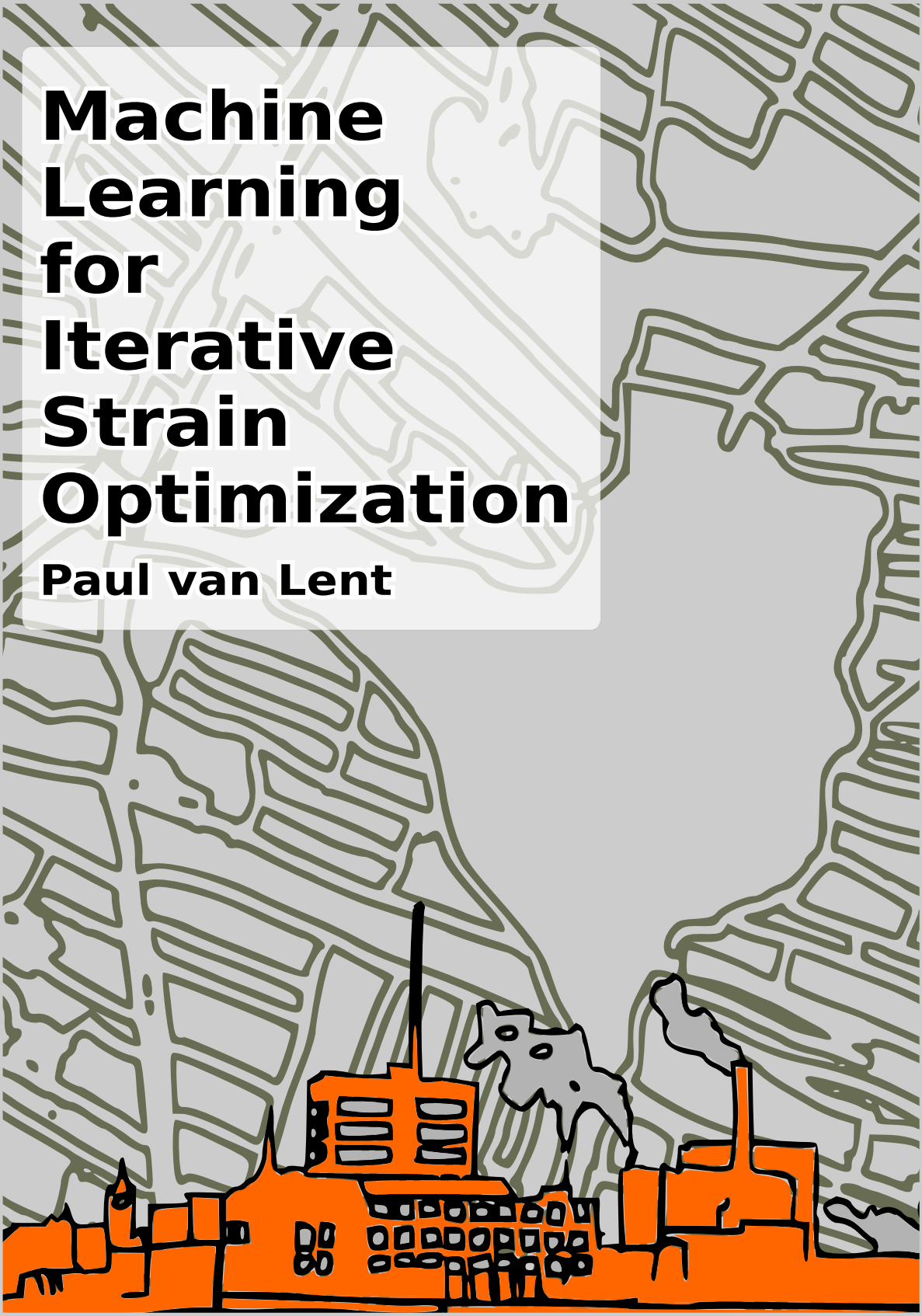
Machine Learning for Iterative Strain Optimization

Paul van Lent



Machine Learning for Iterative Strain Optimization

MACHINE LEARNING FOR ITERATIVE STRAIN OPTIMIZATION

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, Prof.dr.ir. H. Bijl,
chair of the Board for Doctorates
to be defended publicly on
Friday 3 July 2026 at 10:00

Paul Harrie VAN LENT

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Dr. T.E.P.M.F. Abeel,	Delft University of Technology, *promotor*
Prof.dr.ir. M.J.T. Reinders,	Delft University of Technology, *promotor*
Independent members:	
Prof.dr.ir. H.J. Noorman	Delft University of Technology
Prof.dr.ir. L.A.M. van der Wielen	Delft University of Technology
	DTU, Denmark
Prof.dr.ir. R.A. Weusthuis	Wageningen University and Research
Prof.dr. B. Teusink	Vrije Universiteit Amsterdam
Dr. J.M. Weber	Delft University of Technology

The research in this thesis is supported by the AI4b.io program, a collaboration between TU Delft and dsm-firmenich, and is fully funded by dsm-firmenich and the RVO (Rijksdienst voor Ondernemend Nederland).

Keywords:	Metabolic engineering, machine learning, kinetic modeling, hybrid modeling, automated recommendation
Printed by:	Ridderprint (www.ridderprint.nl)
Cover by:	Back cover by Yara de Kok (`https://www.yaradekok.com/`)

Contents

Summary

Fermentation has played an important role in the production of foods such as bread, beer, wine and cheese for thousands of years. These processes rely on micro-organisms that convert sugars into products such as alcohol, lactic acid, and carbon dioxide under oxygen-depleted conditions. Thanks to genetic modification, the variety of substances that can be produced through fermentation has expanded greatly. Fermentative production – also known as bioprocesses – offers a sustainable alternative to fossil fuel-based chemical production. However, metabolic engineering, in which the genetics of microorganisms are specifically modified to produce desired metabolites in high yields, is highly complex, time-consuming and costly. One of the main challenges is the enormous design space created by the complexity of cellular metabolism.

Machine learning can help explore this design space more efficiently, for example, by predicting the performance of strain designs or suggesting new genetic modifications. In this thesis, we focus on improving these two applications. First, we develop a simulation tool that mimics metabolic processes, allowing us to compare different machine learning models and experimental strategies in a fair and cost-efficient manner. We then use the insights gained from these simulations to optimize yeast strains that produce p-Coumaric acid.

One drawback of many machine learning models is their limited transparency: it is often difficult to understand how a prediction is generated. In this thesis, we investigate how such models can be combined with mechanistic, mathematically formulated models. This hybrid approach brings together the predictive accuracy of machine learning and the interpretability of mechanistic models. We demonstrate that this integration results in more understandable models without sacrificing predictive performance.

Together, the methods and software developed in this thesis provide new tools for applying machine learning more effectively in metabolic engineering, with the aim of accelerating the development of sustainable bioprocesses.

Samenvatting

Fermentatie is een proces waarbij micro-organismen suikers onder zuurstofarme omstandigheden omzetten in stoffen zoals alcohol, melkzuur en koolstofdioxide. Dit proces speelt al duizenden jaren een belangrijke rol in de productie van voedingsmiddelen zoals brood, bier, wijn en kaas. Dankzij genetische modificatie is de verscheidenheid aan stoffen die via fermentatie kan worden gemaakt sterk uitgebreid. Fermentatieve productie – ook wel bioprocessen genoemd – biedt bovendien een duurzaam alternatief voor op fossiele brandstoffen gebaseerde chemische productie. Metabolische bouwkunde, waarbij de genetica van micro-organismen doelgericht wordt aangepast om gewenste metabolieten in hoge opbrengst te produceren, is echter zeer complex, tijdrovend en kostbaar. Een van de belangrijkste uitdagingen hierbij is de enorme ontwerpruimte die ontstaat door de complexiteit van het cellulaire metabolisme.

Machinaal leren kan helpen deze ontwerpruimte efficiënter te verkennen, bijvoorbeeld door de productie van stamontwerpen te voorspellen of suggesties te doen voor nieuwe genetische modificaties. In dit proefschrift richten we ons op het verbeteren van deze twee toepassingen. Eerst ontwikkelen we een simulatiegereedschap dat metabole processen nabootst en zo toestaat verschillende machinaal-lerende modellen en experimentele strategieën eerlijk en kostenefficiënt te vergelijken. De inzichten uit deze simulaties zetten we vervolgens in voor het optimaliseren van giststammen die p-coumarinezuur produceren, een voorloper van gezondheidsbevorderende metabolieten. Met behulp van machinaal leren kunnen we hierbij veel meer ontwerpvarianten evalueren en gerichtere aanbevelingen doen dan met klassieke methoden mogelijk is.

Een nadeel van veel machinaal lerende modellen is hun beperkte transparantie: het is vaak lastig te begrijpen hoe een voorspelling tot standkomt. In dit proefschrift onderzoeken we hoe dergelijke modellen kunnen worden gecombineerd met mechanistische, wiskundig beschreven modellen. Deze hybride benadering verenigt de voorspellende nauwkeurigheid van machine learning met de uitlegbaarheid van mechanistische modellen. We tonen aan dat deze integratie leidt tot begrijpelijkere modellen zonder in te leveren op voorspellingskracht.

De methoden en software die in dit proefschrift zijn ontwikkeld, bieden nieuwe mogelijkheden om machinaal leren effectiever toe te passen in de metabolische bouwkunde, met als doel de ontwikkeling van duurzame bioprocessen te versnellen.

1

Introduction

1.1. Biotechnology

1.1.1. Alcoholic origins

In the 19th century, Louis Pasteur made an important discovery by showing that ethanol fermentation was performed by living yeast cells under oxygen-limited conditions [1]. The fermentation process was already used in food production for thousands of years, but was widely considered to be a spontaneous process. This discovery by Pasteur turned out to be influential. Although the enormous diversity of microorganisms was recognized and appreciated for many centuries [2], it had never truly been placed in the forefront of industrial and economic food production processes. Microbiology, the study of the diversity, morphology, and function of microorganisms, transformed from a curiosity-driven science into an applied science with economic implications. Indeed, less than a decade after Pasteur's discovery, large brewing companies such as Carlsberg and Heineken established their own laboratories, where they developed pure yeast strains that continued to be used in beer production until recently [3]. Therefore, the study of microbial fermentation in the context of industrial production, known as *Zymotechnology*, is the earliest example of what is now known as biotechnology [4].

1.1.2. An expanding scope

Biotechnology, broadly defined as the manipulation of living organisms or their components to generate products beneficial to humans, was relatively limited in scope at the beginning of the twentieth century. Biotechnological processes were mainly used for the production of a narrow range of compounds that microorganisms already produced naturally. This included ethanol, lactic acid, acetic acid, and butyric acid [4, 5]. The discovery of penicillin and the subsequent development of it's industrial-scale production demonstrated that microorganisms can play an important role not just in food, but also in the pharmaceutical and chemical industries [6]. Nevertheless, these early biotechnological products were derived predominantly from wild-type microbial strains that occur naturally.

The elucidation of the DNA structure and the emergence of genetic engineering

techniques in the mid-twentieth century fundamentally transformed the field [4, 7]. These advances enabled the targeted modification of microbial metabolism to enhance the production of naturally occurring compounds [8]. This development laid the foundation for the concept of *microbial cell factories*: genetically engineered microorganisms designed to synthesize a diverse array of products. Today, such engineered bioprocesses underpin the sustainable production of materials ranging from fine chemicals to bulk commodities, with an increasing role of biotechnology in industrial and environmental applications [9, 10].

1.1.3. Sustainability and climate action

In contrast to fossil fuel-based chemical production, microbial fermentation uses renewable substrates, which is an important prerequisite for sustainable production [10]. With increasing awareness of the impact of greenhouse gasses on climate change [11, 12], policies have been advocated to mitigate and adapt to climate change [13]. Biotechnology may play an important role in the decarbonization of the chemical industry, as currently 85% of the substrates used for chemical production are based on fossil fuels [10]. However, there are significant challenges related to biobased production. First, renewable substrates may not always be sustainable, as they, for example, compete with agricultural land use. Second, the development of microbial cell factories that produce on industrially relevant scales is a long and costly process, typically costing millions of euros and several hundred person years for their development [14]. For these reasons, only a few hundred out of the roughly seventy thousand commercialized chemicals are produced using a bioprocess [9, 10].

Bioprocess development and its challenges

Bio-based production of many useful chemicals has shown success, especially in the production of specialty chemicals [9, 15]. Although bio-based production of bulk commodities such as fuels has so far been underrepresented due to the competitive economics of fossil fuels, some companies have shown success in the production of these chemicals in niche markets [16]. To understand why biotechnological solutions have not been widespread, we briefly explain bioprocesses and challenges.

A bioprocess consists of an upstream, fermentation, and downstream part (Fig. 1.1). In the upstream part, the raw renewable resources, such as lignocellulose, C1-C2 carbon sources, or glucose, are processed to a substrate ready to be used in fermentation. The choice of resource is important not only for all subsequent steps, but also to ensure how sustainable and economically competitive the process will be [15]. This can be highly dependent on the product: for pharmaceuticals, expensive substrates might be fine, while for bulk commodities, the cost of substrate may be limiting. In the fermentation part, the actual conversion from the substrate to the product is performed by the microorganism. Several factors affect the performance of this conversion. This includes the composition of the fermentation medium [17], the substrate feeding strategy [18], the mode of fermentation (batch, fed-batch, continuous process) [19], the process scale-up to levels relevant to the industrial market and the performance of the microbial strain

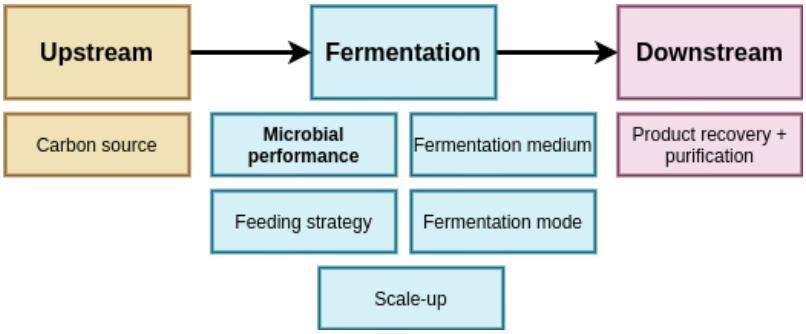


Figure 1.1: **Challenges during bioprocess development** A typical bioprocess consists of upstream processing of the raw renewable resources to substrates that can be used in the bioreactor. Then, the fermentation performs the actual conversion process using microbial cell factories. Finally, the product needs to be separated from the fermentation broth and purified. Several factors related to specific parts of this process need to be considered when developing industrial bioprocesses. Microbial performance in terms of titer, rate, yield is the focus of this thesis.

[20]. In the downstream part, the product needs to be recovered from the fermentation broth. Here, challenges in product recovery and purification can have a large impact on bioprocess performance [15].

Although all of these factors impact the sustainability, economics, and production rate of the bioprocess and need to be considered, this thesis will focus on the microbial strain performance aspect of the bioprocess. More specifically, the role of machine learning in the genetic optimization of strains will be investigated.

1.2. Optimizing microbial strain performance

1.2.1. Metabolism

When it comes to optimizing microorganisms, there are essentially two aspects that need to be considered. First, there are external conditions that can greatly influence the performance of strains [15, 21]. Second, there is the internal metabolism that converts the substrate into a product of interest while also maintaining and growing the cells.

Metabolism is defined as the set of chemical reactions catalyzed by enzymes within a microorganism. From a biological perspective, metabolic reactions can serve three purposes. First, they import substrates to harvest energy for growth and maintenance. These reactions are classified as catabolic, meaning that they release energy during catalysis, both as heat and as chemical energy stored in carriers like ATP and NADH. Second, products formed during the breakdown of the substrate are used as building blocks of biomass, such as lipids, nucleic acids, and amino acids. These are the anabolic reactions that require energy to drive the energetically unfavorable building of

1

macromolecules. Finally, metabolic reactions ensure the elimination of waste products through excretion across the cell membrane. Chains of reactions that fulfill a certain functionality are termed metabolic pathways [22]. The behavior of metabolic pathways can be described using metabolic models, as will be explained in more detail later.

Evolutionary pressures resulted in a metabolism that was optimized for the fitness of the microorganism: it's ability to survive and reproduce. This can result in the maximization of growth rates to outgrow competition [23], but other survival strategies also exist [24]. Metabolism also needs to respond to the changing environment, requiring plasticity in how it is modulated. Modulation occurs through the upregulation and downregulation of enzyme concentrations that are related to specific reactions. This regulation can occur at the transcriptional (e.g., promoters) and translational levels (e.g., ribosomal binding sites) [22]. Therefore, metabolic fluxes — the turnover rate of a reaction from a substrate to an (intermediate) product — are partially determined by enzyme concentration and enzyme catalytic constants (Fig. 1.2A) [25]. Using genetic engineering techniques [26, 27], DNA elements such as promoters or alternative coding sequences can be integrated into the host genome to alter the expression of an enzyme, potentially modifying the flux profile of metabolism [28].

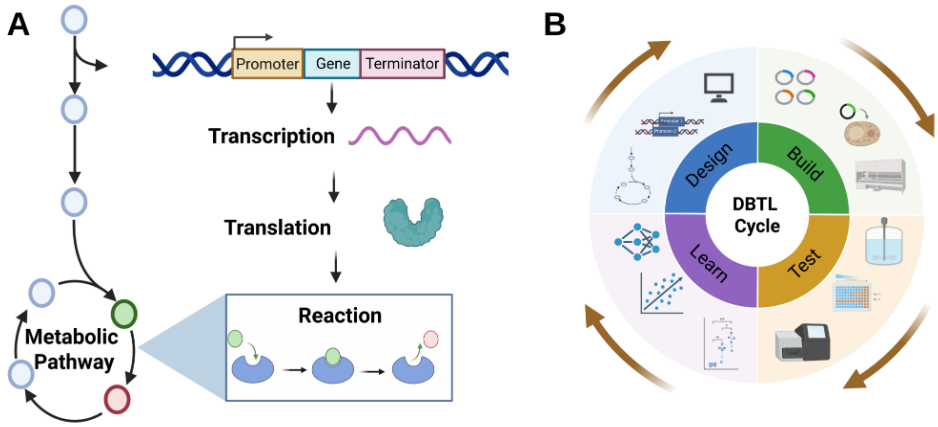


Figure 1.2: **Metabolic flux optimization and the Design-Build-Test-Learn Cycle of metabolic engineering. A)** Metabolism is organized into pathways that consists of sets of reactions catalyzed by enzymes. The metabolite concentrations, enzyme properties and concentration determine the metabolic flux through a certain reaction. These fluxes are partially controlled by the transcription and translation of DNA to protein, which is a highly regulated process on the transcriptional and translation level. **B)** The Design-Build-Test-Learn cycle is an engineering paradigm that aids in strain optimization by navigating the landscape of strain design choices using an iterative approach.

1.2.2. Combinatorial pathway optimization

When the metabolism of the strain to be optimized is fully understood through a mathematical model, rational engineering strategies that sequentially debottleneck rate-limiting steps in the metabolic pathway could be performed. Unfortunately, such a deep understanding of metabolism is not available, certainly not for secondary metabolic pathways that are not part of core metabolic processes, such as glycolysis or the TCA cycle [29].

Combinatorial pathway optimization aims to optimize several elements of the metabolic pathway simultaneously in order to find a globally optimal configuration [30]. This is advantageous in that the chance of finding a globally optimum configuration is higher compared to sequential debottlenecking, while also allowing for the construction of several strains in parallel. This significantly reduces the time it takes to find well-performing strains in metabolic engineering [31, 32]. A consequence of combinatorial optimization is that, by increasing the number of pathways simultaneously optimized, the number of theoretically possible designs grows exponentially. For example, if ten genes with five expression levels are considered, the number of possible strain designs is approximately 9.8 million. This cannot be exhaustively screened and sequenced, even with high-throughput screening approaches [33].

1.2.3. Design-Build-Test-Learn Cycles

The Design-Build-Test-Learn (DBTL) cycle is an engineering paradigm that has been used for many metabolic engineering studies to navigate the large combinatorial design landscape [34]. In the Design phase, metabolic pathway elements that may influence production are identified. A set of combinations is chosen, either randomly or through a precise design approach (Design phase) [35]. Then, these strains are built using genetic engineering tools, followed by the screening of strains for their performance (Build and Test phase). The information gained from these experiments is used to learn which parts of the design resulted in higher performance (Learn phase). Finally, this information is used to guide the designs for the next DBTL cycle. The final two steps of this process, namely learning and recommending new designs, have been increasingly performed using machine learning [36–38].

Although the DBTL cycle has been used in many successful strain optimizations [39, 40], several open questions remain about how it can be optimally deployed. For the design phase, what genes and promoters should we use and how large can the theoretical combinatorial landscape be in order to effectively learn? For the construction and test phase, how many strains should be built and screened while keeping the experimental effort in mind? For the learning phase, what types of supervised learning algorithms should be used to effectively predict? How can information learned from previous cycles be used to recommend strains for the next cycle? The interplay between these questions and the application of machine learning for iterative metabolic engineering plays a central role in this thesis [34].

1.2.4. High-throughput engineering technologies

Effective manipulation of microbial strains using genetic engineering is essential for optimizing the pathways. A crucial development in synthetic biology has been the discovery of the bacterial defense mechanism CRISPR (*clustered regularly interspaced short palindromic repeats*) and its repurposing for genome editing [26]. Due to its high genome editing precision, this technology has had a tremendous impact on the field of biotechnology [27]. The second important scientific/technological advancement has been the possibility of generating large genetic constructs with multiple genetic elements. In combinatorial pathway optimization, pathway elements are simultaneously considered, which requires the integration of several promoter-gene-terminator pairs into a single construct. DNA assembly and cloning protocols have been developed that allow the generation of large DNA constructs, such as *Golden Gate assembly* [41], which can then be integrated into the genome using CRISPR/Cas [42]. These advances have paved the way for large-scale and efficient strain building.

