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

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 proposed 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. In particular, we focus on microbial biocatalysis which has the potential to become a renewable and lower-cost replacement of traditional chemical synthesis processes. The proposed thesis begins with an interdisciplinary review and pespective of the concepts, methodology, and applications of modular design. Then, a conceptual and mathematical framework with associated design algorithms are developed to build modular cell biocatalysts. The proposed framework is used to design modular cell systems for the production of biofuels and biochemicals in major industrial organisms such as *Escherichia coli* and *Clostridium thermocellum*. This modular cell design approach not only brings whole-cell biocatalysis closer to being an industrially competitive technology, but also provides tools to understand the natural modular architectures of metabolic networks designed by evolution for billions of years under physical and biological constraints.

Table of Contents

List of Tables

List of Figures

Chapter 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 to use the same software for different application-dependent hardware, shaping information technology as we know it today. 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,[8] systems and synthetic biology,[72] metabolic engineering,[119, 94] and bioprocessing.[33, 122] Amid many applications to address issues related to health, energy, security, 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 engineered microorganisms, fewer than a dozen are industrially produced.[119] 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.[82, 174] Recently,

modular cell engineering has been proposed as an innovative approach to accelerate the strain engineering process, harnessing a large space of molecules derived from rich and diverse cellular metabolism and pushing whole-cell biocatalysis into an industrially competitive technology.[158, 159]

The goal of this thesis is to advance modular design principles in synthetic biology and metabolic engineering with emphasis on biocatalysis applications. To date, modular design in metabolic engineering has been mostly applied at the pathway level, where enzyme module expression is adjusted to increase target metabolite production.[9, 68, 103, 177] More recently, a system-level modular cell design (ModCell) has been proposed.[158] Unlike conventional modular pathway optimization methods that target one product, ModCell enables rapid construction of multiple production strains each synthesizing 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 **??**.A). The chassis interfaces with the modules through enzymatic and genetic synthesis machinery and precursor metabolites (Figure **??**.B). Modules contain auxiliary regulatory and metabolic pathways (Figure **??**.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 **??**.D).

Executive summary: The proposed thesis is divided into three goals: i) The development of general modular cell design principles and associated mathematical models (Chapters **??**-**??**); ii) the development of scalable algorithms to solve the mathematical models (Chapters **??**-**??**); and iii) the application of the resulting modular cell design tool to understand driving principles of natural biological modularity and to design modular biocatalyst strains based on industrially-relevant hosts and production modules (Chapters **??**, **??**, **??**, **??**). In conclusion, the outcome of this research is to bring modularity principles proven in conventional engineering to metabolic engineering, leading to increased robustness and

efficiency thereby accelerating the strain design process that remains the major roadblock for widespread industrial application of microbial catalysis.

Chapter 2

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

This chapter is based on the publication . As first author I lead its development, implementation, and writting 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.

2.1 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 to use the same software for different application-dependent hardware, shaping information technology as we know it today.{O'Regan, 2018 #135;Ajikumar, 2010 #85} 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 [8], systems and synthetic biology [72]. metabolic engineering [**?**], and bioprocessing [33, 122]. 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 [94], fewer than a dozen molecules are industrially produced [**?**]. 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 [82, 174]. 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 [159].

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.

2.2 Modularity in engineered systems

2.2.1 Basic concepts

Based on the definition by Miller and Elgard [113], 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 [135], 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 [165]. 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 2.1 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 [70].

2.2.2 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 [12, 113]. Modular design has been the core of many innovative technologies across engineering disciplines (Figure 2.1 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 [7, 80]. Likewise, modular buildings in civil engineering enable more rapid and economical construction [75]. 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 [166]. The emerging area of highly automated manufacturing also implements modular production systems [170]. In addition, modular design principles have been applied with great success in abstract engineering disciplines such as software [**?** 48] and management engineering [17]. 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 [7]. Design constraints may also limit the applicability of modularity; for example, a quantitative study [66] 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.

2.2.3 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) [14] 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 2.1 C) [63].

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) [66], node centrality ranking to identify key interactions between modules and interfaces [144], and most importantly, clustering to identify modules (Figure 2.1 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 [63].

2.3 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 [59, 167]. 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.

2.3.1 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 2.2). At the molecular level, DNA transcription activation and rate can be controlled by a variety of modular promoter elements [40]. Additionally, certain proteins are highly modular, such as enzymes (e.g., polyketide synthase and non-ribosomal peptide synthetase [64]) with modular substrate identification elements that enable biosynthesis of a large space of secondary metabolites [79]. At the cellular level, modularity is present in all biomolecule interaction networks [114], including DNAs [**?**], RNAs [147], proteins [145], and metabolites [130]. In all cases, modules are associated with specific cellular functions or pathways, which often interact in a hierarchical manner [130]. 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 [146]. These modules are hierarchically organized into submodules to integrate and contextualize specialized functions of the brain [111]; for instance, visual perception that requires the functional integration of multiple neuron clusters in the cortical columns [125]. 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 [56]. The capability of ecological networks to avoid global failure due to small perturbations has been attributed to their modular structure both theoretically [56] and experimentally [53].

2.3.2 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 19th 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 20th 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 [116] and constraint-based simulations of microbial metabolism [**?**], 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 [133**?** **?** **?**]. Graphs (or networks) can represent these systems [4] and their complex interactions, e.g., metabolites-enzymes [105, 130], genes-diseases [54], protein-protein [148], or a combination of all known protein and genetic interactions [21]. Module analysis of biological networks can provide two insights: (i) identification of modules that represent transferable and self-contained functions [114], for example, a cis-regulatory element and its associated genes [121], the subunits of a protein complex [57, 60], and the genes associated with a disease phenotype [54], and (ii) interactions among modules of a system. For example, by analyzing the metabolic networks of 43 organisms, Ravasz *et al.* [130] 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 [105] 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 [**?**].

2.3.3 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 [13, 119], 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 [100, 127], including the production of valuable metabolites, synthesis of nanomaterials, treatment of disease, and sensors to detect hazardous molecules [13]. 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.* [37] 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 [128].

2.3.4 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 [171] can help explain their contributions to system robustness. For example, the bow-tie architecture of biological networks [46] has been suggested [83] 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 [167]. 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 [77] or simultaneously [26], a modular architecture is favored. Additionally, it has been demonstrated that macroscopic [143] and molecular [139] 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 [51], 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.

2.4 Modular cell engineering

Diverse and complex cellular metabolism encompasses a large space of molecules, providing a potential path towards broader industrialization of biology [31]. 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 [**?** 159]. 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 [9, 68, 103, 177] 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 [**?**]. 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 [51, 158]. 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 2.3 A). The chassis interfaces with the modules through enzyme synthesis machinery and precursor metabolites (Figure 2.3 B). Modules contain auxiliary regulatory and metabolic pathways (Figure 2.3 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 2.3 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.

2.4.1 Recent developments in mathematical formulation of modular cell design

The primer for the modular cell design principles started with the observation [155] that the optimal design of the n-butanol- and isobutanol-producing *E. coli* cells, based on

constraint-based modeling [123] and conventional strain design methods [161], exhibited the same core metabolism (Figure 2.4 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 [158] (Figure 2.4 E). A new formulation of the modular cell design problem, ModCell2 [51], was proposed based on the framework of multi-objective optimization to capture the tradeoffs that may exist among highly diverse biochemical pathways. ModCell2 enables analysis of genome-scale metabolic networks with many production modules and design of optimal modules without extensive prior knowledge of target pathways. ModCell2 was used to demonstrate that a modular cell can be designed to exhibit (i) desirable production phenotypes for many molecules under various culturing conditions beyond the growth-coupled to product synthesis phase [84], such as the stationary-phase phase [85], and (ii) minimal trade-off between compatibility, performance, and robustness (Figure 2.4 H). These results highlight that the systems-level modularization of cellular metabolism can accelerate strain engineering without sacrificing performance requirements. 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 [169] is used to design genomes of minimal cells that can serve as chassis whereas ValveFind [124] 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. [102, 106, 118].

Future development of modular cell design tools will incorporate enzyme kinetics of cellular metabolism [81, 120] to account for potential metabolic burdens [175] in the design of modular cells and exchangeable production modules (Figure 2.4 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) [38]. The lack of experimental data on enzyme catalytic efficiencies needed for these modeling approaches can be addressed through random parameter sampling [38], *omics* integration [41, 78], and machine learning [61]. Recently developed models that predict *in-vivo* enzyme

concentrations from the genetic sequence help bridge the gap between metabolic model predictions and experimental implementation [42, 110, 134].

2.4.2 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 [76], Biocyc [20], Brenda [137], 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 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 [88, 168], 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 [39, 88]. 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 [5, 10, 11, 22, 30, 52, 86, 99, 96, 141, 162, 164, 183]. 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 [131], (ii) redesign of fermentative pathways to produce a large space of designer bioesters used as flavors, fragrances, biofuels, and solvents [90, 133] (Figure 2.4 D), (iii) repurposing of the beta-oxidation pathway for combinatorial biosynthesis of alcohols, dicarboxylic acids, hydroxyl acids, and lactones as industrial platform chemicals [24] (Figure 2.4 F), and (iv) refactoring of polyketide and isoprenoid pathways to explore a large space of secondary metabolites as drugs [3, 47, 109].

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 [**?**]. 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 [47**?** **?** **?**]. 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 [9, 68, 177]. 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.* [173] recently demonstrated a linear correlation between growth rate and product synthesis rate for computationally designed growth-coupled strains [158] (Figure 2.4 G). Pathway optimization through growth-coupled design has been applied successfully [44], and a recent computational study suggest its applicability to many different products and organisms [**?**]. 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.

2.4.3 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 [**?** 156] and (ii) bottom-up approach that seeks to create a new organism derived from a synthetic minimal cell. [65] These two methods draw the same analogy as those in the engineering modular design, classified as "product modularization" and "design with modules" [12]. 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 [51, 158] and experimental [90, 173] 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 [65], 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 [82, 174]. This suggests that modular cell design can serve as a unifying platform for rapid strain engineering (Figure 2.4 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 2.4 K). Recent developments in biosensors coupled with gene expression regulation tools [101, 112, 115, 179, 182] can achieve tunable control over the host metabolism to meet the requirements of each production module (Figure 2.4 C). In practice, such regulatory elements may be implemented in the host or as a part of specific modules.

2.5 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 [55], writing [19, 87], and editing [8] 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 2.4 L) [2, 18, 74, 104, 150]. 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.

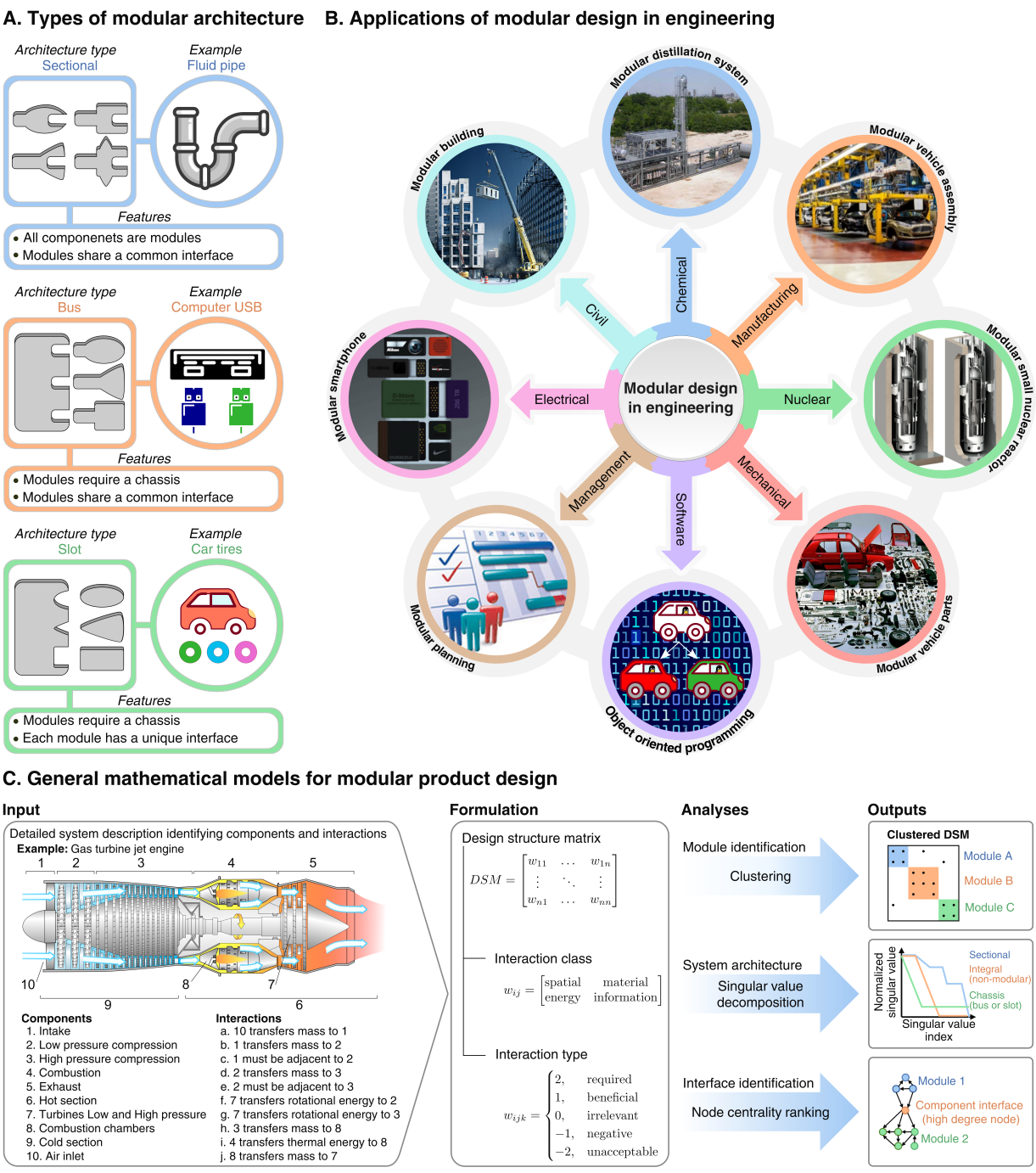


Figure 2.1: Modular design in engineering. (**A**) Common types of modular architectures. (**B**) Current applications of innovative modular design. (**C**) General mathematical framework of modular design. The input is illustrated with a gas turbine jet engine, that intakes air through the front section 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.

