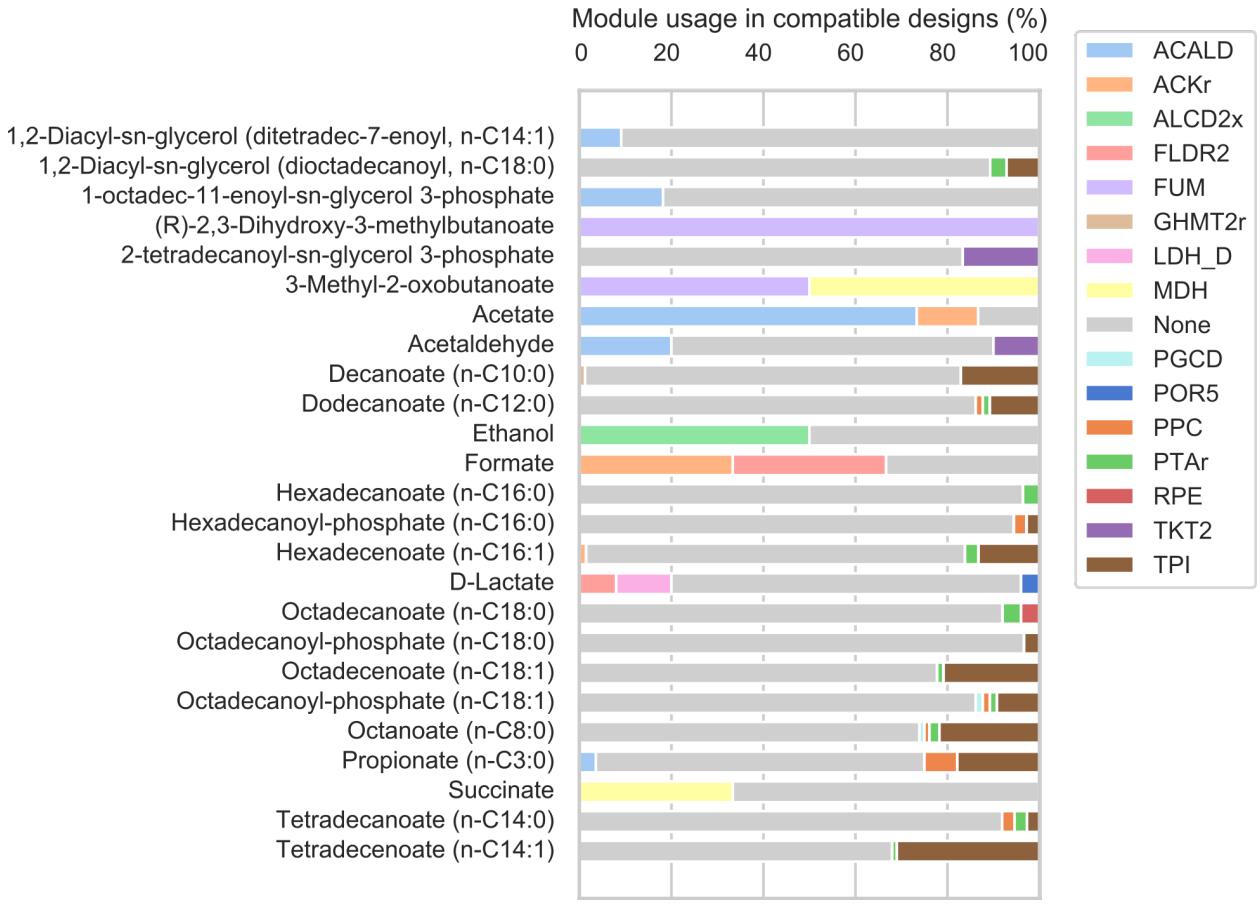


Figure 6.3: Module reaction usage for design parameters $\alpha = 5$, $\beta = 1$. Only designs compatible with the product are considered in the module usage frequency.

only design that features the deletion of FLDR2 and PPC, and it is uniquely compatible with 24 modules, all fatty acids, making this design quite different from designs 121 and 124. FLDR2 is coupled with POR5 to form a pathway for the reduction of pyruvate into acetyl-CoA consuming *nadph* (Figure 6.4 c), a key redox cofactor in fatty acid biosynthesis. PPC deletion is another strategy to increase *nadph* available that has been experimentally validated.[40] Overall, these designs can be efficiently built due to their similarity, and are mainly composed of strategies that have been demonstrated in isolation but also seem applicable to cover large product families.

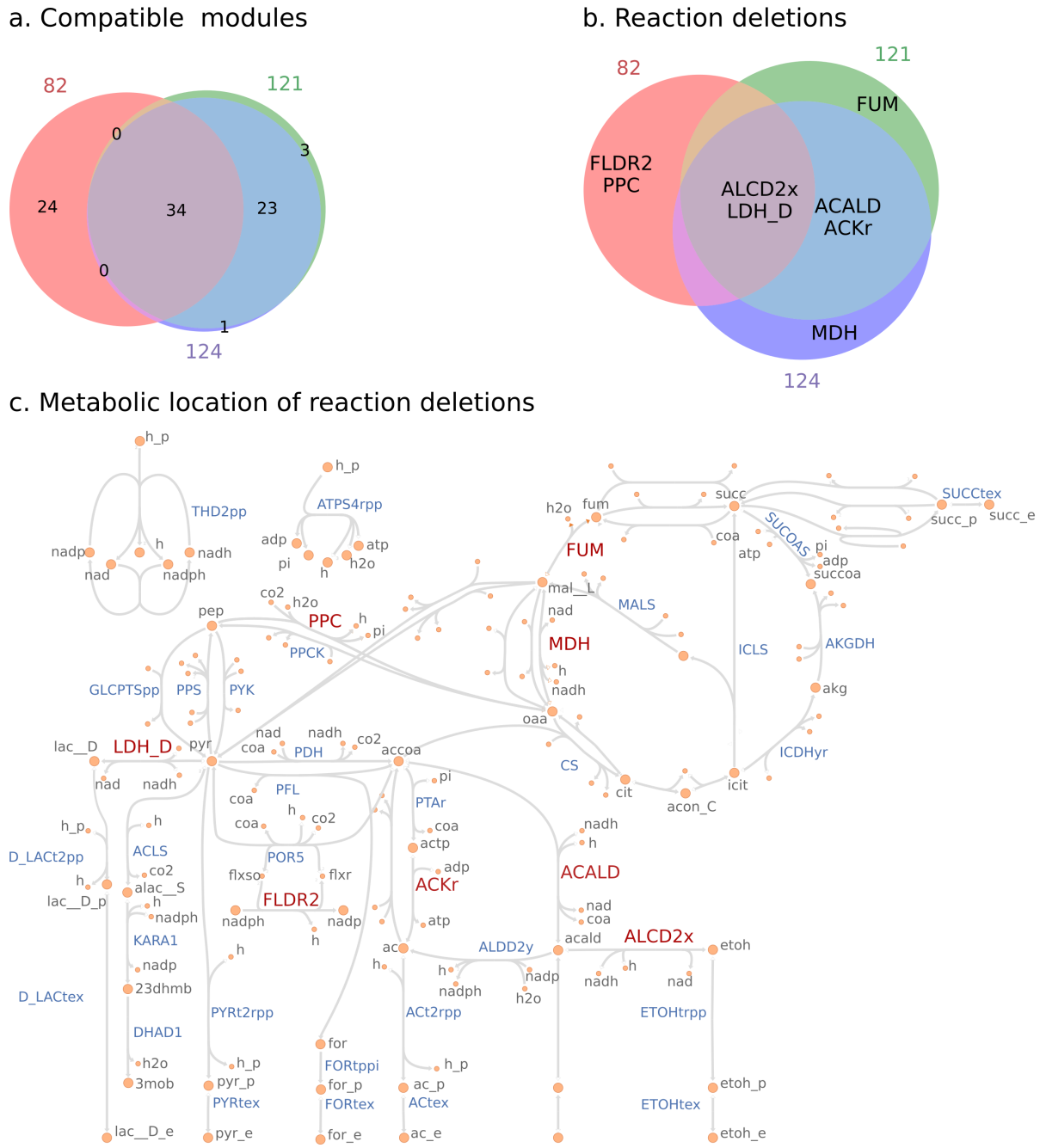


Figure 6.4: Comparison of the designs in the selected minimal cover. (a) Venn diagram of products compatible with each design. The products uniquely compatible with specific designs are (see `http://bigg.ucsd.edu` for abbreviation descriptions): Design 121: *etoh, for, 23dhmb*; Design 124: *succ*; Design 82: *pg140, 2hdecg3p, 2odec11eg3p, 1agpg180, pe140, pg161, pg141, 2hdec9eg3p, pgp161, 2agpg180, 1ddecg3p, pg120, pgp141, pgp140, pe141, ps140, apg120, ps120, pgp120, pe120, lipidX, 2tdecg3p, 2odecg3p, ps141.* (b) Venn diagram of reaction deletions that constitute each design. (c) Metabolic map with reaction deletions colored in red.

6.3.3 Design of *E. coli* modular cells for conversion of hexoses and pentoses

Non-glucose carbon sources can require more genetic manipulations for high compatibility designs We designed modular cells to consume other relevant carbon sources besides glucose also present in biomass feedstocks, including two pentoses, xylose and arabinose, and two more hexoses, galactose and mannose (Figure 6.5 a). For this case study, everything remained the same except for the substrate uptake reaction in the model which was changed to reflect the sole carbon source in each case. We first scanned the distribution of design compatibilities resulting from various combinations of α and β for each carbon source (Figure D4 b-e). All cases plateaued at maximum compatibilities around 50%; however, galactose, arabinose and xylose required at least $\alpha = 10$, $\beta = 2$ to reach this level, while glucose and mannose reached it with only $\alpha = 5$, $\beta = 1$. Hence, we selected $\alpha = 10$, $\beta = 2$ for further analysis. Overall, this simulation reveals the possibility of highly compatible modular cells for various hexose and pentose carbon sources, at the expense of an increased number of genetic manipulations for some of the carbon sources.

The effect of pentose uptake in redox metabolism leads to lower compatibility than hexoses For the set of designs in each carbon source, we examined the total compatible products, i.e., number of unique products compatible in at least one design from the Pareto front. This analysis revealed a group of 26 products (27% of the total 96 compatible products and 16% of the original library size) that are only compatible in designs with hexose carbon sources (Figure 6.5 b). The incompatibility of these 26 products is likely due to the lower reduction potential and different uptake pathways of pentoses with respect to hexoses (Figure 6.5 a). More specifically, analysis of the most deleted reactions in each carbon source revealed several differences in deletions between pentoses and hexoses (Figure 6.5 c). Notably, pentoses do not use TKT2 and MDH reaction deletions, while hexoses make highly frequent use of them. TKT2 is a key component of incorporating pentoses into glycolysis, and hence cannot be deleted by pentose consuming designs. MDH has been observed to be up-regulated under anaerobic conditions when the sole carbon source is pyruvate, galactose, or xylose with respect to glucose.[208] Hence, MDH could be an important source of *nadh*

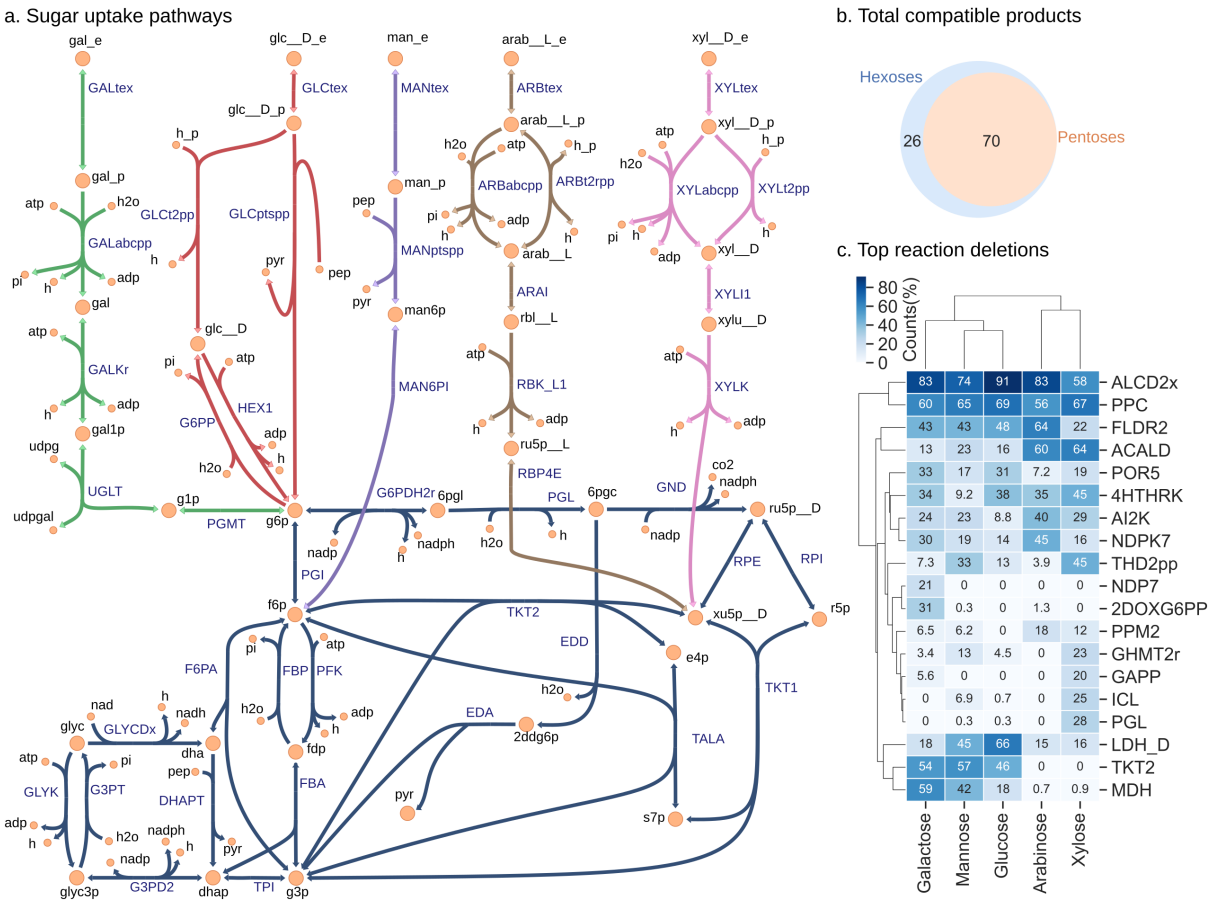


Figure 6.5: Design of modular cells for different carbon sources with design parameters $\alpha = 10$, $\beta = 2$. (a) Sugar uptake, pentose phosphate, Entner-Doudoroff, and upper glycolysis pathways. (b) Venn diagram of total products compatible with designs under pentoses and hexoses. The 26 products uniquely compatible with hexoses are: *1agpg180, 2tdecg3p, 2agpg181, 3c3hmp, 3mob, 2hdecg3p, pe141, ps120, 1agpg160, 2agpg160, 23dhmb, ps141, 1agpe180, 2agpg180, apg120, 2agpe180, pe120, 2odec11eg3p, 4mop, lipidX, 3c2hmp, 2ippm, 2hdec9eg3p, 1agpg181, dha, 2odecg3p.* (c) Top 20 reaction deletions according to deletion frequencies average across carbon sources. The counts for each carbon source correspond to the percentage of designs containing that reaction deletion.

for substrates with less reduction potential. Alternatively, MDH could also be important for *nadph* generation as part of a pathway involving NADP-dependent malic enzyme (ME2) that converts malate to pyruvate generating one mol of *nadph*. Overall, pentose uptake does not use the oxidative phase of the pentose phosphate pathway, the most important source of *nadph* in *E. coli*,[46] hence limiting the products that can be growth-coupled to these carbon sources. Further study of the reactions that limit pentose compatibility could enable strategies to overcome it in certain cases (e.g., create alternative sources of *nadph*[151,193]).

6.3.4 Compatibility towards modules unknown at the time of chassis design

Highly compatible designs are better suited to be re-purposed towards unknown products. To rapidly explore the large space of potential production modules, existing strains could be re-purposed for production of molecules not considered as part of the original design. To examine this scenario, we randomly partitioned the product library into two evenly sized groups, and independently used each partition as input for ModCell-HPC. This was done in triplicates, each corresponding correspond to a different random product partition. Hence, in each replicate there is a group of known products at the time of design and a group of unknown products. For the designs produced by ModCell-HPC, we computed their objective value and then compatibility towards unknown products, which we refer to as *unknown compatibility* of a design, a useful metric to understand the potential to re-purpose a given design. In contrast, *known compatibility* is the compatibility towards known products at the time of design, simply referred to as compatibility in previous cases study. The analysis of unknown compatibility of a new production module with an existing modular cell design is similar to the concept of degree of coupling that was previously introduced in MODCELL based on a different computation framework.[272] The total number of designs for each product group and the unknown compatibility distributions noticeably changed across replicates (Figure 6.6 a). This result reveals the important effect of known products in the resulting designs, which could be further explored to identify "representative products" that can capture the necessary metabolic phenotypes required for certain product families.

Remarkably, there was a high correlation between known and unknown compatibility of a given design (Figure 6.6 b-d). Hence, highly compatible designs are better suited to be re-purposed towards unknown products.

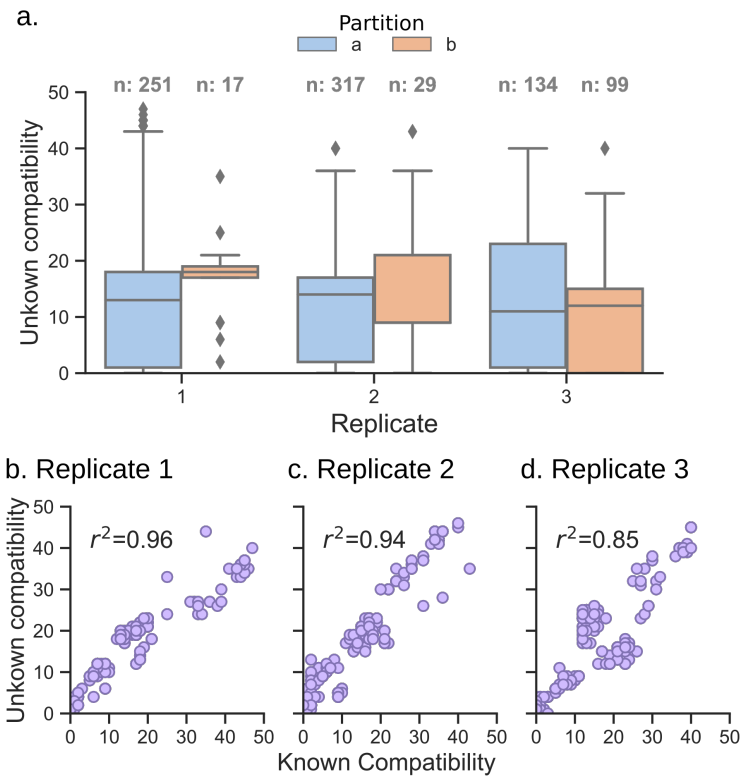


Figure 6.6: Compatibility towards unknown products in 3 random even partitions of the product library. (a) Distribution of unknown compatibility, n corresponds to the number of designs in each case. (b-d) Comparison between unknown and known compatibility of each design for each replicate, r^2 is the Pearson correlation coefficient.