In parallel with increasing strain building capabilities, advances in high-throughput strain testing have been of great importance. The throughput of the DBTL cycle depends on the strain building capabilities, but also on the screening technology used and the subsequent DNA sequencing capacity. The small-molecule screening capacity is largely dependent on the method used, whether it is liquid or gas chromatography (approximately 10^4 strains) or fluorescence activated cell sorting with a biosensor (10^9) [33]. DNA sequencing has become very cheap over the years, now reaching prices of around €0.006 per megabase. The limitation related to DNA sequencing is therefore more in capacity than cost.

1.3. Modeling

Metabolic models are extensively applied in strain optimization. They assist in devising optimal feeding strategies [43], predicting targets for metabolic engineering [44], and providing quantitative insights into metabolism [45]. These models can be roughly categorized into two approaches: mechanistic models and machine learning models. Mechanistic models aim to describe metabolic pathways through mathematical frameworks that relate metabolites and reactions through mass-balances. The advantage of these models is that they are interpretable and explainable [46], but their predictive performance is strongly dependent on how well-understood the system is. Machine learning models take a different approach by focusing on prediction instead of understanding. They aim to model a target variable by learning statistical patterns from datasets. Machine learning models typically excel in predictive performance but are limited in achieving an understanding of the underlying metabolism.

1.3.1. Machine learning

In recent years, machine learning has been applied in virtually every scientific domain. Machine learning algorithms aim to learn patterns from data for the purpose of inference without following explicit instructions on what these patterns should look like, which contrasts with mechanistic modeling approaches. Where mechanistic modeling typically

provides insight into how the output of a model is produced, machine learning methods often offer superior predictive performance with little interpretability.

Machine learning methods can be categorized according to three learning principles. In unsupervised learning, algorithms aim to analyze and cluster unlabeled datasets by uncovering hidden patterns with the goal of understanding the data distribution and retrieving insights. In the context of biotechnology, this can be used to direct metabolic engineering by studying the proteome using principal component analysis [47] or by independent component analysis of the transcriptome [48]. Supervised learning, on the other hand, focuses on learning from labeled data, with the goal of learning a function that maps the input space X, consisting of M samples with F features (measurements), to the output space of the labels Y. This learning problem is defined as a classification problem if the labels are categorical, or as a regression problem if the labels are numerical. Supervised learning algorithms have been widely used in several aspects of the DBTL cycle, including the prediction of metabolic pathway dynamics [49], predicting genetic engineering outcomes [50], and active learning and Bayesian optimization [37, 51]. Finally, reinforcement learning is a type of machine learning in which a decision-maker (agent) interacts with an environment that provides feedback (reward) on the decisions that the agent makes. The goal of reinforcement learning is to find a decision strategy (policy) that optimizes the reward over a given set of decisions. Although the reinforcement learning approach has not been widely adopted in the biotechnological domain [52], it may still be useful to pose the metabolic engineering DBTL cycle as a reinforcement learning-like problem through a Markov Decision Process [53].

1.3.2. Mechanistic modeling: genome-scale models

Genome-scale models aim to describe the entire set of stoichiometry-based and mass-balanced reactions and metabolites while also linking the corresponding genes to individual fluxes [54]. These models have been used for a wide range of applications, including metabolic engineering, enzyme function descriptions, and modeling interactions in microbial communities [55–57]. The typical procedure for constructing genome-scale models consists of building networks from an annotated genome, which are then translated into a stoichiometric matrix $S \in \mathbf{R}^{m \times n}$. The stoichiometric matrix describes how the mass of m metabolites flows through the n reactions. Negative values indicate that a metabolite is consumed by a reaction, while positive values indicate that a metabolite is produced. Typically, lower and upper bounds on the fluxes $\vec{v}$ are set based on experimental data and thermodynamic reversibility patterns [58]. Genome-scale models are often used to perform flux balance analysis, which solves a linear programming problem for an objective function Z, typically a biomass equation, given constraints from the stoichiometric matrix and the flux limits (eq 1.1) [59].

$$\begin{aligned} &\max\ Z \\ &s.t. \\ &S \cdot \vec{v} = 0 \\ &\vec{v} \leq \vec{v} \leq \vec{v}_{ub} \end{aligned} \tag{1.1}$$

The advantages of genome-scale modeling are that kinetic functions for $\vec{v}$ do not need to be specified, which greatly reduces the amount of experimentation required to describe these systems. Furthermore, due to the steady-state assumption, which suggests that there is no time-dependence of the flux state, the optimization process can be solved using linear programming, which is computationally efficient. This allows for actual genome-scale phenotype prediction, which kinetic models typically do not achieve. However, steady state is an assumption that may not be realistic in the context of bioprocesses, as operating conditions such as batch fermentation and fed-batch fermentation are not in a steady state [59]. Furthermore, solutions to maximize the objective function are typically non-unique; therefore, additional constraints from thermodynamics, protein allocation assumptions, and more conservative flux bounds [60, 61] are required. Third, regulatory interactions at the substrate and protein levels are not easily included. Finally, metabolic dynamics cannot be studied using flux balance analysis, although approaches that assume a pseudo steady-state have been proposed [62].

1.3.3. Mechanistic modeling: thermodynamic modeling

Genome-scale models provide a broad description of feasible flux profiles given an objective function and the mass-balance constraints. However, they do not naturally include other physical constraints on fluxes imposed by the laws of thermodynamics. These can be further imposed by an additional set of constraints. Assume that temperature and pressure are constant (an isothermal and isobaric system). In such a system, the thermodynamic driving force is given by the minimization of the Gibbs free energy (eq. 1.2). Assuming that the system is isothermal and isobaric under laboratory conditions, we can set the entropy (H) and pressure (P) term to zero, resulting in a system where the Gibbs free energy dG depends only on the chemical potential μ and the number of particles N (eq. 1.3). The chemical potential is the amount of energy that can be absorbed or released as a result of a change in the number of particles of a given species. As metabolic systems consist of numerous metabolites m, the total Gibbs free energy in a system without interactions — meaning that no reactions are occurring — is given as the sum over all metabolites (eq. 1.4).

$$dG = -HdT + VdP + \mu dN \tag{1.2}$$

$$(dG)_{P,T} = \mu dN \tag{1.3}$$

$$(dG)_{P,T} = \sum_{i=1}^{m} \mu_i dN_i \tag{1.4}$$

The stoichiometric matrix $S \in \mathbf{R}^{m \times n}$ describes the flow of mass through the system. The change in Gibbs free energy is therefore also dependent on the stoichiometric matrix (eq. 1.5). Here, ϕ is a variable that indicates an infinitesimal turnover of the n reactions. By substituting the definition of chemical potential μ for $RT\,ln(C_i/C_0)$, where R is the ideal gas constant [63], the Gibbs free energy of the complete metabolic system can be defined through the concentrations of the metabolites of the products (C_i) and the substrates (C_o) (eq. 1.6). To sustain and grow a biological system, the total Gibbs free energy must be negative [64]. This can be used to impose additional constraints on genome-scale models, such as reaction directionality and removing thermodynamically infeasible flux loops [65].

$$\Delta_r G = \frac{dG}{d\phi} = S^T \mu \tag{1.5}$$

$$\Delta_r G = \Delta_r G^0 + S^T RT\,ln\,(C_i/C_o) \tag{1.6}$$

1.3.4. Mechanistic modeling: kinetic modeling

Although thermodynamics may describe the expected reaction direction, it suggests little about reaction rates and the metabolic states of the biological system [66]. To describe the evolution of metabolic states, kinetic models are typically used. A kinetic model is a system of ordinary differential equations (ODE) that is typically described by the stoichiometric matrix S, a set of flux functions $\vec{v}$ that depend on the metabolic state $m(t)$, possibly time t, and a set of parameters θ (eq. 1.7) [67]. By numerical integration using an ODE solver, the evolution of the metabolic state can be simulated.

$$\frac{dm(t)}{dt} = S \cdot v(t, m(t), \theta) \tag{1.7}$$

$$m(T) = m(0) + \int_0^T S \cdot v(t, m(t), \theta)\,dt \tag{1.8}$$

Kinetic models have several advantages regarding modeling metabolic systems. First, the flux function parameters are directly related to enzyme properties and can be measured in experiments. For example, the Michaelis-Menten equation for a substrate-to-product reaction relates the flux to enzyme (E) and substrate (S) concentrations with two parameters (k_{cat} and K_m) that can be experimentally determined (eq. 1.9) [68, 69]. Due to this direct link between flux and enzyme properties, genetic engineering strategies at the DNA-level can be more straightforwardly linked to consequences at the metabolic level compared to constraint-based models. A second advantage of kinetic modeling is that it allows one to interrogate the dynamic behavior of living systems. Kinetic modeling has been used to understand the behavior of solid tumors and metastases [70], as well as the kinetic behavior of the virus [71]. In the biotechnological domain, kinetic models have been used to optimize bioprocess feeding strategies [43], testing design-of-experiment strategies [35], optimize strains [72], and several other applications [45]. From existing modeling approaches for metabolic networks, kinetic models provide the most detailed and biochemically grounded description of the system [63].

$$v_{MM} = \frac{k_{cat}[E][S]}{K_m + [S]} \tag{1.9}$$

However, this detail comes at the cost of several limitations. The mechanistic insight required to formulate detailed kinetic descriptions of metabolic phenomena is often missing in terms of the reaction stoichiometric matrix S, the analytic form of the flux functions v, and the necessary parameter values θ for equation 1.7 [67]. This has limited the construction of large-scale kinetic models, as the parameters have to be determined experimentally or estimated from experimental data. Second, because of the dependence on numerically solving the system of ODEs, increasingly large models become more challenging to simulate due to numerical instabilities and stiffness. Therefore, while large-scale kinetic models are becoming increasingly likely [73], their practical use have so far been limited to metabolic pathways.

1.4. Motivation and goals of this thesis

Given the remarkable success of machine learning models across many scientific fields, it is natural to ask how they might also be applied in the context of strain development. In this setting, the main objective is to engineer a microbial cell factory that achieves industrially relevant production levels while simultaneously reducing the time required to progress from an initial proof-of-principle strain to this final production strain (Fig. 1.3). This effectively comes down to studying the application of machine learning in the context of the DBTL cycle.

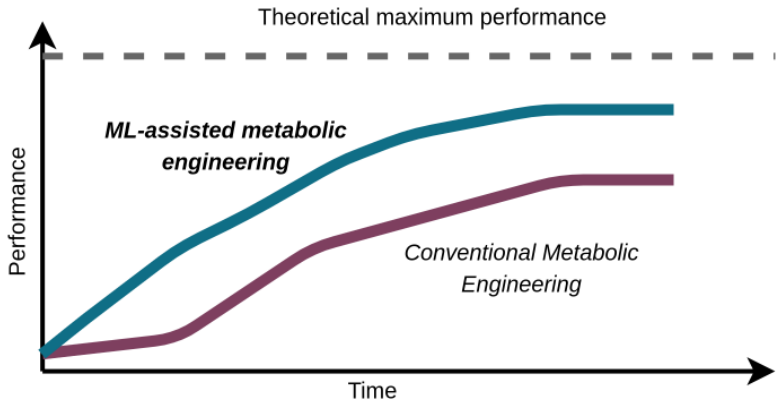


Figure 1.3: **The primary objective of this thesis**. We research the integration of machine learning methods, specifically supervised algorithms and automated recommendation strategies for iterative strain optimization. The ultimate goal is to reduce time and costs of developing a microbial cell factory from a proof-of-principle strain using machine learning-assisted workflows.

As briefly noted in the section on the DBTL cycle, the optimal use of machine learning combined with the DBTL cycle remains largely unresolved. Issues such as experimental design, process-related hyperparameters (e.g., screening throughput and DNA sequencing capacity), and how these factors affect the performance of machine learning models have not been systematically studied. This is mainly because rigorously assessing different DBTL scenarios on real experimental data is time- and resource intensive. For instance, evaluating three machine learning-based recommendation strategies would require building separate sets of strains for the same product task. In practice, this will not be done. An alternative approach is to replace the time- and resource intensive process of real-world experimentation with a surrogate process through simulation, emulating a real-world metabolic engineering process. In the first research direction, we therefore concentrated on creating a kinetic model-based computational framework to evaluate metabolic flux optimization scenarios (**chapter 2**). We examine several scenarios frequently encountered in metabolic engineering to address questions related to experimental design and to determine which types of machine learning models are suitable for automated recommendations.

The second research direction centers on designing DBTL cycles and using machine learning-based strain recommendations to improve strain optimization. Building on **chapter 2**, which indicates that larger combinatorial design spaces are suitable for strain optimization, we construct a large DNA library to optimize p-coumaric acid production in *S. cerevisiae* and propose a machine learning-assisted recommendation strategy (**chapter 3**). In **chapter 6**, we further analyze how the DBTL cycle-related process parameters affect machine learning-assisted optimization performance.

Machine learning-assisted recommendation methods frequently employ supervised models to predict metabolic engineering outcomes [37, 74, 75]. Yet, these models generally operate as black boxes: the internal computations that result in the prediction of metabolite titer, rate, or product yield do not reflect the true behavior of cellular metabolism or the bioprocess. In contrast, kinetic models offer mechanistic insight but are usually difficult to construct and parameterize [67]. A third line of work aims to couple kinetic models with machine learning, resulting in *hybrid* models. Such approaches are increasingly used for modeling partially characterized ODE systems [76–78]. In **chapter 4**, we present software for training systems biology models and hybrid models using JAX/Diffrax. We then apply this framework in **chapter 5** to benchmark several hybrid model architectures on synthetic time-series data and combinatorial pathway optimization datasets.

1.5. Thesis contributions

Chapter 2: In chapter 2, we develop a kinetic model-based framework to perform simulated DBTL cycles to test and optimize machine learning methods for combinatorial pathway optimization, which we term *simulated-DBTL*. We compare a large set of machine learning methods against ground-truth data simulated from a kinetic model. This shows that the ensemble methods Random Forest and XGBoost outperform other

nonlinear and linear methods in predicting production values from strain designs, especially in the low-data regime. Furthermore, we show that for a typically sized optimization problem in metabolic engineering (e.g., six genes and a few promoters), a large initial DBTL cycle followed by a smaller follow-up round, given a certain experimental budget, is preferred over smaller rounds. We therefore hypothesized that machine learning may be leveraged to its fuller potential by expanding the size of the combinatorial design space.

Chapter 3: In chapter 3, the hypothesis posed based on chapter 2 was tested on a real-world example. We designed a DNA library consisting of 19 gene targets paired with 20 promoters for the optimization of p-Coumaric acid in *Saccharomyces cerevisiae*, resulting in a theoretical design space of 170 million design configurations. This significantly expands the space of possible designs compared to the previous literature in metabolic engineering. We performed an initial library transformation round that resulted in a large set of screened and sequenced strains, followed by a machine learning-assisted recommendation strategy for a smaller follow-up round. We experimentally tested the importance of the trade-off between exploration of the combinatorial space versus exploitation of the prediction done by the machine learning model (XGBoost). We show that balancing exploration-exploitation has a prominent impact on p-coumaric acid production of the strains build in the second DBTL cycle. Furthermore, the exploration-exploitation tradeoff becomes more important when a machine learning model is used outside of the training data domain it was originally trained on. In this out-of-distribution prediction scenario, the predictive performance is less guaranteed, which consequently introduces more uncertainty when recommending new strains. Overall, we observe a 137% increase in p-coumaric acid production over the parent strain, with a glucose yield of 0.07 g / g.

Chapter 4: In chapter 4, we develop a simulation and training framework for Systems Biology Markup Language models in *JAX*, using ODE solvers from Diffrax [77, 79]. This allows for automatic differentiation and just-in-time compilation of kinetic models, while also allowing for training of neural ODEs. We show that this can be used to train large-scale kinetic models while also allowing for hybridizing kinetic models with neural networks because of the flexibility of automatic differentiation.

Chapter 5: In chapter 5, we apply the software from chapter 4 to build and evaluate hybrid modeling architectures for time-series data tasks and cross-sectional data modeling, as omics data often lack temporal measurements. We analyze three hybrid model architectures for time-series data, finding that models with stoichiometric constraints perform best. Next, we model genetic perturbations using combinatorial pathway optimization data from chapter 3. Although the predictions of p-Coumaric acid match the machine learning models with better interpretability, the initial mechanistic model did not capture the diauxic growth dynamics of biomass formation [80]. However, after refinement, the architecture effectively predicts p-Coumaric acid and biomass on new strains. This highlights the promise of hybrid modeling for strain optimization, but also underscores existing challenges for broader adoption.

Chapter 6: In chapter 6, we use simulated DBTL cycles to compare a wide range of scenarios *in silico* to optimize the performance of combinatorial pathway optimization by identifying the key variables related to experimental design. This shows that using the machine learning-assisted recommendation strategy presented in chapter 3, the screening-to-sequencing ratio and gene target selection have a major positive impact on optimization performance.

2

Simulated design-build-test-learn cycles for consistent comparison of machine learning methods in metabolic engineering

Paul VAN LENT, Joep SCHMITZ, Thomas ABEEL

Combinatorial pathway optimization is an important tool in metabolic flux optimization. Simultaneous optimization of a large number of pathway genes often leads to combinatorial explosion. Strain optimization is therefore often performed using iterative Design-Build-Test-Learn (DBTL) cycles. The aim of these cycles is to develop a product strain iteratively, every time incorporating learning from the previous cycle. Machine Learning methods provide a potentially powerful tool to learn from data and propose new designs for the next DBTL cycle. However, due to a lack of a framework for consistently testing the performance of machine learning methods over multiple DBTL cycles, evaluating the effectiveness of these methods remains a challenge. In this work, we propose a mechanistic kinetic model-based framework to test and optimize machine learning for iterative combinatorial pathway optimization. Using this framework, we show that Gradient Boosting and Random Forest outperform other tested methods in the low data regime. We demonstrate these methods are robust for training set biases and experimental noise. Finally, we introduce an algorithm for recommending new designs using machine learning model predictions. We show that when the number of strains to be built is limited, starting with a large initial DBTL cycle is favorable over building the same number of strains for every cycle.

This chapter has been published in ACS Synthetic Biology **12.9**, 2588-2599 (2023) [81].

2.1. Introduction

Metabolic engineering focuses on optimizing microorganisms through genetic interventions in metabolic pathways, intending to increase the flux towards a product of interest [82, 83]. Classically, metabolic engineering applies sequential debottlenecking of rate-limiting steps in a pathway of interest [84, 85]. While this method has shown success in many metabolic engineering problems, fundamental limitations to this procedure exist. Despite the vast work on characterizing metabolic networks, there is still a substantial lack of knowledge about many metabolic pathways, especially the regulation of individual pathway elements and cell physiology. As a result, purely rational engineering of pathways remains challenging [86–88]. On top of that, sequential optimization of pathways potentially misses the global optimum configuration of pathway elements (e.g., enzyme concentrations) that maximize the product flux [89].

Recent progress in synthetic biology, genome engineering and high-throughput building and screening of microbial strains now allows for targeting multiple pathway components simultaneously, paving the way for combinatorial pathway optimization [89]. The major advantage of this approach is that there is a reduced chance of missing the optimum pathway configuration, and many successes have been reported for increasing titer/yield/rate (TYR)-values of products using this strategy [90–94]. In combinatorial pathway optimization, strain designs are constructed from a large DNA library consisting of promoters, ribosomal binding sites, coding sequences, and other DNA components that have an effect on enzyme properties or concentrations. These designs are assembled and introduced into a microorganism [95]. Due to the large set of library components, combinatorial explosion of the design space often occurs, making it experimentally infeasible to test every design. Combinatorial pathway optimization is therefore often performed in an iterative fashion using Design-Build-Test-Learn (DBTL) cycles (Fig. 2.1) [92]. The idea is to build an initial set of strain designs and use the generated data from the test phase to learn important characteristics of the pathway. This information is then used to guide engineering in the next cycle. DBTL cycles for microbial strain development are widely adopted. Nevertheless, developing an economically feasible bioprocess is still considered very costly and time consuming. Strategies on how to find the best producing strain with as little experimental efforts as necessary remains an open question. Furthermore, limited availability of public data for multiple DBTL cycles is complicating a systematic comparison of different strategies for their performance [96].

Machine learning has been increasingly used for guiding engineering [82]. For metabolic flux optimization, this ranges from identifying the targets for engineering through unsupervised learning [97] to predicting metabolite concentrations from proteomics data using supervised learning [98]. Another potential application of machine learning is for recommending new strain designs for the next DBTL cycle by learning from a small set of experimentally probed input designs, which would allow (semi)-automated iterative metabolic engineering [99–102]. One example is the Automated Recommendation Tool, which uses an ensemble of machine learning models to create a predictive distribution, from which it samples new designs for the next DBTL cycle given a user-specified exploration/exploitation parameter [101]. While the Automated Recommendation Tool was successfully applied to optimize the production of dodecanol and tryptophan,

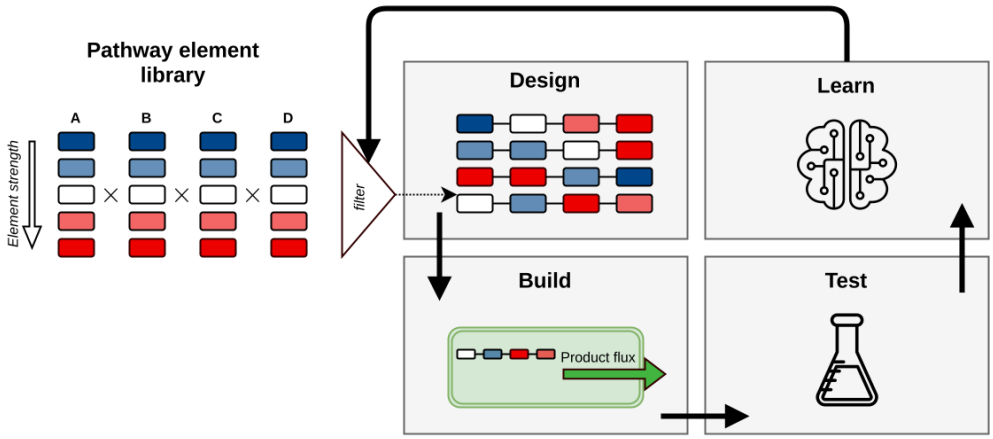


Figure 2.1: **The Design-Build-Test-Learn (DBTL) cycle in metabolic engineering.** In combinatorial pathway optimization, designs are chosen from a large DNA library with pathway elements: promoters, RBSs, CDSs, and other elements that might influence a protein concentration or catalysis rate. Due to combinatorial explosion of the design space, only a small subset of these designs is built and tested experimentally (filter step). From these experiments, important properties for engineering are ideally learned from the experiments and used to design a new set of strains to be build. By iterating this process, the pathway is optimized until the strain is industrially relevant.

instances where the method did not perform well were also reported [101]. This might be attributed to the complexity of the pathways and the lack of data from more than two DBTL cycles [101, 103]. More fundamentally, a framework for comparison of machine learning methods and optimization strategies of the integrated DBTL workflow over multiple iterations is lacking.