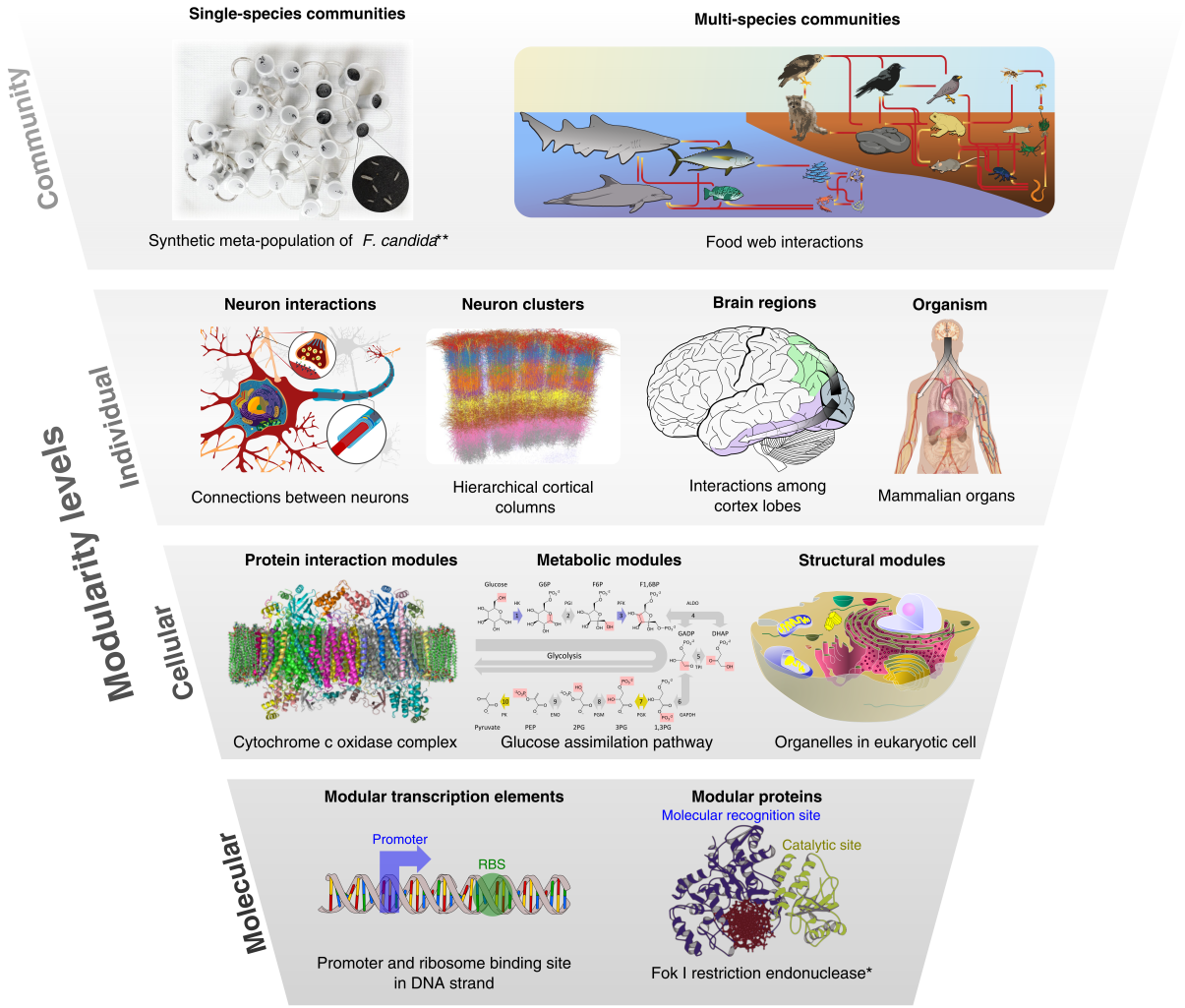


Figure 2.2: Hierarchical modularity across all scales of biology. Images marked with * and ** are adapted from Khosla and Harbury [79] and Gilarranz *et al.* [53], respectively.

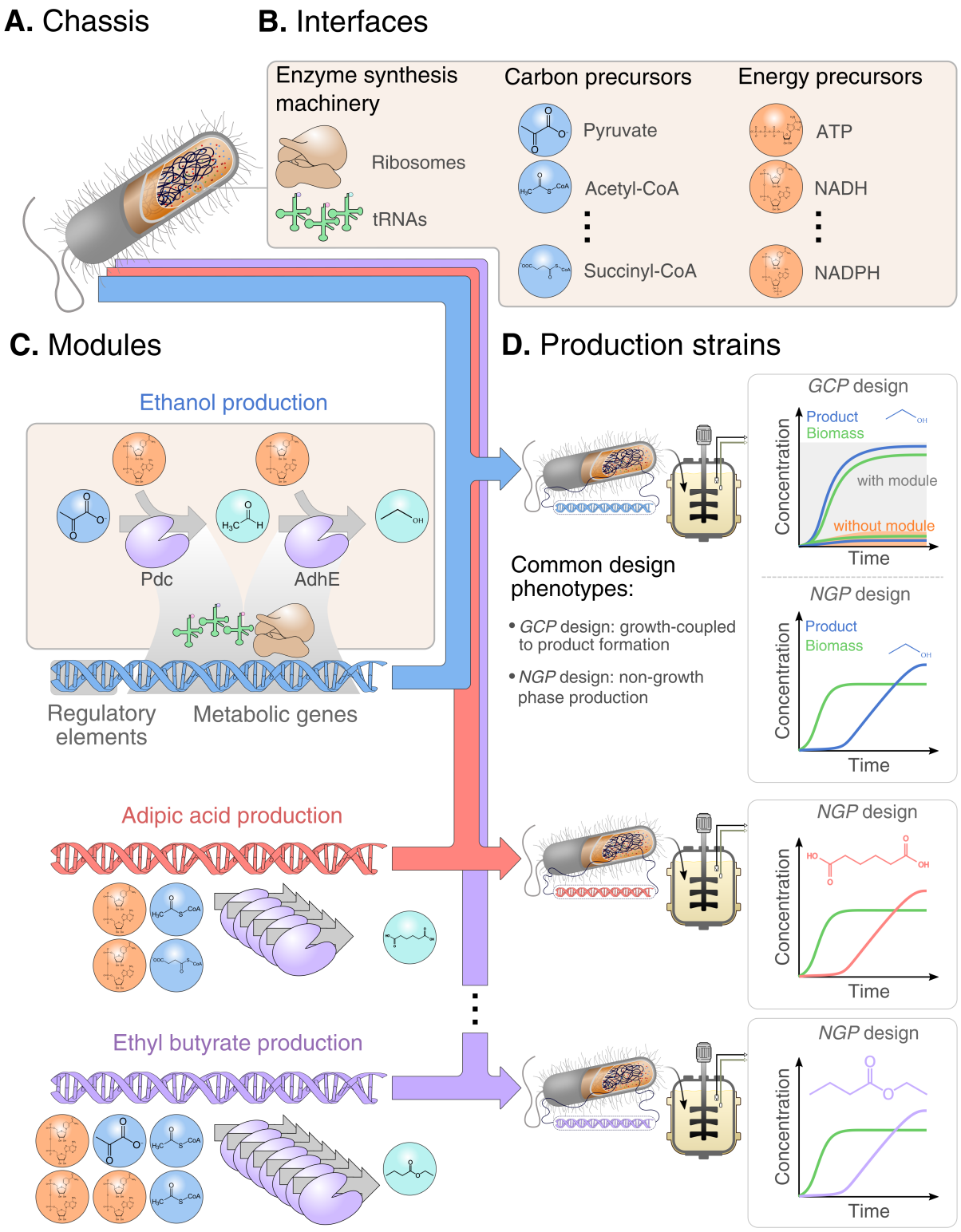


Figure 2.3: Generalized concept of modular cell design. (**A**) Modular (chassis) cell. (**B**) Interfaces. (**C**) Production modules. (**D**) Production strains. A modular cell is designed to provide the necessary precursors for biosynthesis pathway modules that are independently assembled with the modular cell to generate production strains exhibiting desirable phenotypes.

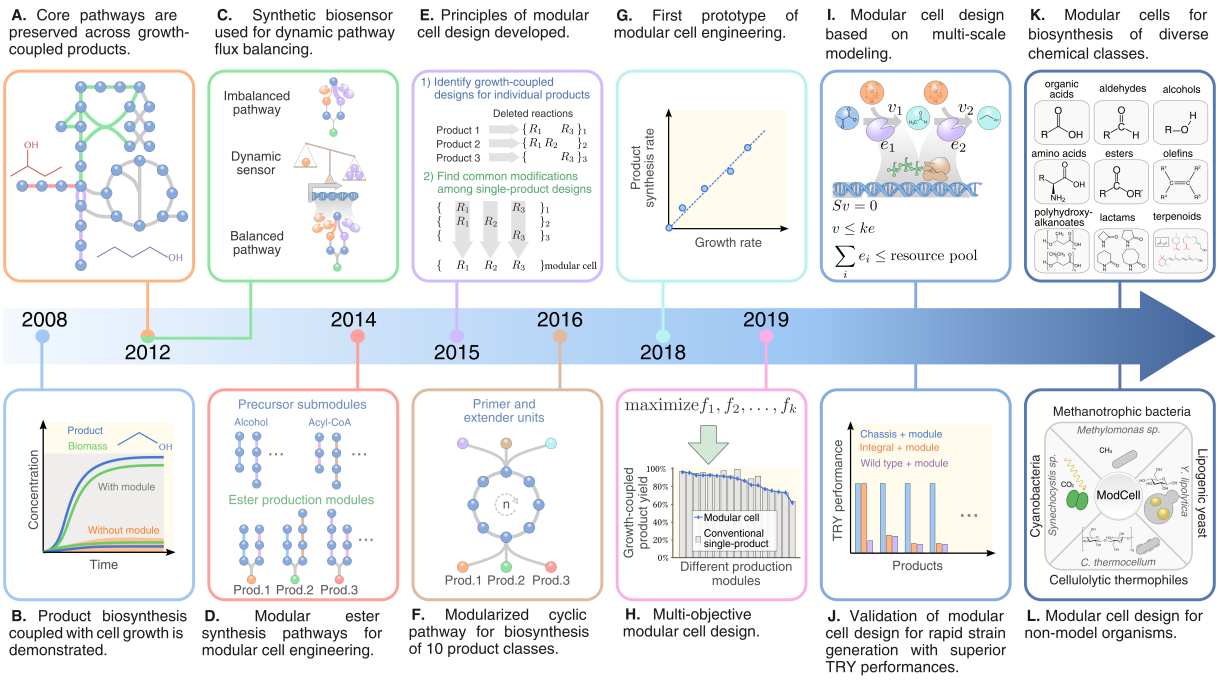


Figure 2.4: Key advances and opportunities in the design and implementation of modular cells and exchangeable production modules. (**A**) The same core metabolic pathways were revealed for butanol and isobutanol growth-coupled production based on elementary mode analysis [155]. (**B**) An *E. coli* cell with minimal metabolic functionality designed to require product (ethanol) synthesis for cell growth was experimentally validated [160]. (**C**) A dynamic sensor-regulator system was designed and implemented to balance metabolic fluxes for enhanced biosynthesis of fatty acid-derived molecules [182]. (**D**) A modular ester fermentative pathway platform derived from alcohol and acyl-CoA pathway submodules was designed and experimentally demonstrated in a chassis strain [90]. (**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 [158]. (**F**). Carbon and energy efficient cyclic pathway was developed to synthesize 10 product classes from a variety of primer and extender precursors [24]. (**G**) Prototypes of modular cell engineering were demonstrated for growth-coupled to product synthesis predicted by MODCELL [158] and pathway optimization by adaptive laboratory evolution [173]. (**H**) Multi-objective optimization-based modular cell design method, named ModCell2, was developed and demonstrated to reveal negligible trade-offs between modular and integral designs [51]. (**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.

Chapter 3

Formulation of conceptual and mathematical framework to design modular cells

This chapter is based on the publication . As first author I lead its development, implementation, and writting of this study. Supplementary Files S1 and S2 are provied in Apendix 1 and 2 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.

3.1 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 [119]. 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 [159]. Thus, innovative technologies to enable rapid and economical strain engineering are needed to harness a large space of industrially-relevant molecules [31].

The modular organization of biological systems has been a source of inspiration for synthetic biology and metabolic engineering [128, 136]. Modular pathway engineering breaks down target pathways into tractable pathway modules that can be finely tuned for optimal production of desirable chemicals [9, 177]. 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 [133] and/or elongation cycles [24**?** , 176] and hence are capable of producing a large library of unique molecules [117]. 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 [158, 159, 173].