Deletion reactions that remove major fermentation byproducts and alter redox metabolism have the highest contribution towards unknown compatibility

To identify the specific genetic intervention strategies that contribute to the unknown compatibility of a design, we defined the unknown compatibility contribution of deletion

reaction j (ucc_j) as follows:

$$ucc_j = \frac{\sum_{h \in \mathcal{H}_j} u_h}{|\mathcal{H}|} \tag{6.14}$$

where $\mathcal{H}_j$ is the subset of designs from a ModCell-HPC solution (Pareto set $\mathcal{H}$) containing deletion reaction j, and u_h is the unknown compatibility of design h. We computed ucc for all 3 replicates and examined the top 10 sorted by mean value (Table 6.3). The main contributors towards unknown compatibility were removal of major fermentative byproducts (lactate, ethanol, and acetate) followed by manipulation of redox pathways (THD2pp, FLDR2, MDH) and metabolic branch points (TKT2, PPC). Indeed byproduct removal strategies are the most common across the metabolic engineering literature.[295] Strain repurposing could be further explored with algorithms specialized for this task, e.g., by identifying module reactions in the unknown modules or using the existing strain as a starting point to identify genetic manipulations instead of a wild-type strain. In our analysis, we have identified that high modular cell compatibility and certain reaction deletions are positive indicators of compatibility towards unknown products.

Table 6.3: Top 10 reactions sorted by mean unknown compatibility contribution (ucc) among replicates.

ID	Name	ucc			
		R. 1	R. 2	R. 3	Mean
LDH_D	D-lactate dehydrogenase	13.2	10.5	11.9	11.9
ALCD2x	Alcohol dehydrogenase (ethanol)	11.5	10.5	11.8	11.3
PTAr	Phosphotransacetylase	4.0	4.8	6.5	5.1
ACALD	Acetaldehyde dehydrogenase (acetylating)	4.5	2.8	2.9	3.4
THD2pp	NAD(P) transhydrogenase (periplasm)	4.7	2.4	2.2	3.1
ACKr	Acetate kinase	3.8	2.2	1.7	2.6
FLDR2	Flavodoxin reductase (NADPH)	2.0	2.2	2.9	2.4
TKT2	Transketolase	2.6	2.0	2.5	2.4
PPC	Phosphoenolpyruvate carboxylase	2.3	2.2	2.5	2.3
MDH	Malate dehydrogenase	2.7	1.1	2.3	2.0

6.4 Conclusions

In this study, we developed ModCell-HPC, a computational method to design modular cells compatible with hundredths of product synthesis modules. We applied ModCell-HPC to design *E. coli* modular cells with a product library of 161 endogenous metabolites. This resulted in many Pareto optimal designs for the production of these molecules, from which we identified three modular cells that include all compatible products. The designs feature strategies consistent with previous experimental studies aimed at optimizing production of a single product, reinforcing our confidence in the value of our simulations. Remarkably, the strategies not only include removal of major byproducts, but also modification of key metabolic branch-points. The modular cells were designed for growth-coupled production, which not only is expected to result in high product yields but also enables high-throughput pathway engineering approaches. Specifically, the modular cell can be simultaneously transformed with a module library to rapidly identify good candidates through adaptive laboratory evolution.[67] We also used ModCell-HPC to design modular cells that utilize different hexoses and pentoses carbon sources. This revealed the limitations of pentoses towards coupling with certain products which might be addressed by redox cofactor engineering. Finally, we identified that high compatibility and certain reaction deletion are important features to re-purpose an existing modular cell towards new modules. Overall, ModCell-HPC is an effective tool towards more efficient and generalizable design of modular cells and platform strains that have recently captured the interest of metabolic engineers.[196]

Supplementary Materials

1. Supplementary figures.

2. Designs for selected parameters $\alpha = 5$, $\beta = 1$.

3. Computer programs used to generate the results of this study.

Future directions

Whole-cell biocatalysis technology can lead to renewable and sustainable manufacturing of chemicals, fuels, and materials. Over the past two decades, researchers at companies and universities have been developing platform strains that can produce various metabolically similar molecules.[196] However, these efforts have been limited by the use of qualitative and reductionistic approaches. Motivated by the potential of modular platform strains, we used multi-objective optimization theory and genome-scale metabolic models to develop several iterations of the ModCell design method (Figure i). We applied ModCell tools to design *E. coli* and *C. thermocellum* modular cells that enable various product synthesis phenotypes in a plug-and-play fashion. These proposed modular cell designs require few genetic manipulations thanks to the natural modular features of metabolic networks. Overall, this effort contributes to the current wave in synthetic biology of tackling problems through more quantitative and holistic approaches. We envision the new design tools will help reduce the cost and time required to develop efficient and robust biocatalytic strains that harness the large space of molecules resulting from natural and synthetic metabolic pathways.

The interdisciplinary nature of computational biology demands that one addresses algorithmic and modeling challenges, while also combining simulation efforts with experimental validation. To bridge the gap between simulations and experiments, a simulation can be used to generate a testable hypothesis, or a simulation can be used to explained existing data. Chapter 5 is an example of the second case, where proteomics data studied through model simulations increases our understanding of redox metabolism in *C. thermocellum* biofuel-producing strains. However, the majority of this work, formulated as a biocatalysis strain design problem, falls under the first case. Hence, the next wave of development of modular

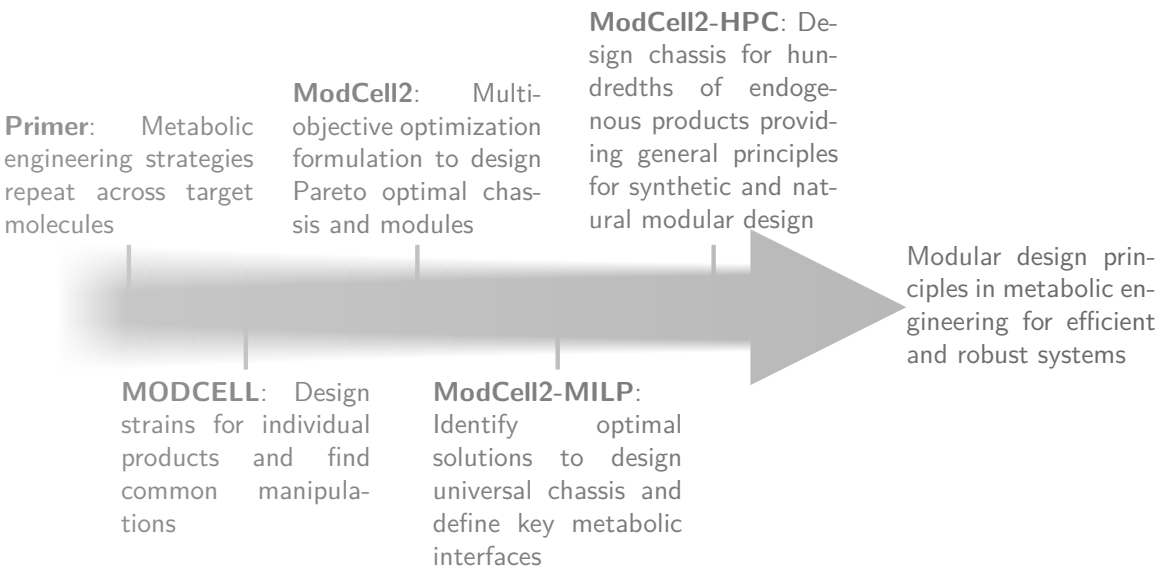


Figure i: Developments in modular cell design tools: Primer (Chapter 1), MODCELL,[272] ModCell2 (Chapter 2), ModCell-MILP (Chapter 4), ModCell-HPC (Chapter 6).

cell design principles should be focused on the implementation and characterization of the strains proposed here, closing the design-build-test-learn cycle.

As with most contemporary research works, this thesis contributes a drop to the ocean of knowledge needed to address the scientific and social challenges of our time. The implementation of modular design principles developed here can be impactful in addressing the challenge of whole-cell biocatalysis, however, there are other important facets of this problem. Most notably, the predictive and explanatory capacities of cellular models remain highly limited by a lack of tools to integrate disparate data types and to efficiently measure enzymatic kinetics. Furthermore, even if the necessary metabolic fluxes and required enzyme concentrations were known, it remains highly challenging to accomplish appropriate enzyme expression and pathway function *in vivo*. More generally, we lack precise comprehensive description of already known phenomena (e.g., metabolic reactions), and unknown or poorly explored biophysical phenomena (e.g., macromolecular crowding) are likely to highly influence cellular function. Hence, it is unlikely that we will develop the capacity to manipulate living organisms for any conceivable biological function in a near future. The good news is that we are rapidly overcoming the key challenges needed for

applied technologies, as evidenced by the many biotechnology companies emerging over the last decade.

As we develop novel technologies, we must also become aware of the ethical and existential risks associated with them. For example, genetic engineering for enhanced cognitive abilities could likely become an expensive medical treatment that increases the wealth gap in society. These concerns are specially relevant for the field of synthetic biology, as tools continue to become more widely available hence enabling DIY biohacking.[16] These developments could pose an existential risk for our current civilization, given that a highly destructive technology becomes sufficiently easy to use, and such technology cannot be "uninvented" or effectively policed. This challenge can be illustrated through The Urn of Inventions metaphor (Figure ii).[26] Briefly, consider technologies to be balls in an urn, and our current strategy is to draw balls as fast as possible, perhaps to obtain wealth, prestige, and citations. If there is a "black ball" technology, which discovery would cause high damage, alternative research strategies should be considered. In summary, while issues like climate change already receive considerable attention, we should become more aware of other dangers of technological development and create policies accordingly.

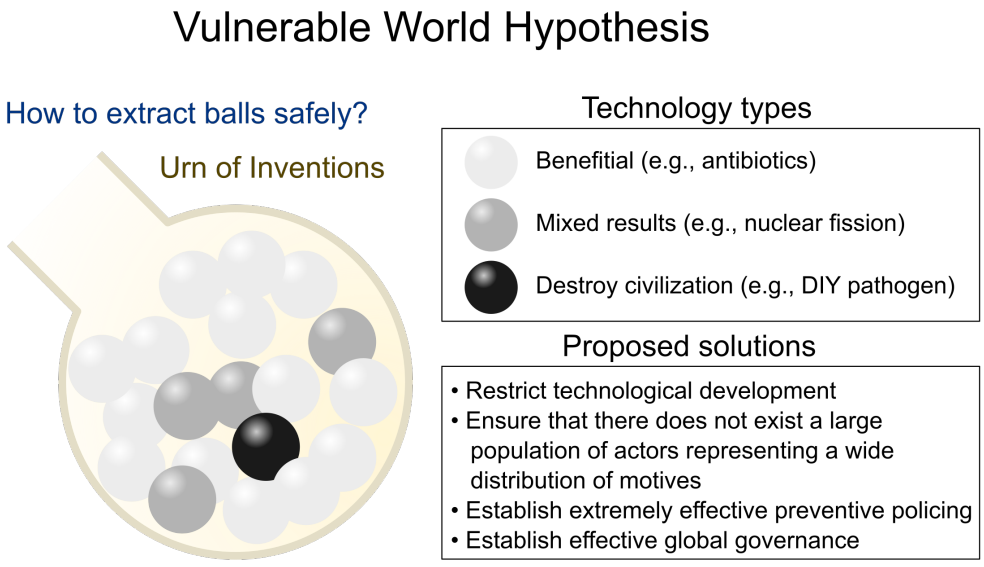


Figure ii: The Vulnerable World Hypothesis illustrated through the Urn of Inventions metaphor.[26]

Bibliography

[1] Abdel-Mawgoud, A. M., Markham, K. A., Palmer, C. M., Liu, N., Stephanopoulos, G., and Alper, H. S. (2018). Metabolic engineering in the host yarrowia lipolytica. *Metabolic engineering*, 50:192–208.

[2] Abelson, H., Sussman, G. J., and Sussman, J. (1996). *Structure and interpretation of computer programs*. Justin Kelly.

[3] Ajikumar, P. K., Xiao, W.-H., Tyo, K. E., Wang, Y., Simeon, F., Leonard, E., Mucha, O., Phon, T. H., Pfeifer, B., and Stephanopoulos, G. (2010). Isoprenoid pathway optimization for taxol precursor overproduction in escherichia coli. *Science*, 330(6000):70–74.

[4] Akinosho, H., Yee, K., Close, D., and Ragauskas, A. (2014). The emergence of clostridium thermocellum as a high utility candidate for consolidated bioprocessing applications. *Frontiers in chemistry*, 2:66.

[5] Akita, H., Nakashima, N., and Hoshino, T. (2016). Pyruvate production using engineered escherichia coli. *AMB Express*, 6(1):94.

[6] Alba, E., Luque, G., and Nesmachnow, S. (2013). Parallel metaheuristics: recent advances and new trends. *International Transactions in Operational Research*, 20(1):1–48.

[7] Alon, U. (2003). Biological networks: the tinkerer as an engineer. *Science*, 301(5641):1866–1867.

[8] Annaluru, N., Muller, H., Mitchell, L. A., Ramalingam, S., Stracquadanio, G., Richardson, S. M., Dymond, J. S., Kuang, Z., Scheifele, L. Z., and Cooper, E. M. (2014). Total synthesis of a functional designer eukaryotic chromosome. *Science*, 344(6179):55–58.

[9] Argyros, D. A., Tripathi, S. A., Barrett, T. F., Rogers, S. R., Feinberg, L. F., Olson, D. G., Foden, J. M., Miller, B. B., Lynd, L. R., Hogsett, D. A., et al. (2011). High ethanol titers from cellulose by using metabolically engineered thermophilic, anaerobic microbes. *Appl. Environ. Microbiol.*, 77(23):8288–8294.

[10] Arkin, A. P., Cottingham, R. W., Henry, C. S., Harris, N. L., Stevens, R. L., Maslov, S., Dehal, P., Ware, D., Perez, F., Canon, S., et al. (2018). Kbase: the united states department of energy systems biology knowledgebase. *Nature biotechnology*, 36(7).

[11] Atsumi, S., Hanai, T., and Liao, J. C. (2008). Non-fermentative pathways for synthesis of branched-chain higher alcohols as biofuels. *Nature*, 451(7174):86.

[12] Atsumi, S., Wu, T.-Y., Eckl, E.-M., Hawkins, S. D., Buelter, T., and Liao, J. C. (2010). Engineering the isobutanol biosynthetic pathway in escherichia coli by comparison of three aldehyde reductase/alcohol dehydrogenase genes. *Applied microbiology and biotechnology*, 85(3):651–657.

[13] Baldea, M., Edgar, T. F., Stanley, B. L., and Kiss, A. A. (2017). Modular manufacturing processes: Status, challenges and opportunities. *AIChE journal*, 63(10):4262–4272.

[14] Barnett, J. and Adger, W. N. (2007). Climate change, human security and violent conflict. *Political geography*, 26(6):639–655.

[15] Barrangou, R. and Doudna, J. A. (2016). Applications of crispr technologies in research and beyond. *Nature biotechnology*, 34(9):933.

[16] Bennett, G., Gilman, N., Stavrianakis, A., and Rabinow, P. (2009). From synthetic biology to biohacking: are we prepared? *Nature Biotechnology*, 27(12):1109–1111.

[17] Biggs, B. W., De Paepe, B., Santos, C. N. S., De Mey, M., and Ajikumar, P. K. (2014). Multivariate modular metabolic engineering for pathway and strain optimization. *Current opinion in biotechnology*, 29:156–162.

[18] Biswas, R., Wilson, C. M., Giannone, R. J., Klingeman, D. M., Rydzak, T., Shah, M. B., Hettich, R. L., Brown, S. D., and Guss, A. M. (2017). Improved growth rate in clostridium thermocellum hydrogenase mutant via perturbed sulfur metabolism. *Biotechnology for biofuels*, 10(1):6.

[19] Biswas, R., Zheng, T., Olson, D. G., Lynd, L. R., and Guss, A. M. (2015). Elimination of hydrogenase active site assembly blocks h 2 production and increases ethanol yield in clostridium thermocellum. *Biotechnology for biofuels*, 8(1):20.

[20] Bitinaite, J., Rubino, M., Varma, K. H., Schildkraut, I., Vaisvila, R., and Vaiskunaite, R. (2007). User^{TM} friendly dna engineering and cloning method by uracil excision. *Nucleic Acids Research*, 35(6):1992–2002.

[21] Blake, W. J., Chapman, B. A., Zindal, A., Lee, M. E., Lippow, S. M., and Baynes, B. M. (2010). Pairwise selection assembly for sequence-independent construction of long-length dna. *Nucleic Acids Research*, 38(8):2594–2602.

[22] Blazeck, J. and Alper, H. (2010). Systems metabolic engineering: Genome-scale models and beyond. *Biotechnology journal*, 5(7):647–659.

[23] Boecker, S., Zahoor, A., Schramm, T., Link, H., and Klamt, S. (2019). Broadening the scope of enforced atp wasting as a tool for metabolic engineering in escherichia coli. *Biotechnology journal*, 14(9):1800438.

[24] Bonvoisin, J., Halstenberg, F., Buchert, T., and Stark, R. (2016). A systematic literature review on modular product design. *Journal of Engineering Design*, 27(7):488–514.

[25] Bordel, S., Agren, R., and Nielsen, J. (2010). Sampling the solution space in genome-scale metabolic networks reveals transcriptional regulation in key enzymes. *PLOS Computational Biology*, 6(7):1–13.

[26] Bostrom, N. (2019). The vulnerable world hypothesis. *Global Policy*, 10(4):455–476.

[27] Brodeur, G., Yau, E., Badal, K., Collier, J., Ramachandran, K., and Ramakrishnan, S. (2011). Chemical and physicochemical pretreatment of lignocellulosic biomass: a review. *Enzyme research*, 2011.

[28] Brophy, J. A. and Voigt, C. A. (2014). Principles of genetic circuit design. *Nature methods*, 11(5):508.

[29] Browning, T. R. (2016). Design structure matrix extensions and innovations: a survey and new opportunities. *IEEE Transactions on Engineering Management*, 63(1):27–52.

[30] Burgard, A. P., Pharkya, P., and Maranas, C. D. (2003). Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnology and bioengineering*, 84(6):647–657.

[31] Callura, J. M., Cantor, C. R., and Collins, J. J. (2012). Genetic switchboard for synthetic biology applications. *Proceedings of the National Academy of Sciences*, 109(15):5850–5855.

[32] Calusinska, M., Happe, T., Joris, B., and Wilmotte, A. (2010). The surprising diversity of clostridial hydrogenases: a comparative genomic perspective. *Microbiology*, 156:1575–1588.

[33] Campagnolo, D. and Camuffo, A. (2010). The concept of modularity in management studies: a literature review. *International journal of management reviews*, 12(3):259–283.

[34] Carroll, A. L., Case, A. E., Zhang, A., and Atsumi, S. (2018). Metabolic engineering tools in model cyanobacteria. *Metabolic engineering*, 50:47–56.

[35] Casini, A., Storch, M., Baldwin, G. S., and Ellis, T. (2015). Bricks and blueprints: methods and standards for dna assembly. *Nature Reviews Molecular Cell Biology*, 16(9):568.

[36] Caspi, R., Altman, T., Billington, R., Dreher, K., Foerster, H., Fulcher, C. A., Holland, T. A., Keseler, I. M., Kothari, A., and Kubo, A. (2013). The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic acids research*, 42(D1):D459–D471.

[37] Causey, T., Shanmugam, K., Yomano, L., and Ingram, L. (2004). Engineering escherichia coli for efficient conversion of glucose to pyruvate. *Proceedings of the National Academy of Sciences*, 101(8):2235–2240.

[38] Chan, S. H., Wang, L., Dash, S., and Maranas, C. D. (2018). Accelerating flux balance calculations in genome-scale metabolic models by localizing the application of loopless constraints. *Bioinformatics*, 34(24):4248–4255.