Much effort has been put into understanding and modelling cellular metabolism[104–106]. We propose to use a mechanistic kinetic model-based framework for the optimization and application of machine learning methods in iterative metabolic engineering. In kinetic modelling, changes in intracellular metabolite concentrations over time are described by ordinary differential equations (ODEs). Each reaction flux is described by a kinetic mechanism that can in principle be derived from the laws of mass action, so that kinetic parameters can directly be interpreted as biologically relevant quantities. This property of kinetic models allows for *in silico* changes in properties of a pathway element, such as increasing the enzyme concentration or changing catalytic properties of an enzyme.[67, 104, 107]. The ODE model is used to simulate data for comparing the performance of machine learning methods. We start by addressing which machine learning algorithms are favored for this particular use case and test the effect of training set biases and measurement noise on predictive performance. Building on

these results, a recommendation algorithm is introduced to automate DBTL cycles. We demonstrate how our framework can be used to optimize strain development workflows over multiple DBTL cycles for different DBTL cycle strategies.

2.2. Results and Discussion

Publicly available datasets of multiple cycles are scarce due to the costly and time-consuming nature of these experiments[99, 101, 103, 108]. This complicates the validation and comparison of machine learning methods, as well as comparing DBTL cycle strategies. For example, the validation of recommendation algorithms requires that separate DBTL cycles are performed for the same metabolic pathway problem. Similarly, testing multiple DBTL cycle strategies for the same metabolic pathway would realistically never be performed due to the costly nature of these experiments. Simulated data offers an advantage over real-world data as many of these practical limitations are overcome. We will first explain how the kinetic model-based framework was developed to represent a metabolic pathway embedded in a model of cell physiology.

2.2.1. Representation of a metabolic pathway using a kinetic modelling approach

To test different metabolic pathway topologies with distinct thermodynamic properties, we integrate a synthetic pathway into a previously established *E. coli* core kinetic model that was implemented in the SKiMpy package [109, 110]. The aim of this study is not necessarily to build the best possible kinetic model of a specific pathway. Instead, we aim to provide a generic representation of a hypothetical pathway which captures pathway behavior (e.g., enzyme kinetics, topology, rate-limiting steps) and is embedded in a physiologically relevant cell and bioprocess model [91, 101, 108].

A schematic representation of the pathway is shown in figure 2.2A. A degradation reaction was added as a boundary condition and the optimization objective is maximizing the production of compound G. The response of the product flux G to perturbations of the enzyme concentrations are shown as panels along the pathway (Fig. 2.2A). Despite the pathway being almost linear, the influence of enzyme concentrations on this flux is nonintuitive. For example, perturbations of enzyme A do not lead to differences in its respective reaction flux, but do lead to 1.5-fold increase in the product flux (Fig. 2.2A, *lower left panel*). Another example is the response to local perturbation in enzyme B. Here, a decreasing reaction flux is observed due to depletion of substrate. However, no significant effect is observed on the product flux (Fig. 2.2A, *upper left panel*). Another observation is that lowering the enzyme concentration in the last step (reaction G) of the pathway leads to an increase in net production (Fig. 2.2A, *lower right panel*). Thus, increasing enzyme concentrations of the individual reactions does not lead to higher fluxes, but instead leads to decrease in flux due to depletion of reaction substrates. These observations stress the importance of combinatorial optimization of pathways, as sequential optimization strategies might have non-intuitive outcomes[89, 98, 111]. To illustrate this, the combinatorial optimization of reaction A and B is shown (Fig. 2.2B). Note that increasing both enzyme concentrations leads to higher product flux then their

individual local perturbations. We therefore conclude that the dynamics of the pathway captures the features of a typical pathway subject to metabolic engineering well (Fig. 2.2B).

On top of being able to represent a metabolic pathway, the cell model should be embedded in a basic bioprocess model. Figure 2.2C shows the modelling of a one liter batch reactor bioprocess, inoculated with one gram of initial biomass [110]. Glucose is consumed until it is depleted and an exponential biomass growth phase is observed. Furthermore, product is formed in the process. After depletion of glucose, no more biomass is formed and the growth rate is zero. We show that this kinetic model captures important characteristics of a batch bioprocess. This could be extended to model other types of bioprocesses, such as fed-batch fermentation [112]. While the implemented pathway considered here has no metabolic burden on the host, these types of effects could be captured by explicitly modelling inhibitory effects of pathway intermediates on the biomass equation (see Fig. S10)[88]. When including these types of interactions, as well as other types of pathways into the kinetic model using ORACLE sampling, physiological relevance of the kinetic parameter sets can be easily verified [110, 113].

2.2.2. Combinatorial pathway optimization strategies can be simulated and used for benchmarking machine learning models

Although machine learning is increasingly being explored for combinatorial pathway optimization, it remains unclear how to optimize the approach over multiple DBTL cycles[82, 99, 101, 102]. For example, questions on the amount of strains that need to be built for effective learning, the effect of biases in the DNA library distributions on predictive performance, and how the DBTL cycle strategy should be set up over multiple cycles, remain unsettled.

Figure 2.3A shows an example of a simulated metabolic engineering scenario for fifty designs, where enzyme levels are varied with respect to the enzyme level of the initial strain. The effect of adjusting enzyme levels was implemented in the model by changing the V_{max} parameters (see *Methods*). We assume here that the enzyme level change can be achieved with a set of DNA elements (e.g., promoters or ribosomal binding sites) from a predefined DNA library. Considerable effort has been put into experimentally quantifying the promoter strength of promoter sequences [114–116], as well as predictive tools that guide promoter sequence engineering to achieve desirable enzyme expression levels[117–120]. While in this study five different enzyme levels were considered, this might not always be attainable in real-world metabolic engineering, due to a lack of promoters for a particular enzyme, or not being able to achieve up/down-regulation levels considered here. This challenge could be overcome in simulation studies by considering less enzyme levels or considering enzyme levels closer to levels of the initial strain.

It can be observed from the strain designs that the highest producing strains have a bias in low promoter strengths for enzyme G and tends to have upregulation of enzyme A (Fig. 2.3A). In combinatorial pathway optimization, it generally occurs that some pathways steps are more important to increase product flux than others, while some have

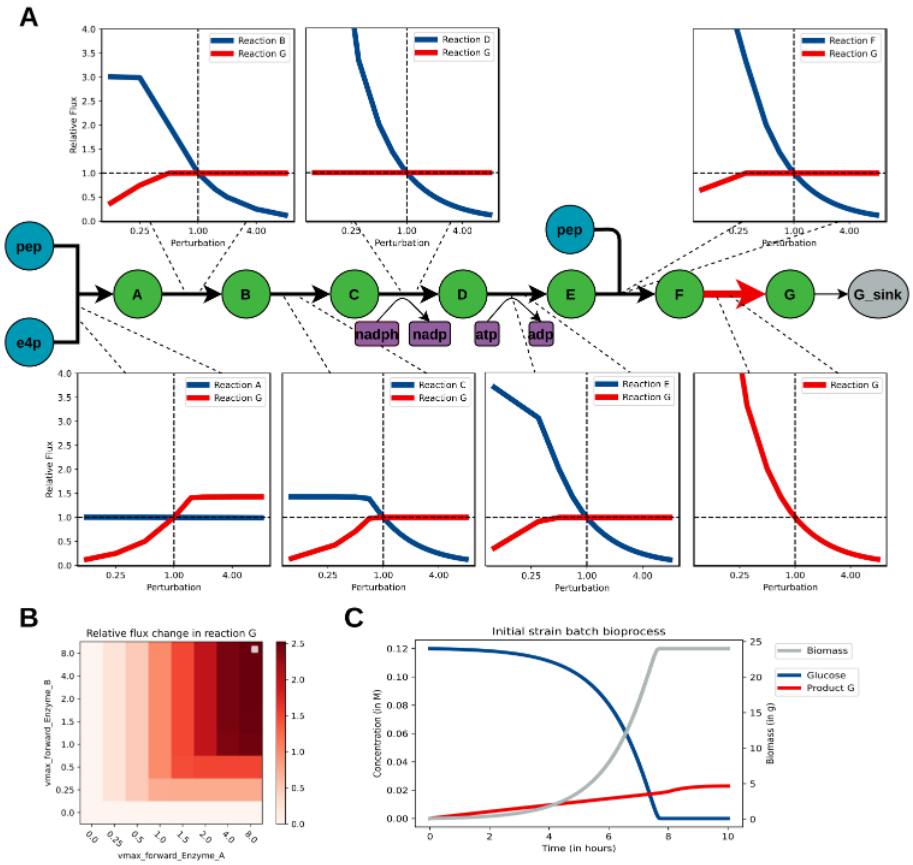


Figure 2.2: **Local enzyme concentration perturbations results in changes in flux.** *(A)* A simplified schematic of the integrated pathway. The pathway is coupled to the core metabolism through the substrates erythrose-4-phosphate (e4p) and phosphoenolpyruvate (pep), as well as the cofactors nadph and atp. For each reaction, response in the steady state flux for the flux through reaction G and its respective reaction is shown, where local enzyme perturbations are performed on the range $[0.125, 8]$. Perturbations are defined with defined with respect to the enzyme concentrations of the initial strain (i.e. $\frac{J([E])}{J([E]_{initial})}$). *(B)* An example of a combinatorial design space for reaction A and B. Note that the behavior of perturbing enzyme A and B is not obvious from the local perturbation shown in A. *(C)* A batch bioprocess of the initial (heterologous) strain with the implemented pathway. Although this model was constructed for intracellular metabolism, a simple batch bioprocess can be modelled, with depletion of glucose, biomass growth, and product formation.

limited sensitivity. We therefore showcase that the simulated data captures key features of

combinatorial optimization in real world data[91, 121]. While in this work only changes in enzyme levels were considered, other enzyme properties (e.g. catalytic properties of the reaction) could be assembled in a similar fashion. This is especially important when the cost of expressing enzymes of a non-native pathway needs to be considered.

The encoded designs along with the simulated product flux can then be used to train a model and compare based on two metrics (Fig. 2.3B). These designs will be used to train different types of models. As an example a multiple linear model and a nonlinear Gradient Boosting Regressor is evaluated using two metrics. The Pearson Correlation (R^2) between the predicted versus simulated product flux gives a general impression of the predictive performance of both methods (Fig. 2.3C). We use the intersection between the predicted top 100 and the simulated top 100 designs as an additional performance metric. This metric captures how well the model has learned the best performing strains (Fig. 2.3D). Together these metrics will be used to consistently evaluate the effect of training set sizes, sampling scenarios, and noise models on predictive performance of machine learning models. Here, the predictive performance of models is only evaluated on the flux through reaction G. When considering other objectives, such as biomass growth or reduction of toxic intermediate concentration in a pathway, models could also be compared for their performance as a multi-objective optimization problem [122]. This can for example be done by defining the target variables as a weighted function of the objectives (see Fig. S9-10).

2.2.3. Ensemble methods excel in the low data regime

Due to a large number of existing machine learning models, we sought to filter out the best performing algorithms from an initial search using default parameters. Eight machine learning models are compared for different training set sizes, where each enzyme level was chosen with equal probability (see *Methods*). The product flux was predicted for the full combinatorial design space (Fig. 2.4A,B).

Figure 2.4A shows the general performance of the machine learning models for increasing number of data points in the training set. Overall, the nonlinear methods outperformed the linear methods for all training set sizes. This could be attributed to the observed non-linear behaviour of the pathway (Fig. 2.2A, *all panels*). The ensemble methods Random Forest and Gradient Boosting Tree outperform the other methods, specifically in the regime where training set sizes are small. This finding is consistent with a previous report that suggests that XGBoost (a regularized Gradient Boosting algorithm) performs well when data is scarce [102]. When training sets are larger, Neural Networks also tend to perform well. Due to the nature of the metabolic engineering experiments, methods that perform well and stable on small training sets are preferred.

Results on the top 100 prediction metric further support the view that ensemble methods and Neural Networks outperform other methods (Fig. 2.4B). Neural Networks here even outperform the ensemble methods. Interestingly, while the linear Support Vector Machine Regressor performs poorly on the general prediction task, prediction of the top 100 designs seems to be similar, if not better, than the ensemble models (Fig. 2.4B). We

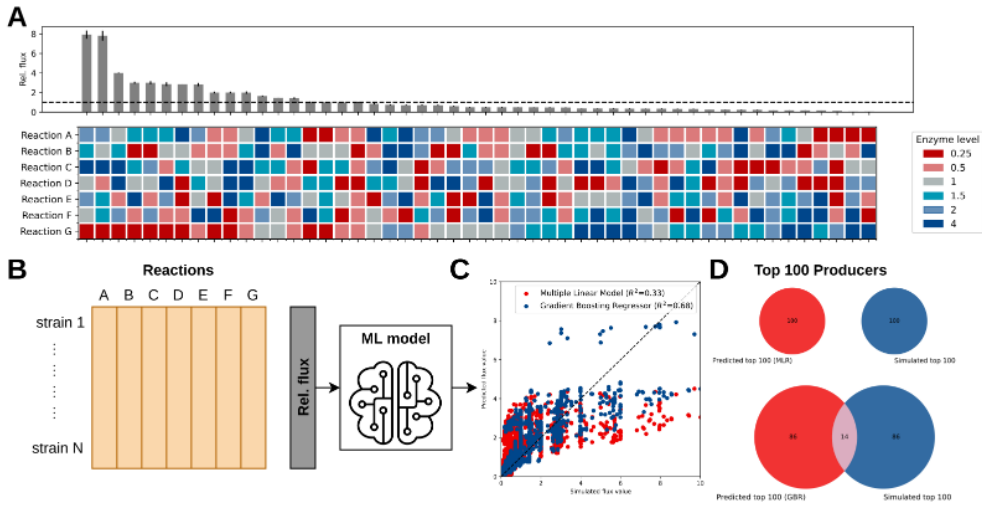


Figure 2.3: **An example of the simulation of large strain libraries**.
(A) Example of fifty simulated combinatorial designs, where each enzyme levels was chosen with equal probability. These levels could be achieved with a promoter from a DNA library. The distribution of enzyme levels of each strain design are shown below the response response curve of the metabolic flux through enzyme G. *(B)* The design space along with the relative flux increase can be used to train a predictive model. *(C)* Predicted versus simulated relative fluxes for a multiple linear regression and gradient boosting regressor, with R^2-values of 0.33 and 0.68, respectively. *(D)* Venn diagrams of the performance of predicting the top 100 designs for the full combinatorial design space compared to the simulated combinatorial space (279.936 designs)

observed that predicting the top 100 best producers with linear Support Vector Machine indeed outperform the ensemble methods, but plateaus in its predictive performance for very large training set sizes (see *Supporting Information*: Fig. S1). Based on these results, we concluded that the ensemble methods Gradient Boosting and Random Forest, along with linear Support Vector Machine and Neural Networks should be further investigated for their performance.

2.2.4. Machine learning models are robust to training set biases and noise

An important aspect for ensuring good generalization of predictions to other unseen designs is that the initial set of constructed strains constitutes a representative sample of the combinatorial design space. In many combinatorial pathway optimization procedures, these first strains are often build using a Design of Experiment approach, such as full factorial or fractional-factorial design setups [91, 123–125]. Alternative to Design of Experiment approaches, designs can be assembled in a probabilistic fashion through the use of DNA libraries [103]. Even though this might not lead to an optimal set

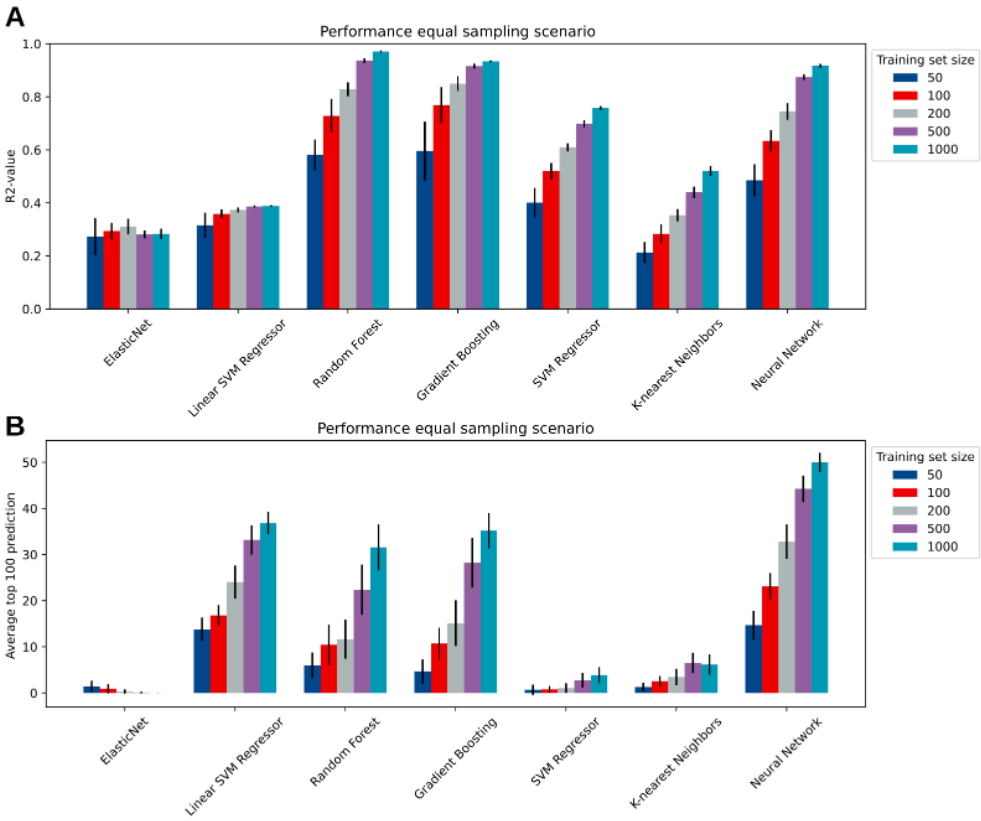


Figure 2.4: **Influence of training set size on predictive performance.** Seven machine learning algorithms (two linear and five non-linear) were tested for their performance on two metrics: a general prediction on the combinatorial design space (R^2-value), and the performance in pointing out the top 100 best performing strains (predicted vs. simulated). *(A)* $R2$-value over the full design space for the seven algorithms with increasing training set size (20 runs per training set size). *(B)* Intersection value between predicted vs. simulated top 100 (averaged predictions, 20 runs per training set size).

of designs (see Fig. S7), it allows for building many strains in a one pot transformation approach. However, this method potentially introduces biases in the distribution of the assembled designs. To understand the effect of enzyme level distribution biases of designs on model performance, we determined three scenarios: a bias for higher enzyme expression levels (strong effect bias), a bias that has enzyme expression levels closer to the initial strain values (mild effect bias), and a scenario where all enzyme levels are equally represented (equal sampling) (Fig. 2.5A).

The general prediction performance (R^2) and the top 100 prediction after Bayesian hyperparameter optimization are shown for two different training set sizes ($N = 50$ and

$N = 200$) (Fig. 2.4B,C,D,E). For the general prediction task, we do not observe a significant difference in performance between the three scenarios, indicating that there is only a marginal effect of the sampling distribution on how well the design space is learned for all algorithms (Fig. 2.4B,C). For the top 100 prediction a difference is observed, where the strong effect bias of enzyme levels tends to predict slightly better on the top 100 (Fig. 2.4C,D). This effect can largely be attributed to the optimization problem that is considered here. The top 100 simulated strains mostly consist of enzyme levels that have a strong perturbation effect with respect to the initial strain, which leads to slightly better performance of the *strong effect bias* distribution. Due to the marginal effect of the enzyme level distributions on the R^2, we therefore expect that this does not generalize to other pathways.

In addition to testing the influence of enzyme level distribution biases, we also tested the influence of noise on predictive performance. A Homoscedastic and heteroscedastic noise model was used on the measured product flux for two different noise percentages (4% and 15%), but no significant effect was found for the four models considered above (Fig. S2,Table S3,S4).

2.2.5. DBTL cycle simulations reveal experimental design principles that may guide real-world metabolic engineering experiments.

While in certain cases the optimal design can immediately be predicted from the initial sampling (i.e. the first DBTL cycle), other cases require multiple DBTL cycles to increase product fluxes to a desirable level. The challenge of effectively suggesting new strain designs for the next DBTL cycle using machine learning remains challenging. Several recommendation algorithms have been introduced in literature[100–102], among which the Automated Recommendation Tool (ART) stands out as a notable example [101]. While a rigorous benchmark of the different recommendation algorithms is outside the scope of this study, an example of how simulated DBTL cycles can be used for this purpose is reported in the Supporting Information (see Fig. S8).