With the arrival of steady-state, constraint-based stoichiometric models of cellular metabolism, various computational algorithms have been developed to guide strain engineering [25, 102, 178]. 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 [**?** 177, 161, 173]. 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 [15] and the other based on the obligate requirement of optimal product synthesis in any growth phase [160]. The distinction between these two types of growth coupling are also referred to weak coupling ($wGCP$) and strong coupling ($sGCP$) [84, 178].

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 [158], which guided several experimental studies [91, 90, 92, 172, 173]. It works by generating $sGCP$ strain designs for each target product based on elementary mode analysis [154], 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 [71]. 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.

3.2 Methods

3.2.1 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 3.1). 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 [158]. 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.

3.2.2 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.

$$\underset{y_j, z_{jk}}{\text{maximize}} \quad (f_1, f_2, \ldots, f_{|\mathcal{K}|})^T \quad \text{subject to} \tag{3.1}$$

$$f_k \in \arg\max \left\{ \sum_{j \in \mathcal{J}_k} c_{jk} v_{jk} \quad \text{subject to} \right. \tag{3.2}$$

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

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

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

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

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

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

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

where 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 (3.9 -3.11); 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 (3.5) 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 (3.3) is used as metabolic constraints for (3.1).[126]. 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 (3.3) and reaction flux bounds (3.4-3.5). 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 3.2).

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}) (3.5). These variables can be experimentally manipulated to constrain the desirable phenotypes of production strains as shown in Figure 3.2. 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 (3.6) , 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, (3.1) 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 3.2). 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 3.2A). The *wGCP* design objective, $f_k^{\mathrm{wGCP}} \in \{0, 1\}$, is calculated as follows:

$$f_k^{\mathrm{wGCP}} = \frac{v_{\mathrm{Pk}}^{\mu}}{v_{\mathrm{Pmaxk}}^{\mu}} \tag{3.9}$$

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 [149] or OptKnock with a tilted objective function [15, 43, 178]. In (3.9), 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 3.2B). The *sGCP* design objective,$f_k^{\mathrm{sGCP}} \in \{0, 1\}$, is calculated as follows:

$$f_k^{\mathrm{sGCP}} = \frac{v_{\mathrm{Pk}}^{\mu}}{v_{P_{\max}k}^{\mu}} \frac{v_{\mathrm{Pk}}^{\overline{\mu}}}{v_{P_{\max}k}^{\overline{\mu}}} \tag{3.10}$$

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 [158]. 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 [44**?**].

The *NGP* design aims to maximize the minimum product rate during the non-growth phase by eliminating carbon fluxes directed to biomass synthesis (Figure 3.2C). The *NGP* design objective, $f_k^{\mathrm{NGP}} \in \{0, 1\}$, is calculated as follows:

$$f_k^{\mathrm{NGP}} = \frac{v_{\mathrm{Pk}}^{\overline{\mu}}}{v_{P_{\max}k}^{\overline{\mu}}} \tag{3.11}$$

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 (3.1-3.8 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:

$$\mathcal{PS} = \{x \in \Omega : \nexists t \in \Omega,\ F(t) \prec F(x)\} \tag{3.12}$$

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 (3.2-3.8), 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***):

$$\mathcal{PF} = \{F(x) : x \in \mathcal{PS}\} \tag{3.13}$$

For the multi-objective strain design framework, the input parameters include α (3.8), 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.

3.2.3 Algorithm and implementation

ModCell2 algorithm

To solve the multi-objective optimization problem for modular cell engineering, we used multi-objective evolutionary algorithms (MOEAs) [29]. MOEAs were selected because they can efficiently handle linear and non-linear problems and do not require preferential specification of design objectives [108]. 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 [35] 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 [62, 138] and F2C2 0.95b [89] 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 [158], 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 [158], and iii) a genome-scale metabolic network of *E. coli* (i.e., iML1515 [43]) 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 [**?**]. 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 titlted objective function [107] to generate *wGCP* designs for each production network, using the open-source algebraic modeling language Pyomo [58]. 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`.

3.2.4 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 [83]. 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:

$$CD = C_{\text{initial}} - C_{\text{final}} \tag{3.14}$$

$$CL = \frac{C_{\text{initial}} - C_{\text{final}}}{C_{\text{initial}}} \tag{3.15}$$

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 [16].

3.3 Results and discussion

3.3.1 Illustrating ModCell2 for modular cell design of a simplified network

An example parent network, adapted from [158], was used to illustrate ModCell2 (Figure 3.3A). Inputs for the multi-objective optimization problem include i) three production networks (Figure 3.3B), comprising of one endogenous production module (module 1) and two heterologous production modules (modules 2 and 3) and ii) design parameters (Figure 3.3C), 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 3.3D). 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 3.3D). 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

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.

3.3.2 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 [158] 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 3.4A).

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 3.4B, 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

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 3.4E). The desirable phenotypic spaces can be further constrained for many products if α is increased from 5 to 6 (Figure 3.4F). 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 3.4G). 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 3.5B). 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 3.5C, 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.

3.3.3 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 3.6A, 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 3.6A, 6C, Supplementary File S4). Among reaction deletions, *LdhA* (86% of designs), *Pta* (38%), and *AdhE* (25%) are the most frequent deletion reactions (Figure 3.6B). This finding is consistent with a comprehensive survey of metabolic engineering publications [174] 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 3.6A). These products together were compatible with different modular cells and exhibited metabolic similarity

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 3.6D). 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 [173].

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 3.6E). 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 [32], 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 [85]. Using ModCell2, we investigated the feasibility to design optimal strains to toggle switch desirable production phenotypes. To design a $wGCP \rightarrow NPG$ metabolic switch, we first used our reference *wGCP* design as a parent strain (Figure 3.7A) 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 3.7B). 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 3.7C). 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.

3.4 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.

Figures

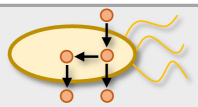
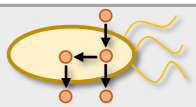
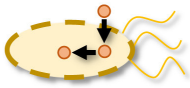
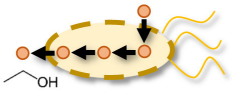
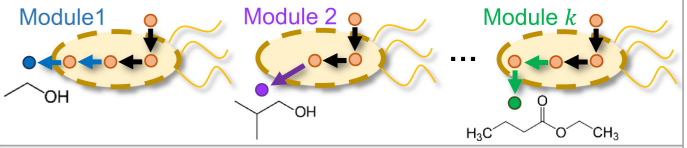
Features	Conventional strain engineering	Modular cell engineering
Parent strain		
Modular cell	Absent	 Optimized common production phenotypes
Exchangeable modules	1	k
Optimal production strains		
Design-build-test cycle	Repeated for every new product	One time for many products

Figure 3.1: 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.

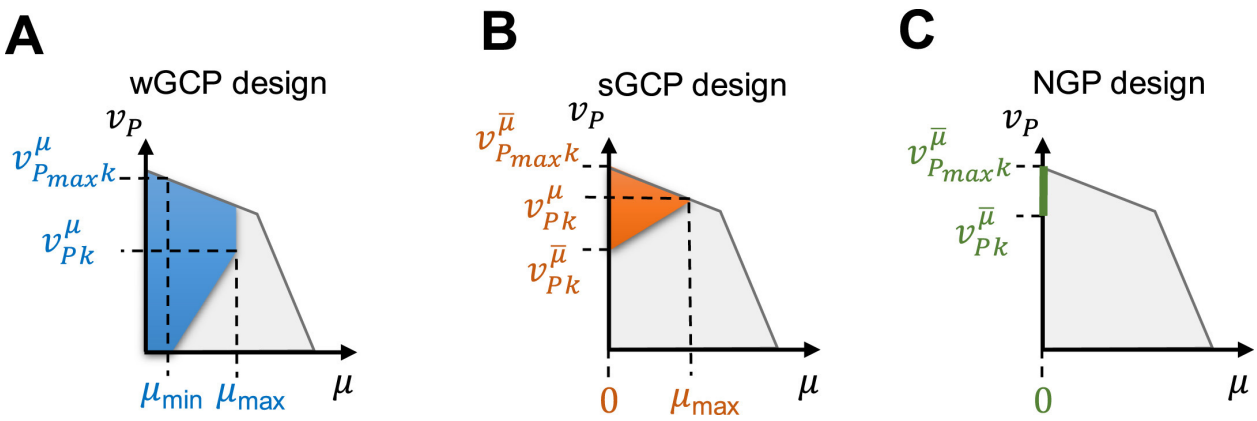
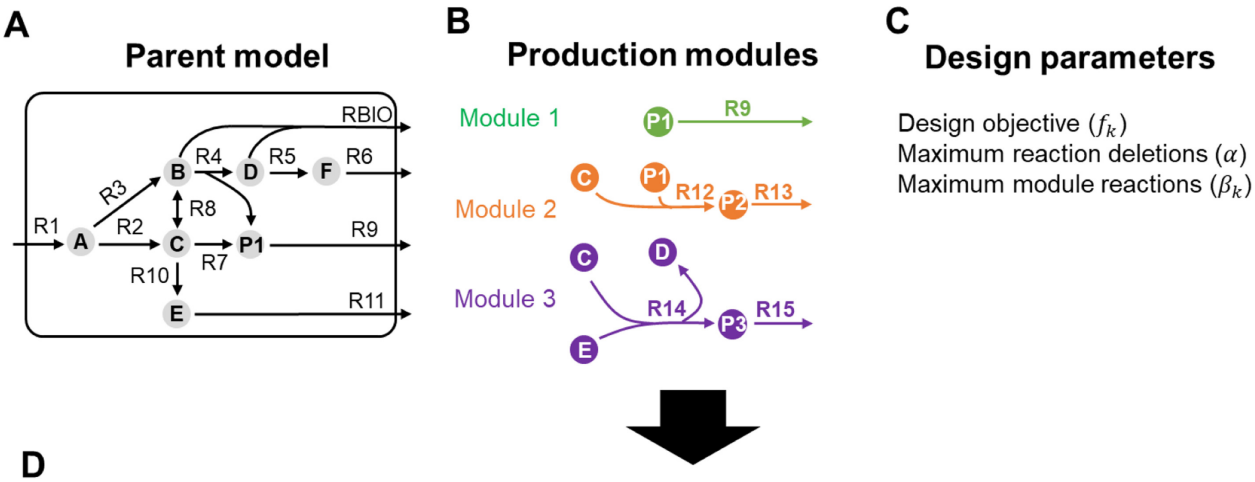


Figure 3.2: Graphical representation of phenotypic spaces for different strain design objectives including **(A)** weak growth coupling ($wGCP$), **(B)** strong growth coupling ($sGCP$), and **(C)** no-growth production (NPG). v^{μ}_{Pk} is the minimum product formation rate at the maximum growth rate 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.



D

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

Objective	α	β_k	d	Reaction deletions	Module reactions P1	P2	P3	f_k P1	P2	P3	Prod. yield vs growth P1	P2	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 3.3: 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.

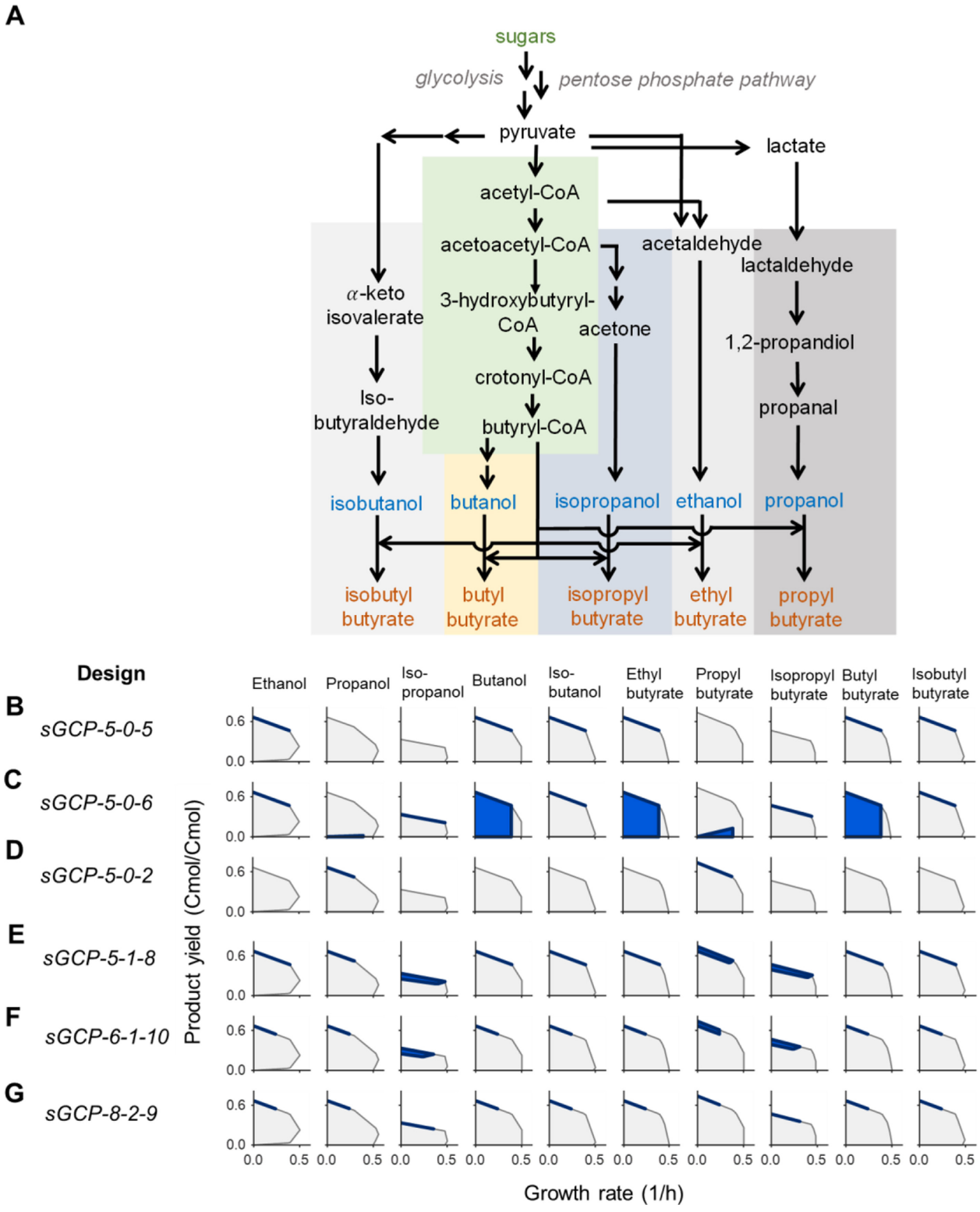


Figure 3.4: The 2-D metabolic phenotypic spaces of different *sGCP* designs using the core metabolic model. (**A**) Metabolic map, (**B**) *sGCP-5-0-5* design, (**C**) *sGCP-5-0-6* design, (**D**) *sGCP-5-0-2* design, (**E**) *sGCP-5-1-8* design, (**F**) *sGCP-6-1-10* design, and (**G**) *sGCP-8-2-9* design. For each panel, the gray and blue areas correspond to the phenotypic spaces of the wildtype and the optimal production strain, respectively.