[39] Chatr-Aryamontri, A., Oughtred, R., Boucher, L., Rust, J., Chang, C., Kolas, N. K., O'Donnell, L., Oster, S., Theesfeld, C., and Sellam, A. (2017). The biogrid interaction database: 2017 update. *Nucleic acids research*, 45(D1):D369–D379.

[40] Chemler, J. A., Fowler, Z. L., McHugh, K. P., and Koffas, M. A. (2010). Improving nadph availability for natural product biosynthesis in escherichia coli by metabolic engineering. *Metabolic Engineering*, 12(2):96 – 104. Metabolic Flux Analysis for Pharmaceutical Production Special Issue.

[41] Chen, W.-H., Qin, Z.-J., Wang, J., and Zhao, G.-P. (2013). The master (methylation-assisted tailorable ends rational) ligation method for seamless dna assembly. *Nucleic Acids Research*, 41(8):e93–e93.

[42] Cheng, J., Yen, G. G., and Zhang, G. (2015). A many-objective evolutionary algorithm with enhanced mating and environmental selections. *IEEE Transactions on Evolutionary Computation*, 19(4):592–605.

[43] Cheong, S., Clomburg, J. M., and Gonzalez, R. (2016). Energy-and carbon-efficient synthesis of functionalized small molecules in bacteria using non-decarboxylative claisen condensation reactions. *Nature biotechnology*, 34:556–561.

[44] Chowdhury, A., Zomorrodi, A. R., and Maranas, C. D. (2014). k-optforce: integrating kinetics with flux balance analysis for strain design. *PLoS computational biology*, 10(2):e1003487.

[45] Chowdhury, A., Zomorrodi, A. R., and Maranas, C. D. (2015). Bilevel optimization techniques in computational strain design. *Computers & Chemical Engineering*, 72:363–372.

[46] Christodoulou, D., Link, H., Fuhrer, T., Kochanowski, K., Gerosa, L., and Sauer, U. (2018). Reserve flux capacity in the pentose phosphate pathway enables escherichia coli's rapid response to oxidative stress. *Cell systems*, 6(5):569–578.

[47] Clune, J., Mouret, J.-B., and Lipson, H. (2013). The evolutionary origins of modularity. *Proc. R. Soc. B*, 280(1755):20122863.

[48] Coello, C. A. C. and Lamont, G. B. (2004). *Applications of multi-objective evolutionary algorithms*, volume 1. World Scientific, Singapore.

[49] Coello, C. A. C., Lamont, G. B., Van Veldhuizen, D. A., et al. (2007). *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer.

[50] Coello Coello, C. A. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11):1245–1287.

[51] Colloms, S. D., Merrick, C. A., Olorunniji, F. J., Stark, W. M., Smith, M. C., Osbourn, A., Keasling, J. D., and Rosser, S. J. (2014). Rapid metabolic pathway assembly and modification using serine integrase site-specific recombination. *Nucleic Acids Research*, 42(4):e23–e23.

[52] Coomes, M. W., Mitchell, B. K., Beezley, A., and Smith, T. E. (1985). Properties of an escherichia coli mutant deficient in phosphoenolpyruvate carboxylase catalytic activity. *Journal of bacteriology*, 164(2):646–652.

[53] Council, N. R. et al. (2015). *Industrialization of biology: a roadmap to accelerate the advanced manufacturing of chemicals*. National Academies Press.

[54] Cramer, S. M. and Holstein, M. A. (2011). Downstream bioprocessing: recent advances and future promise. *Current Opinion in Chemical Engineering*, 1(1):27–37.

[55] Dash, S., Khodayari, A., Zhou, J., Holwerda, E. K., Olson, D. G., Lynd, L. R., and Maranas, C. D. (2017). Development of a core clostridium thermocellum kinetic metabolic model consistent with multiple genetic perturbations. *Biotechnology for biofuels*, 10(1):108.

[56] Dash, S., Mueller, T. J., Venkataramanan, K. P., Papoutsakis, E. T., and Maranas, C. D. (2014). Capturing the response of clostridium acetobutylicum to chemical stressors using a regulated genome-scale metabolic model. *Biotechnology for biofuels*, 7(1):144.

[57] Dash, S., Ng, C. Y., and Maranas, C. D. (2016). Metabolic modeling of clostridia: current developments and applications. *FEMS microbiology letters*, 363(4).

[58] Dash, S., Olson, D. G., Chan, S. H. J., Amador-Noguez, D., Lynd, L. R., and Maranas, C. D. (2019). Thermodynamic analysis of the pathway for ethanol production from cellobiose in clostridium thermocellum. *Metabolic Engineering*, 55:161–169.

[59] De Maeseneire, S., De Mey, M., Vandedrinck, S., and Vandamme, E. (2006). Metabolic characterisation of e. coli citrate synthase and phosphoenolpyruvate carboxylase mutants in aerobic cultures. *Biotechnology letters*, 28(23):1945–1953.

[60] Deb, K. and Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601.

[61] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002a). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197.

[62] Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2002b). Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation.*, volume 1, pages 825–830. IEEE.

[63] Del Vecchio, D., Ninfa, A. J., and Sontag, E. D. (2008). Modular cell biology: retroactivity and insulation. *Molecular systems biology*, 4(1):161.

[64] Deng, Y., Olson, D. G., Zhou, J., Herring, C. D., Shaw, A. J., and Lynd, L. R. (2013). Redirecting carbon flux through exogenous pyruvate kinase to achieve high ethanol yields in clostridium thermocellum. *Metabolic engineering*, 15:151–158.

[65] Dinh, H. V., King, Z. A., Palsson, B. O., and Feist, A. M. (2018). Identification of growth-coupled production strains considering protein costs and kinetic variability. *Metabolic engineering communications*, 7:e00080.

[66] DoD, U. (2015). National security implications of climate-related risks and a changing climate. *US Department of Defense, Washington, DC*.

[67] Dragosits, M. and Mattanovich, D. (2013). Adaptive laboratory evolution–principles and applications for biotechnology. *Microbial cell factories*, 12(1):64.

[68] Dugar, D. and Stephanopoulos, G. (2011). Relative potential of biosynthetic pathways for biofuels and bio-based products. *Nature Biotechnology*, 29(12):1074–1078.

[69] Dynan, W. S. (1989). Modularity in promoters and enhancers. *Cell*, 58(1):1–4.

[70] Ebrahim, A., Brunk, E., Tan, J., O'brien, E. J., Kim, D., Szubin, R., Lerman, J. A., Lechner, A., Sastry, A., and Bordbar, A. (2016). Multi-omic data integration enables discovery of hidden biological regularities. *Nature communications*, 7:13091.

[71] Ebrahim, A., Lerman, J. A., Palsson, B. O., and Hyduke, D. R. (2013). Cobrapy: constraints-based reconstruction and analysis for python. *BMC systems biology*, 7(1):74.

[72] Farasat, I., Kushwaha, M., Collens, J., Easterbrook, M., Guido, M., and Salis, H. M. (2014). Efficient search, mapping, and optimization of multi-protein genetic systems in diverse bacteria. *Molecular systems biology*, 10(6):731.

[73] Feist, A. M. and Palsson, B. Ø. (2008). The growing scope of applications of genome-scale metabolic reconstructions using escherichia coli. *Nature biotechnology*, 26(6):659.

[74] Feist, A. M., Zielinski, D. C., Orth, J. D., Schellenberger, J., Herrgard, M. J., and Palsson, B. Ø. (2010). Model-driven evaluation of the production potential for growth-coupled products of Escherichia coli. *Metabolic engineering*, 12(3):173–186.

[75] Fell, D. A. and Small, J. R. (1986). Fat synthesis in adipose tissue. an examination of stoichiometric constraints. *Biochemical Journal*, 238(3):781–786.

[76] Fischetti, M., Ljubić, I., and Sinnl, M. (2016). Benders decomposition without separability: A computational study for capacitated facility location problems. *European Journal of Operational Research*, 253(3):557–569.

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

[78] Fong, S. S., Nanchen, A., Palsson, B. O., and Sauer, U. (2006). Latent pathway activation and increased pathway capacity enable escherichia coli adaptation to loss of key metabolic enzymes. *Journal of Biological Chemistry*, 281(12):8024–8033.

[79] Fonseca, C. M., Paquete, L., and López-Ibánez, M. (2006). An improved dimension-sweep algorithm for the hypervolume indicator. In *IEEE international conference on evolutionary computation*, pages 1157–1163. IEEE.

[80] Friedlander, T., Mayo, A. E., Tlusty, T., and Alon, U. (2015). Evolution of bow-tie architectures in biology. *PLoS computational biology*, 11(3):e1004055.

[81] Galanie, S., Thodey, K., Trenchard, I. J., Interrante, M. F., and Smolke, C. D. (2015). Complete biosynthesis of opioids in yeast. *Science*, 349(6252):1095–1100.

[82] Gancarz, M. (2003). *Linux and the Unix philosophy*. Digital Press.

[83] Garcia, S. and Trinh, C. T. (2019a). Comparison of multi-objective evolutionary algorithms to solve the modular cell design problem for novel biocatalysis. *Processes*, 7(6).

[84] Garcia, S. and Trinh, C. T. (2019b). Harnessing natural modularity of cellular metabolism to design a modular chassis cell for a diverse class of products by using goal attainment optimization. *bioRxiv*.

[85] Garcia, S. and Trinh, C. T. (2019c). Modular design: Implementing proven engineering principles in biotechnology. *Biotechnology Advances*, 37(7):107403.

[86] Garcia, S. and Trinh, C. T. (2019d). Multiobjective strain design: A framework for modular cell engineering. *Metabolic Engineering*, 51.

[87] Garcia, S. and Trinh, C. T. (2020). Top-down design of microbial catalysis platforms for large libraries of product synthesis modules. *Under preparation*.

[88] García-Sánchez, P., Ortega, J., González, J., Castillo, P. A., and Merelo, J. J. (2016). Addressing high dimensional multi-objective optimization problems by coevolutionary islands with overlapping search spaces. In *European Conference on the Applications of Evolutionary Computation*, pages 107–117. Springer.

[89] Garst, A. D., Bassalo, M. C., Pines, G., Lynch, S. A., Halweg-Edwards, A. L., Liu, R., Liang, L., Wang, Z., Zeitoun, R., Alexander, W. G., et al. (2017). Genome-wide mapping of mutations at single-nucleotide resolution for protein, metabolic and genome engineering. *Nature biotechnology*, 35(1):48.

[90] Geoffrion, A. M. (1972). Generalized benders decomposition. *Journal of optimization theory and applications*, 10(4):237–260.

[91] Giannone, R. J., Wurch, L. L., Heimerl, T., Martin, S., Yang, Z., Huber, H., Rachel, R., Hettich, R. L., and Podar, M. (2015a). Life on the edge: functional genomic response of ignicoccus hospitalis to the presence of nanoarchaeum equitans. *The ISME journal*, 9(1):101.

[92] Giannone, R. J., Wurch, L. L., Podar, M., and Hettich, R. L. (2015b). Rescuing those left behind: recovering and characterizing underdigested membrane and hydrophobic proteins to enhance proteome measurement depth. *Analytical chemistry*, 87(15):7720–7728.

[93] Gibson, D., Young, L., Chuang, R., Venter, J., Hutchison, C., and Smith, H. (2009). Enzymatic assembly of dna molecules up to several hundred kilobases. *Nat Methods*, 6:343 – 345.

[94] Gilarranz, L. J., Rayfield, B., Liñán-Cembrano, G., Bascompte, J., and Gonzalez, A. (2017). Effects of network modularity on the spread of perturbation impact in experimental metapopulations. *Science*, 357(6347):199–201.

[95] Goh, K.-I., Cusick, M. E., Valle, D., Childs, B., Vidal, M., and Barabási, A.-L. (2007). The human disease network. *Proceedings of the National Academy of Sciences*, 104(21):8685–8690.

[96] Goodwin, S., McPherson, J. D., and McCombie, W. R. (2016). Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6):333.

[97] Grilli, J., Rogers, T., and Allesina, S. (2016). Modularity and stability in ecological communities. *Nature communications*, 7:12031.

[98] Guruharsha, K., Rual, J.-F., Zhai, B., Mintseris, J., Vaidya, P., Vaidya, N., Beekman, C., Wong, C., Rhee, D. Y., and Cenaj, O. (2011). A protein complex network of drosophila melanogaster. *Cell*, 147(3):690–703.

[99] Hädicke, O., Bettenbrock, K., and Klamt, S. (2015). Enforced atp futile cycling increases specific productivity and yield of anaerobic lactate production in escherichia coli. *Biotechnology and bioengineering*, 112(10):2195–2199.

[100] Hart, W. E., Laird, C. D., Watson, J.-P., Woodruff, D. L., Hackebeil, G. A., Nicholson, B. L., and Siirola, J. D. (2012). *Pyomo-optimization modeling in python*, volume 67. Springer.

[101] Hart, W. E., Laird, C. D., Watson, J.-P., Woodruff, D. L., Hackebeil, G. A., Nicholson, B. L., and Siirola, J. D. (2017). *Pyomo — Optimization Modeling in Python*, volume 67 of *Springer Optimization and Its Applications*. Springer International Publishing, Cham.

[102] Hartwell, L. H., Hopfield, J. J., Leibler, S., and Murray, A. W. (1999). From molecular to modular cell biology. *Nature*, 402(6761supp):C47.

[103] Havugimana, P. C., Hart, G. T., Nepusz, T., Yang, H., Turinsky, A. L., Li, Z., Wang, P. I., Boutz, D. R., Fong, V., and Phanse, S. (2012). A census of human soluble protein complexes. *Cell*, 150(5):1068–1081.

[104] Heckmann, D., Lloyd, C. J., Mih, N., Ha, Y., Zielinski, D. C., Haiman, Z. B., Desouki, A. A., Lercher, M. J., and Palsson, B. O. (2018). Machine learning applied to enzyme turnover numbers reveals protein structural correlates and improves metabolic models. *Nature Communications*, 9(1):5252.

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

[106] Helmer, R., Yassine, A., and Meier, C. (2010). Systematic module and interface definition using component design structure matrix. *Journal of Engineering Design*, 21(6):647–675.

[107] Henry, C. S., DeJongh, M., Best, A. A., Frybarger, P. M., Linsay, B., and Stevens, R. L. (2010). High-throughput generation, optimization and analysis of genome-scale metabolic models. *Nature biotechnology*, 28(9):977.

[108] Hijaze, M. and Corne, D. (2009). An investigation of topologies and migration schemes for asynchronous distributed evolutionary algorithms. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pages 636–641. IEEE.

[109] Hill, M. D. and Marty, M. R. (2008). Amdahl's law in the multicore era. *Computer*, 41(7):33–38.

[110] Holwerda, E. K., Thorne, P. G., Olson, D. G., Amador-Noguez, D., Engle, N. L., Tschaplinski, T. J., van Dijken, J. P., and Lynd, L. R. (2014). The exometabolome of clostridium thermocellum reveals overflow metabolism at high cellulose loading. *Biotechnology for biofuels*, 7(1):155.

[111] Hsiang, S. M., Meng, K. C., and Cane, M. A. (2011). Civil conflicts are associated with the global climate. *Nature*, 476(7361):438.

[112] Hutchinson, C. R. (2003). Polyketide and non-ribosomal peptide synthases: falling together by coming apart. *Proceedings of the National Academy of Sciences*, 100(6):3010–3012.

[113] Hutchison, C. A., Chuang, R.-Y., Noskov, V. N., Assad-Garcia, N., Deerinck, T. J., Ellisman, M. H., Gill, J., Kannan, K., Karas, B. J., and Ma, L. (2016). Design and synthesis of a minimal bacterial genome. *Science*, 351(6280):aad6253.

[114] Hölttä-Otto, K. and De Weck, O. (2007). Degree of modularity in engineering systems and products with technical and business constraints. *Concurrent Engineering*, 15(2):113–126.

[115] Ishibuchi, H., Sakane, Y., Tsukamoto, N., and Nojima, Y. (2009). Evolutionary many-objective optimization by nsga-ii and moea/d with large populations. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 1758–1763. IEEE.

[116] Ishibuchi, H., Tsukamoto, N., and Nojima, Y. (2008). Evolutionary many-objective optimization: A short review. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 2419–2426. IEEE.

[117] Ishii, N., Nakahigashi, K., Baba, T., Robert, M., Soga, T., Kanai, A., Hirasawa, T., Naba, M., Hirai, K., Hoque, A., et al. (2007). Multiple high-throughput analyses monitor the response of e. coli to perturbations. *Science*, 316(5824):593–597.

[118] Jeschek, M., Gerngross, D., and Panke, S. (2017). Combinatorial pathway optimization for streamlined metabolic engineering. *Current opinion in biotechnology*, 47:142–151.

[119] Jiang, S. and Yang, S. (2017). A strength pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 21(3):329–346.

[120] Jose, A. and Tollenaere, M. (2005). Modular and platform methods for product family design: literature analysis. *Journal of Intelligent manufacturing*, 16(3):371–390.

[121] Jouhten, P., Boruta, T., Andrejev, S., Pereira, F., Rocha, I., and Patil, K. R. (2016). Yeast metabolic chassis designs for diverse biotechnological products. *Scientific reports*, 6.

[122] Jozefowiez, N., Semet, F., and Talbi, E.-G. (2005). Enhancements of nsga ii and its application to the vehicle routing problem with route balancing. In *International Conference on Artificial Evolution (Evolution Artificielle)*, pages 131–142. Springer.

[123] Kabir, M. M., Ho, P. Y., and Shimizu, K. (2005). Effect of ldha gene deletion on the metabolism of escherichia coli based on gene expression, enzyme activities, intracellular metabolite concentrations, and metabolic flux distribution. *Biochemical Engineering Journal*, 26(1):1–11.

[124] Kahl, L. J. and Endy, D. (2013). A survey of enabling technologies in synthetic biology. *Journal of biological engineering*, 7(1):13.

[125] Kalyanmoy, D. (2001). *Multi objective optimization using evolutionary algorithms*. John Wiley and Sons, Chichester, England.

[126] Kalyuzhnaya, M. G., Puri, A. W., and Lidstrom, M. E. (2015). Metabolic engineering in methanotrophic bacteria. *Metabolic engineering*, 29:142–152.

[127] Kamali, M. and Hewage, K. (2016). Life cycle performance of modular buildings: A critical review. *Renewable and Sustainable Energy Reviews*, 62:1171–1183.

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

[129] Kashtan, N., Noor, E., and Alon, U. (2007). Varying environments can speed up evolution. *Proceedings of the National Academy of Sciences*, 104(34):13711–13716.

[130] Kaufman, D. E. and Smith, R. L. (1998). Direction choice for accelerated convergence in hit-and-run sampling. *Operations Research*, 46(1):84–95.

[131] Khodayari, A. and Maranas, C. D. (2016). A genome-scale escherichia coli kinetic metabolic model k-ecoli457 satisfying flux data for multiple mutant strains. *Nature Communications*, 7.

[132] Khosla, C. and Harbury, P. B. (2001). Modular enzymes. *Nature*, 409(6817):247.

[133] Kim, P., Laivenieks, M., Vieille, C., and Zeikus, J. G. (2004). Effect of overexpression of actinobacillus succinogenes phosphoenolpyruvate carboxykinase on succinate production in escherichia coli. *Appl. Environ. Microbiol.*, 70(2):1238–1241.

[134] Kim, Y.-h., Park, L. K., Yiacoumi, S., and Tsouris, C. (2017). Modular chemical process intensification: a review. *Annual review of chemical and biomolecular engineering*, 8:359–380.