To test the behavior of ML algorithms over multiple DBTL cycles, we introduce a fast and model-free recommendation algorithm to automated recommendation. Learned from an initial sampling scenario using any machine learning algorithm, the predicted design space is used to generate a sampling distribution of the enzyme levels for each enzyme (see *Methods*). In short, frequencies of enzyme levels for each enzyme are counted above a certain flux threshold ($J \geq \lambda_*$). The frequencies of the enzyme levels in the subspace can then be followed as a function of the threshold λ_* (Fig. 2.6). To go from this threshold plot to a sampling distribution, the area under the curve (AUC) is taken and normalized for enzyme level of the respective enzyme.

Two enzymes are shown as an example of how the sampling distribution is generated from the learned combinatorial design space. Gradient Boosting Regressor is trained on 200 training samples, but we note that this recommendation algorithm is independent of the used model. We observe from the thresholded frequency plots that enzyme A has a bias towards higher enzyme levels in the space where flux is high. After taking the AUC, it

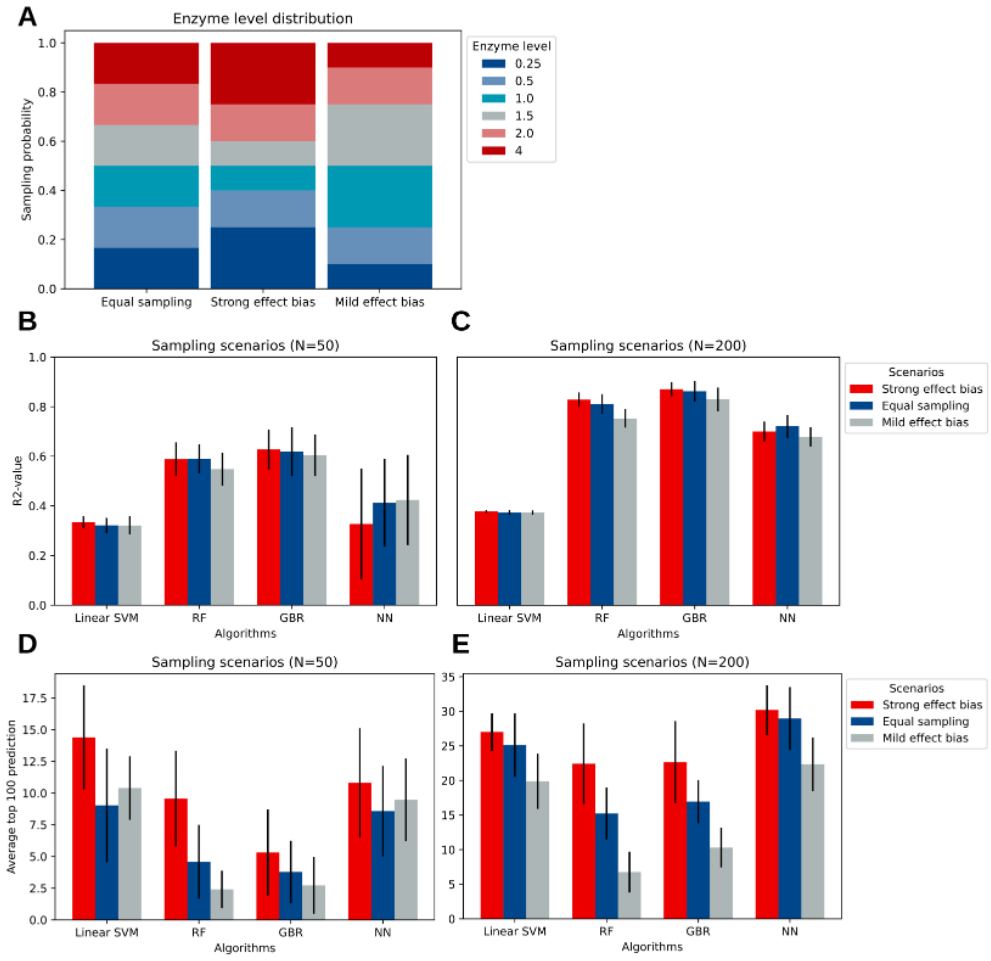


Figure 2.5: **Effect of sampling biases on predictive performance** *(A)* Three different sampling scenarios are defined (see *Methods*). The equal sampling scenario is unbiased, while the strong effect bias and mild effect bias are biased towards certain enzye levels. *(B,C)* R^2 for the four best performing algorithms trained on 50 and 200 samples, respectively. *(D,E)* Top 100 prediction performance for the four best performing algorithms trained on 50 and 200 samples, respectively. A slightly better performance is observed for the strong effect bias sampling scenario, which can be attributed to the fact that the strong enzyme levels with respect to the wild-type are overrepresented in the simulated top 100.

is observed that the new sampling distribution captures this bias, which means that in the next DBTL cycle it is more likely that the higher enzyme levels are chosen (Fig.

2.6A,C). In contrast, enzyme D does not seem to have any bias towards certain enzyme levels as a function of the threshold, with the frequencies being close to equiprobability even in the higher flux range. This leads to an almost equal sampling distribution for enzyme D in the next DBTL cycle (Fig. 2.6B,C). From an experimental perspective, the biased enzyme level distributions can be achieved by adjusting the DNA content in the library transformation. DNA sequencing can then be used to confirm whether there was a successful introduction of the intended bias[126]. Using this approach, it is now possible to completely automate DBTL cycles and use this to test different DBTL cycle scenarios (see Fig. S3-6).

Table 2.1: **Performance of three DBTL cycle scenarios for the Gradient Boosting Regressor.** For all five cycles the predicted top 100 and R^2 are reported.

Strategy	Metric	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5
(50,50,50,50,50)	R^2	0.47 ± 0.15	0.74 ± 0.07	0.81 ± 0.05	0.84 ± 0.04	0.87 ± 0.03
	Top 100 prediction	43.97	62.27	72.7	80.03	82.67
(150,50,25,25)	R^2	0.75 ± 0.07	0.82 ± 0.05	0.84 ± 0.04	0.84 ± 0.03	-
	Top 100 prediction	78.9	82.03	84.3	83.1	-
(25,25,50,150)	R^2	0.32 ± 0.19	0.58 ± 0.07	0.78 ± 0.05	0.87 ± 0.03	-
	Top 100 prediction	16.6	38.9	59.84	83.9	-

We utilize the model-free recommendation algorithm to evaluate the performance of DBTL cycle scenarios under different settings. These settings may include variations in the number of samples per round, utilization of different models, etc.. Here, we showcase three distinct DBTL cycle strategies. For all three scenarios, only 250 strains are allowed to be built over the course of all DBTL cycle rounds: every round building a similar number of strains, a large set of strains in the first cycle and then a decreasing number of strains, and starting with only a few strains and building many strains in the later stages (see *Methods*, Table 2.3). Every scenario is initialized by equal sampling scenario as described above (see Fig. 2.5).

Table 2.1 reports the performance of the three DBTL cycle scenarios for Gradient Boosting Regressor, as this algorithm was shown to outperform the other methods (see *Table S5*). For all scenarios, we do not observe significant differences in the performance of the strategies after all DBTL rounds have been performed. One interesting observation is that the scenario with a large initial building phase already performs well on the top 100 prediction and does not improve drastically over the rounds. This indicates that for some metabolic engineering problems, DBTL cycles are not necessarily required to optimize strains as long as your initial sampling is large. In these cases the design of the next DBTL cycle should increase the size of the design space. This could be extending the range of the considered enzyme levels (e.g., by considering strong promoters) or using other pathway elements that were not included in the initial DNA library transformation.

To conclude, we have developed a framework for a consistent comparison of machine

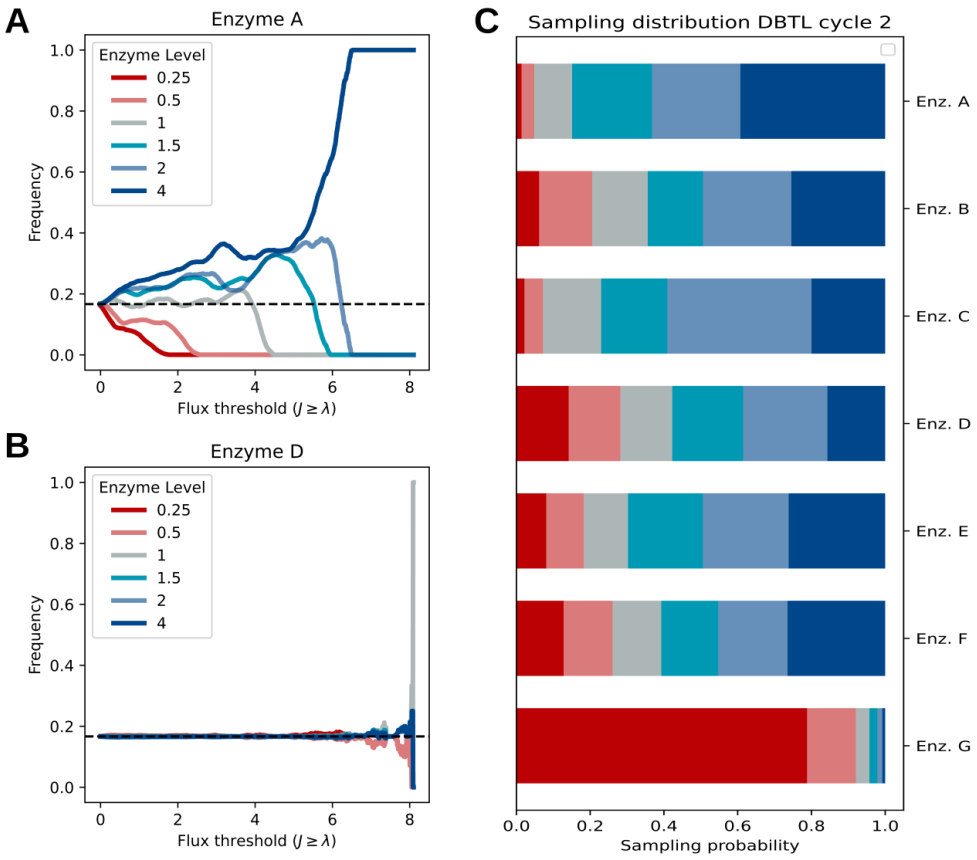


Figure 2.6: **Recommending new designs based on the learned combinatorial space.** For each enzyme, the frequency of each enzyme level is calculated (promoter strength) as a function of the threshold. Then, we take the AUC of all enzyme levels and normalize. *(A,C)* Example of the frequency for enzyme A when a model is trained on 200 samples. As seen, the higher enzyme levels are more frequent than the lower levels and when taking the AUC, this is resembled in the sampling distribution of promoters for the next round. *(B,C)* An example of a feature that is not considered important for the optimization problem. As the threshold increases, we do not see a certain enzyme level (promoter strength) being favored, which can also be observed from the sampling distribution.

learning methods over multiple DBTL cycles. We show that Gradient Boosting Regressor outperforms other methods in the low data regime. After introducing a recommendation algorithm, we show that for this metabolic flux optimization problem multiple DBTL cycles are not always necessary when the first cycle has many built strains. The developed

framework is not limited to this pathway, but can also be extended to other pathways with distinct topologies. Furthermore, strain building methods other than the DNA library transformations considered here (e.g., Design of Experiment approach) could be compared using this framework, similar to how we tested the effect of training set biases (Fig. 2.5, S7)[91, 123–125].

2.3. Methods

An overview of the simulation pipeline used in this study is shown (Fig. 2.7). The input kinetic model with the included synthetic pathway is constructed as described in the next section. Enzyme names and perturbation values are used to construct a combinatorial design list to simulate steady state flux rates compared with respect to the wild-type. In the first DBTL cycle round, an initial sampling scenario is chosen to create training sets. Noise is then added to the target value (product flux) based on a noise model to mimic experimental/technical noise. Next, a machine learning model is trained and the full combinatorial design space is predicted. The performance is measured based on two metrics: the general predictive performance described by the R^2 and the performance in predicting the top 100 designs by comparing it to the simulated top 100 designs. All steps were performed in Python 3.9 and all code used in this study is provided on *https://github.com/AbeelLab/simulated-dbtl*.

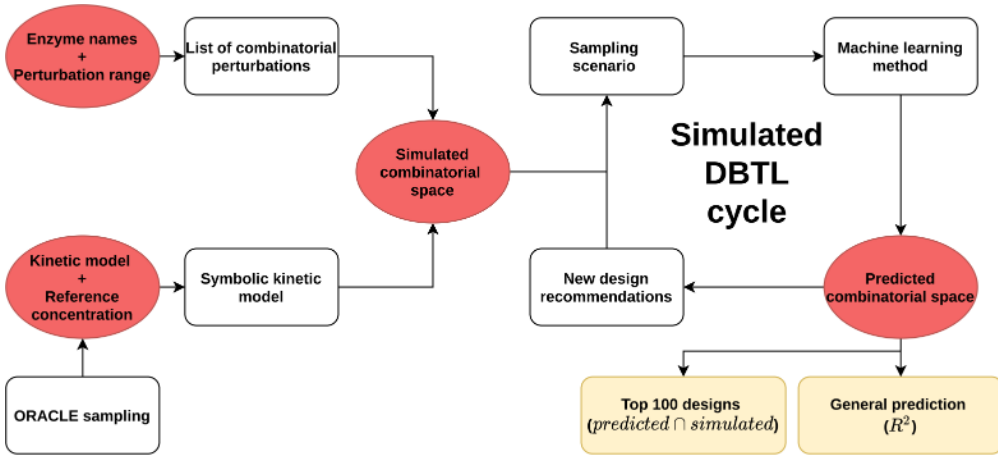


Figure 2.7: **Overview of the pipeline used in this study.** A synthetic pathway is integrated into the kinetic model to be used for ODE simulations. From this space we test multiple sampling scenarios and train seven models. Two different performance metrics are used to evaluate. As the simulation of the full design space is computationally intensive, the intermediate files of the pipeline are saved (shown in red)

2.3.1. Including synthetic pathways in a *E. coli* core kinetic model

We use a previously published kinetic model of the core metabolism of *E. coli*, that was reformulated to be used in the Symbolic Kinetic Models in Python (SKiMpy) package [109, 110]. SKiMPY is a recently developed Python package for semi-automated generation of kinetic models, along with a variety of tools for downstream analysis [110]. The model consists of 64 metabolites involved in 65 reactions, of which 49 metabolites have an ordinary differential equation (ODE). The other metabolites are considered constant boundary conditions and are not directly involved in the pathway that was implemented.

We wish to include a pathway with thermodynamic properties that is reminiscent of a biological pathway into the core metabolism of *E. coli*. With this in mind, we use the structural thermokinetic modelling framework to generate kinetic models that are consistent with stoichiometric and thermodynamic constraints, as well as a biomass optimization objective often used in constraint-based models[127–129]. We start by adding a pathway to the stoichiometry of the core model and add thermodynamic information from a provided database to impose a flux directionality profile [128, 130, 131]. Additional constraints on the steady state metabolite concentrations of the pathway are imposed (see *Supporting Information*). Based on these constraints, the biomass objective that was provided is optimized and a sample of fluxes, concentrations, and equilibrium constants, are taken from the feasible solution space. Next, from the reaction stoichiometry and thermodynamic properties kinetic mechanisms were identified that are available in SKiMpy [110]. Finally, we back-calculate and sample kinetic parameters from the fluxes, concentrations, and kinetic mechanisms and check whether they lead to stable behaviour of the ODE-system using ORACLE parameter sampling [132]. Additional details on the stoichiometry and constraints is further described below and in the Supporting Information (see Table S1).

The included synthetic pathway

A seven-reaction pathway that was inspired on the shikimate pathway is included, with phosphoenolpyruvate and erythrose-4-phosphate as substrates (Table 2.2)[133]. Metabolite G is considered as a boundary condition that is kept constant. The optimization objective is the maximization of the flux through reaction G. Thermodynamic information was added from a thermodynamic database that was included in the pyTFA package [130]. Further information on the kinetic parameters are reported in the Supporting Information (see Table S2).

2.3.2. Metabolic engineering scenarios

The goal is to generate synthetic metabolomics data from the previously described kinetic model for scenarios that are realistic in industrial metabolic engineering [91, 92, 103]. We therefore perturb the V_{max} of a particular enzyme in the pathway, solve the initial value problem of the ODE-system until it reaches steady state, and calculate the relative increase/decrease of the target flux with respect to the wild-type.

Table 2.2: **An overview of the properties of synthetic pathway**. Newly included metabolites are capitalized, while the already modelled metabolites are shown with lower case letters. Thermodynamic flux analysis was performed using the pyTFA package to add additional constraints on the feasible parameter space for sampling steady-state consistent parameter sets [130, 132]. The kinetic mechanisms used are Irreversible Michaelis Menten (Irrev. MM), Reversible Michaelis Menten (Rev. MM), and Generalized Reversible Hill (Gen. Rev. Hill). Kinetic parameters for each reaction are reported in the Supporting Information.

Reaction	Name	$dG(in\,kJ/mol)$	Kinetic mechanism
$pep + e4p \rightarrow A + pi$	Reaction A	-12.03	Irrev. MM
$A \rightarrow B$	Reaction B	-16.12	Irrev. MM
$B \leftrightarrow C$	Reaction C	-8.47	Rev. MM
$C + nadph \leftrightarrow D + nadp$	Reaction D	1.10	Gen. Rev. Hill
$D + atp \rightarrow E + adp$	Reaction E	-6.40	Gen. Rev. Hill
$pep + E \leftrightarrow F + pi$	Reaction F	0.41	Gen. Rev. Hill
$F \rightarrow G + pi$	Reaction G	-11.77	Irrev. MM

Vmax as a proxy of enzyme concentration

To simplify the combinatorial designs used in this study, the focus is on changing the concentration of the N pathway enzymes by changing the V_{max} of the enzyme in the kinetic model. For each enzyme the maximum flux is given by $V_{max} = k_{cat} * [E]$, where k_{cat} is the turnover rate of the substrate by the respective enzyme and $[E]$ is the enzyme concentration. As k_{cat} is a constant for each gene variant, it follows that modulation of the enzyme concentration by the relative gene up/down-regulation results in a proportional change in the V_{max} (eq. 2.1).

$$V_{max} \propto [E] \tag{2.1}$$

Combinatorial design scenarios

To model combinatorial designs, two assumptions are made. First, enzyme levels can be effectively manipulated using some combinatorial DNA library on the range $[0.25 - 4]$. Second, the order of strengths for these DNA elements is known *a priori*. In real metabolic engineering, this prior information could for example be predicted using a promoter strength prediction algorithm [120, 134]. Enzyme levels are represented relative to the enzyme concentration of the initial strain. For example, when in a metabolic engineering process N enzymes of a pathway are considered, a two-fold upregulation in the second enzyme and two-fold downregulation in the fourth enzyme would be encoded as (eq. 2.2):

$$Design = [1, 2, 1, 0.5, 1, ..., N] \tag{2.2}$$

From the combinatorial design space containing P^N designs, only a small subset can be experimentally probed. In the equal sampling scenario, we choose each enzyme

level with equal probability and assemble the design [103]. To mimic the effect of biases in the building of strains (e.g., due to experimental limitations), we define two alternative sampling scenarios. The strong effect bias sampling scenario means that stronger enzyme levels with respect to the initial strain are assigned higher probabilities to be sampled. Conversely, the weak effect bias sampling scenario means that we assign a higher probability to enzyme levels closer to the initial strain enzyme level.

2.3.3. Machine learning methods

All models are trained on the three scenarios. The features are the enzymes, where values indicate the enzyme level with respect to the initial strain. Strain designs encoded as described in the previous section and are used as instances. The target variable is the product flux through enzyme G. For the linear methods, we chose Elastic-Net and linear Support Vector Machine (linear SVM) Regressor. For the non-linear models we chose Random Forest (RF) Regressor, Gradient Boosting (GBR) Regressor, Feed-Forward Neural Network (NN), Support Vector Regressor, K-NN Regressor, and Gradient Boosting Regressor[135].

Neural Networks, Random Forest, and Gradient Boosting require hyperparameter optimization in order to prevent overfitting. Hyperparameters were chosen by using Bayesian hyperparameter optimization through a five-fold cross-validation on the training set, using twenty iterations. For this, *scikit-optimize* was used[136].

Model validation

Model performance was evaluated based on two metrics: how well machine learning methods predict on unseen designs from the full combinatorial set, and the performance in predicting the top 100 highest producing strains. The Pearson Correlation Coefficient (R^2) between the predicted versus simulated steady state flux was used to address the general prediction performance. For the performance in finding the global optimum, the fluxes are predicted based on the trained model and the set of the top 100 best producing predicted designs are compared to the actual top 100 best producing strains from the simulated combinatorial design space (eq. 2.3).

$$score = top\,100_{predicted} \cap top\,100_{simulated} \tag{2.3}$$

As the sampling scenarios are probabilistic in nature, we perform the training set scenario and test set simulation twenty times to estimate a mean and variance R^2.

Noise models

Metabolic networks are inherently stochastic and measurement errors in determining metabolite concentrations and reaction fluxes exist. Two different noise percentages were chosen for testing the robustness of machine learning models to noise. Thus, after the mutant is simulated and compared with respect to to the wild-type, noise is added by drawing a value from a normal distribution for a homoscedastic case and heteroscedastic case ($\sigma = 0.04, 0.15$).

2.3.4. Design-Build-Test-Learn cycles

Machine learning models were trained on a small subset of designs that were sampled from an initial sampling scenario, as described in the preceding section. After training, models are used to predict the product flux of the combinatorial design space. We developed the following algorithm to recommend designs for the next round based on the machine learning model predictions.