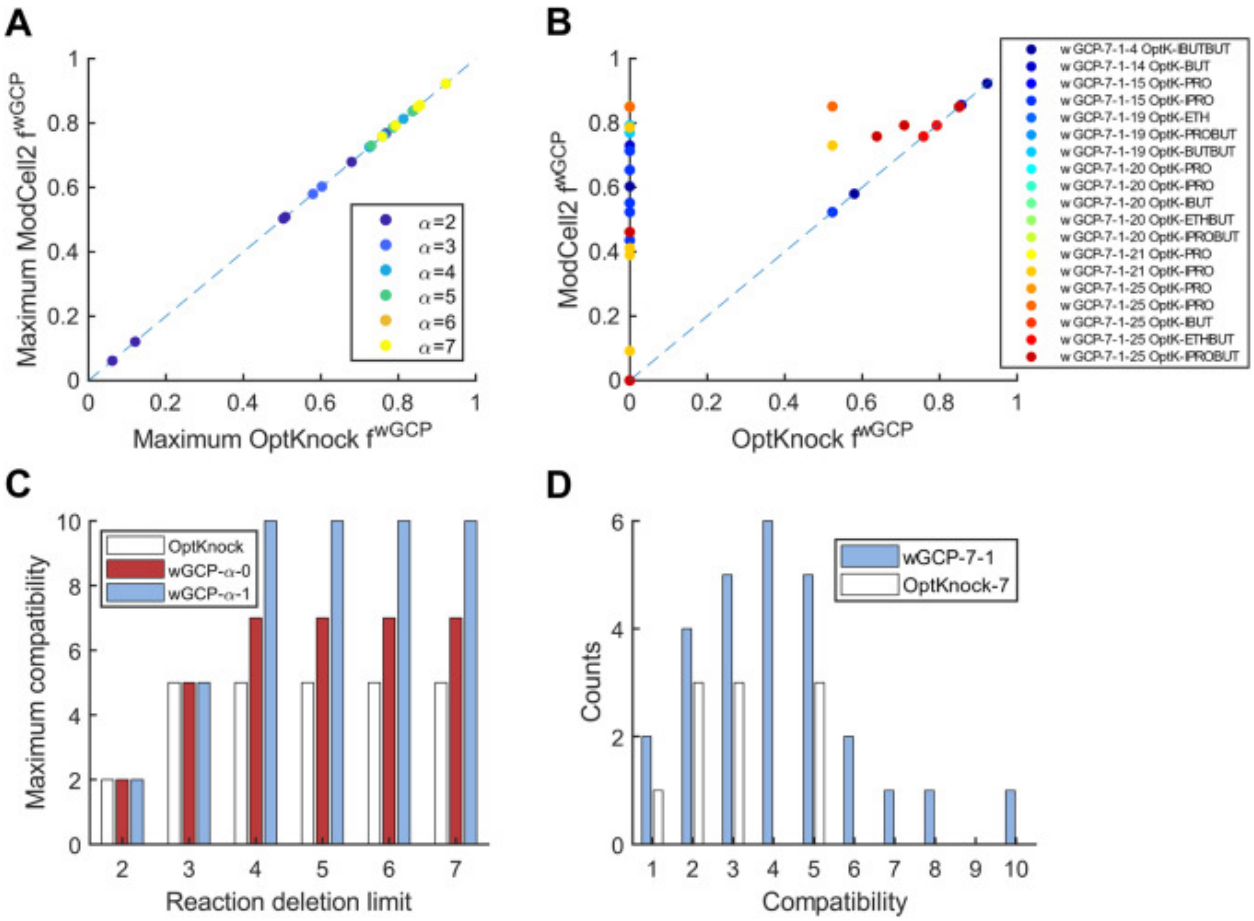


Figure 3.5: Comparison of strain design by OptKnock and Modcell2. (**A**) A correlation between the maximum objective values for each product generated by OptKnock and the equivalent values attained by ModCell2. Each point is colored based on the number of reaction deletions, with warmer colors corresponding 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).

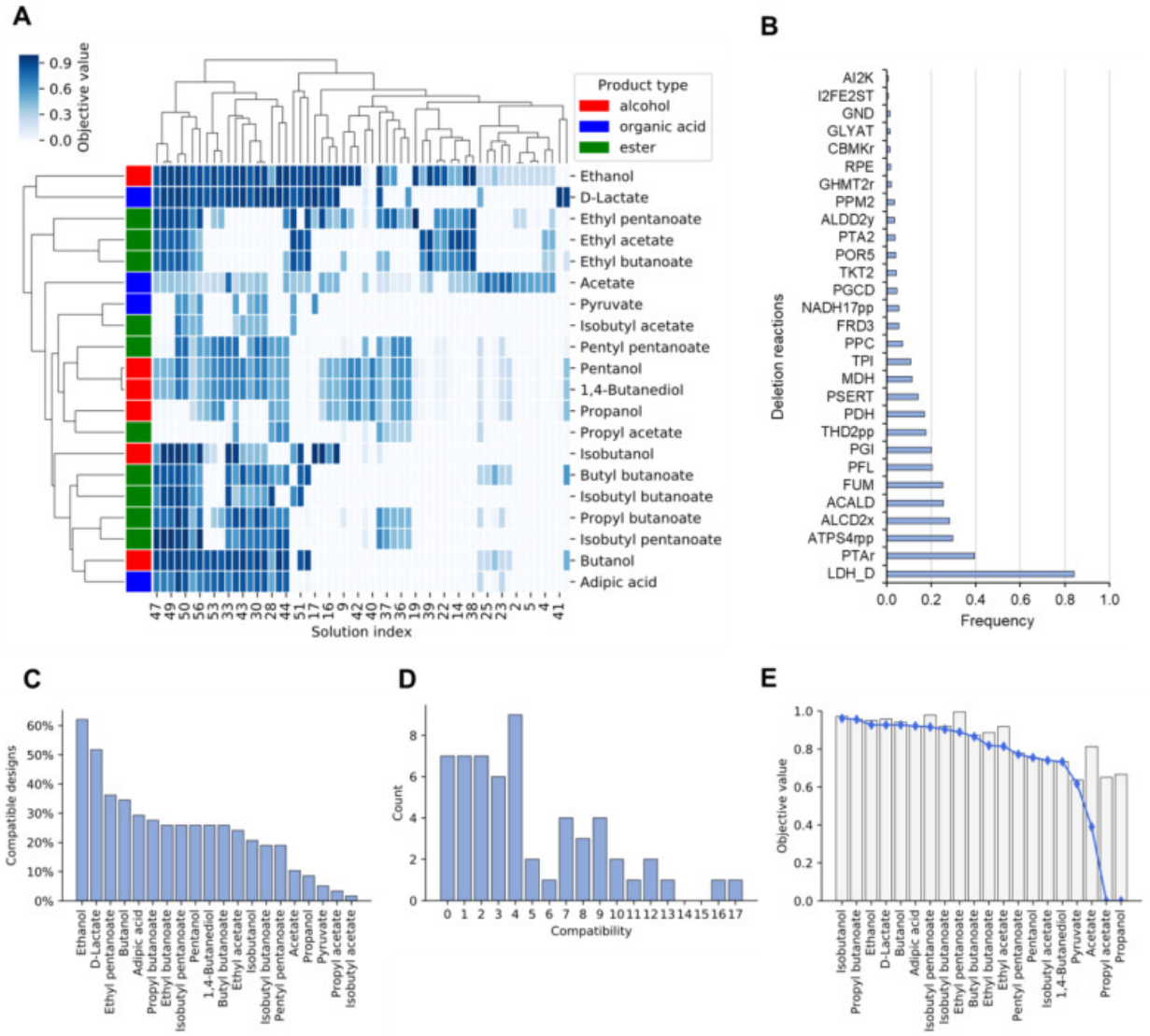


Figure 3.6: Analysis of *wGCP* designs with genome-scale model. **(A)** Pareto front of *wGCP-4-0-d*. The columns correspond to different designs labeled by their design index, *d*, where the rows correspond to different products. **(B)** Frequency of the top deletion reactions. **(C)** Product compatibility 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.

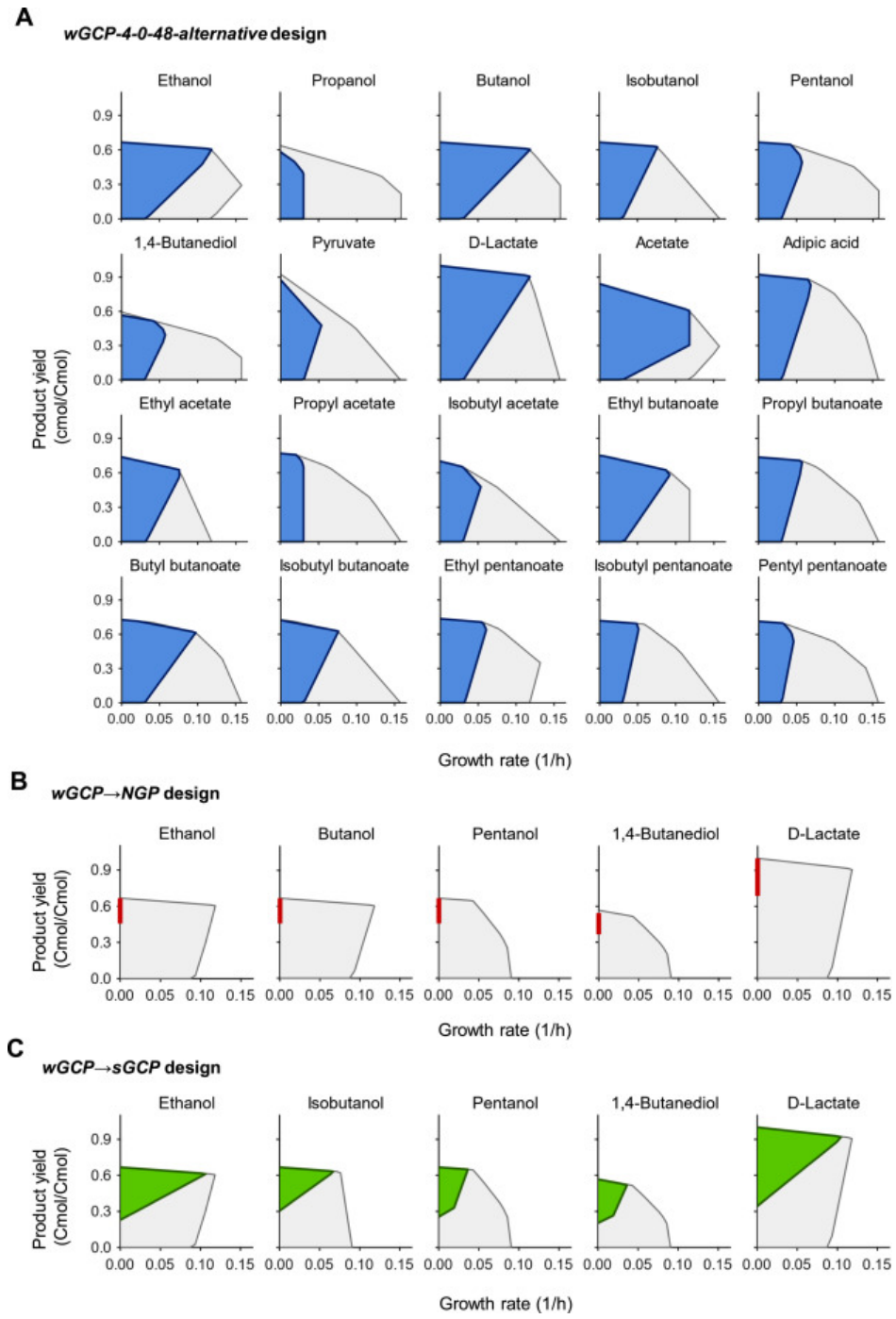


Figure 3.7: Production phenotypes of (**A**) the wild type (gray) and the representative, highly-compatible design *wGCP-4-0-48-alternative* (blue), (**B**) the *wGCP→NPG* design, *sup-NGP-5-0-23*, and (**C**) the *wGCP→sGCP* design, *sup-sGCP-6-0-39*.