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

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

[137] King, Z. A., O'Brien, E. J., Feist, A. M., and Palsson, B. O. (2017). Literature mining supports a next-generation modeling approach to predict cellular byproduct secretion. *Metabolic Engineering*, 39:220–227.

[138] Kitano, H. (2004). Biological robustness. *Nature Reviews Genetics*, 5(11):826.

[139] Klamt, S. and Mahadevan, R. (2015). On the feasibility of growth-coupled product synthesis in microbial strains. *Metabolic engineering*, 30:166–178.

[140] Klamt, S., Mahadevan, R., and Hädicke, O. (2018). When do two-stage processes outperform one-stage processes? *Biotechnology journal*, 13(2):1700539.

[141] Kok, S. d., Stanton, L. H., Slaby, T., Durot, M., Holmes, V. F., Patel, K. G., Platt, D., Shapland, E. B., Serber, Z., and Dean, J. (2014). Rapid and reliable dna assembly via ligase cycling reaction. *ACS Synthetic Biology*, 3(2):97–106.

[142] Kosuri, S. and Church, G. M. (2014). Large-scale de novo dna synthesis: technologies and applications. *Nature methods*, 11(5):499.

[143] Kridelbaugh, D. M., Nelson, J., Engle, N. L., Tschaplinski, T. J., and Graham, D. E. (2013). Nitrogen and sulfur requirements for clostridium thermocellum and caldicellulosiruptor bescii on cellulosic substrates in minimal nutrient media. *Bioresource technology*, 130:125–135.

[144] Kumar, A., Wang, L., Ng, C. Y., and Maranas, C. D. (2018). Pathway design using de novo steps through uncharted biochemical spaces. *Nature communications*, 9(1):184.

[145] Larhlimi, A., David, L., Selbig, J., and Bockmayr, A. (2012). F2c2: a fast tool for the computation of flux coupling in genome-scale metabolic networks. *BMC Bioinformatics*, 13:57.

[146] Layton, D. S. and Trinh, C. T. (2014). Engineering modular ester fermentative pathways in escherichia coli. *Metabolic Engineering*, 26:77–88.

[147] Layton, D. S. and Trinh, C. T. (2016a). Expanding the modular ester fermentative pathways for combinatorial biosynthesis of esters from volatile organic acids. *Biotechnology and bioengineering*.

[148] Layton, D. S. and Trinh, C. T. (2016b). Microbial synthesis of a branched-chain ester platform from organic waste carboxylates. *Metabolic Engineering Communications*, 3:245–251.

[149] Lee, J. and Trinh, C. T. (2018). De novo microbial biosynthesis of a lactate ester platform. *bioRxiv*, page 498576.

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

[151] Lee, W.-H., Kim, M.-D., Jin, Y.-S., and Seo, J.-H. (2013). Engineering of nadph regenerators in escherichia coli for enhanced biotransformation. *Applied microbiology and biotechnology*, 97(7):2761–2772.

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

[153] Li, B., Li, J., Tang, K., and Yao, X. (2015a). Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)*, 48(1):13.

[154] Li, K., Wang, R., Zhang, T., and Ishibuchi, H. (2018). Evolutionary many-objective optimization: A comparative study of the state-of-the-art. *IEEE Access*, 6:26194–26214.

[155] Li, M. and Elledge, S. (2007). Harnessing homologous recombination in vitro to generate recombinant dna via slic. *Nat Methods*, 4(3):251 – 6.

[156] Li, M., Yang, S., and Liu, X. (2014). Shift-based density estimation for pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 18(3):348–365.

[157] Li, M., Yang, S., and Liu, X. (2015b). Bi-goal evolution for many-objective optimization problems. *Artificial Intelligence*, 228:45–65.

[158] Li, M. Z. and Elledge, S. J. (2005). Magic, an in vivo genetic method for the rapid construction of recombinant dna molecules. *Nature Genetics*, 37(3):311–319.

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

[160] Lim, W. A. (2010). Designing customized cell signalling circuits. *Nature reviews Molecular cell biology*, 11(6):393.

[161] Lin, H., Vadali, R. V., Bennett, G. N., and San, K.-Y. (2004). Increasing the acetyl-coa pool in the presence of overexpressed phosphoenolpyruvate carboxylase or pyruvate carboxylase enhances succinate production in escherichia coli. *Biotechnology progress*, 20(5):1599–1604.

[162] Lin, P. P., Mi, L., Morioka, A. H., Yoshino, K. M., Konishi, S., Xu, S. C., Papanek, B. A., Riley, L. A., Guss, A. M., and Liao, J. C. (2015). Consolidated bioprocessing of cellulose to isobutanol using clostridium thermocellum. *Metabolic engineering*, 31:44–52.

[163] Liu, D., Evans, T., and Zhang, F. (2015). Applications and advances of metabolite biosensors for metabolic engineering. *Metabolic engineering*, 31:35–43.

[164] Lo, J., Olson, D. G., Murphy, S. J.-L., Tian, L., Hon, S., Lanahan, A., Guss, A. M., and Lynd, L. R. (2017). Engineering electron metabolism to increase ethanol production in clostridium thermocellum. *Metabolic engineering*, 39:71–79.

[165] Loder, A. J., Zeldes, B. M., Garrison, G. D., Lipscomb, G. L., Adams, M. W., and Kelly, R. M. (2015). Alcohol selectivity in a synthetic thermophilic n-butanol pathway is driven by biocatalytic and thermostability characteristics of constituent enzymes. *Appl. Environ. Microbiol.*, 81(20):7187–7200.

[166] Long, M. R., Ong, W. K., and Reed, J. L. (2015). Computational methods in metabolic engineering for strain design. *Current opinion in biotechnology*, 34:135–141.

[167] Lowe, R., Shirley, N., Bleackley, M., Dolan, S., and Shafee, T. (2017). Transcriptomics technologies. *PLoS computational biology*, 13(5):e1005457.

[168] Lu, H., Li, F., Sánchez, B. J., Zhu, Z., Li, G., Domenzain, I., Marcišauskas, S., Anton, P. M., Lappa, D., Lieven, C., et al. (2019). A consensus s. cerevisiae metabolic model yeast8 and its ecosystem for comprehensively probing cellular metabolism. *Nature communications*, 10(1):1–13.

[169] Lu, H., Villada, J. C., and Lee, P. K. (2018). Modular metabolic engineering for biobased chemical production. *Trends in biotechnology*.

[170] Lynd, L. R., Guss, A. M., Himmel, M. E., Beri, D., Herring, C., Holwerda, E. K., Murphy, S. J., Olson, D. G., Paye, J., and Rydzak, T. (2016). Advances in consolidated bioprocessing using clostridium thermocellum and thermoanaerobacter saccharolyticum. *Industrial Biotechnology: Microorganisms*, pages 365–394.

[171] Ma, H.-W. and Zeng, A.-P. (2003). The connectivity structure, giant strong component and centrality of metabolic networks. *Bioinformatics*, 19(11):1423–1430.

[172] Ma, Z.-Q., Dasari, S., Chambers, M. C., Litton, M. D., Sobecki, S. M., Zimmerman, L. J., Halvey, P. J., Schilling, B., Drake, P. M., Gibson, B. W., et al. (2009). Idpicker 2.0: Improved protein assembly with high discrimination peptide identification filtering. *Journal of proteome research*, 8(8):3872–3881.

[173] Machado, D. and Herrgård, M. (2014). Systematic evaluation of methods for integration of transcriptomic data into constraint-based models of metabolism. *PLoS Comput Biol*, 10(4):e1003580.

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

[175] Maervoet, V. E. and Briers, Y. (2017). Synthetic biology of modular proteins. *Bioengineered*, 8(3):196–202.

[176] Maranas, C. D. and Zomorrodi, A. R. (2016). *Optimization Methods in Metabolic Networks*. John Wiley & Sons, Hoboken, New Jersey.

[177] Marler, R. T. and Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395.

[178] Märtens, M. and Izzo, D. (2013). The asynchronous island model and nsga-ii: study of a new migration operator and its performance. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 1173–1180.

[179] Martin, V. J., Pitera, D. J., Withers, S. T., Newman, J. D., and Keasling, J. D. (2003). Engineering a mevalonate pathway in escherichia coli for production of terpenoids. *Nature biotechnology*, 21(7):796–802.

[180] Mathworks (2019). Matlab documentation gamultiobj algorithm. `https://www.mathworks.com/help/gads/gamultiobj-algorithm.html`. Accessed: 2019-02-04.

[181] McAnulty, M. J., Yen, J. Y., Freedman, B. G., and Senger, R. S. (2012). Genome-scale modeling using flux ratio constraints to enable metabolic engineering of clostridial metabolism in silico. *BMC systems biology*, 6(1):42.

[182] Meng, H., Wang, J., Xiong, Z., Xu, F., Zhao, G., and Wang, Y. (2013). Quantitative design of regulatory elements based on high-precision strength prediction using artificial neural network. *PLoS One*, 8(4):e60288.

[183] Meunier, D., Lambiotte, R., and Bullmore, E. T. (2010). Modular and hierarchically modular organization of brain networks. *Frontiers in neuroscience*, 4:200.

[184] Meyer, A. J., Segall-Shapiro, T. H., Glassey, E., Zhang, J., and Voigt, C. A. (2018). Escherichia coli "marionette" strains with 12 highly optimized small-molecule sensors. *Nature chemical biology*, page 1.

[185] Miller, T. D. and Elgard, P. (1998). Defining modules, modularity and modularization. In *Proceedings of the 13th IPS research seminar, Fuglsoe*. Aalborg Universiy.

[186] Milne, C. B., Eddy, J. A., Raju, R., Ardekani, S., Kim, P.-J., Senger, R. S., Jin, Y.-S., Blaschek, H. P., and Price, N. D. (2011). Metabolic network reconstruction and genome-scale model of butanol-producing strain clostridium beijerinckii ncimb 8052. *BMC systems biology*, 5(1):130.

[187] Mitra, K., Carvunis, A.-R., Ramesh, S. K., and Ideker, T. (2013). Integrative approaches for finding modular structure in biological networks. *Nature Reviews Genetics*, 14(10):719.

[188] Monk, J. M., Lloyd, C. J., Brunk, E., Mih, N., Sastry, A., King, Z., Takeuchi, R., Nomura, W., Zhang, Z., Mori, H., et al. (2017). iml1515, a knowledgebase that computes escherichia coli traits. *Nature biotechnology*, 35(10):904.

[189] Moser, F., Borujeni, A. E., Ghodasara, A. N., Cameron, E., Park, Y., and Voigt, C. A. (2018). Dynamic control of endogenous metabolism with combinatorial logic circuits. *Molecular systems biology*, 14(11):e8605.

[190] Nagarajan, H., Sahin, M., Nogales, J., Latif, H., Lovley, D. R., Ebrahim, A., and Zengler, K. (2013). Characterizing acetogenic metabolism using a genome-scale metabolic reconstruction of clostridium ljungdahlii. *Microbial cell factories*, 12(1):118.

[191] Neidhardt, F. C., Ingraham, J. L., and Schaechter, M. (1990). *Physiology of the bacterial cell: a molecular approach*, volume 20. Sinauer Associates Sunderland, MA.

[192] Ng, C. Y., Chowdhury, A., and Maranas, C. D. (2016). A microbial factory for diverse chemicals. *Nat Biotech*, 34:513–515.

[193] Ng, C. Y., Farasat, I., Maranas, C. D., and Salis, H. M. (2015a). Rational design of a synthetic entner–doudoroff pathway for improved and controllable nadph regeneration. *Metabolic engineering*, 29:86–96.

[194] Ng, C. Y., Khodayari, A., Chowdhury, A., and Maranas, C. D. (2015b). Advances in de novo strain design using integrated systems and synthetic biology tools. *Current opinion in chemical biology*, 28:105–114.

[195] Nielsen, D. R., Leonard, E., Yoon, S.-H., Tseng, H.-C., Yuan, C., and Prather, K. L. J. (2009). Engineering alternative butanol production platforms in heterologous bacteria. *Metabolic engineering*, 11(4-5):262–273.

[196] Nielsen, J. and Keasling, J. (2016). Engineering Cellular Metabolism. *Cell*, 164(6):1185–1197.

[197] Niu, D., Tian, K., Prior, B. A., Wang, M., Wang, Z., Lu, F., and Singh, S. (2014). Highly efficient l-lactate production using engineered escherichia coli with dissimilar temperature optima for l-lactate formation and cell growth. *Microbial cell factories*, 13(1):78.

[198] Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media, United States of America.

[199] Noor, E., Flamholz, A., Bar-Even, A., Davidi, D., Milo, R., and Liebermeister, W. (2016). The protein cost of metabolic fluxes: Prediction from enzymatic rate laws and cost minimization. *PLOS Computational Biology*, 12(11):e1005167.

[200] Ohta, H., Kinoshita, K., Saeki, M., Hayashi, S., and Obayashi, T. (2008). Atted-ii provides coexpressed gene networks for arabidopsis. *Nucleic Acids Research*, 37(suppl_1):D987–D991.

[201] Olson, D. G., Hörl, M., Fuhrer, T., Cui, J., Zhou, J., Maloney, M. I., Amador-Noguez, D., Tian, L., Sauer, U., and Lynd, L. R. (2017). Glycolysis without pyruvate kinase in clostridium thermocellum. *Metabolic engineering*, 39:169–180.

[202] Olson, D. G., McBride, J. E., Shaw, A. J., and Lynd, L. R. (2012). Recent progress in consolidated bioprocessing. *Current opinion in biotechnology*, 23(3):396–405.

[203] O'Regan, G. (2018). The system/360 revolution. In *The Innovation in Computing Companion*, pages 243–248. Springer.

[204] Palsson, B. Ø. (2015). *Systems biology: constraint-based reconstruction and analysis*. Cambridge University Press, United Kingdom.

[205] Pandit, A. V., Srinivasan, S., and Mahadevan, R. (2017). Redesigning metabolism based on orthogonality principles. *Nature communications*, 8:15188.

[206] Papanek, B., Biswas, R., Rydzak, T., and Guss, A. M. (2015). Elimination of metabolic pathways to all traditional fermentation products increases ethanol yields in clostridium thermocellum. *Metabolic engineering*, 32:49–54.

[207] Park, H.-J. and Friston, K. (2013). Structural and functional brain networks: from connections to cognition. *Science*, 342(6158):1238411.

[208] Park, S.-J., Cotter, P. A., and Gunsalus, R. P. (1995). Regulation of malate dehydrogenase (mdh) gene expression in escherichia coli in response to oxygen, carbon, and heme availability. *Journal of bacteriology*, 177(22):6652–6656.

[209] Patti, G. J., Yanes, O., and Siuzdak, G. (2012). Innovation: Metabolomics: the apogee of the omics trilogy. *Nature reviews Molecular cell biology*, 13(4):263.

[210] Peng, L., Arauzo-Bravo, M. J., and Shimizu, K. (2004). Metabolic flux analysis for a ppc mutant Escherichia coli based on 13C-labelling experiments together with enzyme activity assays and intracellular metabolite measurements. *FEMS Microbiology Letters*, 235(1):17–23.

[211] Peters, N. K. (2018). Bioenergy research centers. Technical report, USDOE Office of Science (SC), Washington, DC (United States).

[212] Pharkya, P., Burgard, A. P., and Maranas, C. D. (2004). Optstrain: a computational framework for redesign of microbial production systems. *Genome research*, 14(11):2367–2376.

[213] Pharkya, P. and Maranas, C. D. (2006). An optimization framework for identifying reaction activation/inhibition or elimination candidates for overproduction in microbial systems. *Metabolic engineering*, 8(1):1–13.

[214] Price, N. D., Papin, J. A., Schilling, C. H., and Palsson, B. O. (2003). Genome-scale microbial in silico models: the constraints-based approach. *Trends in Biotechnology*, 21:162–169.

[215] Pryciak, P. M. (2009). Designing new cellular signaling pathways. *Chemistry & biology*, 16(3):249–254.

[216] Purnick, P. E. M. and Weiss, R. (2009). The second wave of synthetic biology: from modules to systems. *Nature reviews Molecular cell biology*, 10:410–422.

[217] Ragauskas, A. J., Williams, C. K., Davison, B. H., Britovsek, G., Cairney, J., Eckert, C. A., Frederick, W. J., Hallett, J. P., Leak, D. J., Liotta, C. L., et al. (2006). The path forward for biofuels and biomaterials. *Science*, 311(5760):484–489.

[218] Rangaiah, G. P. (2009). *Multi-objective optimization: techniques and applications in chemical engineering*, volume 1. World Scientific, Singapore.

[219] Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., and Barabási, A.-L. (2002). Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551–1555.

[220] Rehm, B. H. (2010). Bacterial polymers: biosynthesis, modifications and applications. *Nature Reviews Microbiology*, 8(8):578.

[221] Reimers, A.-M., Lindhorst, H., and Waldherr, S. (2017). A protocol for generating and exchanging (genome-scale) metabolic resource allocation models. *Metabolites*, 7(3):47.

[222] Riquelme, N., Von Lücken, C., and Baran, B. (2015). Performance metrics in multi-objective optimization. In *Latin American Computing Conference (CLEI)*, pages 1–11. IEEE.

[223] Roberts, S. B., Gowen, C. M., Brooks, J. P., and Fong, S. S. (2010). Genome-scale metabolic analysis of clostridium thermocellum for bioethanol production. *BMC systems biology*, 4(1):31.

[224] Rodriguez, G. M., Tashiro, Y., and Atsumi, S. (2014). Expanding ester biosynthesis in escherichia coli. *Nature Chemical Biology*, 10:259–265.

[225] Rydzak, T., McQueen, P. D., Krokhin, O. V., Spicer, V., Ezzati, P., Dwivedi, R. C., Shamshurin, D., Levin, D. B., Wilkins, J. A., and Sparling, R. (2012). Proteomic analysis of clostridium thermocellum core metabolism: relative protein expression profiles and growth phase-dependent changes in protein expression. *BMC microbiology*, 12(1):214.

[226] Salimi, F., Zhuang, K., and Mahadevan, R. (2010). Genome-scale metabolic modeling of a clostridial co-culture for consolidated bioprocessing. *Biotechnology journal*, 5(7):726–738.

[227] Salis, H. M., Mirsky, E. A., and Voigt, C. A. (2009). Automated design of synthetic ribosome binding sites to control protein expression. *Nature biotechnology*, 27(10):946.

[228] Salvador, F. (2007). Toward a product system modularity construct: literature review and reconceptualization. *IEEE Transactions on engineering management*, 54(2):219–240.

[229] Sanchez-Pascuala, A., de Lorenzo, V., and Nikel, P. (2017). Refactoring the embden–meyerhof–parnas pathway as a whole of portable glucobricks for implantation of glycolytic modules in gram-negative bacteria. *ACS synthetic biology*, 6(5):793–805.

[230] Sauro, H. M. (2008). Modularity defined. *Molecular systems biology*, 4.

[231] Scheer, M., Grote, A., Chang, A., Schomburg, I., Munaretto, C., Rother, M., Söhngen, C., Stelzer, M., Thiele, J., and Schomburg, D. (2010). Brenda, the enzyme information system in 2011. *Nucleic acids research*, 39(suppl_1):D670–D676.

[232] Schellenberger, J., Lewis, N. E., and Palsson, B. Ø. (2011a). Elimination of thermodynamically infeasible loops in steady-state metabolic models. *Biophysical journal*, 100(3):544–553.