Recommending new designs for the next cycle

A trained model is used to predict all possible combinatorial designs, which for a pathway with N enzymes and each enzyme having for example P enzyme levels (that might be achieved by a set of promoters) results in P^N predictions. The strain designs are sorted in ascending order by their predicted flux. A threshold λ_* is introduced, which is defined between zero and the maximum predicted flux ($0 \leq \lambda_* \leq y_{max}$). The threshold λ_* is increased from zero to y_{max} given a predefined number of evaluations. At each evaluation, only combinatorial designs where the predicted flux is higher than λ_* are considered ($y_{pred} \geq \lambda_*$). For each enzyme, the frequency of enzyme levels is counted and normalized by the total number of designs where $y_{pred} \geq \lambda_*$.

When $\lambda_* = 0$, the frequency of each enzyme level is given by $p = 1/P$. This is because all combinatorial strain designs have a product flux higher than zero, and therefore each enzyme level is equiprobable. As λ_* is increased, the statistical contribution of each enzyme level to the remaining part of the combinatorial design space can be followed. Promoters that have no significant contribution to higher product flux are expected to be underrepresented.

To transform the threshold plot to a distribution that can be used for sampling new strain designs, the area under the curve for each enzyme level is determined and normalized. This results in a probability distribution for each considered enzyme as a machine learning guided way to sample new designs. Finally, we retrain the machine learning model in the next DBTL cycle, also including data from past cycles [103].

DBTL cycle strategies

Three DBTL cycles strategies were compared for the four best performing machine learning algorithms. For a fair comparison, at most 250 strains are built over multiple cycles.

The first scenario is one where for each DBTL cycle round an equal amount of strains are built. The second scenario consists of building many strains in the first cycle and then decrease for each round. Finally, a reversed scenario is defined, where we start with building only few strains, and then increase (Table 2.3). Each metabolic engineering experiment consisting of five cycles is initialized by an equal sampling scenario of enzyme levels. 15% homoscedastic noise is added to each measurement and each scenario is performed 30 times. To assess the performance of the DBTL cycle strategies, the top 100 prediction score and R^2 are reported to give a general impression of the

performance for each strategy. Additional performance measures are reported in the Supporting Information.

Table 2.3: **DBTL cycle scenarios.** 250 strains were built over multiple rounds

	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5
Scenario 1	50	50	50	50	50
Scenario 2	150	50	25	25	-
Scenario 3	25	25	50	150	-

2.4. Supporting Information

Additional data, results, and supplementary material can be found in the original publication (see QR-code).

3

Machine learning-assisted pathway optimization in large combinatorial design spaces: a p-Coumaric acid case study

Paul VAN LENT, Rianne VAN DER HOEK, Sara MORENO PAZ, Moniek JONKER, Irsan KOOI, Priscilla ZWARTJENS Joep SCHMITZ, Thomas ABEEL

Combinatorial pathway optimization is a powerful approach in metabolic engineering to improve strain performance. While machine learning (ML) has shown promise in guiding the Design-Build-Test-Learn (DBTL) cycle, most applications have been limited to small design spaces, thereby restricting the potential of predictive and exploration-exploitation strategies. In this work, we applied two DBTL cycles to optimize p-Coumaric acid production in Saccharomyces cerevisiae. The first cycle involved constructing a large combinatorial library of 18 genes and 20 promoters (170 million possible designs). In the second cycle, we employed a gradient bandit-based machine learning recommendation strategy, tuned to balance exploration and exploitation. Our results show that this balanced strategy outperforms greedy, feature importance-based approaches, leading to greater diversity in strain performance and improved top-producer identification. Notably, applying the same strategy to an alternative parent strain yielded the highest p-Coumaric acid titer (1.23 g/L), a 2.37-fold improvement over the original. These findings highlight the value of ML-guided exploration in large design spaces and demonstrate that balancing exploration and exploitation is critical for successful strain optimization.

This chapter has been accepted in ACS Synthetic Biology [137]

3.1. Introduction

Transitioning chemical production to sustainable processes is critical for addressing climate and biodiversity challenges [138]. Microbial cell factories offer an environmentally friendly alternative for producing pharmaceuticals, feed additives, and bulk chemicals [139]. However, the lengthy and costly development of proof-of-concept strains limits their widespread adoption [140, 141].

Advances in genome engineering and high-throughput screening have paved the way for combinatorial pathway optimization [30]. This strategy targets multiple pathway elements simultaneously to identify configurations that maximize titer, rate, and yield (TRY-values) [20]. While it overcomes the limitations of sequential optimization [30], it also creates a combinatorial explosion of possible designs, making exhaustive screening impractical, even with high-throughput methods [33]. To mitigate this challenge, metabolic engineers often choose to limit the number of pathway elements simultaneously considered [142] and employ the Design-Build-Test-Learn (DBTL) cycle to navigate the vast combinatorial landscape [39] (Fig. 3.1A). In the DBTL cycle, an initial set of designs is introduced into a parent strain (Design phase) and evaluated for TRY performance (Build and Test phases). Insights from these results are used to inform subsequent design iterations (Learn phase), which may use the same parent strain or a higher-performing derivative. Conceptually, this iterative process resembles a Markov decision framework (Fig. 3.1B), where strains represent states, pathway designs are actions, and production outcomes serve as rewards. The overarching goal—maximizing TRY values over successive cycles—parallels reinforcement learning strategies [52, 143].

Machine learning (ML) is increasingly integrated into the DBTL cycle, supporting tasks such as predicting design outcomes [39], assessing gene importance [50], and the automated recommendation of new strains [51, 144]. Among these, automated design recommendation is particularly promising, as it can close the loop from Learn-to-Design and has demonstrated successful applications [37, 38]. However, there remains a limitation in how these machine learning recommendation strategies are applied. First, while advances in genome engineering and high-throughput screening have allowed for targeting many pathway elements simultaneously, the design configuration space often remains limited to a few genes. With configuration spaces on the order of thousands of designs, high-throughput screening and sequencing can achieve relatively dense sampling. This dense sampling of the combinatorial design space may result in increasing production for the recommended strains only marginally, as the strains that were built and used in the training data are already close to optimum of the design space by random chance [81]. Consequently, when the design space is small, the need to balance the exploitation of model predictions with the exploration of new regions — a principle central to Bayesian optimization and reinforcement learning — becomes less important as the design space is already sufficiently explored.

In this study, we examine the impact of substantially expanding the pathway configuration space on design recommendations. Our target is p-Coumaric acid (pCA), an aromatic amino acid derivative synthesized from phenylalanine and tyrosine, serving

as a precursor to numerous bioactive compounds with commercial relevance [145]. We constructed a large library in Saccharomyces cerevisiae comprising 18 genes and 20 promoters, yielding approximately 170 million possible configurations. By design, sampling this space is extremely sparse (5×10^{-6}%), even with high-throughput screening, creating conditions where exploration–exploitation trade-offs become critical [30, 32, 38]. To address this, we implemented a ML-assisted recommendation strategy based on the gradient bandit algorithm [143], using XGBoost as the predictive model [146]. We experimentally assessed how varying the exploration–exploitation balance influences strain performance and investigated whether the model can predict outcomes beyond its training distribution, a common scenario in iterative optimization when a different parent strain is used. Our findings indicate that an initial large library transformation enables the training of a robust predictive model for subsequent recommendations. Balancing exploration and exploitation improves both gene diversity and pCA production compared to greedy approaches based solely on feature importance [50]. Furthermore, combining a high-performing parent strain with a balanced recommendation strategy that accounts for model uncertainty yields the best results. The top strain achieved a 2.37-fold improvement over the previous parent strain [50], with a titer of 1.23 g/L and a yield of 0.07 g/g on glucose.

3.2. Results

3.2.1. Library design using the *yeast8* genome-scale model

Supervised machine learning has been used to predict the TRY values of strains [37, 50, 147], yet it typically focuses on a limited number of genes and promoters. This restriction weakens the full predictive capabilities that machine learning offers, as strain performance may only slightly surpass the training data. We suggest expanding the combinatorial design space by selecting a wider range of gene targets and promoter values. To select gene targets from the vast array of reactions in yeast metabolism, we utilized parsimonious flux balance analysis (pFBA) on the *yeast8* consensus genome-scale model [148, 149]. pFBA seeks to find the minimal total flux through the metabolic networks that maximizes the objective.

The selection of genes involved comparing reaction flux ratios between two pFBA objectives, pCA and biomass, identifying reactions that divert flux from biomass to pCA ($\frac{v_{Biomass}}{v_{pCA}} > 1$) (Fig. 3.2A). Excluding sink, exchange, and non-targetable reactions, 33 targets were identified. These targets are speculated to shift carbon flux from biomass to the product, suggesting that increasing their protein expression could elevate pCA levels. The targets were validated concerning their impact on pCA, the aromatic amino-acid biosynthesis pathway, and reaction thermodynamics (see Table Supporting Information1). Consequently, 18 genes were chosen for the DNA library, including two from glycolysis (FBA, ENO2), seven from the pentose phosphate pathway (ZWF1, SOL3/4, GND1/2, RKI, TKL), eight from the aromatic amino-acid biosynthesis pathway (ARO1, ARO4, ARO2, PHA2, ARO8, PAL1, CPR1, C4H), and one involved in oxoglutarate and citrate transfer between the mitochondrion and cytosol (YHM2) (Fig. 3.2B).

Variety in input feature values is crucial for machine learning model predictions. We therefore used twenty promoters, previously quantified by GFP fluorescence, with intensities spanning several orders of magnitude [50] (Fig. 3.2C). This resulted in a library design comprising 28 promoter-gene cassettes, amounting to approximately 170 million design configurations.

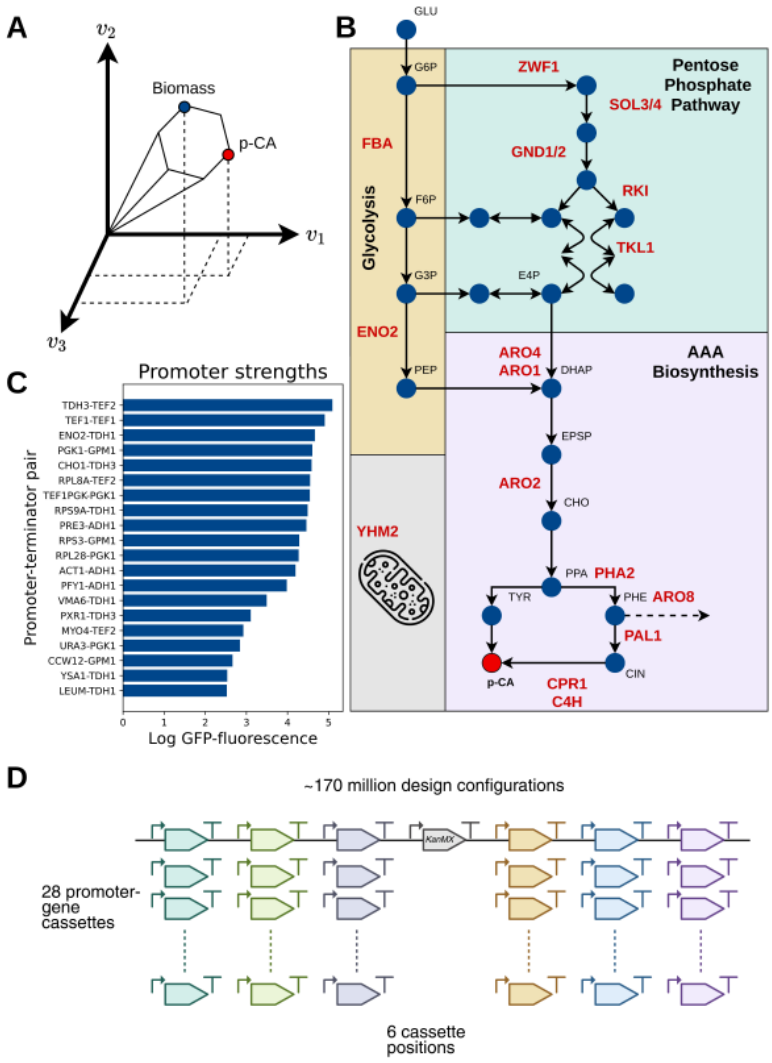


Figure 3.2: **Feature selection using the constraint-based model yeast8**. A) Using the steady-state flux ratio between pCA and biomass as the objective for parsimonious flux balance analysis to identify potential targets ($\frac{v_{Biomass}}{v_{pCA}} > 1$) (see Methods). B) A simplified diagram illustrating the pathway from glucose to pCA, with gene targets identified by pFBA highlighted in red. C) A collection of already measured promoters for the library [50]. D) The library contains 28 promoter-gene-terminator cassettes per position and includes 6 engineered positions, with a grey-marked KanMX selection marker. This setup offers a combinatorial action space of roughly 170 million designs, though 18 out of 168 cassette constructions were unsuccessful.

3.2.2. Supervised modeling of the pCA pathway to predict strain designs

Despite the increasing capabilities of high-throughput screening and sequencing, exhaustively sampling all combinatorial strain designs is both impractical and unnecessary. Effective strain recommendations for subsequent DBTL cycles depend on a reliable supervised model capable of predicting untested designs [150]. To train this model, we transformed a DNA library with a parent strain that resulted from a previous pCA optimization study [50], randomly picked 3000 colonies, and screened their pCA production using Nuclear Magnetic Resonance (NMR). The pCA production was adjusted and normalized against the parent strain (SHK0066) [50]. There was a wide variance in pCA production among the strains, with 5% producing over double the amount of SHK0066 and 27% producing less (Fig. 3.3A). To verify the performance of the highest producers, we re-screened the top 86 strains with four biological replicates, resulting in more conservative estimates; though they remained high producers (see Fig. Supporting Information3A). These remeasured top producers and data from a previous DBTL cycle were used in the training data [50].

After a library transformation, the precise genetic makeup of the strain remains unknown after NMR screening. To address this, we selected a representative sample across the production spectrum and conducted whole genome sequencing to identify integrated library cassettes (see Methods) (Fig. 3.3A). We further processed the copy count matrix by aggregating cassettes sharing identical genes into numerical values based on promoter strengths, effectively reducing the dataset's dimensionality from 169 library cassettes to 18 genes (Fig. 3.3B). Our findings revealed that, aside from CPR1, all genes were prevalent in the sequenced strains, exceeding 10% abundance (Fig. 3.3C). This observation aligns with CPR1's limited inclusion in the library design, where it appeared in only five out of 150 successful cassettes. Despite the library's intention to introduce six genes per strain, unexpected homologous recombination led to strains with over 20 introduced genes (see Discussion). Although these strains are likely genetically unstable, they could still provide valuable information for training the machine learning model as they result in strains that were not purposefully designed. For example, a strain with many integrated genes of the same copy could have higher expression levels than could be purposefully be designed. We therefore included these strains in the dataset.

For modeling, we employed the ensemble method XGBoost with a density-based weighted loss function, emphasizing underrepresented data points via kernel density estimation [151] (see Methods). Model predictions were validated through cross-validation, showing a high Pearson correlation (PCC = 0.83) between predicted and actual pCA production on the validation sets, indicating robust predictive performance. Despite this, there was some under-prediction of high-yield strains and over-prediction of low-yield strains (Fig. 3.3D). To further verify the model, we identified genes that contribute to its performance and enhance pCA production. Six feature importance methods were used to rank genes (see Methods) for the XGBoost model. A Borda consensus ranking of the first 8 genes is reported in figure 3.3E [152]. The genes PAL1, ARO4, and PHA2 were highlighted as the top genes that have a strong influence on pCA, despite observed variability across feature importance rankings (see Supporting

Information4). These genes were previously found to boost pCA production in *S. cerevisiae* [50, 153, 154], supporting the model's reliability for further application in recommending strains.

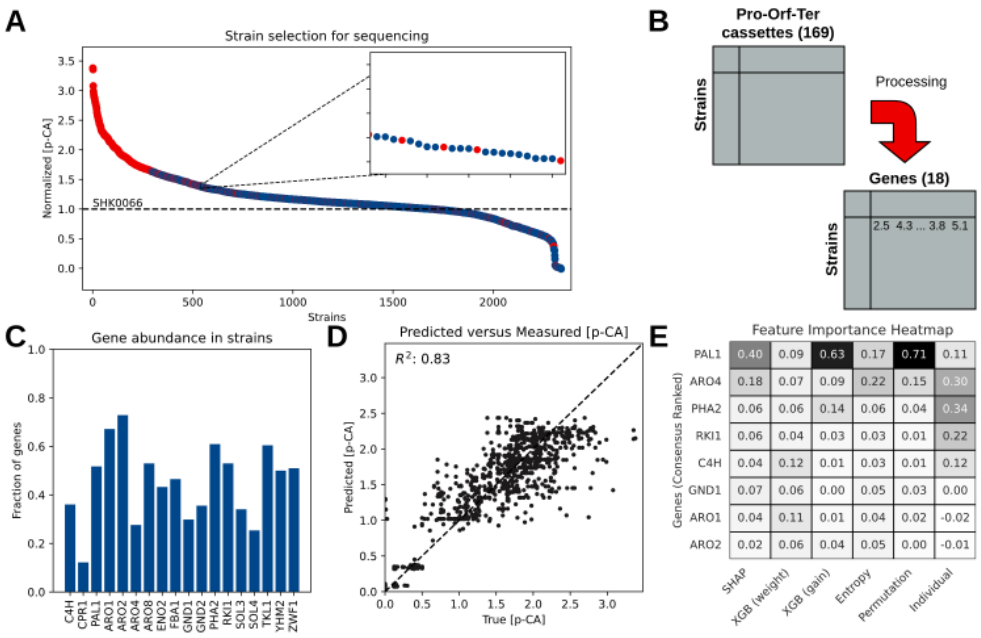


Figure 3.3: **Screening, sequencing, and processing for supervised machine learning.** A) Selection of strains for screening and sequencing. Out of 3000 screened colonies for pCA production, 2350 strains met the quality criteria of the methodological workflow. pCA levels were compared to the parent strain SHK0066, derived from two previous DBTL cycles [50]. From these, six 96-well plates, highlighted in red, were chosen for sequencing, including top producers and a representative sample of lower producers, to minimize machine learning bias using XGBoost[146]. B) After sequencing and preprocessing (refer to Methods), the cassette count matrix was converted into a gene numeric matrix, quantifying promoter strengths via GFP. C) Gene abundance in sequenced strains. This fraction indicates gene prevalence across all strains. All library genes were present in the sequenced strains. D) Predicted vs. measured pCA production in test folds during cross-validation. A high Pearson correlation coefficient signifies accurate processing for pCA prediction. E) Feature importance rankings as determined by six methods (see Methods). The y-axis is the consensus gene ranking across these methods [152]. The values in the heatmap are the relative importance per gene.

3.2.3. A model-based exploration-exploitation strategy for strain recommendation

To improve pCA production using the XGBoost model, it is important to implement a strain recommendation strategy suitable for the complex combinatorial design landscape. This strategy should create a practical action space anticipated by the model from a metabolic engineering viewpoint (Fig. 3.1B) and address the uncertainty of model predictions. Although the model performs well in cross-validation (Fig. 3.3D), its accuracy often declines with new strain designs due to biological and technical variability in pCA concentration, particularly in high-performing strains (see Fig. Supporting Information3A), and possible overfitting. More fundamentally, this decrease is due to the fact that new parent strains, which are outside the model's training data distribution, are often used in later DBTL cycles. Thus, balancing model prediction exploitation with design space exploration for better returns is key [143]. We propose a ML-assisted recommendation strategy that effectively balances exploration and exploitation to successfully optimize pathways.

We create a design space by combining four promoters with eighteen genes, resulting in 72 different cassettes (Fig. 3.4A). For each pathway design, four gene cassettes are selected (see Methods). The predicted pCA production for these *in silico* strain designs is determined using the aforementioned model. A softmax distribution is applied to sample actions, giving each action a sampling probability based on the model's predicted pCA production. The exploration-exploitation parameter β in the formula of figure 3.4A influences the sampling bias: a higher β increases the bias towards model-driven recommendations for high-yield strains. To illustrate the impact of increasing β, 25 strains were sampled from this softmax distribution with varying β values. As the exploitation increases, the number of unique genes decreases (Fig. 3.4B), while both average and maximum strain performance improve (Fig. 3.4C). This indicates that strains sampled in highly exploitative contexts become more similar, which can be problematic if the supervised model has biases, potentially missing more optimal producing strains. There is a clear trade-off between the genetic diversity of sampled strains and their average predicted production (Fig. 3.4D).

3.2.4. Experimental validation reveals a successful automated recommendation strategy and importance of balancing exploration and exploitation

The exploration-exploitation trade-off is a key concept in Bayesian optimization and reinforcement learning [143, 155]. This trade-off holds particular importance in the recommendation strategy previously discussed and is also relevant in metabolic engineering [37, 52, 144]; yet, experimental validation on a consistent optimization target remains sparse. To evaluate its role in effective recommendation strategies, we examined four scenarios with varying degrees of exploitiveness. For SHK0066, this consists of an explorative, balanced, and exploitative scenario, with β values set at 0, 2.5, and 5, respectively (Fig. 3.4D). A fourth scenario involved choosing the best predicted strain, SHK0073, with four integrated genes, as the parent strain using a balanced approach.

In each scenario, twenty-five strains were constructed with four biological replicates (Fig. 3.5A). The dashed lines indicate the parent SHK0066 and the top-performing strain selected via a greedy recommendation method for the six genes included (see Methods). The average production of strains using parent SHK0066 rises with increased exploitative approaches. Several strains in both the balanced and exploitative scenarios outperform the greedy recommendation strategy, highlighting the need for caution when relying solely on predictive models greedily. The balanced scenario exhibits the greatest variance, while the exploitation scenario shows the least variance in pCA production. This variance in the balanced scenario is anticipated: strains with greater genetic diversity display a wider range of pCA production compared to strains with similar genetics. These findings confirm the trade-off inherent in this recommendation strategy.

In the final scenario, using the top producer SHK0073 with four genes included, the average pCA production reached even higher levels, with several strains significantly outperforming the greedy recommendation strategy. The best strain achieved production levels 2.37 times greater than SHK0066, 1.33 times more than the greedy strategy (which included six genes), and 1.18 times more than the best strain from the initial library transformation round (which included ten genes via homologous recombination) (Fig. Supporting Information3B). Note that there is an even higher variance in production than in the balanced scenario of SHK0066.