Chapter 4

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

This chapter is based on the publication . As first author I lead its development, implementation, and writting 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.

4.1 Introduction

Multi-objective optimization is a powerful mathematical toolbox widely used in engineering disciplines to solve problems with multiple conflicting design objectives [27]. 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 [129]. 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 [159, 94]. However, the current strain design process is prohibitively laborious and expensive for broad industrial application [119]. To overcome this challenge, recent studies have proposed the application of modular design principles commonly used in engineering [12] to microbial biocatalysis [155, 158, 51, 50]. 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 [157, 155, 173] and esters [90, 91, 92, 172, 93] 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 [28], (ii) compatibility with non-linear objectives and constraints, and (iii) unbiased sampling of Pareto optimal solutions without a need to pre-specify objective preference [108]. 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 [95]. For the case of ModCell problem, the popular MOEA NSGAII [mat, 73] was used to design a modular cell under 20 different production modules [51]. 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 [35] and SPEA2 [185]. New MOEAs are benchmarked against libraries of artificial problems with known solutions [184, 36], 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.

4.2 Methods

4.2.1 Multi-objective modular cell design

Modular cell design enables rapid generation of optimal production strains with desirable phenotypes from a modular (chassis) cell [51], 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 [51]:

$$\max_{y_j, z_{jk}} \quad (f_1, f_2, \ldots, f_{|\mathcal{K}|})^T \quad \text{s.t.} \tag{4.1}$$

$$f_k \in \arg\max \left\{ \frac{1}{f_k^{max}} \sum_{j \in \mathcal{J}_k} c_{jk} v_{jk} \quad \text{s.t.} \right. \tag{4.2}$$

$$\sum_{j \in \mathcal{J}_k} S_{ijk} v_{jk} = 0 \qquad \text{for all } i \in \mathcal{I}_k \tag{4.3}$$

$$l_{jk} \leq v_{jk} \leq u_{jk} \qquad \text{for all } j \in \mathcal{J}_k \tag{4.4}$$

$$l_{jk} d_{jk} \leq v_{jk} \leq u_{jk} d_{jk} \qquad \text{for all } j \in \mathcal{C} \tag{4.5}$$

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

$$z_{jk} \leq (1 - y_j) \qquad \text{for all } j \in \mathcal{C},\ k \in \mathcal{K} \tag{4.6}$$

$$\sum_{j \in \mathcal{C}} z_{jk} \leq \beta_k \qquad \text{for all } k \in \mathcal{K} \tag{4.7}$$

$$\sum_{j \in \mathcal{C}} (1 - y_j) \leq \alpha \tag{4.8}$$

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 (4.1). 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 (4.2-4.5) [123]. 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 (4.2) (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 (4.3), where S_{ijk} represents the stoichiometric coefficient of metabolite i in reaction j of production network k, (ii) flux bound constraints (4.4) that determine reaction reversibility

and available substrates, where l_{jk} and u_{jk} are lower and upper bounds respectively, and (iii) genetic manipulation constraints (4.5), 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 (4.3-4.4), 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 (4.6). The numbers of module-reaction additions and reaction deletions in the chassis are constrained by the parameters β_k (4.7) and α (4.8), respectively, to avoid unnecessary genetic manipulations that are generally time-consuming to implement and can lead to unforeseen phenotypes.

4.2.2 Optimal solutions for a multi-objective optimization problem

Optimal solutions for a MOP (4.1-4.8) 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 (4.2-4.8), 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:

$$PS := \{x \in X : \nexists\, x' \in X, F(x') \prec F(x)\} \tag{4.9}$$

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

$$PF := \{F(x) : x \in PS\} \tag{4.10}$$

4.2.3 MOEA selection

To find the best MOEAs for ModCell2, we evaluated a recent and comprehensive set of MOEAs implemented in the PlatEMO platform [152]. 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 4.1). It should be noted that gamultiobj is an alternative implementation of the NSGAII algorithm available in Matlab.

Table 4.1: Summary of MOEAs used in this study

Abbreviation	Name	Notes	Reference
NSGAII	Non-dominated sorting genetic algorithm 2	Highly applied MOEA	[35]
gamultiobj	Matlab implementation of NSGAII	Used in the original ModCell2 study[51]	[mat]
MOEAIGDNS	Multi-objective evolutionary algorihtm based on an enhanced inverted generational distance metric	General MOEA with an implementation that works well with discrete variables	[153]
ARMOEA	Adapation to reference points multi-objective evolutionary algorithm	Many-objective EA based on MOEAIGDNS	[151]
EFRRR	Ensemble fitness ranking with ranking restriction	Many-objective EA	[181]
MaOEADDFC	Many-objective evolutionary algorithm based on directional diversity and favorable convergence	Many-objective EA	[23]
SPEAR	Strength Pareto evolutionary algorithm based on reference direction	Many-objective EA	[69]
tDEA	θ-dominance evolutionary algorithm	Many-objective EA	[180]
BiGE	Bi-goal evolution	Many-objective EA	[98]
NSGAIII	Non-dominated sorting genetic algorithm 3	Many-objective EA	[34]
SPEA2SDE	Strength Pareto evolutionary algorithm 2 with shift-based density estimation	Many-objective EA	[97]

4.2.4 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 [132]. 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 [140], 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 4.1a):

$$HV = \bigcup_{i \in I} \text{Volume}(\text{Box}(\mathcal{PF}_i, \vec{1})) \tag{4.11}$$

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 4.1b):

$$GD = \frac{\sum_{i \in I} d_i}{|I|} \tag{4.12}$$

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 4.1b):

$$IGD = \frac{\sum_{k \in K} \hat{d}_k}{|K|} \tag{4.13}$$

Δ_p: This metric combines *GD* and *IGD* metric and thus has superior properties [140]:

$$\Delta_p = \max(GD, IGD) \tag{4.14}$$

C: This metric determines the fraction of $\mathcal{PF}^*$ captured by the solution $\mathcal{PF}$ (Figure 4.1c):

$$C = \frac{|\mathcal{PF} \cap \mathcal{PF}^*|}{|\mathcal{PF}^*|} = \frac{|\{k \in K : \exists i \in I \text{ such that } \mathcal{PF}^*_{kj} = \mathcal{PF}_{ij} \text{ for all } j \in J\}|}{|K|} \tag{4.15}$$

ϵ: This metric is the additive epsilon indicator [186] 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 4.1d):

$$\epsilon = \inf\{\epsilon \in \mathbb{R} : \text{for all } i \in I \, \exists k \in K \text{ such that } \mathcal{PF}_{ij} + \epsilon \geq \mathcal{PF}^*_{kj} \text{ for all } j \in J\} \tag{4.16}$$

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 4.1e). 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 4.1f).

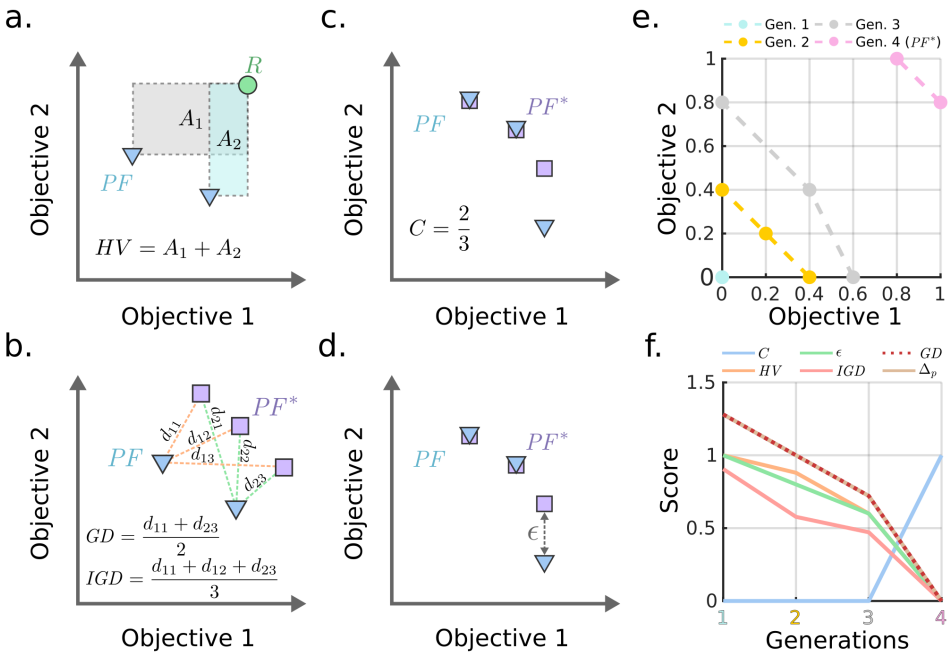


Figure 4.1: (**a-d**) Conceptual illustration of performance metrics of MOEAs for a two-objectives design problem. PF and PF^* correspond to the Pareto front approximation and the best Pareto front available, respectively. The reference point R must always dominate all solutions in PF. (**e-f**) An example of Pareto fronts with 2 dimensions and associated metrics. The 4th generation corresponds to $\mathcal{PF}^*$ used as a reference for comparison.

4.2.5 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.

4.2.6 Metabolic models

For all simulations, we used a core *E. coli* model, downloaded from the BiGG database (https://bigg.ucsd.edu) [81], that captures the most important metabolic pathways [123]. The product synthesis pathways for each module correspond to native *E. coli* pathways together with well-characterized heterologus pathways for the synthesis of propanol [163],

butanol [142], isobutanol [6], and pentanol [163]. The metabolic reactions associated with these pathways are described in the software implementation (Supplementary Material 1).

4.2.7 Implementation

The simulations were performed using the ModCell2 software framework [51]. The MOEAs are implemented in the PlatEMO Matlab library [152], except *gamultiobj* which is implemented as part of the Matlab Optimization Toolbox. HV was calculated using the *hv package* [45]. 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.

4.3 Results and Discussion

4.3.1 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 4.2a). 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 4.2b, e, h, i) and by *gamultiobj* after 150 generations (Figure 4.2k), while the remaining algorithms could not attain C values above 0.8 (Figure 4.2c, d, f, g, j, l). In particular, MaOEADDFC and BiGE obtained the worst C, ϵ, and Δ_p values (Figure 4.2m). 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 4.2o). Due to the simplicity of the problem, every algorithm except MaOEADDFC, tDEA, and BiGE converged to very similar Pareto fronts (Figure 4.2n-x), and 5 of them reached $C = 1$, indicating convergence to the reference Pareto front (Figure 4.2y).

4.3.2 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 4.3a-k). The SPEA2SDE algorithm displayed consistent improvement of C as generations progressed, and quickly reached the smallest values of ϵ and Δ_p (Figure 4.3h). 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 4.3a, d). However, SPEA2SDE approached $C \cong 0.6$, which is twice the value reached by the next best-performing methods (Figure 4.3l). 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.

4.3.3 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 [67]. 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 4.4a, d, g, h, j). These 5 algorithms also yielded very similar final values across all metrics (Figure 4.4l). The remaining algorithms converged to considerably lower C values (Figure 4.4b, c, e, f, i, k).

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 4.5). 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.

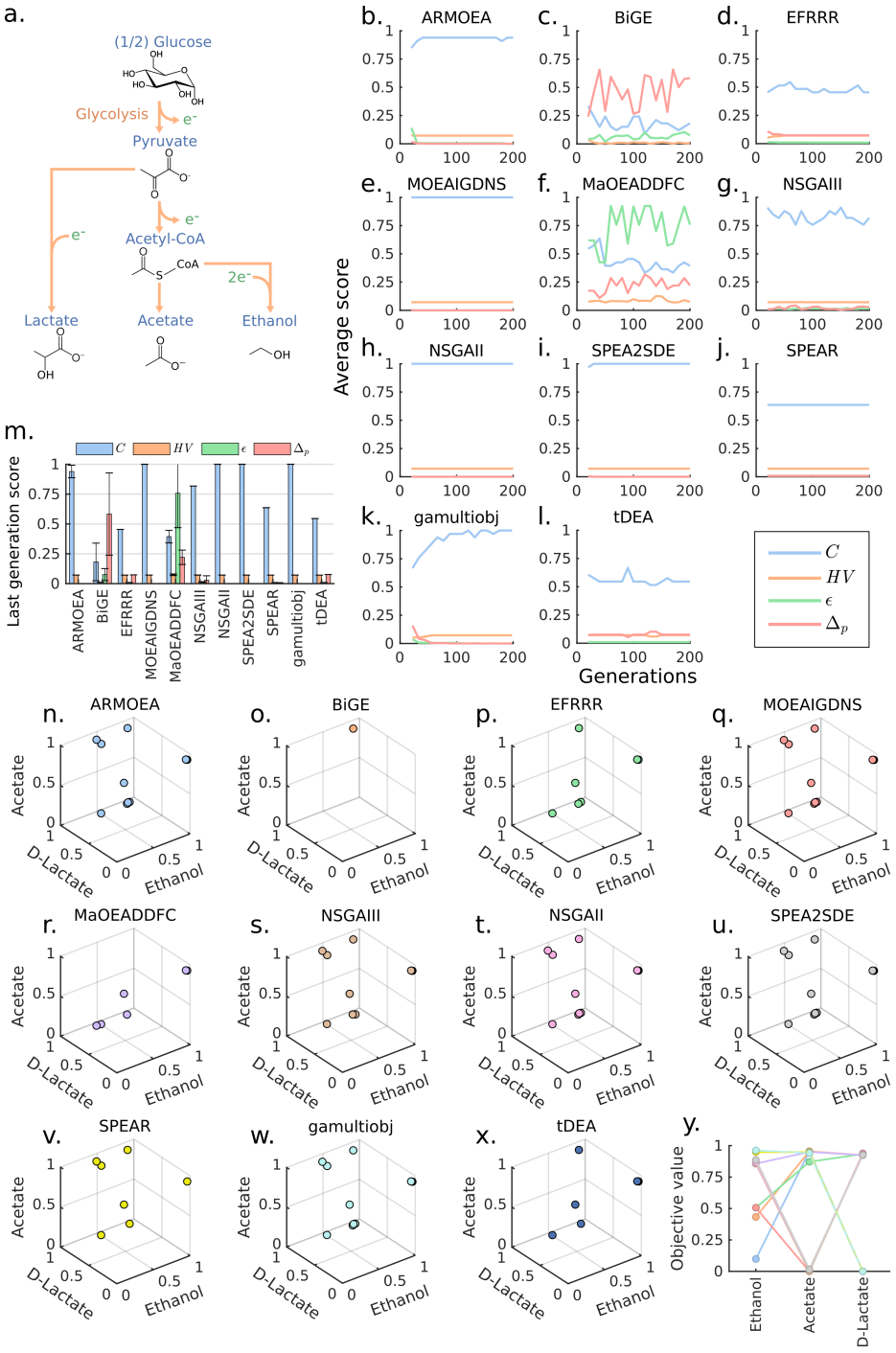


Figure 4.2: Comparison of MOEAs for a 3-objectives design problem. (**a**) The simplified metabolic pathways for conversion of glucose to the target products. Reducing equivalents are presented with e^-. (**b-l**) Generation-dependent performance metrics for various MOEAs. (**m**) Performance metrics 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.