[233] Schellenberger, J., Que, R., Fleming, R. M. T., Thiele, I., Orth, J. D., Feist, A. M., Zielinski, D. C., Bordbar, A., Lewis, N. E., and Rahmanian, S. (2011b). Quantitative prediction of cellular metabolism with constraint-based models: the cobra toolbox v2. 0. *Nature protocols*, 6:1290–1307.

[234] Schuetz, R., Zamboni, N., Zampieri, M., Heinemann, M., and Sauer, U. (2012). Multidimensional optimality of microbial metabolism. *Science*, 336(6081):601–604.

[235] Schutze, O., Esquivel, X., Lara, A., and Coello, C. A. C. (2012). Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522.

[236] Senger, R. S. and Papoutsakis, E. T. (2008). Genome-scale model for clostridium acetobutylicum: Part i. metabolic network resolution and analysis. *Biotechnology and bioengineering*, 101(5):1036–1052.

[237] Seo, H., Lee, J.-W., Garcia, S., and Trinh, C. T. (2019). Single mutation at a highly conserved region of chloramphenicol acetyltransferase enables isobutyl acetate production directly from cellulose by clostridium thermocellum at elevated temperatures. *Biotechnology for biofuels*, 12(1):245.

[238] Serrano-Bermúdez, L. M., Barrios, A. F. G., Maranas, C. D., and Montoya, D. (2017). Clostridium butyricum maximizes growth while minimizing enzyme usage and atp production: metabolic flux distribution of a strain cultured in glycerol. *BMC systems biology*, 11(1):58.

[239] Shafiee, S. and Topal, E. (2009). When will fossil fuel reserves be diminished? *Energy policy*, 37(1):181–189.

[240] Shao, Z., Zhao, H., and Zhao, H. (2009). Dna assembler, an in vivo genetic method for rapid construction of biochemical pathways. *Nucleic Acids Research*, 37(2):e16.

[241] Shen, C. R., Lan, E. I., Dekishima, Y., Baez, A., Cho, K. M., and Liao, J. C. (2011). Driving forces enable high-titer anaerobic 1-butanol synthesis in escherichia coli. *Applied and Environmental Microbiology*, 77(9):2905–2915.

[242] Shetty, R. P., Endy, D., and Knight, T. F. (2008). Engineering biobrick vectors from biobrick parts. *Journal of biological engineering*, 2(1):5.

[243] Shi, A., Zhu, X., Lu, J., Zhang, X., and Ma, Y. (2013). Activating transhydrogenase and nad kinase in combination for improving isobutanol production. *Metabolic engineering*, 16:1–10.

[244] Shoval, O., Sheftel, H., Shinar, G., Hart, Y., Ramote, O., Mayo, A., Dekel, E., Kavanagh, K., and Alon, U. (2012). Evolutionary trade-offs, pareto optimality, and the geometry of phenotype space. *Science*, page 1217405.

[245] Siddiquee, K. A. Z., Arauzo-Bravo, M., and Shimizu, K. (2004). Metabolic flux analysis of pykf gene knockout escherichia coli based on 13 c-labeling experiments together with measurements of enzyme activities and intracellular metabolite concentrations. *Applied microbiology and biotechnology*, 63(4):407–417.

[246] Slusarczyk, A. L., Lin, A., and Weiss, R. (2012). Foundations for the design and implementation of synthetic genetic circuits. *Nature Reviews Genetics*, 13(6):406–420.

[247] Sosa, M. E., Eppinger, S. D., and Rowles, C. M. (2007). A network approach to define modularity of components in complex products. *Journal of mechanical design*, 129(11):1118–1129.

[248] Spirin, V. and Mirny, L. A. (2003). Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences*, 100(21):12123–12128.

[249] Sporns, O. and Betzel, R. F. (2016). Modular brain networks. *Annual review of psychology*, 67:613–640.

[250] Stephanopoulos, G. and Vallino, J. J. (1991). Network rigidity and metabolic engineering in metabolite overproduction. *Science*, 252(5013):1675–1681.

[251] Stuart, J. M., Segal, E., Koller, D., and Kim, S. K. (2003). A gene-coexpression network for global discovery of conserved genetic modules. *Science*, 302(5643):249–255.

[252] Szegezdi, J. and Csizmadia, F. (2007). Method for calculating the pka values of small and large molecules. In *Abstracts of Papers of The American Chemical Society*, volume 233. AMER CHEMICAL SOC 1155 16TH ST, NW, WASHINGTON, DC 20036 USA.

[253] Szklarczyk, D., Morris, J. H., Cook, H., Kuhn, M., Wyder, S., Simonovic, M., Santos, A., Doncheva, N. T., Roth, A., and Bork, P. (2017). The string database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic acids research*, 45(D1):D362–D368.

[254] Tabb, D. L., Fernando, C. G., and Chambers, M. C. (2007). Myrimatch: highly accurate tandem mass spectral peptide identification by multivariate hypergeometric analysis. *Journal of proteome research*, 6(2):654–661.

[255] Tan, G.-Y. and Liu, T. (2017). Rational synthetic pathway refactoring of natural products biosynthesis in actinobacteria. *Metabolic engineering*, 39:228–236.

[256] Taverner, T., Karpievitch, Y. V., Polpitiya, A. D., Brown, J. N., Dabney, A. R., Anderson, G. A., and Smith, R. D. (2012). Danter: an extensible r-based tool for quantitative analysis of-omics data. *Bioinformatics*, 28(18):2404–2406.

[257] Temme, K., Zhao, D. H., and Voigt, C. A. (2012). Refactoring the nitrogen fixation gene cluster from klebsiella oxytoca. *Proceedings of the National Academy of Sciences of the United States of America*, 109(18):7085–7090.

[258] Tepper, N. and Shlomi, T. (2010). Predicting metabolic engineering knockout strategies for chemical production: accounting for competing pathways. *Bioinformatics*, 26:536–543.

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

[260] Thompson, R. A., Dahal, S., Garcia, S., Nookaew, I., and Trinh, C. T. (2016). Exploring complex cellular phenotypes and model-guided strain design with a novel genome-scale metabolic model of clostridium thermocellum dsm 1313 implementing an adjustable cellulosome. *Biotechnology for biofuels*, 9(1):194.

[261] Thompson, R. A., Layton, D. S., Guss, A. M., Olson, D. G., Lynd, L. R., and Trinh, C. T. (2015). Elucidating central metabolic redox obstacles hindering ethanol production in clostridium thermocellum. *Metabolic engineering*, 32:207–219.

[262] Thompson, R. A. and Trinh, C. T. (2017). Overflow metabolism and growth cessation in clostridium thermocellum dsm1313 during high cellulose loading fermentations. *Biotechnology and bioengineering*, 114(11):2592–2604.

[263] Tian, L., Papanek, B., Olson, D. G., Rydzak, T., Holwerda, E. K., Zheng, T., Zhou, J., Maloney, M., Jiang, N., Giannone, R. J., et al. (2016a). Simultaneous achievement of high ethanol yield and titer in clostridium thermocellum. *Biotechnology for biofuels*, 9(1):116.

[264] Tian, Y., Cheng, R., Zhang, X., Cheng, F., and Jin, Y. (2018). An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility. *IEEE Transactions on Evolutionary Computation*, 22(4):609–622.

[265] Tian, Y., Cheng, R., Zhang, X., and Jin, Y. (2017). Platemo: A matlab platform for evolutionary multi-objective optimization. *IEEE Computational Intelligence Magazine*, 12(4):73–87.

[266] Tian, Y., Zhang, X., Cheng, R., and Jin, Y. (2016b). A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 5222–5229. IEEE.

[267] Tokuyama, K., Ohno, S., Yoshikawa, K., Hirasawa, T., Tanaka, S., Furusawa, C., and Shimizu, H. (2014). Increased 3-hydroxypropionic acid production from glycerol, by modification of central metabolism in escherichia coli. *Microbial cell factories*, 13(1):64.

[268] Trinh, C. and Srienc, F. (2009). Metabolic engineering of escherichia coli for efficient conversion of glycerol to ethanol. *Appl Environ Microbiol*, 75(21):6696 – 6705.

[269] Trinh, C. T. (2012). Elucidating and reprogramming escherichia coli metabolisms for obligate anaerobic n-butanol and isobutanol production. *Applied microbiology and biotechnology*, 95(4):1083–1094.

[270] Trinh, C. T., Carlson, R., Wlaschin, A., and Srienc, F. (2006). Design, construction and performance of the most efficient biomass producing e. coli bacterium. *Metabolic engineering*, 8(6):628–638.

[271] Trinh, C. T., Li, J., Blanch, H. W., and Clark, D. S. (2011). Redesigning escherichia coli metabolism for anaerobic production of isobutanol. *Appl. Environ. Microbiol.*, 77(14):4894–4904.

[272] Trinh, C. T., Liu, Y., and Conner, D. J. (2015). Rational design of efficient modular cells. *Metabolic engineering*, 32:220–231.

[273] Trinh, C. T. and Mendoza, B. (2016). Modular cell design for rapid, efficient strain engineering toward industrialization of biology. *Current Opinion in Chemical Engineering*, 14:18–25.

[274] Trinh, C. T., Unrean, P., and Srienc, F. (2008). Minimal escherichia coli cell for the most efficient production of ethanol from hexoses and pentoses. *Applied and Environmental Microbiology*, 74(12):3634–3643.

[275] Trinh, C. T., Wlaschin, A., and Srienc, F. (2009). Elementary mode analysis: a useful metabolic pathway analysis tool for characterizing cellular metabolism. *Applied Microbiology and Biotechnology*, 81(5):813–826.

[276] Trubitsyna, M., Michlewski, G., Cai, Y., Elfick, A., and French, C. E. (2014). Paperclip: rapid multi-part dna assembly from existing libraries. *Nucleic Acids Research*, page gku829.

[277] Tseng, H.-C. and Prather, K. L. (2012). Controlled biosynthesis of odd-chain fuels and chemicals via engineered modular metabolic pathways. *Proceedings of the National Academy of Sciences*, page 201209002.

[278] Tsuge, K., Matsui, K., and Itaya, M. (2003). One step assembly of multiple dna fragments with a designed order and orientation in bacillus subtilis plasmid. *Nucleic Acids Research*, 31(21):e133–e133.

[279] Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research policy*, 24(3):419–440.

[280] Venayak, N., von Kamp, A., Klamt, S., and Mahadevan, R. (2018). Move identifies metabolic valves to switch between phenotypic states. *Nature communications*, 9(1):5332.

[281] von Kamp, A. and Klamt, S. (2014). Enumeration of smallest intervention strategies in genome-scale metabolic networks. *PLOS Computational Biology*, 10(1):1–13.

[282] von Kamp, A. and Klamt, S. (2017a). Growth-coupled overproduction is feasible for almost all metabolites in five major production organisms. *Nature communications*, 8:15956.

[283] von Kamp, A. and Klamt, S. (2017b). Growth-coupled overproduction is feasible for almost all metabolites in five major production organisms. *Nature communications*, 8:15956.

[284] Vujić, J., Bergmann, R. M., Škoda, R., and Miletić, M. (2012). Small modular reactors: Simpler, safer, cheaper? *Energy*, 45(1):288–295.

[285] Wagner, A. and Fell, D. A. (2001). The small world inside large metabolic networks. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1478):1803–1810.

[286] Wagner, G. P., Pavlicev, M., and Cheverud, J. M. (2007). The road to modularity. *Nature Reviews Genetics*, 8(12):921–931.

[287] Wallenius, J., Viikilä, M., Survase, S., Ojamo, H., and Eerikäinen, T. (2013). Constraint-based genome-scale metabolic modeling of clostridium acetobutylicum behavior in an immobilized column. *Bioresource technology*, 142:603–610.

[288] Wang, L., Dash, S., Ng, C. Y., and Maranas, C. D. (2017). A review of computational tools for design and reconstruction of metabolic pathways. *Synthetic systems biotechnology*, 2(4):243–252.

[289] Wang, L. and Maranas, C. D. (2018). Mingenome: An in silico top-down approach for the synthesis of minimized genomes. *ACS synthetic biology*, 7(2):462–473.

[290] Weyer, S., Schmitt, M., Ohmer, M., and Gorecky, D. (2015). Towards industry 4.0-standardization as the crucial challenge for highly modular, multi-vendor production systems. *Ifac-Papersonline*, 48(3):579–584.

[291] Whitacre, J. M. (2012). Biological robustness: paradigms, mechanisms, and systems principles. *Frontiers in genetics*, 3:67.

[292] Wierzbicki, M., Niraula, N., Yarrabothula, A., Layton, D. S., and Trinh, C. T. (2016). Engineering an escherichia coli platform to synthesize designer biodiesels. *Journal of biotechnology*, 224:27–34.

[293] Wilbanks, B., Layton, D., Garcia, S., and Trinh, C. (2017). A prototype for modular cell engineering. *ACS Synthetic Biology*, page acssynbio.7b00269.

[294] Wilkins, M. R., Sanchez, J.-C., Gooley, A. A., Appel, R. D., Humphery-Smith, I., Hochstrasser, D. F., and Williams, K. L. (1996). Progress with proteome projects: why all proteins expressed by a genome should be identified and how to do it. *Biotechnology and genetic engineering reviews*, 13(1):19–50.

[295] Winkler, J. D., Halweg-Edwards, A. L., and Gill, R. T. (2015). The laser database: Formalizing design rules for metabolic engineering. *Metabolic Engineering Communications*, 2:30–38.

[296] Wu, G., Yan, Q., Jones, J. A., Tang, Y. J., Fong, S. S., and Koffas, M. A. (2016). Metabolic burden: cornerstones in synthetic biology and metabolic engineering applications. *Trends in biotechnology*, 34(8):652–664.

[297] Xu, P., Gu, Q., Wang, W., Wong, L., Bower, A. G. W., Collins, C. H., and Koffas, M. A. G. (2013). Modular optimization of multi-gene pathways for fatty acids production in e. coli. *Nat Commun*, 4:1409.

[298] Yadav, V. G., De Mey, M., Giaw Lim, C., Kumaran Ajikumar, P., and Stephanopoulos, G. (2012). The future of metabolic engineering and synthetic biology: Towards a systematic practice. *Metabolic Engineering*, 14:233–241.

[299] Yang, L., Cluett, W. R., and Mahadevan, R. (2011). Emilio: a fast algorithm for genome-scale strain design. *Metabolic engineering*, 13:272–281.

[300] Yang, Y., Lin, Y., Wang, J., Wu, Y., Zhang, R., Cheng, M., Shen, X., Wang, J., Chen, Z., and Li, C. (2018). Sensor-regulator and rnai based bifunctional dynamic control network for engineered microbial synthesis. *Nature communications*, 9(1):3043.

[301] Yim, H., Haselbeck, R., Niu, W., Pujol-Baxley, C., Burgard, A., Boldt, J., Khandurina, J., Trawick, J. D., Osterhout, R. E., and Stephen, R. (2011a). Metabolic engineering of escherichia coli for direct production of 1, 4-butanediol. *Nature chemical biology*, 7:445–452.

[302] Yim, H., Haselbeck, R., Niu, W., Pujol-Baxley, C., Burgard, A., Boldt, J., Khandurina, J., Trawick, J. D., Osterhout, R. E., Stephen, R., et al. (2011b). Metabolic engineering of escherichia coli for direct production of 1, 4-butanediol. *Nature chemical biology*, 7(7):445.

[303] Yoo, M., Bestel-Corre, G., Croux, C., Riviere, A., Meynial-Salles, I., and Soucaille, P. (2015). A quantitative system-scale characterization of the metabolism of clostridium acetobutylicum. *MBio*, 6(6):e01808–15.

[304] Yu, J., Xia, X., Zhong, J., and Qian, Z. (2014). Direct biosynthesis of adipic acid from a synthetic pathway in recombinant escherichia coli. *Biotechnology and bioengineering*, 111(12):2580–2586.

[305] Yuan, Y., Xu, H., Wang, B., and Yao, X. (2016a). A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(1):16–37.

[306] Yuan, Y., Xu, H., Wang, B., Zhang, B., and Yao, X. (2016b). Balancing convergence and diversity in decomposition-based many-objective optimizers. *IEEE Transactions on Evolutionary Computation*, 20(2):180–198.

[307] Zhang, F., Carothers, J. M., and Keasling, J. D. (2012a). Design of a dynamic sensor-regulator system for production of chemicals and fuels derived from fatty acids. *Nature biotechnology*, 30(4):354.

[308] Zhang, Y., Werling, U., and Edelmann, W. (2012b). Slice: a novel bacterial cell extract-based dna cloning method. *Nucleic Acids Research*, 40(8):e55.

[309] Zhang, Y.-H. P. and Lynd, L. R. (2005). Cellulose utilization by clostridium thermocellum: bioenergetics and hydrolysis product assimilation. *Proceedings of the National Academy of Sciences*, 102(20):7321–7325.

[310] Zhao, J., Baba, T., Mori, H., and Shimizu, K. (2004). Global metabolic response of escherichia coli to gnd or zwf gene-knockout, based on 13 c-labeling experiments and the measurement of enzyme activities. *Applied microbiology and biotechnology*, 64(1):91–98.

[311] Zhao, J. and Shimizu, K. (2003). Metabolic flux analysis of escherichia coli k12 grown on 13c-labeled acetate and glucose using gc-ms and powerful flux calculation method. *Journal of biotechnology*, 101(2):101–117.

[312] Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., and Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49.

[313] Zhou, J., Olson, D. G., Argyros, D. A., Deng, Y., van Gulik, W. M., van Dijken, J. P., and Lynd, L. R. (2013). Atypical glycolysis in clostridium thermocellum. *Appl. Environ. Microbiol.*, 79(9):3000–3008.

[314] Zhou, S., Causey, T., Hasona, A., Shanmugam, K., and Ingram, L. (2003). Production of optically pure d-lactic acid in mineral salts medium by metabolically engineered escherichia coli w3110. *Appl. Environ. Microbiol.*, 69(1):399–407.

[315] Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195.

[316] Zitzler, E., Laumanns, M., and Thiele, L. (2001). Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103.

[317] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and Da Fonseca Grunert, V. (2002). Performance assessment of multiobjective optimizers: An analysis and review. *TIK-Report*, 139.

Appendices

A Supplementary Material 1 for Chapter 2

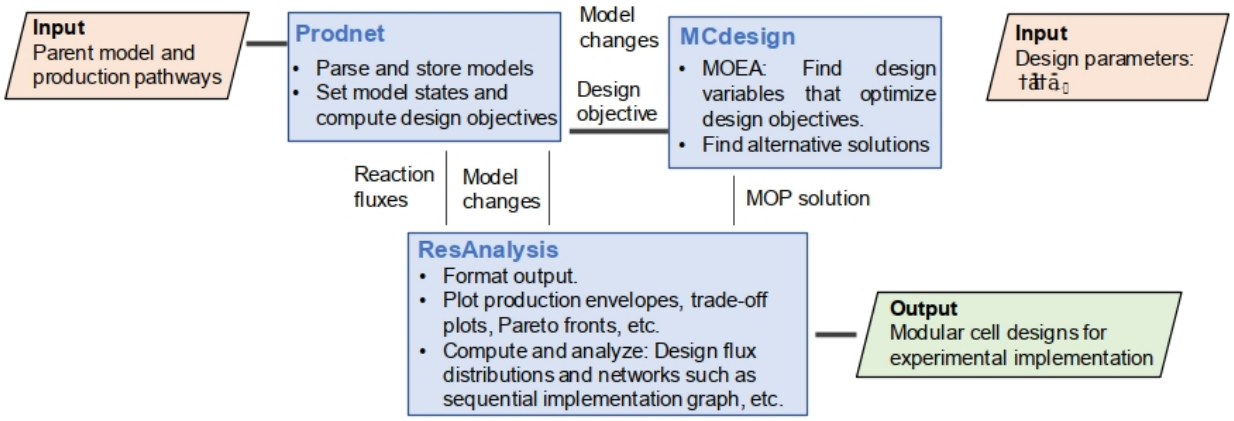


Figure A1: Software architecture of ModCell2. The Prodnet class preprocesses production network models and computes design objectives. The MCdesign class serves as an interface between the MOEA optimization method and metabolic models. The ResAnalysis class loads the Pareto set computed by MCdesign and performs analyses to identify the most promising designs.