The higher variation in production performance for this scenario could be explained in two ways: 1) high variation is generally expected in this balanced scenario, and 2) the model was not specifically trained on this host strain. Notably, production variance was higher than that in the SHK0066 balanced scenario. This latter point highlights the importance of incorporating exploration into the approach: models trained on previous strains may not perform well outside of the training data distribution, a common limitation in machine learning. Therefore, addressing this uncertainty is crucial when using automated recommendation strategies.

3.2.5. Deeper validation of predictive-ness and feature importance

While the outlined recommendation strategy seems effective, questions persist about the model's predictive performance and explainability. High predictive performance is crucial because it enables a more exploitative recommendation approach. Explainability matters for understanding biological processes and ensuring the model's learning is biologically grounded [156].

When validating the model's predictive performance for pCA production, we find that the overall trend in the scenarios is accurately predicted (Fig. 3.5B). This is reflected in a strong correlation between predicted and measured pCA concentration across all scenarios (Table 3.1). However, large differences in correlation are observed for the individual scenarios. For the balanced exploration-exploitation scenario with parent strain SHK0073, a high correlation is observed. For the scenarios with parent strain

SHK0066, the correlation diminishes with increasing exploitativeness. This finding can likely be explained by two factors. First, while the trained XGBoost model may capture the overall distribution well, it might fail to predict more fine-grained differences between strains accurately. Second, measurement noise has a more dominant impact on the correlation coefficients for scenarios where the pCA variance is low. These results suggest that the model generally captures behavior well, but higher variation from exploratory designs is crucial to avoid overfitting or overestimation.

Scenario	**Pearson Correlation**	**Spearman Rank Correlation**
All	0.63	0.78
$\beta = 0$ (SHK0066)	0.28	0.43
$\beta = 2.5$ (SHK0066)	0.15	0.32
$\beta = 5$ (SHK0066)	0.02	0.16
$\beta = 2.5$ (SHK0073)	0.55	0.76

Table 3.1: Correlations of the predicted versus true pCA production of figure 3.4B.

We conducted two experiments to determine whether the feature importance results from the model could be experimentally validated. In the first experiment, individual promoter-gene cassettes were integrated into SHK0066 to verify the feature importance rankings shown in figure 3.3E. Genes such as PHA2, ARO4, PAL1, and C4H positively influenced pCA production (t-test, p-value <0.05) (Fig. 3.5C) and were identified by at least two feature importance methods. Although the rankings differed by method, this suggests that the model's learned representation aligns with biological reality. The second experiment involved a greedy recommendation approach, building the best predicted strain by incrementally adding genes (Fig. 3.5D). Despite observing increased production with more genes, the model overestimated this improvement. The top strain from this method produced less pCA than several strains suggested by our recommendation algorithm, indicating that caution is needed with purely greedy strain construction.

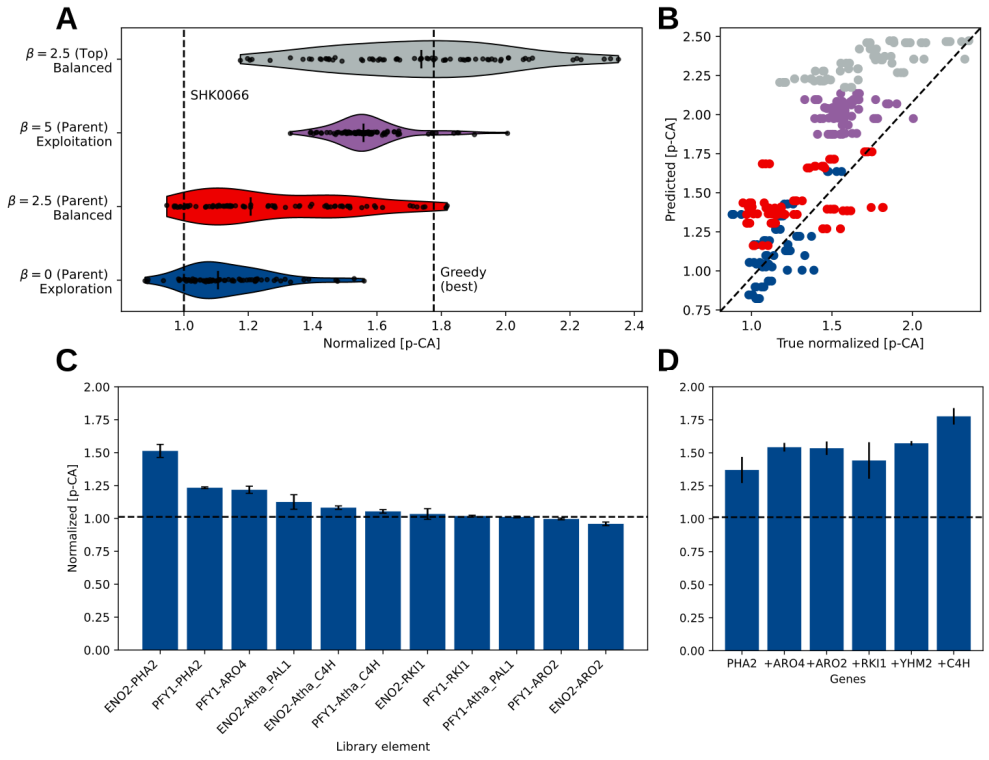


Figure 3.5: **Experimental validation of the impact of balancing exploration and exploitation in extensive combinatorial spaces.** A) Scenarios characterized by exploration, balance, and exploitation ($\beta = [0, 2.5, 5]$) were tested using parent strain SHK0066. The values for pCA production are presented relative to the production by SHK0066 [50]. A balanced scenario was evaluated for the top predicted strain SHK0073. B) A comparison of predicted versus actual [pCA] production across the four scenarios. The XGboost algorithm effectively predicted the general trend, although it underestimated the variability in actual [pCA] production. C) The experimental impact of individual genes included in the library design was assessed. Promoters of varying strength, PFY1 (weak) and ENO2 (strong), were utilized. The genes PHA2, ARO4, C4H, and PAL1 significantly improved pCA production (Wilcoxon rank sum test, $p < 0.05$). D) Construction of strains using a greedy approach based on the model's best-predicted strain. Six strains were developed with an increasing number of cassettes predicted to maximize production.

3.3. Discussion

This study examines the potential of ML-assisted recommendation tools in metabolic engineering to navigate the vast combinatorial space that cannot be explored experimentally. As the combinatorial design space expands, the focus shifts from pure exploitation of model predictions toward a more deliberate balance between exploration and exploitation. Our findings suggest that recommendation strategies incorporating both exploration and exploitation outperform purely greedy approaches that rely exclusively on model prediction (figure 3.5A, dashed line). Importantly, the optimal degree of exploitation appears to depend strongly on the predictive accuracy and certainty of the ML model. In the case of the parent strain SHK0066, where predictive performance was relatively strong, a more exploitative strategy identified the highest pCA-producing strains. However, when model predictions are uncertain or less reliable, heavily exploitative strategies risk converging on local optima, limiting the discovery of superior strain designs elsewhere in the design space. In such scenarios, a balanced exploration–exploitation strategy could become advantageous, as it increases the probability of identifying globally optimal solutions. This is particularly relevant when transitioning to new parent strains, where model uncertainty may be greater. Under these conditions, a balanced approach represents a more robust and risk-mitigating strategy for strain optimization.

The combinatorial design space in this study was constructed from eighteen genes and twenty promoters, arranged across six library positions with twenty-eight cassettes per position. The targeted genes were identified through a combination of pFBA-based feature selection and additional filtering using using database resources and existing literature. This led to the inclusion of several previously characterized genes [157]. While this approach ensured biological relevance, it may bias target selection toward well-studied genes. Future studies could benefit from more data-driven gene selection strategies that are less dependent on prior knowledge, potentially enabling the discovery of novel targets and further improving metabolic pathway optimization [47]. For the promoters, we selected twenty unique promoters that were quantified by GFP fluorescence in previous work [50]. The large set of promoters was chosen for two reasons: (i) to reduce the likelihood of homologous recombination by assigning distinct promoters to each position, and (ii) to introduce varying expression levels. This increases the informational content of the eighteen genes for modeling, despite further inflating the combinatorial space. Although these measures aimed to minimize recombination, such events were still anticipated due to the repetitive nature of the library design. Indeed, some strains unexpectedly contained more than ten genes instead of the intended six, resulting in genetic instability. For example, the two top-performing strains from the first DBTL cycle integrated 26 and 23 genes (including KanMX markers), respectively (Fig. Supporting Information3B). Despite their instability, these strains provided valuable insights and were included in XGBoost model training.

It is important to note that while GFP fluorescence serves as a convenient proxy for expression level, the magnitude of a promoter's effect can differ between targeted enzymes. Similarly, the expression strength may be influenced by the DNA library

position of the integrated promoter. In our modeling framework, we assumed that for a given gene, promoter strength and enzyme expression are related by a monotonically increasing function. This assumption proved sufficient to achieve solid predictive performance, likely due to the inherent flexibility of the supervised machine learning model. Nevertheless, future work could benefit from directly linking promoter strengths to proteomic measurements, which may better capture enzyme-specific expression dynamics and further improve model accuracy [37].

A key aspect of this work is the balance between exploration and exploitation within the combinatorial design space. Although the importance of this effect has been noted in prior work [37, 51, 158], consistent experimental validation of the exploration-exploitation tradeoff has been only limitedly explored in the metabolic engineering literature [38]. While the exploitative scenario ($\beta = 5$) produced high-performing strains, it did so at the cost of genetic diversity among the proposed designs. This reduction in diversity may lead to suboptimal decisions when model performance is lower than that observed for the pCA model used here. Consequently, accounting for model uncertainty is essential when determining the appropriate balance between exploration and exploitation. Although we demonstrate the significance of this balance, we do not prescribe a specific method for achieving it (see Fig. Supporting Information5). One potential approach could involve maximizing the product of genetic diversity and average production, as illustrated in figure 3.4D, similar to the strategy employed by the MODIFY algorithm for directed evolution [75].

For the second round, we selected specific strains recommended by the algorithm. This resulted in strains with a pCA titer of 1.23 g/L and a yield of 0.07g/g glucose, which was a 2.37 times improvement over the original parent strain SHK0066 [50]. While higher pCA production has been reported [159], the strain designs reported in this study may serve as a basis for further improvement to enable industrial use [21]. This would include the need for incorporating bioprocess conditions in the model formulation. We expect that culture and process conditions can be incorporated using mechanistic modeling approaches, such that they generalize better during bioprocess scale-up than pure machine learning approaches [160].

As the methodology for constructing the strains in the second round differs from library transformation, the number of strains that can be built is substantially smaller than what is achievable through a library-based approach using comparable experimental resources. However, this recommendation strategy could also be adapted for library design using the approach illustrated in Figure 3.4A. An additional step would involve generating a probability distribution for each library element at each position. Experimentally, this can be implemented by adjusting the concentration of library cassettes, thereby modulating the probability of integration during library transformation. We experimentally verified that cassette concentration can influence integration likelihood (see Fig. Supporting Information2). This approach would enable larger-scale follow-up rounds.

To evaluate feature importance methods, we experimentally determined the individual

contribution of specific genes to p-CA production (Fig. 3.5C). Three genes — PHA2, ARO4, and PAL1 —emerged as key contributors from the consensus of all feature importance approaches [152]. PHA2 catalyzes the conversion of prephenate to phenylpyruvate and has previously been reported to significantly influence pCA production [159, 161]. AR04, which catalyzes the first step in the shikimate pathway, is known to be critical for the parent strain SHK0066 used in this study and in other pCA yeast cell factories [50, 162]. PAL1, originating from *Arabidopsis thaliana*, offers an alternative to the TAL-route [159, 162] and was also included in the parent strain [50]. For the other considered genes, more variability between feature importance methods for gene importance rankings was observed (Fig. Supporting Information4). For these additional candidates, some of which were experimentally verified, their effects on pCA production were also shown to be less consistent (Fig. 3.3E). Some genes showed a modest positive effect (e.g., YHM2), whereas others exhibited no measurable improvement or even a slight negative impact on production (e.g., ARO2 and RKI1). These results indicate that feature importance rankings may not always correspond to improvements in production.

3.4. Methods

3.4.1. Organisms and media

S. cerevisiae strains originate from CEN.PK113-7D. P-Coumaric (pCA) production strain BMP+PAL-C4H-CPR (SHK0066) from study [50] was used as starter strain. Strains were grown at 30°C in Yeast Extract Phytone Dextrose media (YEPhD, 2% Difco™ phytone peptone (Becton-Dickinson (BD), Franklin Lakes, NJ, USA), 1% Bacto™ Yeast extract (BD) and 2% D-glucose (Sigma Aldrich, St Louis, MO, USA)). When required, antibiotics were added to the media at appropriate concentrations: 200 µg/ml nourseothricin (Jena Bioscience, Germany), 200 µg/ml geneticin (G418, Sigma-Aldrich). E. coli DH10B (New England BioLabs, Ipswich, MA, USA) was used as cloning strain and grown at 37°C in 2*Peptone Yeast Extract media (2*PY, 1.6% tryptone peptone (BD), 1% Bacto™ yeast extract (BD) and 0.5% NaCl (Sigma Aldrich)). When required, antibiotics were added to the media at appropriate concentrations: 100 µg/ml ampicillin (Sigma-Aldrich), 50 µg/ml neomycin (Sigma-Aldrich). Solid medium was prepared by addition of Difco™ granulated agar (BD) to the medium to a final concentration of 1.5% (w/v).

3.4.2. Cassette construction

Gene Targets

To determine which gene targets should be included in the library, parsimonious flux balance analysis (pFBA) is performed [149] on the genome-scale model Yeast8 [148]. We use the pFBA solution for a pCA objective and the biomass objective to calculate a flux ratio. This flux ratio indicates which fluxes are rerouted when diverting carbon flux from biomass towards pCA and was used to select genes for further validation ($\frac{v_{Biomass}}{v_{pCA}} > 1$). After excluding sink, exchange, and non-targetable reactions, 33 potentially interesting gene targets remain.

We further aimed to substantiate gene targets by performing literature enrichment

analysis using the STRING database [163], thermodynamic information from the equilibrator database [164], and inhibitor information from the BRENDA database [165]. Finally, we assess the importance of each gene through a literature review (see table SI1). This resulted in a set of nineteen validated genes that are used in the library. Open reading frames (ORF) were codon pair optimized for *S. Cerevisiae* [166]. Cassettes were designed as described in [167] and constructed as described in [50]

Promoters

A set of twenty previously established promoters, with varying strengths quantified through GFP fluorescence, was used in the library design [50]. The expression strength of these promoters spans several orders of magnitude and can be used to regulate gene expression. We assume that the expression pattern might differ between genes with the same promoter, but that the relative expression per gene is similar.

Library design

The number of positions in the library was limited to six factors to ensure sufficient transformation efficiency. To avoid excluding certain gene combinations during transformation, we decided that for each position in the construct, all genes should be present with two distinct promoters [50]. For genes that were previously considered in the reference strain (C4H, CPR1, PAL1, ARO4), we chose one promoter per library position. This library consists of 168 cassettes, of which 150 were successfully built. While incorporating all genes ensures that we do not exclude certain combinations, it might lead to increased homologous recombination events during transformation. The library design can be found in the supporting information.

3.4.3. S. cerevisiae strain construction

Transformation

Strains were constructed as described in [50]. In short, host strain SHK0066, pre-expressing Cas9 was used for CRISPR mediated integration of the cassette library into the target locus INT28, see supplementary info for gRNA design and genome locus position. Cassette libraries were prepared with equimolar amounts of cassettes (100ng/kb). Per cassette library previously considered genes (C4H, CPR1, PAL1, ARO4) were added with higher dosage as follows; library 1: 300ng/kb, library 2: 200ng/kb, library 3: 100ng/kb, library 4: 50ng/kb, the library design can be found in the supporting information. Linear gRNA (210ng/kb) and linear backbone with homologous sites to the gRNA (35ng/kb) were transformed together with the cassette libraries into the host strain following LiAC/ssDNA/PEG method [168].

For design-based transformation, the cassettes were prepared per pathway design in equimolar amounts and transformed with the linear gRNA and linear backbone following the same LiAC/ssDNA/PEG method. The cassette design with on five prime and three prime side 50 base- pair connector sequences homologous to INT28 facilitate in vivo recombination of cassettes into the genome.

The transformed cells were plated on a Qtray (Nunc) containing yephD agar and appropriate antibiotics. 48-dividers (Genetix) were used in the Qtray when a pathway design was done. Qtrays were incubated for three days at 30°C and picked with Qpix 450 (Molecular Devices) into 96 MTP (Nunc) containing YephD agar with appropriate antibiotics.

3.4.4. pCA screening in *S. cerevisiae*

Strain fermentation

pCA production experiments were conducted as follows; colonies were grown in a microtiter plate (MTP) containing YephD agar medium and incubated for 48 hours at 30°C, 750 rpm, and 80% humidity. Cultures were inoculated in a half deep well plate (HDWP, Thermo Fisher Scientific, AB-1127) containing 350μl YephD and appropriate antibiotics as a preculture. Strains were reinoculated in main culture into HDWP containing 350ul Verduyn Luttik with 2% glucose [169] and incubated for 48hrs at 30°C, 750 rpm, 80% humidity. Every plate contains a medium control and 16 times the parent strain SHK0066 for later normalization.

pCA measurement with NMR

For Flow-NMR measurements, 250μl of the end of the fermentation broth was sampled into a 96-deepwell plate and mixed with 500μl of acetonitrile using the EPmotion (Eppendorf) to initiate cell lysis. The samples were centrifuged at 4000 rpm for 10 minutes, and 500 μl of supernatant was sampled into a new deep well plate. 70% of the liquid was evaporated, and 100μl of 1 g/L internal standard 1,1-difluoro-1-trimethylsilanyl methylphosphoric acid (FSP, Bridge Organics) was added. The samples were lyophilized to remove non-deuterated solvents. To the lyophilized samples, 600 μl of D2O (Cambridge Isotope Laboratories (DLM-4)) was added and homogenized. The samples were analyzed on a CTC PAL3 Dual-Head Robot RTC/RSI 160 cm robotic autosampler (CTC Analytics AG, Zwingen, Switzerland) fluidically coupled to a Bruker spectrometer Avance III HD 500 MHz UltraShield [26]. 1H spectra were recorded with standard pulse program (zgcppr) with following parameters: 16 scans, 2 dummy scans, 33k data points, 16.4 ppm spectral width, 1.2 s relaxation delay (d1), 8 μs 90° pulse, 2s acquisition time, 15 Hz water suppression, and fixed receiver gain (rg) of 64 Spectra were processed and analyzed using Topspin 4.1.4 (Bruker). Spectral phasing was applied and spectra were aligned to 3-(trimethylsilyl)-1-propanesulfonic acid-d6 sodium salt (DSS-d6, Sigma-Aldrich) at 0 ppm. Auto baseline correction was applied on the full spectrum width. Additional third-order polynomial baseline correction for selected regions was applied if needed. The amount of pCA (doublet, 6.38 ppm, n=2H) was calculated relative to the signal of FSP [50].

3.4.5. Strain quality control using whole genome sequencing

Choice of screened strains for sequencing

Based on the screening of 3000 strains, we chose a representative set of strains for sequencing. We chose three 96-wells plates for the top producers, two 96-wells plates for the strains that showed improved pCA production but that are not in the top, and one 96-wells plate for strains that performed worse than the parent strain.

Whole genome sequencing

S. cerevisiae cells (OD600 of 5-10) were pelleted and lysed as follows; 140mg Zirconia beads (0.5mm diameter, Biospec products) and 400μl ZymoBIOMICS Lysis buffer (Zymo Research) were added to the cell pellet. Cells were lysed for 5 minutes at 1500 rpm with the Geno/Grinder 2010 (Cole-Parmer). gDNA was further isolated following ZymoBIOMICS 96 DNA kit (Zymo research) according to manufacturers protocol. The isolated genomic DNA was quantified using the Tapestation 4200 (Agilent) with a genomic DNA screentape (Agilent) and Nanodrop (Thermofisher Scientific). Genomic DNA was sequenced per strain or per library pool of transformants using the native barcoding kit (SQK-NBD114.96) from Oxford Nanopore Technologies on a Promethion device (FLO-PRO114M) according to manufacturer's protocol. For pool sequencing super-accuracy basecalling was used and for strain sequencing high accuracy basecalling.

Data processing of individual clones

DNA sequencing reads obtained from picked individual colonies after transformation, were cleaned using Porechop [170]. Cleaned reads were then assembled into genomes using Canu [171], with the genome size parameter set to 12 Mbp, and were followed by one round of polishing by Racon [172] and Medaka [173]. To identify where the inserted BioBricks were located, annotations of the cassette sequences (such as promoter, open-reading-frame, terminator annotations) were transferred to the genome assemblies by blast-based sequence alignment, leading to genome assembly annotations in GFF format. Cassette annotations were only transferred to the genome assemblies if at least 95% of the cassette annotation was included in an alignment hit to the genome, with at least 95% identity.

3.4.6. Machine learning-assisted recommendation strategy

Ensemble learning tree methods have been shown to perform well on combinatorial pathway optimization data [51, 81]. We therefore trained an XGBoost [146], a regularized gradient boosting algorithm, with a modified loss function that is more specific to our purpose.

Density-based weighting of loss function

Machine learning methods typically assume that data is balanced; that is, the target variable is uniformly distributed. However, in production screening data, this is often not the case. High performing strains are typically underrepresented, while strains with minor increases in performance are overrepresented. To mitigate the effect of this imbalanced

distribution on the mean squared error (MSE) [174], we used DenseWeight [151]. DenseWeight weighs the influence of data instances on the loss function according to the target variable frequencies estimated through kernel density estimation. DenseWeight requires setting two hyperparameters that determine how much the density estimation should influence the weighting, which were set to $\alpha = 0.5$ and $\epsilon = 0.01$.