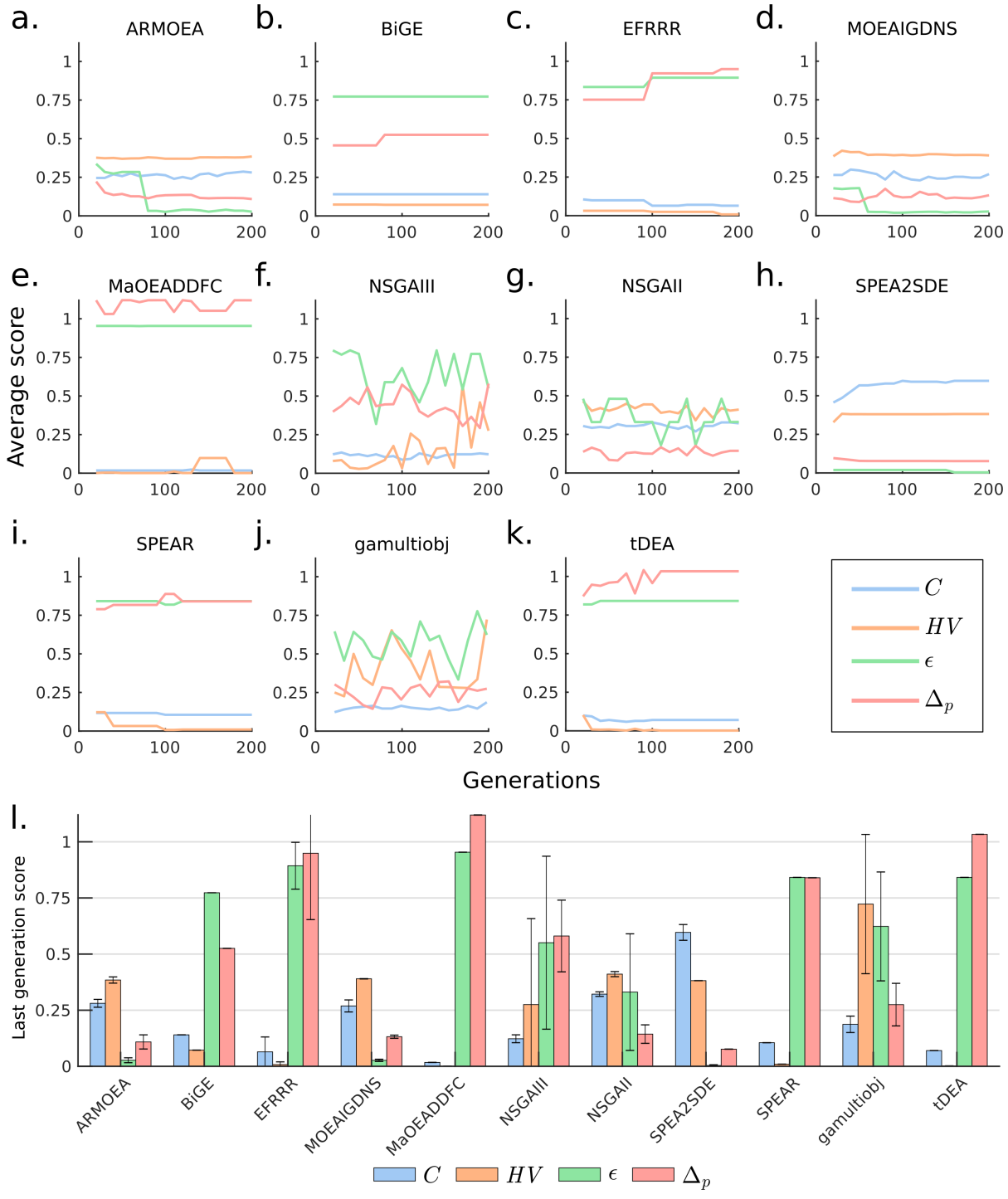


Figure 4.3: Comparison of MOEAs for a 10-objective design problem. (**a-k**) Generation-dependent performance metrics for various MOAEs. (**l**) Performance metrics for various MOEAs at the last generation.

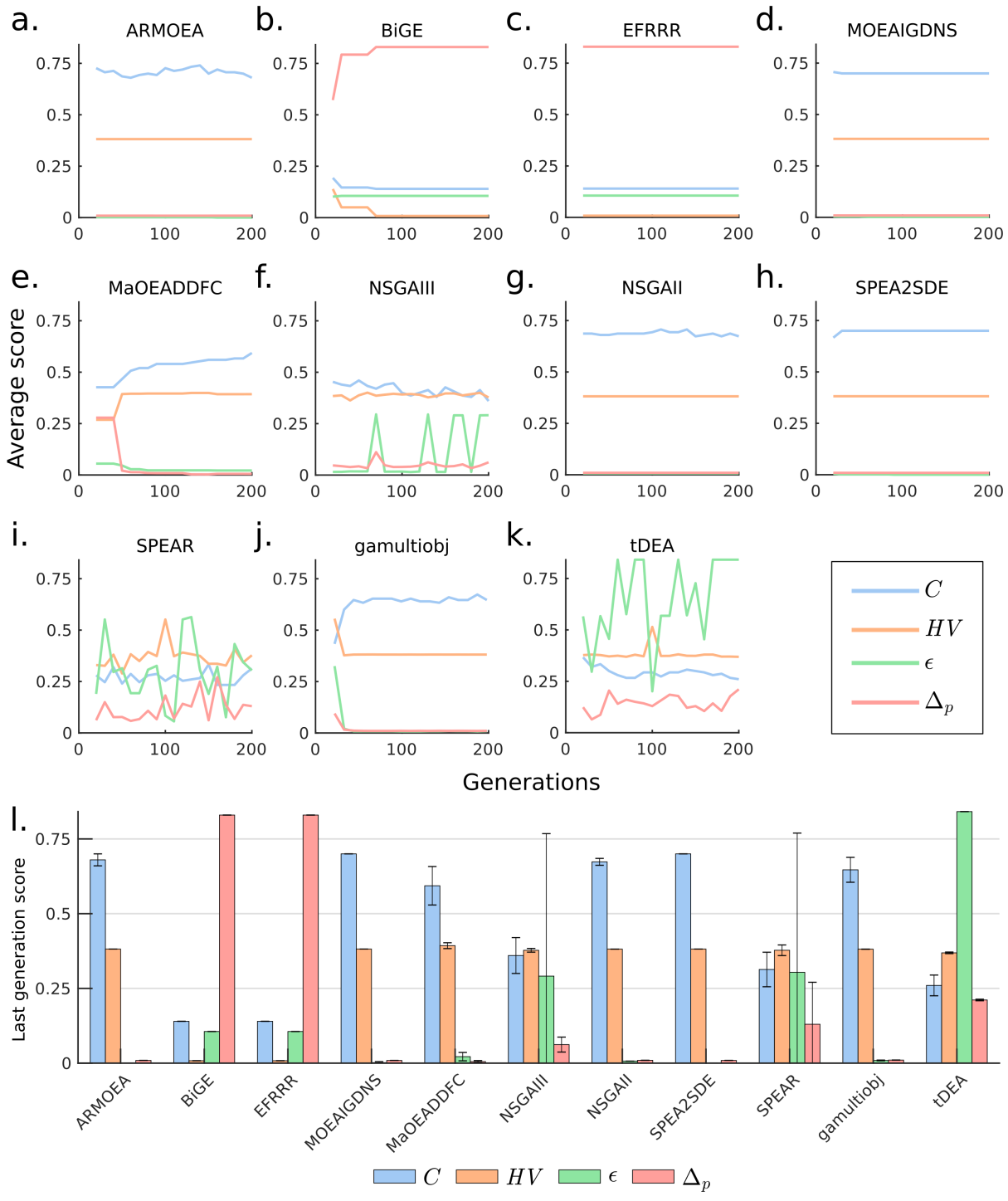


Figure 4.4: Comparison of MOEAs for a 10-objectives design problem with larger population sizes (**a-k**) Generation-dependent performance metrics for various MOAEs. (**l**) Performance metrics for various MOEAs at the last generation.

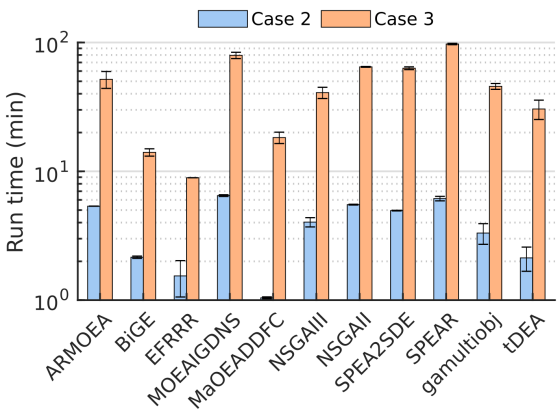


Figure 4.5: Wall-clock run times for the 10-objectives design problem with population sizes of 100 (Case 2) and 1000 (Case 3).

4.4 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.

Chapter 5

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

Chapter 6

Genome-scale metabolic network reconstruction of *C. thermocellum* to design modular cells for consolidated bioprocessing

Chapter 7

Identification of generalized modularity principles in natural systems with many objectives

Chapter 8

Conclusions

Bibliography

[mat] Matlab documentation gamultiobj algorithm. https://www.mathworks.com/help/gads/gamultiobj-algorithm.html. Accessed: 2019-02-04. 50, 55

[2] 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. 18

[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. 15

[4] Alon, U. (2003). Biological networks: the tinkerer as an engineer. *Science*, 301(5641):1866–1867. 10

[5] 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. 15

[6] 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. 59

[7] 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. 7

[8] Barrangou, R. and Doudna, J. A. (2016). Applications of crispr technologies in research and beyond. *Nature biotechnology*, 34(9):933. 1, 5, 18

[9] 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. 2, 12, 16, 24

[10] 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. 15

[11] 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. 15

[12] 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. 7, 17, 49

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

[14] Browning, T. R. (2016). Design structure matrix extensions and innovations: a survey and new opportunities. *IEEE Transactions on Engineering Management*, 63(1):27–52. 8

[15] 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. 25, 29

[16] 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. 34

[17] 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. 7

[18] Carroll, A. L., Case, A. E., Zhang, A., and Atsumi, S. (2018). Metabolic engineering tools in model cyanobacteria. *Metabolic engineering*, 50:47–56. 18

[19] 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. 18

[20] 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. 15

[21] 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. 10

[22] 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. 15

[23] 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. 55

[24] 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. 15, 22, 24

[25] Chowdhury, A., Zomorrodi, A. R., and Maranas, C. D. (2015). Bilevel optimization techniques in computational strain design. *Computers & Chemical Engineering*, 72:363–372. 25

[26] Clune, J., Mouret, J.-B., and Lipson, H. (2013). The evolutionary origins of modularity. *Proc. R. Soc. B*, 280(1755):20122863. 12

[27] Coello, C. A. C. and Lamont, G. B. (2004). *Applications of multi-objective evolutionary algorithms*, volume 1. World Scientific, Singapore. 49

[28] 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

[29] 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. 31

[30] 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. 15

[31] Connelly, T., Chang, C., Clarke, L., Ellington, A., Hillson, N., Johnson, R., Keasling, J., Laderman, S., Ossorio, P., and Prather, K. (2015). Industrialization of biology: A roadmap to accelerate the advanced manufacturing of chemicals. 12, 24

[32] 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. 39

[33] Cramer, S. M. and Holstein, M. A. (2011). Downstream bioprocessing: recent advances and future promise. *Current Opinion in Chemical Engineering*, 1(1):27–37. 1, 5

[34] 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. 55

[35] 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. 31, 50, 55

[36] 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. 50

[37] Del Vecchio, D., Ninfa, A. J., and Sontag, E. D. (2008). Modular cell biology: retroactivity and insulation. *Molecular systems biology*, 4(1):161. 11

[38] 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. 14

[39] Dugar, D. and Stephanopoulos, G. (2011). Relative potential of biosynthetic pathways for biofuels and bio-based products. *Nature Biotechnology*, 29(12):1074–1078. 15

[40] Dynan, W. S. (1989). Modularity in promoters and enhancers. *Cell*, 58(1):1–4. 9

[41] 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. 14

[42] 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. 15

[43] 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. 29, 32

[44] 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. 16, 29

[45] 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. 59

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

[47] 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. 15, 16

[48] Gancarz, M. (2003). *Linux and the Unix philosophy*. Digital Press. 7

[49] 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).