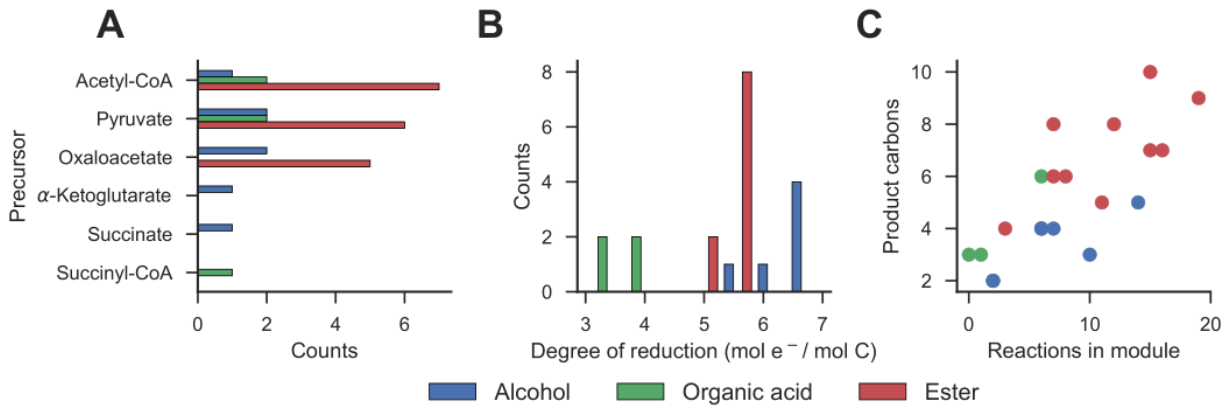


Figure A2: Properties of 20 production modules used in the E. coli genome-scale metabolic model for biosynthesis of 6 alcohols, 4 organic acids, and 10 esters. (A) Distribution of precursor metabolites. (B) Distribution of degrees of reduction of target products. (C) Correlation between the number of product carbons and the number of reactions in production modules. Alcohols include ethanol, propanol, butanol, isobutanol, pentanol, and 1,4-butandiol; acids include pyruvate, D-lactate, acetate, and adipic acid; and esters include ethyl acetate, propyl acetate, isobutyl acetate, ethyl butanoate, propyl butanoate, butyl butanoate, isobutyl butanoate, ethyl pentanoate, isobutyl pentanoate, and pentyl pentanoate.

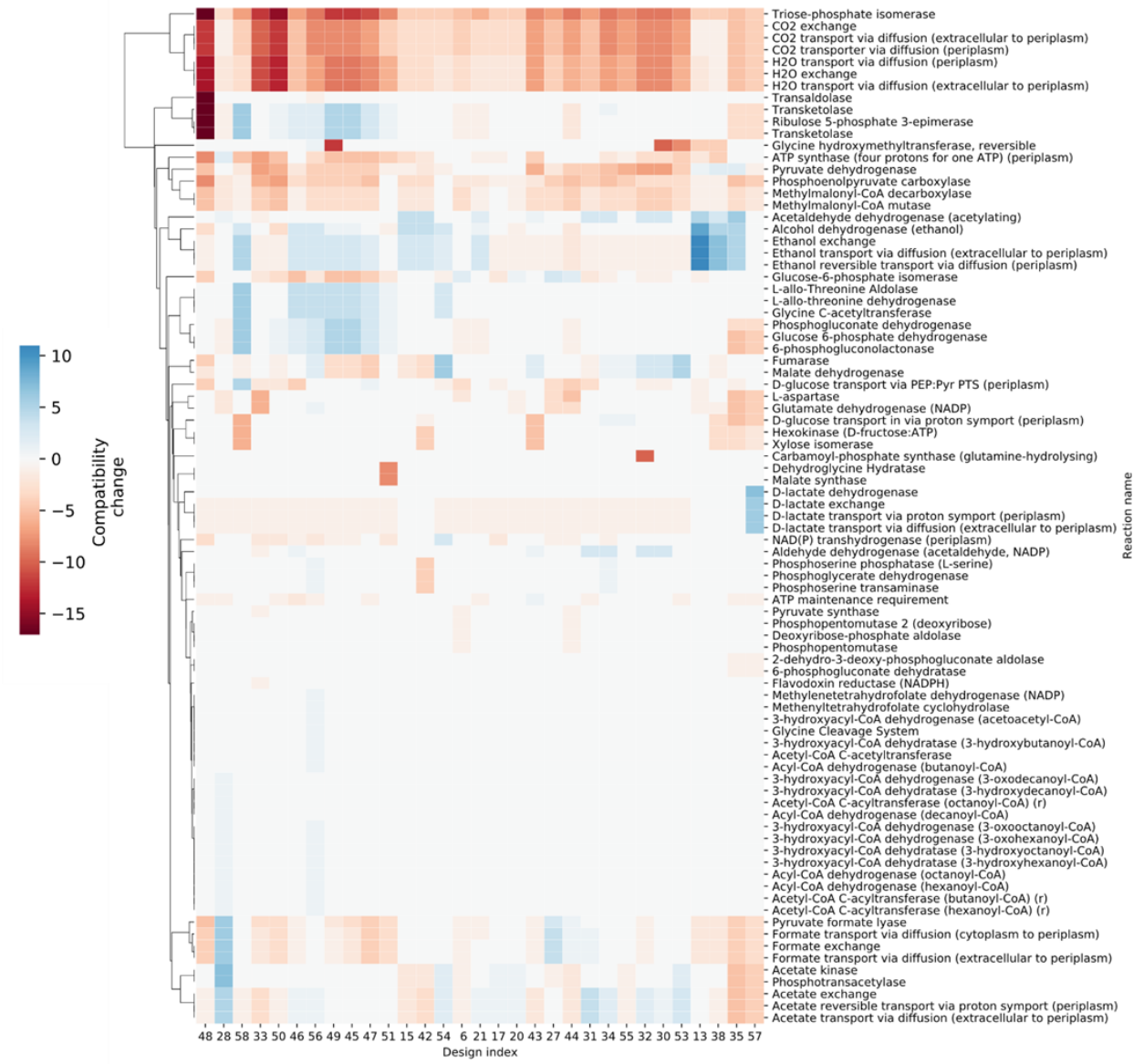


Figure A3: Robustness analysis for wGCP-4-0 designs for the E. coli genome scale model. Only the designs that are compatible with 4 or more products (compatibility 4) were considered. Each column corresponds to a design whereas each row corresponds to a single-reaction deletion. Included in the heat map are all reaction deletions with a compatibility change that is not 0 in at least one product.

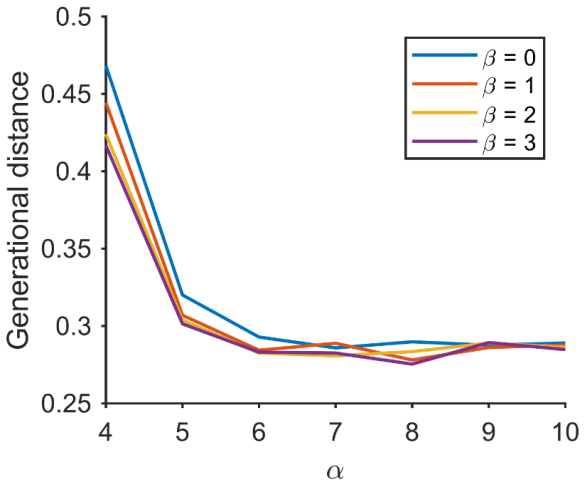


Figure A4: Generational distances between the calculated Pareto fronts and the reference utopia point. The generational distance is calculated as follows: $GD = \frac{|d|_2}{|d|_1}$where $d_j = \left|\mathcal{PF}^j - \mathcal{PF}^*\right|_2$ where $\mathcal{PF}$ is the calculated pareto front and $\mathcal{PF}^* = \vec{1}$ is the utopia point. A smaller value of GD indicates the overall objective values in the Pareto front are closer to the utopia point. The calculation was performed for the iML1515 model with 20 products using the wGCP objective, various and values, and a run time of 10 h for all cases.

B Supplementary Material 2 for Chapter 2

B.1 Solution method: Multiobjective Evolutionary Algorithm

Definitions

Terms

Parent network: A parent network is a metabolic model of a host organism that is used to construct a modular cell.

Production module: A production module is a metabolic pathway that is added to a modular cell to synthesize a target product.

Production network: A production network is a combination of a parent network and a production module.

MOEA: Multiobjective evolutionary algorithm.

Sets

$\mathcal{I}_k$: Set of metabolite indices in production network k.

$\mathcal{J}_k$: Set of reaction indices in production network k.

$\mathcal{K}$: Set of production network indices.

$\mathcal{C}$: Set of candidate reaction deletion indices, where $\mathcal{C} \subseteq \mathcal{J}^{parent} \subseteq \mathcal{J}_k$, $\forall k \in \mathcal{K}$.

Continuous variables

v_{jk} : Flux of reaction j in production network k.

v_{Pk} : Flux of target product (P) reaction in production network k.

v_{Xk} : Flux of biomass (X) synthesis reaction in production network k.

f_k : Objective function for production network k.

f_k^{wGCP} : *wGCP* objective function for production network k.

f_k^{sGCP} : *sGCP* objective function for production network k.

f_k^{NGP} : *NGP* objective function for production network k.

p_k : Penalty objective function for production network k.

q^{enum} : Objective function for enumerating alternative solutions.

Binary variables

y_j : Reaction deletion indicator that takes a value of 0 if reaction j is deleted in a modular cell, and 1 otherwise.

z_{jk} : Endogeonous, module-specific reaction indicator that takes a value of 1 if reaction j is added back to the production module in network k, and 0 otherwise.

$d_{jk} = y_j \vee z_{jk}$: Modeling variable which takes a value of 1 if reaction j may carry flux in production network k, and 0 otherwise.

Parameters

S_{ijk} : Stoichiometric coefficient of metabolite i in reaction j of production network k.

l_{jk} : Lower bound flux for reaction j in production network k.

u_{jk} : Upper bound flux for reaction j in production network k.

α : Maximum number of deletion reactions in a modular cell.

β_k : Maximum number of endogenous module-specific reactions in the module of production network k.

ϵ : Small scalar used for tilting the biomass objective function to obtain the minimum product rate available at the maximum growth rate. In the simulation, we used $\epsilon = 0.0001$.

Solver

We used the *gamultiobj()* solver, an implementation of NSGA-II[61], from the MATLAB Optimization Toolbox to solve our combinatorial multiobjective optimization, formulated as an unconstrained multiobjective problem of the form:

$$\max(p_1, p_2, \ldots, p_{|\mathcal{K}|})^T \tag{B1}$$

with a *bitstring* population type where each individual is a binary vector corresponding to the design variables y_j (reaction deletions) and z_{jk} (endogenous module-specific reactions). To enforce the constraints on the number of deletion and endogenous module reactions, a penalty function p_k, instead of f_k, (Section B.1) and customized genetic operators (Section B.1) were used, respectively.

Penalty Objective function

To restrict the maximum number of reaction deletions, we optimized the following penalty function p_k instead of f_k:

$$p_k = \begin{cases} \dfrac{f_k}{\sum_{j \in C}(1 - y_j)} & \text{if } \sum_{j \in C}(1 - y_j) > \alpha \\ f_k & \text{otherwise} \end{cases} \tag{B2}$$

The penalty function is designed to decrease f_k of an individual proportionally to the number of deletion reactions exceeding the set limit α. Implementation of this penalty function helps the optimization problem converge rapidly because favorable deletion reaction candidates are likely kept to obtain desirable solutions. After simulation, only solutions satisfying the maximum reaction deletion constraint are preserved to obtain the Pareto set of our original problem.

Design objective computation

Depending on desirable applications, the following design objectives are considered:

$$f_k^{wGCP} = \frac{v_{Pk}^{\mu}}{v_{P_{max}k}^{\mu}} \in [0,1], \qquad \forall k \in \mathcal{K} \tag{B3}$$

$$f_k^{sGCP} = \frac{v_{Pk}^{\mu}}{v_{P_{max}k}^{\mu}} \frac{v_{Pk}^{\bar{\mu}}}{v_{P_{max}k}^{\bar{\mu}}} \in [0,1], \forall k \in \mathcal{K} \tag{B4}$$

$$f_k^{NGP} = \frac{v_{Pk}^{\bar{\mu}}}{v_{P_{max}k}^{\bar{\mu}}} \in [0,1], \qquad \forall k \in \mathcal{K} \tag{B5}$$

In (B3)-(B5), the terms v^{μ}_{Pk}, $v^{\mu}_{P_{max}k}$, $v^{\bar{\mu}}_{Pk}$, and $v^{\bar{\mu}}_{P_{max}k}$ are computed by solving the following linear programming problems:

$$v^{\mu}_{Pk} \quad \in \arg\max\{v_{Xk} - \epsilon v_{Pk} : v_k \in \Pi^{\mu}_k(d_{jk})\} \tag{B6}$$

$$v^{\mu}_{P_{max}k} \in \arg\max\{v_{Pk} : v_k \in \Pi^{\mu}_k(d_{jk} = 1,\ \forall j \in \mathcal{J}_k)\} \tag{B7}$$

$$v^{\bar{\mu}}_{Pk} \quad \in \arg\min\{v_{Pk} : v_k \in \Pi^{\bar{\mu}}_k(d_{jk})\} \tag{B8}$$

$$v^{\bar{\mu}}_{P_{max}k} \in \arg\max\{v_{Pk} : v_k \in \Pi^{\bar{\mu}}_k(d_{jk} = 1,\ \forall j \in \mathcal{J}_k)\} \tag{B9}$$

The maximum product synthesis fluxes in (B7) and (B9), used to normalize the design objectives in (B3)-(B5), only need to be computed once for each network prior to solving the multiobjective problem (MOP).

In $(B6) - (B9), \Pi^{\mu}_k$ is the space of steady-state reaction fluxes of production network k where a minimum cell growth is required, defined as follows:

$$\begin{aligned}
\Pi^{\mu}_k(d_{jk}) := \{&v_{jk} \in \mathbb{R}\, \forall j \in \mathcal{J}_k : \\
&\sum_{j \in \mathcal{J}_k} S_{ijk} v_{jk} = 0, && \forall i \in \mathcal{I}_k && \text{(B10)}\\
&l_{jk} \le v_{jk} \le u_{jk}, && \forall j \in \mathcal{J}_k && \text{(B11)}\\
&l_{jk} d_{jk} \le v_{jk} \le u_{jk} d_{jk}, && \forall j \in \mathcal{C} && \text{(B12)}\\
&v_{Xk} \ge \text{minimum growth rate}\} && && \text{(B13)}
\end{aligned}$$

Constraints (B10)-(B11) correspond to mass balance and flux bounds, as described in the main text. Constraint (B12) ensures that reaction j cannot carry any flux, if it is deleted in the modular cell and not present in module k. Constraint (B13) specifies any minimum growth rate requirement.

When the design goals involve the stationary phase (B8)-(B9), the space of steady-state reaction fluxes for production network k, $\Pi^{\bar{\mu}}_k$, is defined as follows:

$$\Pi_k^{\bar{\mu}}(d_{jk}) := \{v_{jk} \in \mathbb{R}\, \forall j \in \mathcal{J}_k :$$

$$\sum_{j \in \mathcal{J}_k} S_{ijk} v_{jk} = 0, \qquad \forall i \in \mathcal{I}_k \tag{B14}$$

$$l_{jk} \leq v_{jk} \leq u_{jk}, \qquad \forall j \in \mathcal{J}_k \tag{B15}$$

$$l_{jk} d_{jk} \leq v_{jk} \leq u_{jk} d_{jk}, \qquad \forall j \in \mathcal{C} \tag{B16}$$

$$v_{Xk} = 0\} \tag{B17}$$

If any of the linear programs associated with f_k becomes infeasible, i.e., $\Pi_k^{\mu} = \emptyset$ or $\Pi_k^{\bar{\mu}} = \emptyset$, then f_k is set to 0.

Termination criteria

We implemented a non-domination termination criterion to determine when simulation must stop to retrieve a solution, as described in Algorithm 1.

Algorithm 1: Non-domination termination criterion for MOEA. PF: Pareto front, PS: Pareto set.

```
[PF, PS] = solveMOP(initialPoint = ∅, stall_generations, ...)
total_generations = 0
do
    PF_old = PF
    PS_old = PS
    [PF, PS] = solveMOP(initialPoint = PS_old, stall_generations, ...)
    total_generations = total_generations + stall_generations
while any(PF dominates PF_old) and total_generations ≤ max_total_generations
 and run_time ≤ max_run_time
```

Based on this criterion, the solution is retrieved if new non-dominated solutions cannot be found for a predefined number of stall generations. For our study, we used highly conservative, empirical values of 500 and 1000 stall generations with runtime limits of 1-2h and 10-15h for core and genome scale models, respectively.

Customized genetic operators to handle endogenous module-specific reactions

We modified the default scattered crossover and uniform mutation operators of *gamultiobj()* to enforce the constraint on the number of endogenous module reactions and improve convergence. First, we ensured both crossover and mutation operators to produce only individuals that meet the maximum module reaction constraint, i.e., $\sum_{j\in\mathcal{J}_k} z_{jk} \leq \beta_k,\ \forall k \in \mathcal{K}$. Next, we required that only reactions deleted in the modular cell can be used as endogenous module-specific reactions, i.e., $z_{jk} \leq 1-y_j,\ \forall j \in \mathcal{J},\ k \in \mathcal{K}$. Finally, we specified the crossover operator to perform crossover on the variables associated with reaction deletions and endogenous module reactions separately, for each production network.

Parameters

All MOEA parameters, except the population size, were left as default. In our study, we set the empirically conservative values for population sizes of 200 and 400 for core and genome-scale models to converge in 2 h and 15 h of simulation time, respectively.

Enumeration of alternative solutions

If a solution w produces the same objective vector as a Pareto optimal solution x^*, i.e., $f(w) = f(x^*)$ and $w \neq x^*$, we say that w constitutes an alternative solution of x^*. To enumerate alternative solutions for a specific Pareto optimal design, we iteratively solve a minimization problem of the form:

$$\min q^{enum} \tag{B18}$$

using MATLAB's genetic algorithm *ga()*. Using the Jaccard similarity metric[1], we define q^{enum}, which takes a value of 0 if an alternative solution is found, as follows:

[1]The Jaccard similarity between vectors r and s, $\mathbf{jacc}(r,s)$, corresponds to the fraction of common elements between r and s. If r and s are the same, the Jaccard similarity is 1; if both vectors do not share any elements, then it takes a value of 0.

$$q^{enum} = \begin{cases} M & \text{if } \{y_j : j \in \mathcal{J}_k\} \in \text{ExcludedSol} \quad \text{(B19)} \\ 1 - \mathbf{jacc}(f^*, f) + \sum_{j \in \mathcal{C}} (1 - y_j) & \text{if } \sum_{j \in \mathcal{C}} (1 - y_j) > \alpha \quad \text{(B20)} \\ 1 - \mathbf{jacc}(f^*, f) & \text{otherwise} \quad \text{(B21)} \end{cases}$$

Initially, ExcludedSol will contain at least a target solution for which we are interested in finding alternative solutions. A large scalar M is returned if the current set of deletions $\{y_j\}$ has been found previously, and hence cannot be an alternative solution (B19). Likewise, a set of deletions $\{y_j\}$ may be a valid solution candidate but have more deletions than allowed (B20). In that case, the negated Jaccard similarity is penalized according to the number of reaction deletions.