XGboost requires the calculation of the gradient and hessian of the loss function for efficient training, and due to the modified loss function, it needs to be adjusted. The modified loss function ($L_{DensLoss}$) is the mean squared error weighted by a function ($f_w(y_i)$) that is dependent on the target variable density and its hyperparameters. The loss function, Jacobian, and Hessian with respect to a data instance y_i are then given by (eq. 3.1-3.3).

$$L_{\text{DenseLoss}}(\alpha) = \frac{1}{N}\sum_{i=1}^{N} f_w(y_i) \cdot \text{MSE}(y_{\text{pred}}, y_i) \tag{3.1}$$

$$\frac{\partial L_{\text{DenseLoss}}}{\partial y_i} = 2(y_{\text{pred}} - y_i) \cdot f_w(y_i) \tag{3.2}$$

$$\frac{\partial^2 L_{\text{DenseLoss}}}{\partial y_i^2} = 2 \cdot f_w(y_i) \tag{3.3}$$

XGboost training

Bayesian hyperparameter optimization was performed on the training data using scikit-optimize [175]. This resulted in the following hyperparameters (*reg_lambda=1*, *max_depth=2*, *tree_method="auto"*). The customized loss function was used as described above. Two hundred rounds of boosting were performed with an early stopping criterion of forty. This means that if the validation mean squared error does not improve after 40 rounds, training is stopped. Model validation was performed on a ten percent random split using sklearn and was evaluated using the Pearson correlation coefficient between the true and predicted values. We further validated the model using an external validation set of designs (see below).

Recommending new strains using gradient bandit principles

To recommend new strains for the second round, we start by defining the action space (Figure 3.1B). In the case of $N = 20$ genes with $P = 4$ cassette positions, we choose $X = 4$ different levels of promoter strength, which result in $\binom{N \times X}{P} = 1028790$ actions. Actions that have multiple integrations of the same gene are filtered out. Then, XGBoost is used to predict the potential outcome of the set of actions given the reference strain. We chose two reference strains: the parent of the first library transformation round (SHK0066) and the best predicted action strain for four pathway positions.

As a recommendation strategy based on predicted outcomes, we used a gradient bandit approach [143] using the softmax distribution with an exploration/exploitation (β) parameter (eq. 3.4).

$$Pr(A_i) = \frac{e^{\beta y_i}}{\sum e^{\beta y_i}} \tag{3.4}$$

The exploration/exploitation parameter β is an important parameter for automated recommendation and represents a trade-off between the diversity of the set of chosen designs in terms of gene content and the expected performance of the strains in terms of production [75]. Here, we have tested three exploration/exploitation values ($\beta = 0$, $\beta = 2.5$, $\beta = 5$) to explore this trade-off in more detail.

Feature importance methods

Six feature importance methods were used to retrieve gene importance rankings from the trained model. Two methods are intrinsic to the XGBoost model: the *weight* method ranks features based on the number of times a feature is used to split data across the trees, and the *gain* method bases this on the average information gain across all splits in which the feature is used. The three other methods are model independent. One popular method for explaining machine learning models is the SHAP values [176]. We also tested feature importance ranking based on the area under the curve of gene-promoter threshold plots, as used in [81]. Furthermore, a permutation importance method from sklearn was used [177]. Finally, the individual contribution of each gene was calculated by predicting the outcome of a single promoter-gene perturbation on the trained model.

3.4.7. Code and data availability

All generated data is accessible for future research. This comprises DNA sequences of the strains from the initial DBTL cycle, NMR screening results of the 3000 strains, and processed datasets for machine learning. These resources are available in the 4tu repository (DOI: `https://doi.org/10.4121/fa0782ab-760d-4fa7-babf-09bdaab0f509`. The related code is accessible on GitHub `https://github.com/AbeelLab/ml-assisted-p-coumaric-acid-optimization`.

3.5. Supporting Information

The following data, results, and supplementary material can be found in the original publication and the 4tu repository (see QR-code).

- Supporting Information (.pdf) file consisting of additional information on parent strain lineage, literature validation of chosen gene targets, whole genome sequencing library pool analysis, and rescreening of the top producers from the first DBTL cycle, as well as a rebuild of the top 2 producers.
- DNA sequencing (.fasta) files can be found in the 4tu-repository. Code and processed files required for reproducing this work can be found in the GitHub repository (see Code and data availability).
- List of promoters, ORFs, and terminator sequences used in the study (.xlsx): integration sites used for strain construction; the list of strains used in this study

and their genotypes; the list of plasmids used in this study; the list of primers used in this study; and sgRNA design
(*SupplData.xlsx*).

- Library design files (.xlsx) for DBTL round 1 and the follow-up machine learning-assisted recommendations.
Round1_LibTrans_design.xlsx,Round2_ML_assisted_design_AI4bioDoE.xlsx.

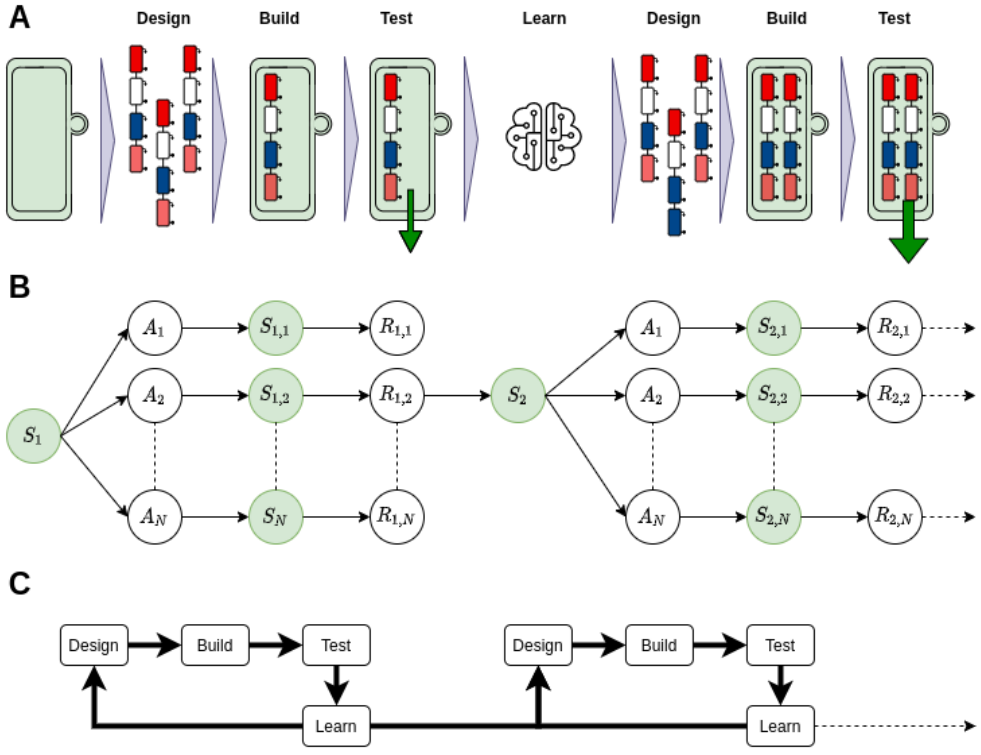


Figure 3.1: **Iterative metabolic engineering from an experimental, computational, and engineering viewpoint.** A) The initial strain undergoes transformation with various pathway designs, integrating these pathways into its genome, resulting in the production of a desired compound (green arrow). By utilizing multiple built strains, a predictive regressor is trained to forecast production outcomes of new designs. This process repeats until strains achieve industrial production levels or reach a theoretical maximum. B) Iterative metabolic engineering can be framed as a Markov Decision Process (MDP). A selection of actions A (designs) is used to transition to a new state S (new strains), yielding a reward R (production value). Supervised machine learning can be employed to determine the optimal action for a given state S_1, or to proceed further within the MDP model. The role of ML-assisted recommendations is to efficiently navigate this MDP using a model-based policy that chooses which action to take in order to improve the reward. C) The DBTL cycle in metabolic engineering describes the steps (arrows) involved in advancing through the MDP. In the Learn phase, the experimental results can be used to inform genetic designs based on the current host strain (left arrow from learn to design). Alternatively, new genetic designs can be used on a different host strain that resulted from the previous DBTL cycle (right arrow from learn to design).

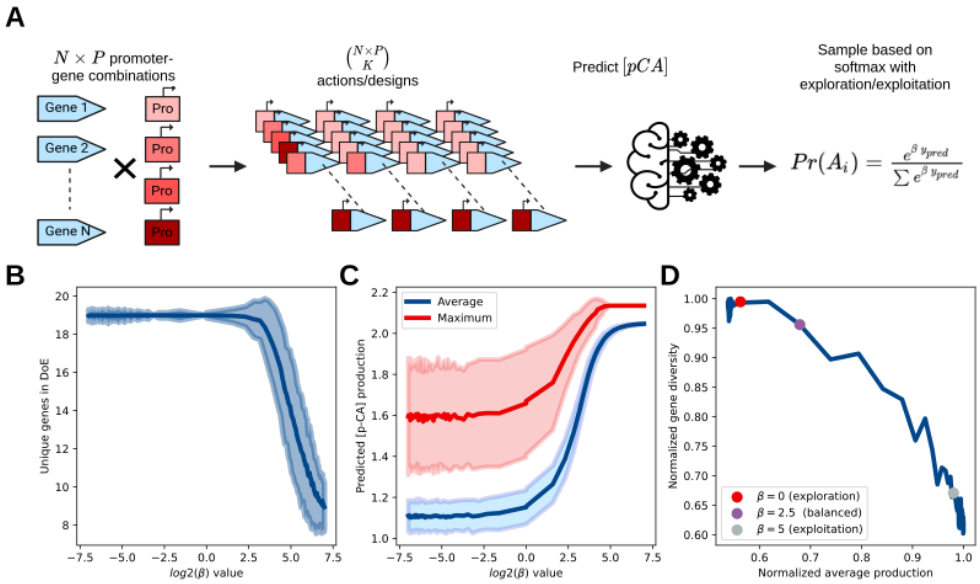


Figure 3.4: **Exploring machine learning-enhanced automated recommendation with a balance between exploration and exploitation.** A) Overview of the strategy: four promoters (P) and eighteen genes (N) form a set based on pathway positions (K) to create an action space. This model forecasts pCA outcomes, with designs sampled from a softmax distribution featuring an exploration-exploitation parameter (β). B) Impact of exploration/exploitation on design diversity. For each β value, 25 samples were drawn, repeated 100 times (shaded region). Increasing β reduces gene set diversity. C) Average and maximum predicted production relative to β. More exploitation leads to higher average predicted production. D) Balance between gene diversity and average production. Three β settings were selected for experimental validation of the recommendation algorithm and to study the effect of the exploration-exploitation trade-off in practice.

4

Jaxkineticmodel: neural ordinary differential equations inspired parameterization of kinetic models

Paul VAN LENT, Olga BUNKOVA, Bálint MAGYAR, Léon PLANKEN, Joep SCHMITZ, Thomas ABEEL

Motivation: *Metabolic kinetic models are widely used to model biological systems. Despite their widespread use, it remains challenging to parameterize these Ordinary Differential Equations (ODE) for large scale kinetic models. Recent work on neural ODEs has shown the potential for modeling time-series data using neural networks, and many methodological developments in this field can similarly be applied to kinetic models.*
Results: *We have implemented a simulation and training framework for Systems Biology Markup Language (SBML) models using JAX/Diffrax, which we named jaxkineticmodel. JAX allows for automatic differentiation and just-in-time compilation capabilities to speed up the parameterization of kinetic models, while also allowing for hybridizing kinetic models with neural networks. We show the robust capabilities of training kinetic models using this framework on a large collection of SBML models with different degrees of prior information on parameter initialization. We furthermore showcase the training framework implementation on a complex model of glycolysis. Finally, we show an example of hybridizing kinetic model with a neural network if a reaction mechanism is unknown. These results show that our framework can be used to fit large metabolic kinetic models efficiently and provides a strong platform for modeling biological systems.*
Implementation: *Implementation of jaxkineticmodel is available as a Python package at* `https://github.com/AbeelLab/jaxkineticmodel`.

This chapter has been published in PLOS Computational Biology **21.7**, e1012733 (2025) [178].

4.1. Introduction

Kinetic modeling is a useful tool for describing biological systems in a quantitative manner, with many applications in the biotechnological and medical domain [45, 179]. In the biotechnological domain, the application of kinetic models includes assessing metabolic control of pathways [180], simulation of metabolic engineering scenarios [181, 182], and optimizing feeding strategies on the bioprocess level [183]. Effective deployment of kinetic models for these purposes requires a representative description of the biological process in mathematical equations, as well as fitting the model parameters to available data. The encountered data in this domain is typically limited and either steady-state or dynamic in nature.

Metabolic kinetic models are described by ODEs that describe the change over time of metabolites (m) by a right-hand-side formula that consists of the mass balances imposed by the stoichiometric matrix (S) and a vector of reaction flux functions ($\vec{v}$) (eq. 4.1). The process of finding a model that reproduces observed data therefore consists of establishing the stoichiometric matrix [54, 184], determining kinetic mechanisms [185], and parametrization of the flux functions.

$$\frac{dm(t)}{dt} = S \cdot v(t, m(t), \theta) \tag{4.1}$$

Fitting parameters to large-scale kinetic models can be challenging [186]. While biological systems operate on many different timescales, biologically relevant parameters can vary orders of magnitude [187]. Additionally, the ODEs observables might be insensitive to many of these parameters, sometimes referred to as sloppiness [188]. Furthermore, many kinetic models in systems biology have unidentifiable parameters, which complicates the fitting process even more [189]. Finally, many biological systems are known to be stiff [190], which leads to problems when numerically solving the ODEs [191]. These challenges complicate the use of standard parameter estimation methods.

Several parameter estimation methods have been proposed and implemented in publicly available software packages to tackle this difficult task (see Supporting information for a qualitative assessment). Some methods focus specifically on fitting steady-state fluxomics and metabolomics data; through gradient-based optimization [192, 193], sampling-based approaches [110, 132], generative modeling [194, 195], or Bayesian approaches [196]. Other toolboxes provide more general purpose parameter inference methods for metabolic modeling, such as pyPESTO [197]. pyPESTO provides an interface to many different optimization methods: local versus global, gradient-free versus gradient-based optimizers. A particular efficient method that is integrated in pyPESTO are the multi-start gradient-based methods, which uses AMICI for efficient sensitivity computation [198]. These multi-start, gradient-based methods have shown to perform well on large-scale fitting problems [199, 200].

Recently, Neural ODEs were introduced [201] and applied to modeling time-series concentration data [202], as well as in other applications in the biological domain [203]. The idea behind a neural ODE is to replace the right-hand-side of equation 4.1 by a neural network and then use a numerical solver to predict a time-series. The neural network is then trained using back-propagation with the adjoint state method; an efficient method for estimating gradients. These methods are efficient compared to forward gradient computation, as they do not scale with the number of model parameters: a feature that is important in neural networks or large kinetic models (see Supporting information and Supporting information in Supporting information) [194, 201]. Even though neural ODEs lack the necessary mechanistic structure required in biotechnological/medical applications, many of the techniques for training, such as memory efficient adjoint gradient computation for time-series data, can similarly be applied to metabolic kinetic models. Furthermore, hybrid approaches, where a mechanistic model is augmented with a neural network to model dynamics that are not mechanistically understood, are increasingly being studied and used [76, 204, 205].

In this work, we have implemented a *JAX*-based training and model building framework [206] for systems biology models, which we named *jaxkineticmodel* (Fig 4.1). *JAX* has just-in-time compilation, automatic differentiation, and parallelization capabilities. This paves the way to large-scale kinetic model training (see Supporting information4 for a comparison against other popular parameterization methods). Even though similar features have been implemented in packages for the *Julia* programming language [207], having Python packages for systems biology purposes is valuable due to its widespread use and support [208]. Furthermore, while a previous python package *SBMLtoODEjax* was developed that provides an interface between SBML and *JAX* [208], several unique features are implemented. First, *jaxkineticmodel* supports a larger set of SBML models (see Supporting information in Supporting information), is compatible with numerical solvers from *Diffrax* [209], optimizers from *Optax* [210] and has the capability to integrate mechanistic and neural network components [76]. The default training framework is tailored to systems biology, with support for the Systems Biology Markup Language, as well as manual model building using predefined kinetic mechanisms.[211]. As a default setting, training is performed by gradient descent in log parameter space [187] with a stiff numerical solver [212] and a custom loss function to deal with metabolic scale differences. Gradients are calculated using an efficient adjoint state method from the *Diffrax* package [209]. To further stabilize training, we perform gradient clipping, a method that is used in stabilizing the training process of recurrent neural networks and neural ODEs [213]. We apply this implementation on a large collection of SBML models to answer questions on robustness of the training procedure in terms of convergence properties. Furthermore, we show the training of a large-scale kinetic model of glycolysis (141 parameters) to model feast/famine feeding strategy datasets [214]. Finally, we show how *jaxkineticmodel* can be used for hybrid models that have mechanistic and neural network components, an increasingly important application of universal differential equations in systems biology [76, 204].

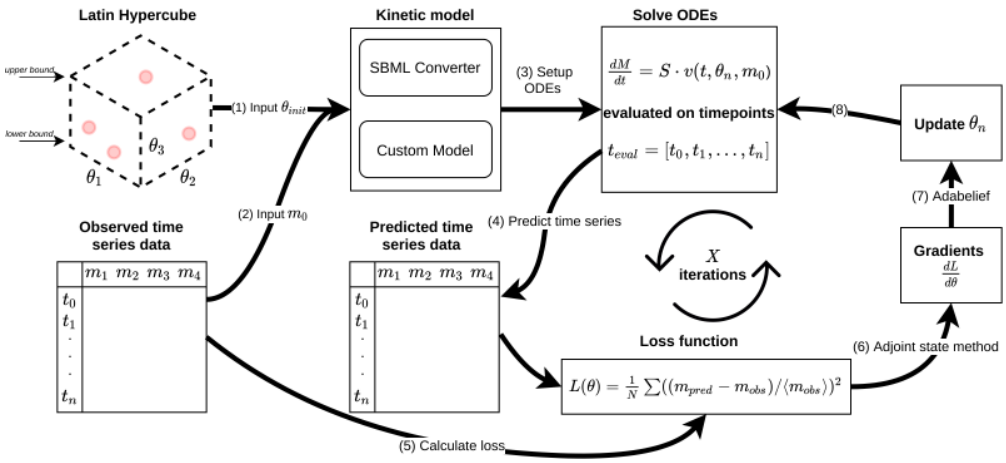


Figure 4.1: **Overview of the implemented simulation tool and training framework in *Diffrax*.** Latin Hypercube sampling is used to initialize parameters given a lower- and upper bound value (1). The initial conditions are retrieved from the observed dataset (2) and the ODEs are set up based on an imported SBML model, or a self-implemented model (3). After predicting a time-series dataset given the initial guess (4), the mean-centered loss is calculated (5). The gradients that are calculated through the adjoint state method (6) are then used to update parameters using AdaBelief (7)[215]. This process is repeated for X steps or until convergence (8). Implementation details on specific aspects are discussed in *Design and Implementation*.

4.2. Design and Implementation

4.2.1. Implementation

An overview of the training framework is shown in Fig 4.1. All experiments, code, and results, as well as documentation are publicly available. Internally, *jaxkineticmodel* uses *Sympy* [216], *JAX* [206], and *LibSBML* [211] for compatibility with many numerical solvers in *Diffrax* [209]. Documentation is available on Github.

Kinetic model setup

We provide two options to set up the system of ODEs required prior to training. A model can be constructed by using kinetic mechanisms that have been provided in *jaxkineticmodel* (see Supporting information in Supporting information), or by using the SBML-to-JAX converter. Kinetic models can also be exported to the SBML format after parameterization [211].

Scaling of species in the loss function

Biological systems exhibit large-scale differences in metabolite concentrations, therefore resulting in mean squared error loss J being dominated by large absolute error, even though relative error might be small [191]. We therefore implemented as a default a mean-centered loss function (eq. 4.2), but other loss functions can be passed as well.

$$J(m_{pred}, m_{observed}) = \frac{1}{N}\sum(\frac{m_{pred} - m_{observed}}{\langle m_{observed} \rangle})^2 \tag{4.2}$$

Specifics of the gradient descent algorithm

The AdaBelief optimizer was used during training [215]. Training was further stabilized by clipping the gradient global norm to prevent the exploding gradient problem often encountered in neural ODEs and recurrent neural networks[213]. This requires setting a maximum global norm hyperparameter ($\hat{g}$), which was set to $\hat{g} = 4$. As previous reports show that systems biology models are better fitted in a log-transformed parameter space [187, 217], we applied the log-transformation of parameters before passing it through AdaBelief. We note however that users can change the optimizer to any optimizer provided in *Optax* [210].

Specifics of the solver

Simulations were performed using the Kvaerno5 stiff ODE solver, which was already implemented in *Diffrax* [212]. The relative and absolute error tolerances were 10^{-8} and 10^{-11}, respectively. The initial time step dt_0 was set to 10^{-10}. *Jaxkineticmodel* is compatible with many numerical solvers for ODEs in *Diffrax*. For the adjoint state method, the default implementation provided by *Diffrax* is used, but users can also change this according to their preferences [209].

4.2.2. Data

SBML models

To rigorously test properties of *jaxkineticmodel*, we generate time-series metabolomics datasets from SBML models. The advantages of using synthetic data are that we can: 1) compare learned parameters against true parameters; and 2) test the method against a large collection of biological systems. The twenty-six models were retrieved from a previously reported set of benchmark models [187] and the Biomodels database [218]. These models are both dynamic models and steady-state models. Relevant properties of the models are summarized in Table 4.1. Models were simulated on a time interval $t = [0, t_{end}]$ with ten data points per specimen for the ground truth parameters. We chose ten observations as in many biotechnological domains, time-series data is only sparsely sampled. No noise model was used for the observations.