[50] Garcia, S. and Trinh, C. T. (2019b). Modular design: Implementing proven engineering principles in biotechnology. *Biotechnology Advances*, 37(7):107403. 49

[51] Garcia, S. and Trinh, C. T. (2019c). Multiobjective strain design: A framework for modular cell engineering. *Metabolic Engineering*, 51. 12, 13, 14, 17, 22, 49, 50, 51, 55, 59

[52] 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. 15

[53] 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. 9, 20

[54] 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. 10

[55] 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. 18

[56] Grilli, J., Rogers, T., and Allesina, S. (2016). Modularity and stability in ecological communities. *Nature communications*, 7:12031. 9

[57] 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. 10

[58] 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. 32

[59] Hartwell, L. H., Hopfield, J. J., Leibler, S., and Murray, A. W. (1999). From molecular to modular cell biology. *Nature*, 402(6761supp):C47. 8

[60] 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. 10

[61] 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. 14

[62] Heirendt, L., Arreckx, S., Pfau, T., Mendoza, S. N., Richelle, A., Heinken, A., Haraldsdottir, H. S., Keating, S. M., Vlasov, V., Wachowiak, J., et al. (2017). Creation and analysis of biochemical constraint-based models: the cobra toolbox v3. 0. *arXiv preprint arXiv:1710.04038*. 31

[63] 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. 8

[64] 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. 9

[65] 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. 17

[66] 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. 7, 8

[67] 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. 60

[68] Jeschek, M., Gerngross, D., and Panke, S. (2017). Combinatorial pathway optimization for streamlined metabolic engineering. *Current opinion in biotechnology*, 47:142–151. 2, 12, 16

[69] 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. 55

[70] Jose, A. and Tollenaere, M. (2005). Modular and platform methods for product family design: literature analysis. *Journal of Intelligent manufacturing*, 16(3):371–390. 6

[71] 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. 25

[72] Kahl, L. J. and Endy, D. (2013). A survey of enabling technologies in synthetic biology. *Journal of biological engineering*, 7(1):13. 1, 5

[73] Kalyanmoy, D. (2001). *Multi objective optimization using evolutionary algorithms*. John Wiley and Sons, Chichester, England. 50

[74] Kalyuzhnaya, M. G., Puri, A. W., and Lidstrom, M. E. (2015). Metabolic engineering in methanotrophic bacteria. *Metabolic engineering*, 29:142–152. 18

[75] Kamali, M. and Hewage, K. (2016). Life cycle performance of modular buildings: A critical review. *Renewable and Sustainable Energy Reviews*, 62:1171–1183. 7

[76] Kanehisa, M. and Goto, S. (2000). Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30. 15

[77] 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. 12

[78] 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. 14

[79] Khosla, C. and Harbury, P. B. (2001). Modular enzymes. *Nature*, 409(6817):247. 9, 20

[80] 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. 7

[81] King, Z. A., Lu, J., Dräger, A., Miller, P., Federowicz, S., Lerman, J. A., Ebrahim, A., Palsson, B. O., and Lewis, N. E. (2015). Bigg models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic acids research*, 44(D1):D515–D522. 14, 58

[82] 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. 1, 5, 17

[83] Kitano, H. (2004). Biological robustness. *Nature Reviews Genetics*, 5(11):826. 11, 33

[84] Klamt, S. and Mahadevan, R. (2015). On the feasibility of growth-coupled product synthesis in microbial strains. *Metabolic engineering*, 30:166–178. 14, 25

[85] Klamt, S., Mahadevan, R., and Hädicke, O. (2018). When do two-stage processes outperform one-stage processes? *Biotechnology journal*, 13(2):1700539. 14, 39

[86] 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. 15

[87] Kosuri, S. and Church, G. M. (2014). Large-scale de novo dna synthesis: technologies and applications. *Nature methods*, 11(5):499. 18

[88] 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. 15

[89] 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. 31

[90] Layton, D. S. and Trinh, C. T. (2014). Engineering modular ester fermentative pathways in escherichia coli. *Metabolic Engineering*, 26:77–88. 15, 17, 22, 25, 49

[91] 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*. 25, 49

[92] 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. 25, 49

[93] Lee, J. and Trinh, C. T. (2018). De novo microbial biosynthesis of a lactate ester platform. *bioRxiv*, page 498576. 49

[94] 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. 1, 5, 49

[95] Li, B., Li, J., Tang, K., and Yao, X. (2015a). Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)*, 48(1):13. 50

[96] Li, M. and Elledge, S. (2007). Harnessing homologous recombination in vitro to generate recombinant dna via slic. *Nat Methods*, 4(3):251 – 6. 15

[97] 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. 55

[98] Li, M., Yang, S., and Liu, X. (2015b). Bi-goal evolution for many-objective optimization problems. *Artificial Intelligence*, 228:45–65. 55

[99] 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. 15

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

[101] Liu, D., Evans, T., and Zhang, F. (2015). Applications and advances of metabolite biosensors for metabolic engineering. *Metabolic engineering*, 31:35–43. 18

[102] 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. 14, 25

[103] Lu, H., Villada, J. C., and Lee, P. K. (2018). Modular metabolic engineering for biobased chemical production. *Trends in biotechnology*. 2, 12

[104] 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. 18

[105] Ma, H.-W. and Zeng, A.-P. (2003). The connectivity structure, giant strong component and centrality of metabolic networks. *Bioinformatics*, 19(11):1423–1430. 10

[106] 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. 14

[107] Maranas, C. D. and Zomorrodi, A. R. (2016). *Optimization Methods in Metabolic Networks*. John Wiley & Sons, Hoboken, New Jersey. 32

[108] Marler, R. T. and Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395. 31, 50

[109] 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. 15

[110] 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. 15

[111] Meunier, D., Lambiotte, R., and Bullmore, E. T. (2010). Modular and hierarchically modular organization of brain networks. *Frontiers in neuroscience*, 4:200. 9

[112] 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. 18

[113] Miller, T. D. and Elgard, P. (1998). Defining modules, modularity and modularization. In *Proceedings of the 13th IPS research seminar, Fuglsoe*. Aalborg Universiy. 6, 7

[114] 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. 9, 10

[115] 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. 18

[116] Neidhardt, F. C., Ingraham, J. L., and Schaechter, M. (1990). *Physiology of the bacterial cell: a molecular approach*, volume 20. Sinauer Associates Sunderland, MA. 10

[117] Ng, C. Y., Chowdhury, A., and Maranas, C. D. (2016). A microbial factory for diverse chemicals. *Nat Biotech*, 34:513–515. 24

[118] Ng, C. Y., Khodayari, A., Chowdhury, A., and Maranas, C. D. (2015). Advances in de novo strain design using integrated systems and synthetic biology tools. *Current opinion in chemical biology*, 28:105–114. 14

[119] Nielsen, J. and Keasling, J. (2016). Engineering Cellular Metabolism. *Cell*, 164(6):1185–1197. 1, 11, 24, 49

[120] 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. 14

[121] 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. 10

[122] 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. 1, 5

[123] Palsson, B. Ø. (2015). *Systems biology: constraint-based reconstruction and analysis*. Cambridge University Press, United Kingdom. 14, 52, 58

[124] Pandit, A. V., Srinivasan, S., and Mahadevan, R. (2017). Redesigning metabolism based on orthogonality principles. *Nature communications*, 8:15188. 14

[125] Park, H.-J. and Friston, K. (2013). Structural and functional brain networks: from connections to cognition. *Science*, 342(6158):1238411. 9

[126] 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. 28

[127] Pryciak, P. M. (2009). Designing new cellular signaling pathways. *Chemistry & biology*, 16(3):249–254. 11

[128] 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. 11, 24

[129] Rangaiah, G. P. (2009). *Multi-objective optimization: techniques and applications in chemical engineering*, volume 1. World Scientific, Singapore. 49

[130] 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. 9, 10

[131] Rehm, B. H. (2010). Bacterial polymers: biosynthesis, modifications and applications. *Nature Reviews Microbiology*, 8(8):578. 15

[132] 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. 56

[133] Rodriguez, G. M., Tashiro, Y., and Atsumi, S. (2014). Expanding ester biosynthesis in escherichia coli. *Nature Chemical Biology*, 10:259–265. 10, 15, 24

[134] 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. 15

[135] Salvador, F. (2007). Toward a product system modularity construct: literature review and reconceptualization. *IEEE Transactions on engineering management*, 54(2):219–240. 6

[136] Sauro, H. M. (2008). Modularity defined. *Molecular systems biology*, 4. 24

[137] 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. 15

[138] 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. (2011). Quantitative prediction of cellular metabolism with constraint-based models: the cobra toolbox v2. 0. *Nature protocols*, 6:1290–1307. 31

[139] Schuetz, R., Zamboni, N., Zampieri, M., Heinemann, M., and Sauer, U. (2012). Multidimensional optimality of microbial metabolism. *Science*, 336(6081):601–604. 12

[140] 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. 56, 57

[141] 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. 15

[142] 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. 59

[143] 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. 12

[144] 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. 8

[145] 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. 9

[146] Sporns, O. and Betzel, R. F. (2016). Modular brain networks. *Annual review of psychology*, 67:613–640. 9

[147] 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. 9

[148] 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. 10

[149] Tepper, N. and Shlomi, T. (2010). Predicting metabolic engineering knockout strategies for chemical production: accounting for competing pathways. *Bioinformatics*, 26:536–543. 29

[150] 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. 18

[151] 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. 55

[152] 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. 54, 59

[153] Tian, Y., Zhang, X., Cheng, R., and Jin, Y. (2016). A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 5222–5229. IEEE. 55

[154] 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. 25

[155] 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. 13, 22, 49

[156] 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. 17

[157] 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. 49

[158] Trinh, C. T., Liu, Y., and Conner, D. J. (2015). Rational design of efficient modular cells. *Metabolic engineering*, 32:220–231. 2, 13, 14, 16, 17, 22, 24, 25, 26, 29, 32, 34, 35, 49

[159] 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. 2, 6, 12, 24, 49

[160] 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. 22, 25

[161] 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. 14, 25

[162] 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. 15

[163] 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. 58, 59

[164] 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. 15

[165] Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research policy*, 24(3):419–440. 6

[166] Vujić, J., Bergmann, R. M., Škoda, R., and Miletić, M. (2012). Small modular reactors: Simpler, safer, cheaper? *Energy*, 45(1):288–295. 7

[167] Wagner, G. P., Pavlicev, M., and Cheverud, J. M. (2007). The road to modularity. *Nature Reviews Genetics*, 8(12):921–931. 8, 12

[168] 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. 15

[169] 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. 14

[170] 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. 7

[171] Whitacre, J. M. (2012). Biological robustness: paradigms, mechanisms, and systems principles. *Frontiers in genetics*, 3:67. 11

[172] 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. 25, 49

[173] Wilbanks, B., Layton, D., Garcia, S., and Trinh, C. (2017). A prototype for modular cell engineering. *ACS Synthetic Biology*, page acssynbio.7b00269. 16, 17, 22, 24, 25, 38, 49

[174] 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. 1, 5, 17, 37

[175] 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. 14

[176] 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. 24

[177] 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. 2, 12, 16, 24, 25

[178] Yang, L., Cluett, W. R., and Mahadevan, R. (2011). Emilio: a fast algorithm for genome-scale strain design. *Metabolic engineering*, 13:272–281. 25, 29

[179] 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. 18

[180] 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. 55

[181] 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. 55

[182] 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. 18, 22

[183] Zhang, Y., Werling, U., and Edelmann, W. (2012b). Slice: a novel bacterial cell extract-based dna cloning method. *Nucleic Acids Research*, 40(8):e55. 15

[184] Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195. 50

[185] Zitzler, E., Laumanns, M., and Thiele, L. (2001). Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103. 50

[186] 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. 57

Appendices

A Supplementary Material 1 for Chapter 3

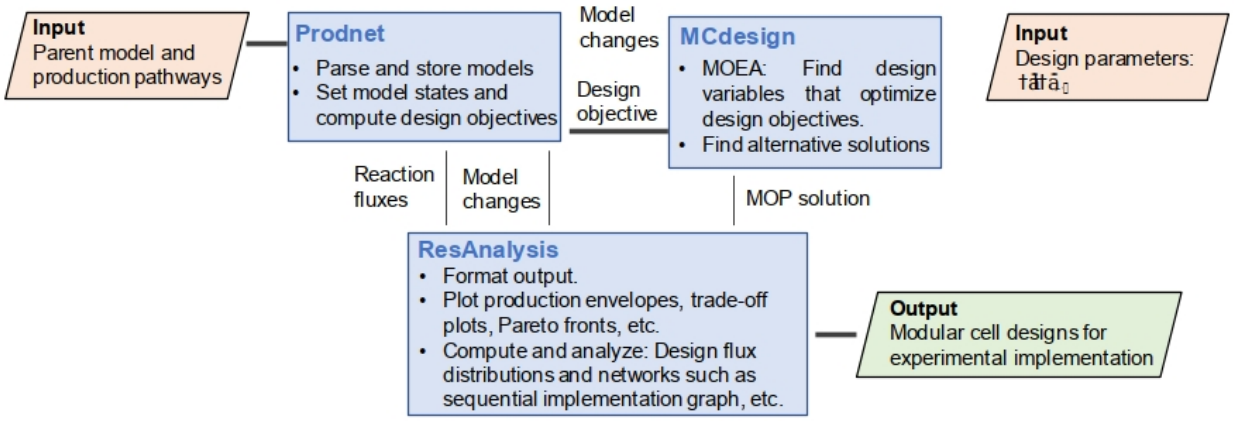


Figure A1: Software architecture of ModCell2. The Prodnet class preprocesses production network models and computes design objectives. The MCdesign class serves as an interface between the MOEA optimization method and metabolic models. The ResAnalysis class loads the Pareto set computed by MCdesign and performs analyses to identify the most promising designs.