Optimizing algorithm performance

Variable declaration. To minimize the number of free variables in the optimization problem, we created binary variables, y_j, only for the reaction candidate set C instead of all reactions in the parent model. Similarly, endogenous module reaction variables, z_{jk}, were only created for $j \in \mathcal{C}$ if $\beta_k > 0$.

Selection of starting population. To accelerate convergence in our simulation, we used a predetermined starting population of individuals, if possible. A starting population can be derived from a previously obtained result; for instance, the solutions from $\alpha = 6$ can be used as a starting population to find solutions for $\alpha = 7$. In some cases, we also used design strategies determined experimentally or orginated from other strain design algorithms (e.g., Optknock).

Parallelization. To increase the simulation speed, we performed the objective function computations in parallel. This parallelization alleviated the bottlenecks of solving 1 linear programming problem (LP) in *wGCP* (or *NGP*) and 2 LPs in *sGCP*.

Archive of solutions. Since computing objective functions is one critical bottleneck, we used a table (archive mapping design variables to design objectives) of previously evaluated individuals to avoid repeating this calculation. The size of the table is determined by the amount of memory available. For instance, we stored at most 50,000 solutions, which can be handled for 20 design objectives by a personal computer. When the table becomes full, it is erased to allow for higher quality individuals to be archived.

B.2 Specifying the Set of Deletion Reaction Candidates for Manipulation

The set of deletion reaction candidates, C, is a subset of all reactions in the parent network, excluding reactions infeasible to eliminate in practice and irrelevant to desirable phenotypes, as described below. Some criteria used in[74] were adapted and implemented in our study.

Macromolecule-associated reactions. These reactions involve macromolecules whose biologically relevant roles are not well represented in the model (e.g., glycogen) or do not impact the optimal design of target product biosynthesis pathways. We identified macromolecule-associated reactions by screening metabolite IDs and formulas, for instance, those with total carbon number above 10 except currency metabolites (e.g., ATP, acetyl-CoA, etc.)

Non-metabolic reactions. These reactions belong to the functional categories such as ion transport, tRNA charging, etc. We identified these non-metabolic reactions by screening the reaction-subsystem annotation in the parent network.

Modeling reactions. These reactions are sink reactions and/or reactions which are not well characterized. We identified the modeling reactions by screening reaction IDs and sbo terms in the parent network.

Transport reactions. These reactions involve metabolites transported across cellular compartments. Most of these reactions are not included in C due to their unspecific

annotation or non-enzymatic mechanisms except some well-annotated reactions such as ATP synthase or NAD(P) transhydrogenase. We identified these transport reactions by screening metabolites appearing in multiple compartments.

Exchange reactions. These reactions are pseudo-reactions used to simulate steady-state conditions. We identified these reactions based on the characteristics they only have either substrates or products.

Orphan reactions. These reactions do not have known encoding enzymes. We determined them by screening gene-protein-reaction associations in the parent model.

Essential reaction. The essential reactions are the reactions whose removal from the model makes the maximum growth rate fall below the minimum acceptable value (i.e, 10-20% of the predicted maximum growth rate). We identified these reactions by performing flux balance analysis combined with single reaction deletions.

Blocked reactions. These reactions carry a flux of 0 mmol/gCDW/hr across all production networks (Figure B1.A). We found these blocked reactions by performing flux variability analysis.

Reactions in fully correlated sets (co-sets). Sets of reactions that have linearly correlated fluxes are classified as co-sets. These reactions can belong to a linear pathway or more than one associated pathways. For each co-set, only one potential candidate reaction is needed to be considered in the reaction deletion candidate set. In our analysis, we considered all co-sets present in a master network, containing all production modules, to prevent potentially useful reaction deletions to be excluded from the candidate set (Figure B1.B). We found the co-sets by flux coupling analysis.

Special consideration for NGP designs. The NGP design objective does not involve the growth phase, unlike wGCP and sGCP. Thus, the set of reaction deletion candidates for NGP designs are determined as outlined above except: i) essential reactions that should

be considered in the candidate set and ii) blocked reactions that are determined under non-growth conditions (i.e. biomass flux is constrained to be 0).

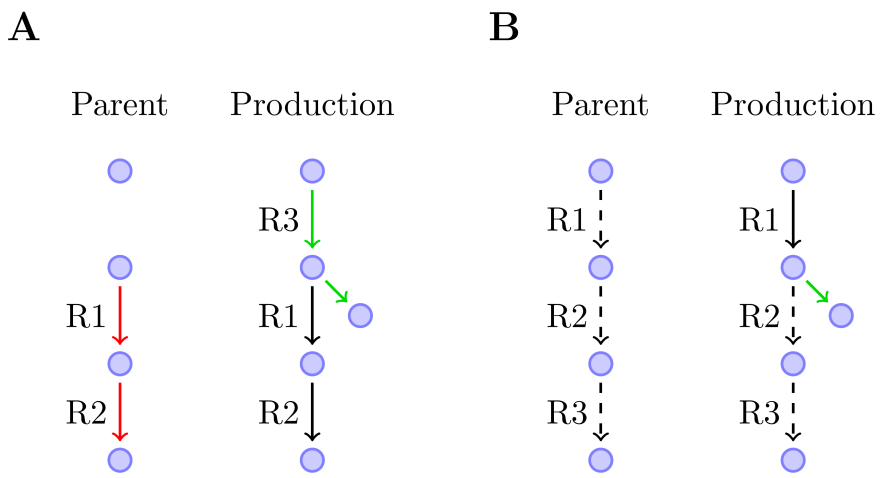


Figure B1: (A) Blocked reactions. The red arrows represent reactions originally blocked in the parent model, because the substrate of R1 cannot be produced. When heterologus reactions (green arrows) are added to produce a target chemical, the originally blocked reactions may become active and drain an intermediate of the production pathway. (B) Reaction co-sets. The dashed arrows are used to indicated a fully correlated set. The addition of heterologous reactions (green arrows), alters the co-set definition and has important effect in deletion candidates. If R1 is considered as a deletion candidate, instead of R2 or R3, that would prevent the elimination of a potentially undesired pathway.

C Supplementary Material 1 for Chapter 5

Table C1: The 70 consistent reactions in the Δ*hydG*-Δ*ech* case study and their associated fold changes. The biomass reaction is not included due to size. This table is continued in the following pages.

ID	Formula	Fold change		
		proteomics	*pFBA*	*FVAcenter*
MDH	mal__L_c + nad_c ↔ h_c + nadh_c + oaa_c	1.4	0.2	0.1
PEPCK_re	co2_c + gdp_c + pep_c → gtp_c + oaa_c	0.9	0.1	0.1
VOR2b	3mob_c + coa_c + 2.0 fdxo_42_c → co2_c + 2.0 fdxr_42_c + h_c + ibcoa_c	0.9	12.3	0.8
PFL	coa_c + pyr_c → accoa_c + for_c	0.5	0.2	0.8
FRNDPR2r	2.0 fdxr_42_c + h_c + nadh_c + 2.0 nadp_c ↔ 2.0 fdxo_42_c + nad_c + 2.0 nadph_c	0.3	0.2	2.3
ALCD2x	acald_c + h_c + nadh_c → etoh_c + nad_c	0.1	0.8	0.6
IBUTCOARx	h_c + ibcoa_c + nadh_c → 2mppal_c + coa_c + nad_c	0.1	12.3	0.8
ACALD	accoa_c + h_c + nadh_c → acald_c + coa_c + nad_c	0.1	0.8	1.1
ALCD23xi	2mppal_c + h_c + nadh_c → ibutoh_c + nad_c	0.1	12.3	0.8
ME2	mal__L_c + nadp_c → co2_c + nadph_c + pyr_c	0.1	0.3	0.1
PSCVT	pep_c + skm5p_c → 3psme_c + pi_c	-0.1	-0.3	0.7
PTAr	accoa_c + pi_c ↔ actp_c + coa_c	-0.1	-2.7	-0.1
HSOR	3.0 h_c + 3.0 nadph_c + so3_c → 3.0 h2o_c + h2s_c + 3.0 nadp_c	-0.1	-0.3	-1.2
TRDR	h_c + nadph_c + trdox_c → nadp_c + trdrd_c	-0.1	-0.3	0.8
IGPS	2cpr5p_c + h_c → 3ig3p_c + co2_c + h2o_c	-0.1	-0.3	0.2
GLUDy	glu__L_c + h2o_c + nadp_c ↔ akg_c + h_c + nadph_c + nh4_c	-0.1	-0.5	-0.2
ECH	2.0 fdxr_42_c + 3.0 h_c ↔ 2.0 fdxo_42_c + h2_c + h_e	-0.1	-15.6	-13.5

Table C1 Continued

ID	Formula	Fold change		
		proteomics	*pFBA*	*FVAcenter*
ALLAS	24.0 ala__D_c + 24.0 atp_c + 24.0 cdpglyc_c + dg12dg_c + 24.0 h2o_c → ala_lta_c + 24.0 amp_c + 24.0 cmp_c + 48.0 h_c + 24.0 ppi_c	-0.1	-0.2	0.0
HSDxi	aspsa_c + h_c + nadh_c → hom__L_c + nad_c	-0.1	-0.3	0.8
PRMICI	prfp_c → prlp_c	-0.1	-0.3	-0.1
OCBT	cbp_c + orn__L_c ↔ citr__L_c + h_c + pi_c	-0.2	-0.3	-0.3
UAGCVT	pep_c + uacgam_c → pi_c + uaccg_c	-0.2	-0.3	-0.1
ANS	chor_c + gln__L_c → anth_c + glu__L_c + h_c + pyr_c	-0.2	-0.3	0.2
DHAD2	23dhmp_c → 3mop_c + h2o_c	-0.2	-0.3	0.6
ASPO2y	asp__L_c + nadp_c → h_c + iasp_c + nadph_c	-0.3	-0.3	-0.1
NADK	atp_c + nad_c → adp_c + h_c + nadp_c	-0.3	-0.3	-0.1
METS	5mthf_c + hcys__L_c → met__L_c + thf_c	-0.3	-0.3	0.8
IMPD	h2o_c + imp_c + nad_c ↔ h_c + nadh_c + xmp_c	-0.3	-0.3	-0.3
MG2abc	atp_c + h2o_c + mg2_e → adp_c + h_c + mg2_c + pi_c	-0.3	-0.3	-0.1
PDHam1hi	h_c + pyr_c + thmpp_c → 2ahethmpp_c + co2_c	-0.4	-0.3	12.4
ACAS_2ahbut	2ahethmpp_c + 2obut_c → 2ahbut_c + thmpp_c	-0.4	-0.3	0.6
GF6PTA	f6p_B_c + gln__L_c → gam6p_c + glu__L_c	-0.4	-0.3	0.0
CHRS	3psme_c → chor_c + pi_c	-0.4	-0.3	0.7
SERH	3ig3p_c + ser__L_c → g3p_c + h2o_c + trp__L_c	-0.4	-0.3	0.2
ALATA_L	akg_c + ala__L_c ↔ glu__L_c + pyr_c	-0.5	-0.3	-12.9
NNDPR	h_c + prpp_c + quln_c → co2_c + nicrnt_c + ppi_c	-0.5	-0.3	-0.1
TMDS	dump_c + mlthf_c → dhf_c + dtmp_c	-0.5	-0.3	-0.1
PHETA1	akg_c + phe__L_c ↔ glu__L_c + phpyr_c	-0.6	-0.3	0.8
TYRTA	akg_c + tyr__L_c ↔ 34hpp_c + glu__L_c	-0.6	-0.3	0.6
GMPS	atp_c + nh4_c + xmp_c → amp_c + gmp_c + 3.0 h_c + ppi_c	-0.6	-0.3	-0.3

Table C1 Continued

ID	Formula	Fold change		
		proteomics	*pFBA*	*FVAcenter*
SKK	atp_c + skm_c → adp_c + h_c + skm5p_c	-0.6	-0.3	0.7
ACOTA	acorn_c + akg_c ↔ acg5sa_c + glu__L_c	-0.6	-0.3	-0.3
PPDK	amp_c + 2.0 h_c + pep_c + ppi_c → atp_c + pi_c + pyr_c	-0.7	-10.0	0.1
GLUPRT	gln__L_c + h2o_c + prpp_c → glu__L_c + h_c + ppi_c + pram_c	-0.7	-0.3	-0.3
KARI	2ahbut_c ↔ cpd10162_c	-0.8	-0.3	0.6
KARI_23dhmp	23dhmp_c + nadp_c ↔ cpd10162_c + h_c + nadph_c	-0.8	-0.3	0.6
ARGSL	argsuc_c ↔ arg__L_c + fum_c	-0.8	-0.3	-0.3
LEUTA	4mop_c + glu__L_c → akg_c + leu__L_c	-0.8	-0.3	0.4
ILETA	akg_c + ile__L_c ↔ 3mop_c + glu__L_c	-0.8	-0.3	0.6
VALTA	akg_c + val__L_c ↔ 3mob_c + glu__L_c	-0.8	-1.4	-1.5
ARGSS	asp__L_c + atp_c + citr__L_c ↔ amp_c + argsuc_c + 2.0 h_c + ppi_c	-0.8	-0.3	-0.3
SHSL2	h2s_c + suchms_c → hcys__L_c + succ_c	-0.9	-6.0	0.8
AHSL	achms_c + cys__L_c ↔ ac_c + cyst_L_c + h_c	-0.9	-10.6	-0.3
SHSL1	cyst_L_c + h_c + succ_c ↔ cys__L_c + suchms_c	-0.9	-10.6	0.4
ACKr	actp_c + adp_c → ac_c + atp_c	-0.9	-2.7	-0.1
QULNS	dhap_c + iasp_c → 2.0 h2o_c + h_c + pi_c + quln_c	-0.9	-0.3	-0.1
NADS2	atp_c + dnad_c + gln__L_c + h2o_c → amp_c + glu__L_c + 2.0 h_c + nad_c + ppi_c	-0.9	-0.3	-0.1
FE3abc	atp_c + fe3_e + h2o_c → adp_c + fe3_c + h_c + pi_c	-1.0	-0.3	-0.1
ASPTA	akg_c + asp__L_c ↔ glu__L_c + oaa_c	-1.0	-0.3	0.2
CTPS1	atp_c + nh4_c + utp_c → adp_c + ctp_c + 2.0 h_c + pi_c	-1.2	-0.3	0.0
ACGK	acglu_c + atp_c → acg5p_c + adp_c	-1.2	-0.3	-0.3

Table C1 Continued

ID	Formula	Fold change		
		proteomics	*pFBA*	*FVAcenter*
IGPDH	eig3p_c → h2o_c + imacp_c	-1.2	-0.3	-0.1
AGPR	acg5sa_c + nadp_c + pi_c ↔ acg5p_c + h_c + nadph_c	-1.3	-0.3	-0.3
PHEt2r	h_e + phe__L_e ↔ h_c + phe__L_c	-1.5	-0.3	0.8
UAG4Ei	uacgam_c → udpacgal_c	-1.5	-0.3	-0.1
CYSS	acser_c + h2s_c → ac_c + cys__L_c	-1.8	-0.3	-0.5
BIF	2.0 fdxr_42_c + 3.0 h_c + nadh_c ↔ 2.0 fdxo_42_c + 2.0 h2_c + nad_c	-1.8	-13.8	-12.5
UMPK	atp_c + h_c + ump_c → adp_c + udp_c	-2.1	-0.3	0.0
SULabc	atp_c + h2o_c + so4_e → adp_c + h_c + pi_c + so4_c	-4.6	-0.3	0.8

Table C2: pFBA simulated fluxes from Δ*hydG*-Δ*ech* case study. Only includes reactions involving NADPH or exchange reactions that have different flux between wild-type and mutant. The biomass reaction is not included due to size. Additionally, we also excluded reactions with the same fold change magnitude as the biomass reaction (|FC| = 0.26), because their fluxes are likely fully correlated.

ID	Formula	Fluxes (mmol/gCDW/hr)		
		W.T.	*Mut.*	FC
EX_ibutoh_e	ibutoh_e →	0.0	0.49	12.26
KARA1	alac__S_c + h_c + nadph_c → 23dhmb_c + nadp_c	0.22	0.59	1.42
EX_etoh_e	etoh_e →	1.09	1.88	0.78
EX_h2o_e	h2o_e ↔	-0.38	0.5	0.42
EX_co2_e	co2_e →	2.07	2.77	0.42
ME2	mal__L_c + nadp_c → co2_c + nadph_c + pyr_c	2.9	3.51	0.28
EX_for_e	for_e →	0.48	0.56	0.23
FRNDPR2r	2.0 fdxr_42_c + h_c + nadh_c + 2.0 nadp_c ↔ 2.0 fdxo_42_c + nad_c + 2.0 nadph_c	-1.02	-1.17	0.2
EX_nh4_e	nh4_e ↔	-0.73	-0.53	-0.48
GLUDy	glu__L_c + h2o_c + nadp_c ↔ akg_c + h_c + nadph_c + nh4_c	-0.61	-0.42	-0.53
EX_h_e	h_e ↔	2.93	1.19	-1.3
EX_val__L_e	val__L_e →	0.18	0.06	-1.54
ICDHyr	icit_c + nadp_c → akg_c + co2_c + nadph_c	0.21	0.04	-2.27
EX_ac_e	ac_e →	0.93	0.17	-2.46
EX_lac__L_e	lac__L_e →	0.05	0.01	-2.49
EX_succ_e	succ_e →	0.41	0.0	-12.01
EX_h2_e	h2_e →	2.2	0.0	-14.43

Table C3: Reaction deletions sorted by appeareance frequency (counts) in the designs of the Pareto front for $\alpha = 6, \beta = 0$.

ID	Name	Formula	Counts (%)
PGM	Phosphoglycerate mutase	2pg_c ↔ 3pg_c	75
H2ASE_syn	Bidirectional [NiFe] Hydrogenase (Fe-H2)	h2_c + nadp_c ↔ h_c + nadph_c	75
ECH	(FeFe)-hydrogenase, ferredoxin dependent, membrane-bound	2.0 fdxr_42_c + 3.0 h_c ↔ 2.0 fdxo_42_c + h2_c + h_e	66.7
BIF	Bifurcating Hydrogenase	2.0 fdxr_42_c + 3.0 h_c + nadh_c ↔ 2.0 fdxo_42_c + 2.0 h2_c + nad_c	66.7
GLUDy	Glutamate dehydrogenase (NADP)	glu__L_c + h2o_c + nadp_c ↔ akg_c + h_c + nadph_c + nh4_c	50
FRNDPR2r	Ferredoxin: nadp reductase (NFN)	2.0 fdxr_42_c + h_c + nadh_c + 2.0 nadp_c ↔ 2.0 fdxo_42_c + nad_c + 2.0 nadph_c	41.7
RNF	Ferredoxin:NAD oxidoreductase (membrane bound)	2.0 fdxr_42_c + 2.0 h_c + nad_c ↔ 2.0 fdxo_42_c + h_e + nadh_c	33.3
PEPCK_re	Phosphoenolpyruvate carboxykinase (GTP)	co2_c + gdp_c + pep_c → gtp_c + oaa_c	33.3
ALCD2x	Alcohol dehydrogenase (ethanol)	acald_c + h_c + nadh_c → etoh_c + nad_c	25
ACALD	Acetaldehyde dehydrogenase (acetylating)	accoa_c + h_c + nadh_c → acald_c + coa_c + nad_c	25
PPDK	Pyruvate, phosphate dikinase	amp_c + 2.0 h_c + pep_c + ppi_c → atp_c + pi_c + pyr_c	25
GLUSy	Glutamate synthase (NADPH)	akg_c + gln__L_c + h_c + nadph_c → 2.0 glu__L_c + nadp_c	16.7
PFL	Pyruvate formate lyase	coa_c + pyr_c → accoa_c + for_c	16.7
LDH_L	L-lactate dehydrogenase	h_c + nadh_c + pyr_c → lac__L_c + nad_c	16.7
POR	Pyruvate-ferredoxin oxidoreductase	coa_c + 2.0 fdxo_42_c + pyr_c → accoa_c + co2_c + 2.0 fdxr_42_c + h_c	8.3
CEPA	Cellobiose phosphorylase	cellb_c + pi_c → g1p_c + glc__D_c	8.3
GMPS	GMP synthase	atp_c + nh4_c + xmp_c → amp_c + gmp_c + 3.0 h_c + ppi_c	8.3
AHSL	O-Acetyl-L-homoserine succinate-lyase	achms_c + cys__L_c ↔ ac_c + cyst_L_c + h_c	8.3

D Supplementary Material 1 for Chapter 6

Figure D1: Solution improvement process.