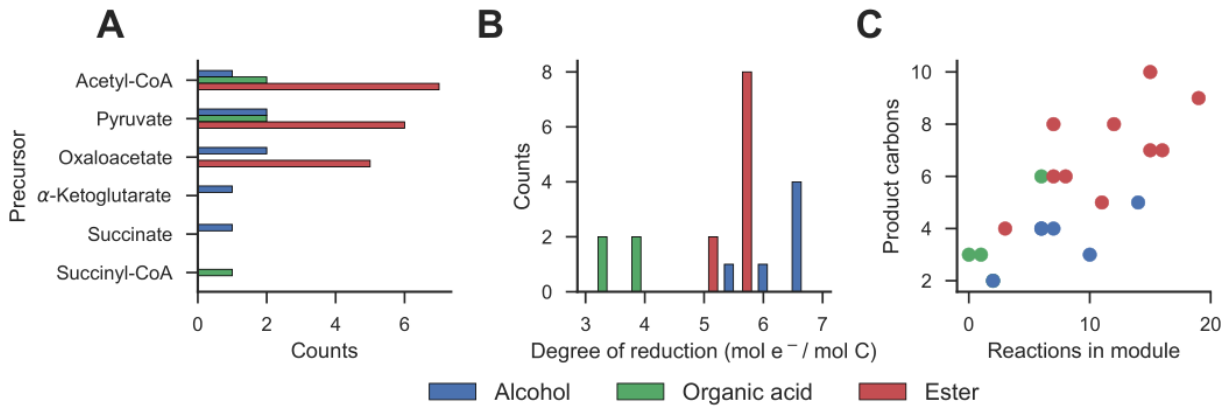


Figure A2: Properties of 20 production modules used in the E. coli genome-scale metabolic model for biosynthesis of 6 alcohols, 4 organic acids, and 10 esters. (A) Distribution of precursor metabolites. (B) Distribution of degrees of reduction of target products. (C) Correlation between the number of 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.

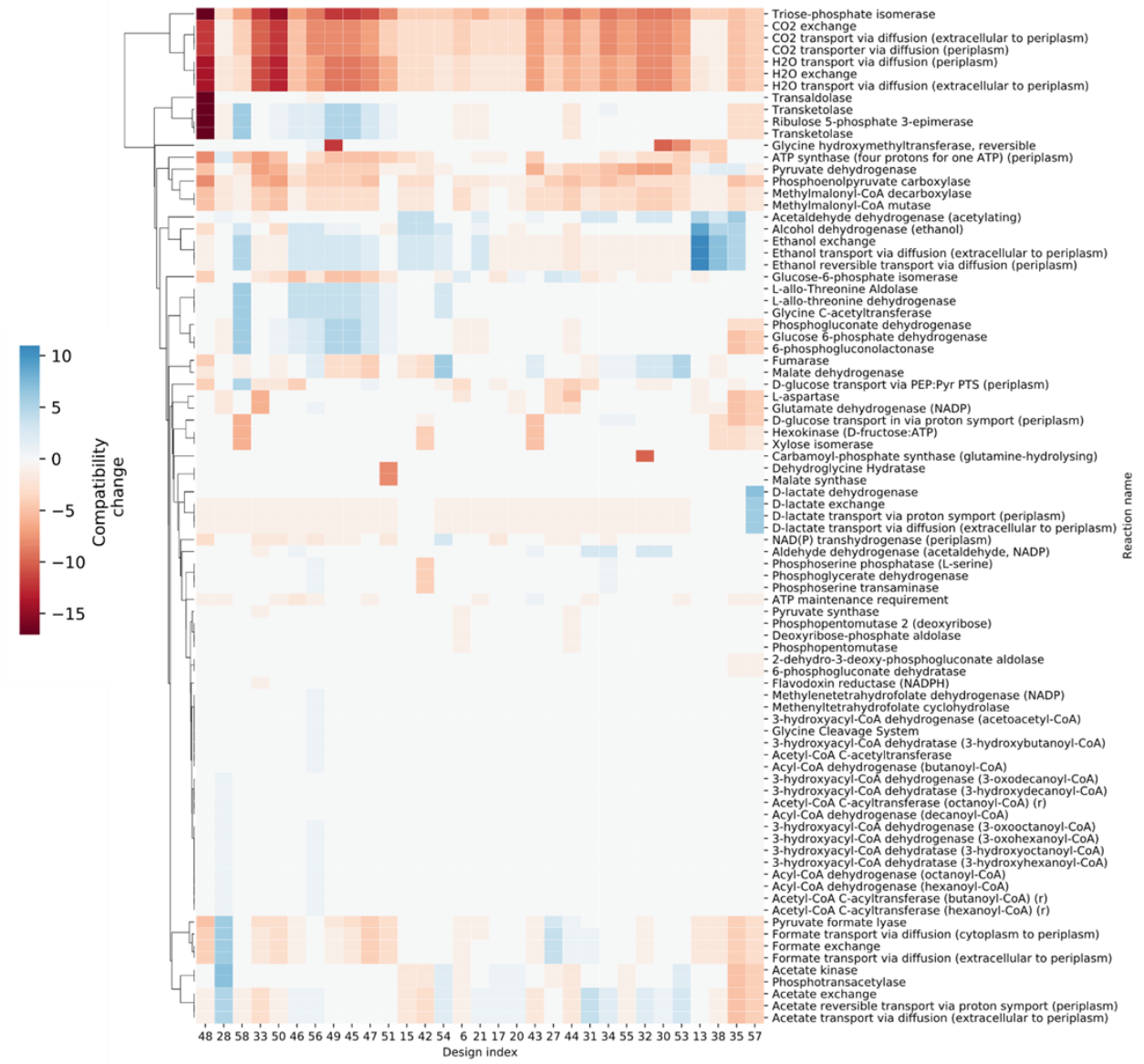


Figure A3: Robustness analysis for wGCP-4-0 designs for the E. coli genome scale model. Only the designs that are compatible with 4 or more products (compatibility 4) were considered. Each column corresponds to a design whereas each row corresponds to a single-reaction deletion. Included in the heat map are all reaction deletions with a compatibility change that is not 0 in at least one product.

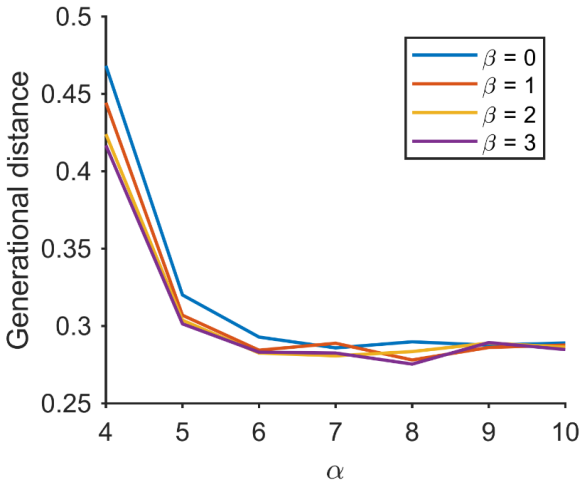


Figure A4: Generational distances between the calculated Pareto fronts and the reference utopia point. The generational distance is calculated as follows: $GD = \frac{|d|_2}{|d|_1}$where $d_j = \left|\mathcal{PF}^j - \mathcal{PF}^*\right|_2$ where $\mathcal{PF}$ is the calculated pareto front and $\mathcal{PF}^* = \vec{1}$ 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 3

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 [**?**], 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_{Pk}^{μ}, $v_{P_{max}k}^{\mu}$, $v_{Pk}^{\bar{\mu}}$, and $v_{P_{max}k}^{\bar{\mu}}$ are computed by solving the following linear programming problems:

$$v_{Pk}^{\mu} \quad \in \arg\max\{v_{Xk} - \epsilon v_{Pk} : v_k \in \Pi_k^{\mu}(d_{jk})\} \tag{B6}$$

$$v_{P_{max}k}^{\mu} \in \arg\max\{v_{Pk} : v_k \in \Pi_k^{\mu}(d_{jk} = 1,\ \forall j \in \mathcal{J}_k)\} \tag{B7}$$

$$v_{Pk}^{\bar{\mu}} \quad \in \arg\min\{v_{Pk} : v_k \in \Pi_k^{\bar{\mu}}(d_{jk})\} \tag{B8}$$

$$v_{P_{max}k}^{\bar{\mu}} \in \arg\max\{v_{Pk} : v_k \in \Pi_k^{\bar{\mu}}(d_{jk} = 1,\ \forall j \in \mathcal{J}_k)\} \tag{B9}$$

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), Π_k^{μ} is the space of steady-state reaction fluxes of production network k where a minimum cell growth is required, defined as follows:

$$\Pi_k^{\mu}(d_{jk}) := \{v_{jk} \in \mathbb{R}\, \forall j \in \mathcal{J}_k :$$

$$\sum_{j \in \mathcal{J}_k} S_{ijk} v_{jk} = 0, \qquad \forall i \in \mathcal{I}_k \tag{B10}$$

$$l_{jk} \leq v_{jk} \leq u_{jk}, \qquad \forall j \in \mathcal{J}_k \tag{B11}$$

$$l_{jk} d_{jk} \leq v_{jk} \leq u_{jk} d_{jk}, \qquad \forall j \in \mathcal{C} \tag{B12}$$

$$v_{Xk} \geq \text{minimum growth rate}\} \tag{B13}$$

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, $\Pi_k^{\bar{\mu}}$, 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:

$$q^{enum} = \begin{cases} M & \text{if } \{y_j : j \in \mathcal{J}_k\} \in \text{ExcludedSol} \quad \text{(B19)} \\ 1 - \mathbf{jacc}(f^*, f) + \sum_{j \in \mathcal{C}} (1 - y_j) & \text{if } \sum_{j \in \mathcal{C}} (1 - y_j) > \alpha \quad \text{(B20)} \\ 1 - \mathbf{jacc}(f^*, f) & \text{otherwise} \quad \text{(B21)} \end{cases}$$

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 $\beta_k > 0$.

[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.

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 [**?**] 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).

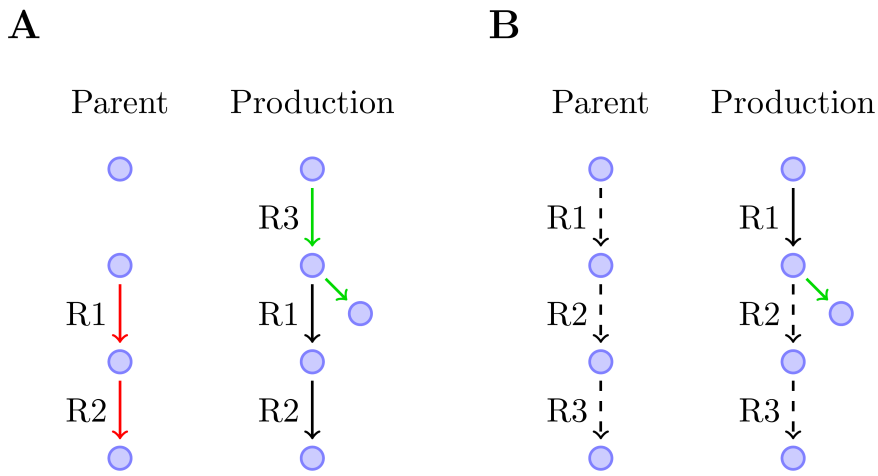


Figure B1: (A.) Blocked reactions. The red arrows represent reactions originally blocked in the parent model, because the substrate of R1 cannot be produced. When heterologus reactions (green arrows) are added to produce a target chemical, the originally blocked reactions may become active and drain 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.

Vita

Sergio Garcia is originally from Murcia, a region in the south eastern part of Spain. After finishing highschool... he began his studies in Chemical Engineering at the university of Murcia, founded in 12... among the first universities funded in the world. Although the chemical engineering program was only a few decades old at the time. Sergio spent his junior year abroad at the University of Tennesse Knoxville (UTK) throught the International ... (ISEP) scholarship, and here he became involved in undegraduate research, later resulting in one publication. He then return to finish his degree in Murcia, where he also pursued undegraduate research in the Department of Biochemistry and Molecular Biology. After completion of his degree he returned to UTK to pursue a Ph.D in Chemical and Biomolecular Engineering. ... Sergio enjoys studying music in the classical and jazz generes.