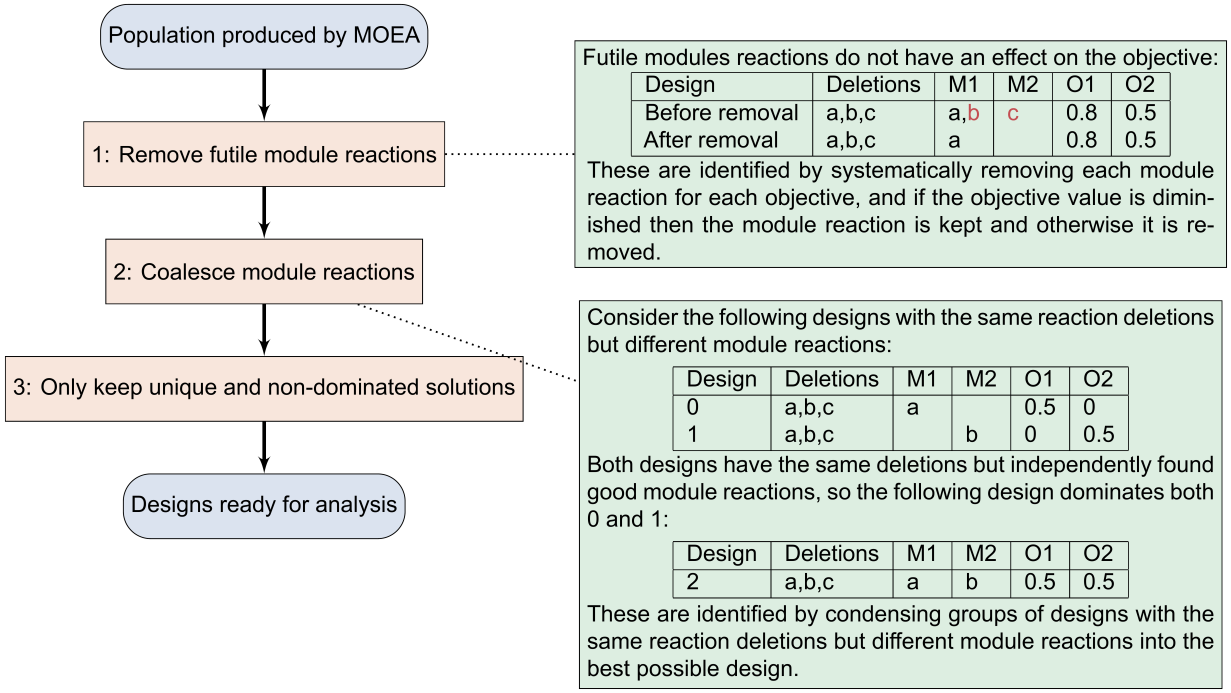


Figure D2: Chemical properties of the product library. DoR is the degree of reduction (mol e^-/ mol C), which is computed assuming a constant valency of 4,1,-2, and 5 for C,H,O, and P, respectively. For reference, ethanol has 2 carbon atoms, a molecular weight of 46 g/mol, and a DoR of 6 (mol e^-/mol C). The molecular weight and the number of carbon atoms have a Pearson correlation coefficient (pcc) of 0.98, while DoR and the molecular weight only have a pcc of 0.42.

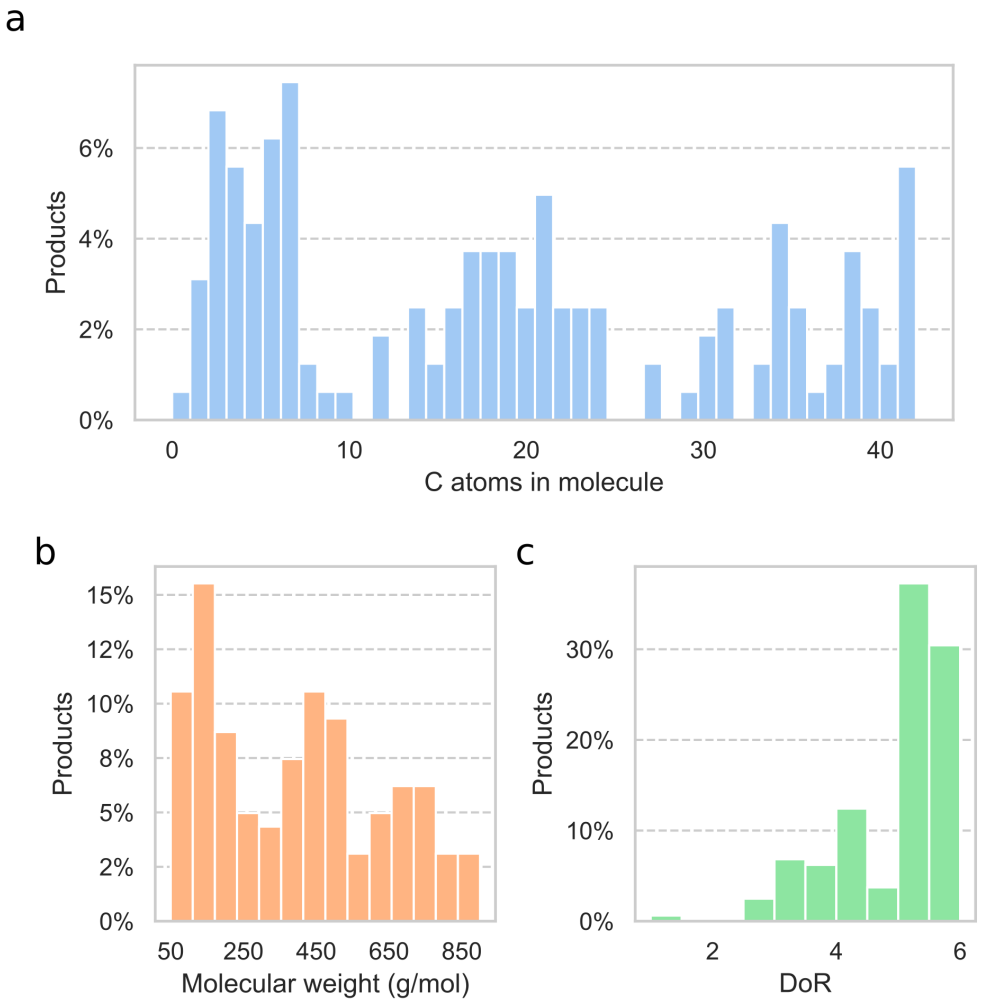


Figure D3: ModCell-HPC benchmarking with 20 products and design parameters $\alpha = 6$, $\beta = 1$. For a given run time, this analysis scans through all the combinations of migration interval, migration policy, and population size. Note that coverage values are not directly comparable between run times since the use a different reference Pareto front.

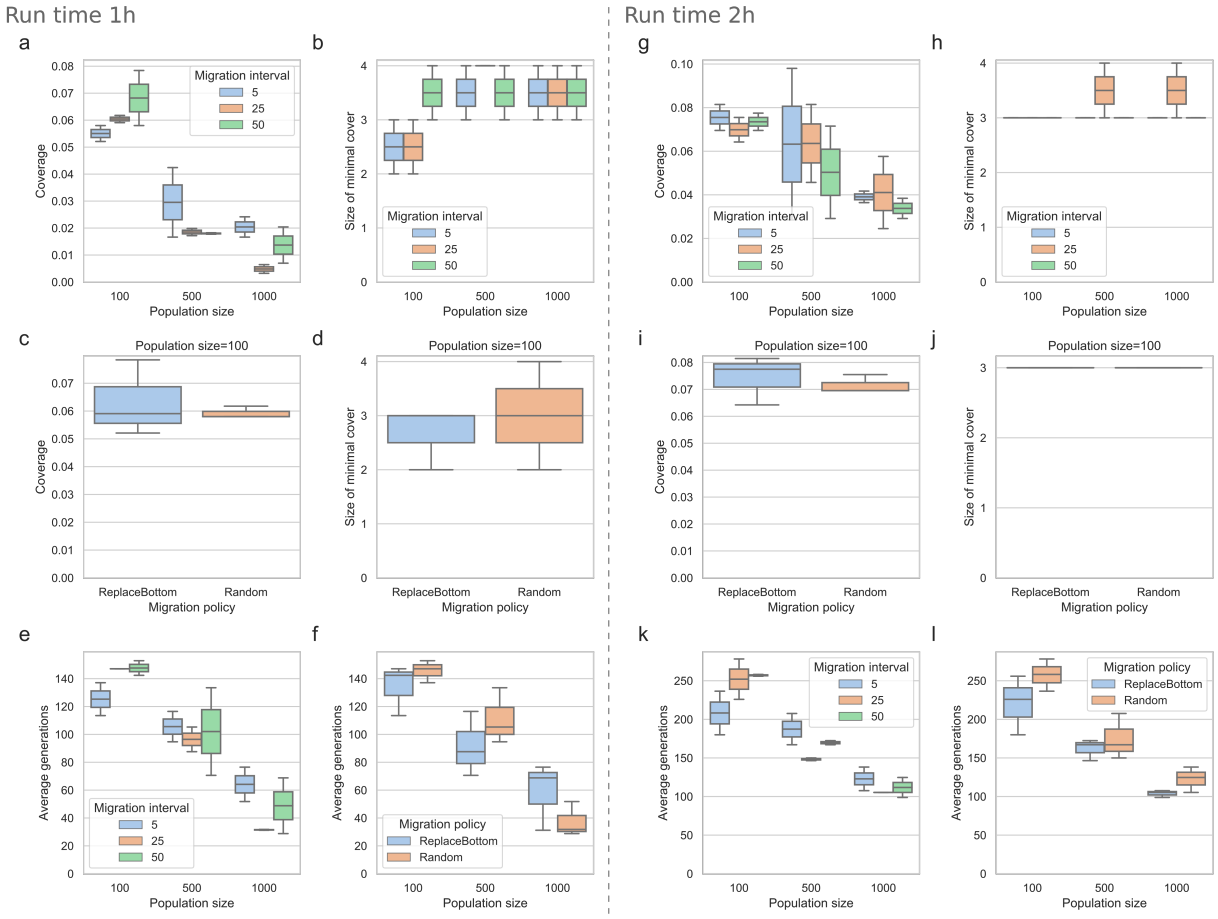


Figure D4: Compatibility of all designs in a Pareto front as a result of the design parameters. Each panel corresponds to a unique carbon source as the only difference in model configuration.

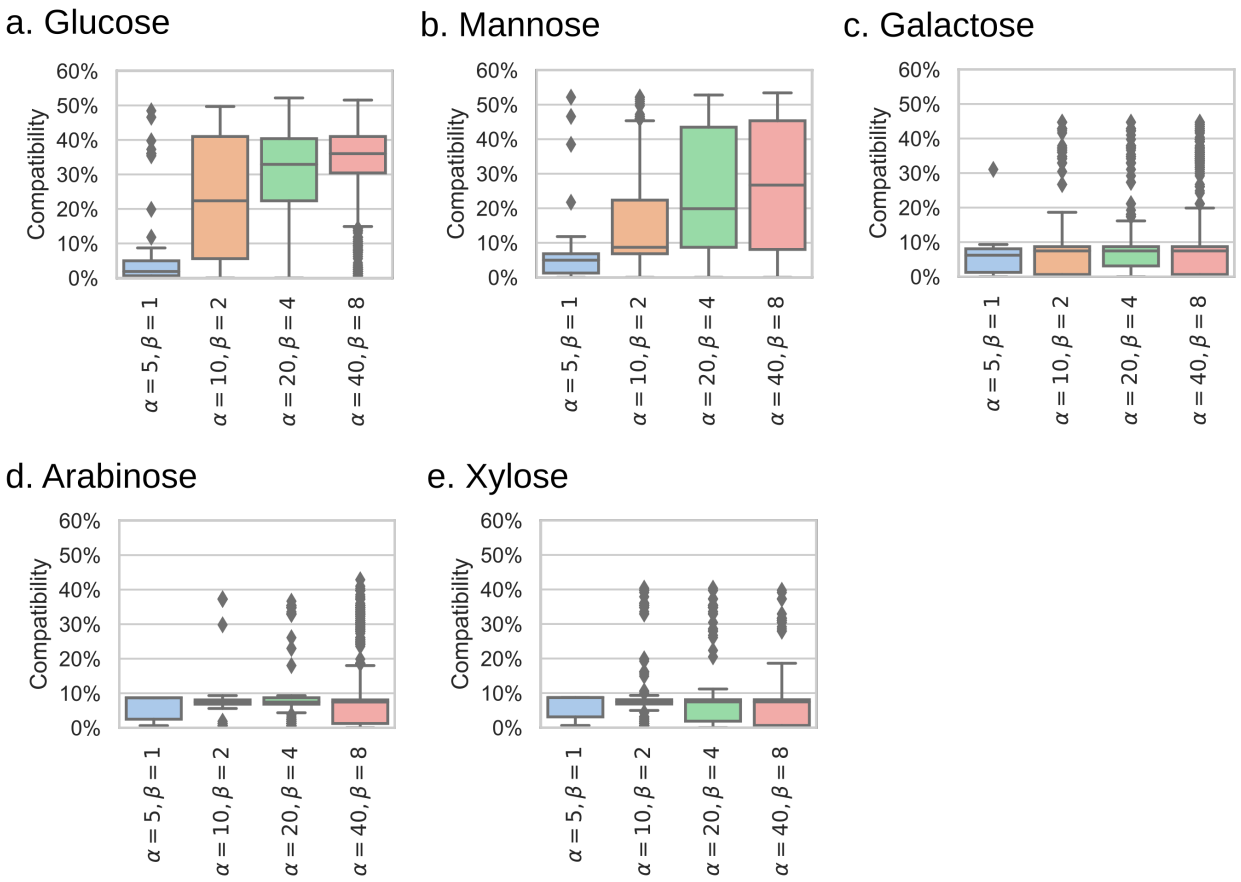
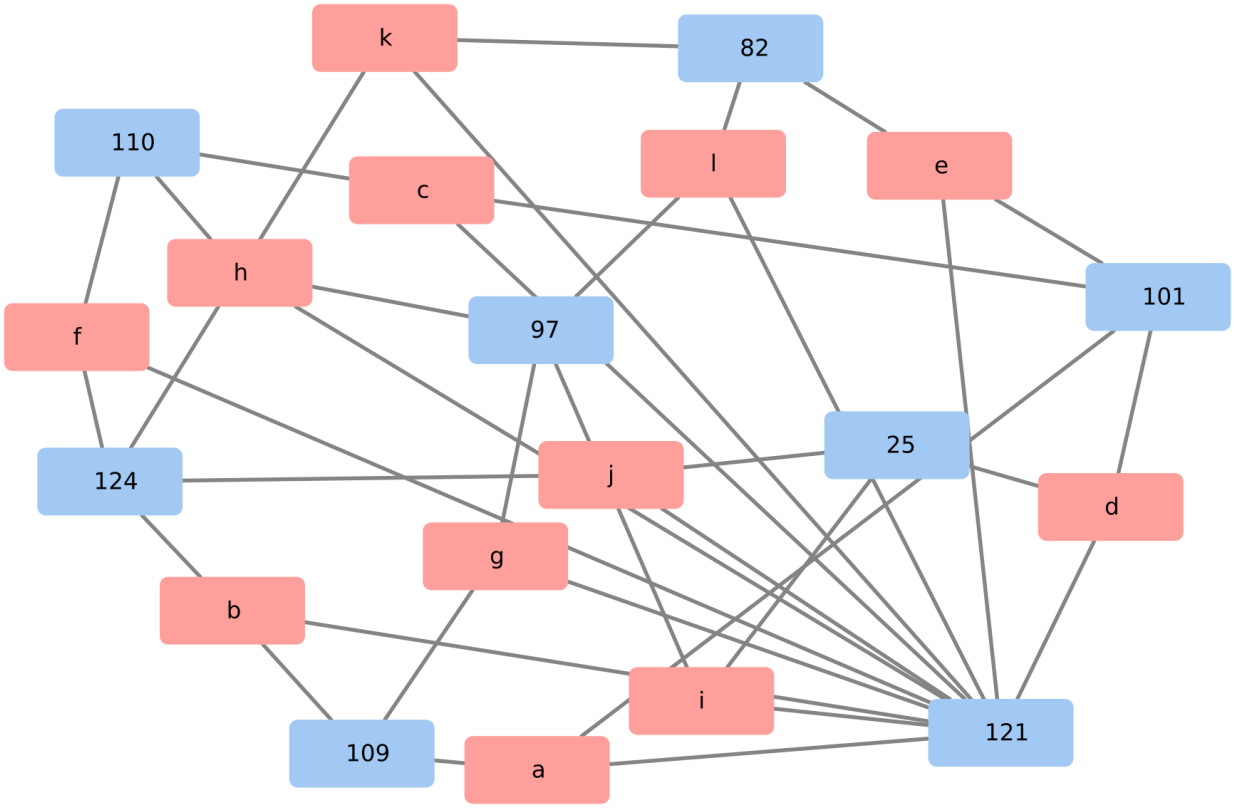


Figure D5: Bipartite graph representing minimal covers for design parameters $\alpha = 5$ and $\beta = 1$. Covers are colored in red and labeled with letters, while designs are colored in blue. All minimal covers are: a: [101, 109, 121], b: [109, 121, 124], c: [101, 110, 121], d: [25, 101, 121], e: [82, 101, 121], f: [110, 121, 124], g: [97, 109, 121], h: [97, 110, 121], i: [25, 97, 121], j: [25, 121, 124], k: [82, 121, 124], l: [82, 97, 121].



Vita

Sergio Garcia is originally from Murcia, a city of 450,000 inhabitants in southeastern Spain. After finishing high school, he began his studies in chemical engineering at the University of Murcia. This university was founded in 1272, among the first in the world, although the Chemical Engineering program began in 1995. Sergio spent his junior year abroad at the University of Tennessee Knoxville (UTK) through the International Student Exchange Program scholarship, where he became involved in undergraduate research. He then returned to finish his degree in Murcia, where he also pursued undergraduate research at the Department of Biochemistry and Molecular Biology. After completion of his undergraduate degree, he returned to UTK to pursue a PhD in chemical engineering. In his free time, Sergio enjoys playing music, meditation, and cycling.