Feast/famine cycle and steady-state data

Along synthetic data, we applied the parameterization framework to datasets of a feast/famine cycle that was previously reported [214]. Feast/famine experiments are a stimulus-response experiment that consists of a 20 second feeding phase, followed by 380 seconds without feed. Three datasets were available for fitting a glycolysis model.

4.2.3. Evaluation of training framework

Gradient averaging for parameterization using multiple datasets

To showcase the practical usability of Neural ODEs, we aim to fit multiple datasets to the glycolysis model (see Supporting information). These datasets are both steady-state metabolomics for different dilution rates, as well as dynamic data in the form of a glucose pulse. In order to simultaneously fit all datasets, we calculate the gradient $dJ/d\theta$ for each individual dataset and average them before updating using AdaBelief. This is a popular method to train a global model from local datasets in federated learning [244].

4.2.4. Evaluating the training process on simulated data

Three properties of the training process are tested after training: 1) the initialization success rate of parameters; 2) the convergence properties through the relative improvement over the initialization; and 3) the distance of trained parameters to the true optimum. All experiments are performed three times.

Initialization success percentage

Datasets are simulated with the true parameters (θ_{true}) that are specified in each SBML model. Latin hypercube sampling [245] is used to initialize 100 parameters within

Table 4.1: **Overview of the SBML models used in this study.** The collection of models was retrieved from Biomodels [218] and a previously reported collection of SBML benchmark models [187]. The number of parameters, number of species, simulated range, and data points are reported.

Model	Parameters	Species	t_{end}	Data points
Garde et. al 2020 [219]	6	6	6	60
Smallbone et. al (2013) [220]	10	2	10	20
Bruno et. al (2016) [221]	10	6	100	60
Beer et. al (2014) [222]	12	4	8000	40
Patil et. al (2023) [223]	13	12	200	60
Palani et. al (2011) [224]	15	5	2500	50
Crauste et. al (2017) [225]	16	5	15	60
Sneyd et. al (2002) [226]	16	6	2	60
Becker et. al (2010) [227]	17	6	15	60
Brannmark et. al (2010) [228]	18	9	50	90
Ray et. al (2013) [229]	20	6	25	60
Elowitz et. al (2000) [230]	22	8	30	80
Fiedler et. al (2016) [231]	24	6	10	60
Borghans et. al (1997) [232]	24	3	3	30
Fujita et. al (2010) [233]	26	9	2000	90
Bertozzi et. al (2020) [234]	36	3	200	30
Hass et. al (2017) [235]	37	9	100	90
Raia et. al (2011) [236]	45	14	150	140
Smallbone et. al (2011) [237]	52	6	10	60
Weber et. al (2015) [238]	53	7	10	70
Zheng et. al (2012) [239]	62	15	100	150
Isensee et. al (2018) [240]	63	25	100	250
Chassagnole et. al (2002) [241]	117	36	10	360
Messiha et. al (2014) [242]	192	28	10	280
Mosbacher (2023) [243]	280	95	10	980

lower and upper bounds of the true parameter values, i.e., $\theta_{true}/X \leq \theta_{true} \leq X\theta_{true}$. To investigate the effect of priors on initialization success, we chose priors for five different values of X: 2, 5, 10, 50, and 100. The initialization success rate, that is, the first iteration of stochastic gradient descent that leads to an estimate of the error, is calculated as a percentage of the total number of initializations.

Convergence of successful parameter initializations

For successful initializations, training is performed with 3000 iterations of stochastic gradient descent (AdaBelief) per initial parameter guess. To reduce computation time, training is stopped early when the loss function is below the loss threshold $\lambda = 10^{-6}$.

To quantify convergence properties, we look at the relative improvement in the mean squared error after training with respect to the initialization loss, as well as the percentage of successful training ($\lambda < 0.001$).

Global norm of gradients

The magnitude of the gradients with respect to the parameters is followed during the training process using the L2 norm (eq. 4.3). This allows to investigate the training process in more detail for different models.

$$||\nabla J(\theta)|| = \sqrt{(\frac{\partial J}{\partial \theta_1})^2 + (\frac{\partial J}{\partial \theta_2})^2 + ... + (\frac{\partial J}{\partial \theta_n})^2} \tag{4.3}$$

4.2.5. Fitting the glycolysis model

To showcase the usefulness of the proposed framework we fit feast/famine datasets to a previously established kinetic model of glycolysis [246] . The model was reimplemented to be compatible with *JAX* to make parameters trainable [206]. The parameter initialization was done from parameters retrieved from literature. The model consists of 29 metabolites and 38 reactions, totaling 141 parameters. Twenty distinct rate laws were implemented as reusable *JAX* classes. Details on the implementation are reported in Supporting information in Supporting information.

4.2.6. Hybrid modeling example

A minimal model of metabolic oscillations in biofilms was used to showcase the usefulness of automatic differentiation capabilities in *JAX/Diffrax* [219]. Reaction names in the original model were replaced by symbolic names for conciseness (v_1-v_{10}). In this setup, we assume that a certain reaction and its stoichiometry are unknown, a scenario that is often encountered in real-world applications. The i-th reaction of the model of the fully mechanistic description (eq. 4.1) is then replaced by a neural network (eq. 4.4).

$$\frac{dm(t)}{dt} = S^{(-i)} \cdot v^{(-i)}(t, m(t), \theta) + NN(t, y, w, b) \tag{4.4}$$

We perform a masking experiment for every reaction in the model, with 100 time-points and a noise percentage of 0.05%. All reactions and the stoichiometry were individually masked.

4.3. Results

4.3.1. Training SBML models using *Diffrax*

In order to analyze the behavior of the parametrization using techniques from neural ODEs, an efficient and easy-to-use ODE simulation tool for systems biology models

with automatic differentiation options for calculating adjoint sensitivities was required. We have implemented a *JAX*-based simulation tool and training framework that is compatible with Systems Biology Markup Language (SBML) models in *Diffrax*[206, 209] (Fig 4.1). SBML models are the standard accepted format for saving systems biology models in a reproducible manner [211].

The training input consists of three information modes. The kinetic model is loaded from an SBML format or a manually implemented *JAX*-compatible class that allows for Just-In-Time (JIT) compiling. The observed time-series data that is used for fitting is used to get the initial conditions from t_0. Finally, an initial parameter guess is required, which was obtained using Latin Hypercube Sampling [245]. Due to nonlinearities in the formulation of systems biology models and the potential non-convexity of the solution space, multiple parameter initialization are required.

For the training process, the ODEs are solved for timepoints that are observed in the dataset and the loss function is calculated. Due to large differences in metabolite concentration ranges, a mean-centered loss function was used to ensure roughly equal contribution of metabolites to the mean squared error. Finally, N iterations of stochastic gradient descent using AdaBelief can be performed [215].

While kinetic models typically have a mechanistic structure of the right-hand-side of $dm(t)/dt$, similar tools that are used to train Neural ODEs (e.g., the adjoint state method) can be applied. The goal of the study performed here is to address properties on the convergence of local gradient-based methods for a large collection of systems biology models.

4.3.2. Loss convergence analysis reveals robust training of systems biology models

Training kinetic models by gradient descent requires an initial guess of parameters. This requires setting a lower- and upper bound of parameters for the initialization and sampling the space using any sampling method. Due to the dependency of the loss function on numerical integration of the ODEs, not every parameter initialization might be successful. This can be attributed to either stiffness of the dynamical system or unstable behavior of the systems given that particular initial guess [191]. We therefore aim to understand how loss convergence is affected by parameter priors and model size, as well as how robust the training process is to these features by quantifying the percentage of successfully trained models given the parameter initializations.

To motivate the main analysis of SBML models, we show the influence of parameter bounds on one model when sampling 100 initializations using Latin Hypercube Sampling (Fig 4.2A)[245]. The percentage of models that are below the loss threshold are reported for five different parameter bound priors. It is observed for most models that with larger bounds, the percentage of succesfully trained models is either negatively affected

or not affected. This behavior is expected, as starting your parameter initialization closer to the true optimum would be an easier problem. We also compare the convergence success among five different systems biology models with a fixed prior bound ($\frac{1}{10}\theta_{true} \leq \theta_{true} \leq 10\theta_{true}$) (Fig 4.2B). This allows for comparing the performance of the parametrization framework across a large collection of Systems Biology models. In order to compare many models across different bounds, we chose a loss threshold of 10^{-3}.

4.3.3. Initialization success and the training process is stable across models and priors

Fig 4.2C shows a heatmap for 25 SBML models for five different priors. The left column of each model is the initialization success percentage, while the right column is the percentage of models after training that were below the loss threshold (10^{-3}). Overall, we see a high initialization success percentage for most SBML models. For many models, the loss function can be calculated independent of the bounds used in this study. For six models, the behavior of decreasing initialization success given the priors is observed, in line with what would be expected [222, 228, 230, 232, 241, 243]. Interestingly, a reversed pattern is observed for two models, where the initialization success increases with larger bounds [240, 242]. For one model, the initialization success is low, independent of the prior [225].

The training process consists of 3000 iterations of stochastic gradient descent using AdaBelief and a global norm clipping with a learning rate of 10^{-3}, without any batching of training data [213, 215]. The right column of each model for Fig 4.2C shows the percentage of successfully trained parameter given the initialization success. Here, the effect of priors is observed more clearly. For the smallest bounds ($[1/2\theta_{true}, 2\theta_{true}]$), we observe good convergence to the global minimum for most models. In some cases [243], the loss during the initialization is already below the chosen loss threshold, therefore having a similar percentage of successful initializations to successful training convergence. Parameter initializations that are not successfully trained can occur due to several reasons. First, the number of steps of the optimizer might not be enough for the loss to be below the threshold. Second, during optimization the gradient descent might lead to a parameter set that is not numerically solvable due to stiffness or divergence issues. When we increase the parameter bounds, it is typically observed that a decreasing number of parameter initializations are successfully trained. Generally speaking, we do not observe a clear relation between the number of parameters (or state variables) and the difficulty of training these models, except that the computation time increases for larger scale models.

To further observe whether the models fit data such that it properly captures the metabolite concentration dynamics, we simulate three models with losses below the threshold given in the heatmap for their best fit parameters with bounds $[\frac{1}{10}\theta_{true}, 10\theta_{true}]$ and compare it to the simulated data points (Fig 4.2D-F). It can be observed that for the

model from Beer et. al (2014) and Messiha et. al (2014) the dynamics are fairly close if not perfectly matching the simulated data points (Fig 4.2D,E). For the model from Garde et. al (2020), despite the loss being below the threshold, the trained dynamics are not close to the true dynamics, but rather finds a heavily oscillating parameter set that then matches the data. This behavior is not observed when the prior is between one-fifth and five of the true parameters. This suggests that evaluating the fit solely based on the objective can be misleading and additional visual inspection of the dynamics is needed. Furthermore, for some models prior information could be more important than for others, and methods to filter parameters based on spectral analysis of the Jacobian matrix [132, 194] could be beneficial.

Overall, initialization success is stable across a wide variety of models, which increases the likelihood of effectively learning parameters. While we see for some models a dependency on the bounds, this effect can be mitigated by increasing the initial sampling size. The loss after training shows a clearer effect of the priors. This indicates that when parameterizing kinetic models, the prior can be of high importance. Although it might in practice be difficult to define strict bounds for parameters *a priori*, using kinetic databases like Brenda might be a way to guess an initial parameter. Additionally, analysis of sloppiness in systems biology, as well as identifiability analysis, might aid in increasing success of model fitting (see Supporting information in Supporting information) [188, 247].

4.3.4. Large scale kinetic models can be trained using *jaxkineticmodel*: an example of glycolysis

While we have shown that neural differential equations could be used for a large variety of different SBML models, real time-series metabolomics data has additional complexity in terms of heterogeneity of the measured metabolites as well as noise. We therefore reimplemented a kinetic model of glycolysis in *JAX* [206, 246]. This model consists of 20 metabolites, 37 reactions, and 141 parameters. The kinetic mechanisms used are implemented as *JAX* compatible classes for well-known equations (e.g., Michaelis Menten) that can be reused for other purposes (see Supporting information in Supporting information).

The model was initialized with literature values reported in the previous MATLAB implementation [246]. 10000 iterations of stochastic gradient descent using AdaBelief were performed, resulting in the fit as reported in Fig 4.3. The panels show the dynamic response of a glucose pulse during a feast-famine cycle for some previously reported datasets [214]. The striped lines are inferred dynamic responses of metabolites, for which no training data was available. The glucose pulse obviously leads to an increase through the upper- and lower glycolysis, as well as a pulse through the glycerophospholipid pathway. In the first steps ATP is formed, which is well-captured by the dynamic response of the model. There is also a drop in NAD+ due to an increased activity through the lower glycolysis, where NAD+ is consumed by *glyceraldehyde-3-phosphate dehydrogenase*.

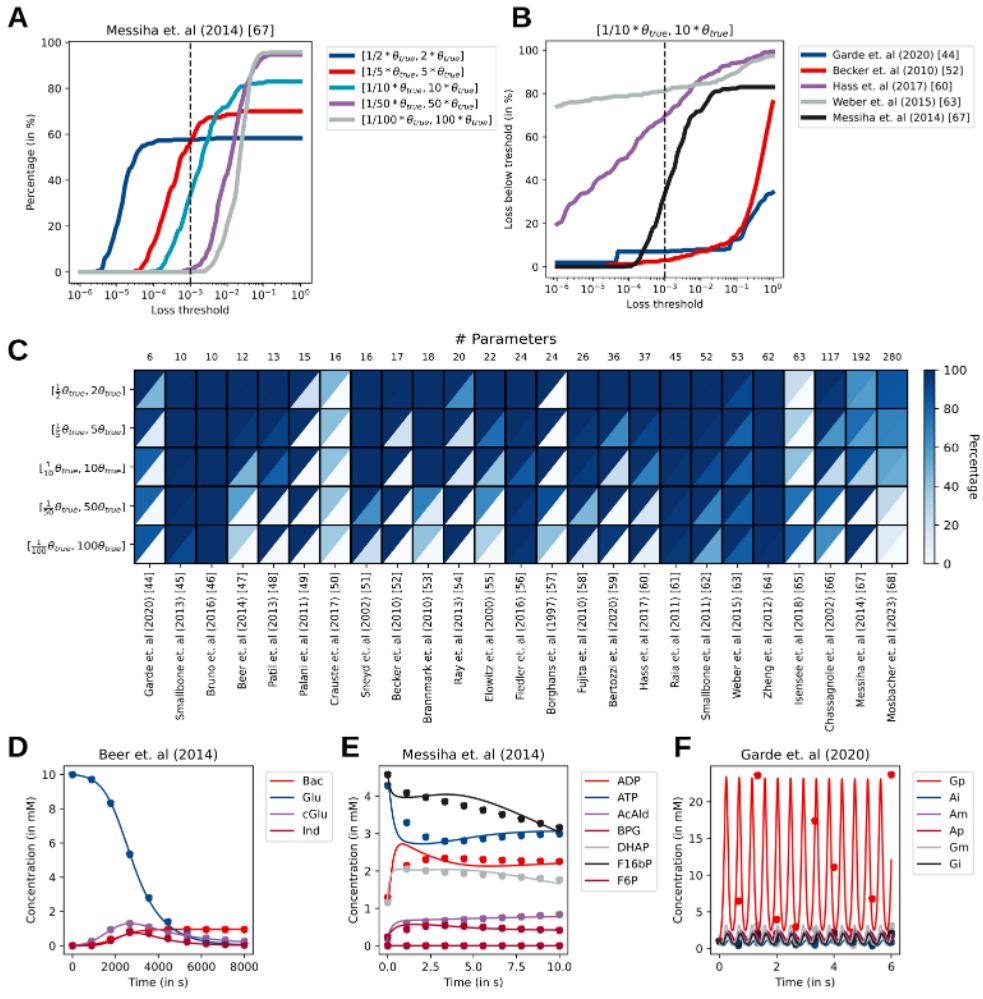


Figure 4.2: **Analysis of convergence properties for a collection of SBML models.** Latin hypercube sampling is performed for five different lower and upper bound priors of the parameters and training is performed. A) Percentage of successful convergence of an example model, where success is defined as the percentage of models below the loss threshold on the x-axis. B) A similar plot, but now with a fixed prior and five SBML models. C) Heatmap for the initialization success percentage (upper diagonal) and training success (lower diagonal) for $loss < 10^{-3}$. The number of successfully trained models is dependent on the initialization success and is therefore always a lower percentage. D,E,F) Examples of fitted SBML models after training.

Overall, a good fit is observed between modeled and measured data. Even though this is a

relatively large and complicated model of glycolysis, the fitting process for one dataset took approximately four hours (for 10000 update steps) on one CPU with 10GB memory. This shows that the implementation could be used to fit large-scale kinetic models. Furthermore, the training process is easy to extend to fit multiple datasets simultaneously (see Supporting information in Supporting information).

4.3.5. Flexibility of automatic differentiation with *jaxkineticmodel* allows for hybridizing kinetic models with neural ODEs.

Although *jaxkineticmodel* is capable of parameterizing large kinetic models, several other parameterization methods reported in the literature can perform similarly for mechanistic models (see Supporting information). One package that is methodologically similar to *jaxkineticmodel* is PyPESTO with gradient computations from AMICI [197, 198]. Instead of automatic differentiation, AMICI uses symbolically derived parameter sensitivity equations in combination with the efficient CVODE solver to perform the forward and adjoint simulation [248]. This results in PyPESTO/AMICI outperforming *jaxkineticmodel* in terms of computation time, which can primarily be attributed to the use of CVODE (see Supporting information and Supporting information in Supporting information). However, integrating neural networks with mechanistic models, an aspect that is of increasing interest to the scientific machine learning community when modeling biological systems [76, 204, 205], becomes practically impossible when the neural network components need to be symbolically derived.

To present this flexibility of automatic differentiation, we show an example of how mechanistic models can be combined with neural networks, using a minimal model of metabolic oscillations in biofilms [219]. The original model consists of six species and ten reactions with an oscillatory dynamic (Fig 4.4A). Suppose that a stoichiometry and mechanism of a reaction are unknown, for example the reaction that transforms G_m into A_i (ν_9). This missing reaction mechanism changes the dynamic behavior of the model (Fig 4.4B). By hybridizing the kinetic model with a neural network for ν_9, the dynamic behavior of the original model can be recovered after training (Fig 4.4C). This can be performed for all reactions in the model, where for eight out of ten reactions the expected dynamic behavior could be recovered (Supporting information in Supporting information). In the other two reactions, the dynamics of the masked model is diverging and get stuck in a local minimum when training. Further work is required to understand how training hybrid models can be successful in all cases.

4.4. Availability and Future Directions

Large-scale kinetic models have the potential to explain systems level behavior of biological systems, but their parametrization has proven to be challenging [186]. Furthermore, the dynamics of biological systems are often only partially understood from a mechanistic perspective, stressing the need for hybrid approaches [76, 204, 205]. In

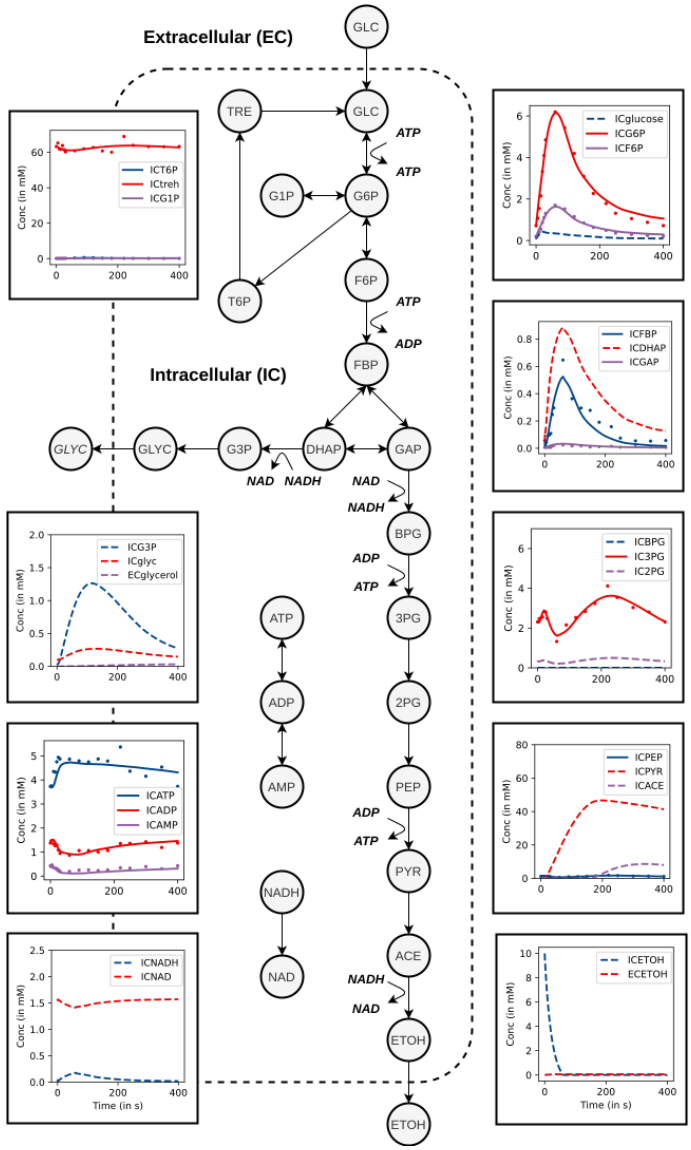


Figure 4.3: **Fitting a feast/famine cycle to a glycolysis model.** A simplified schematic of glycolysis, along with panels of the dynamic response of glycolysis to a glucose pulse [214]. Intracellular (IC) and extracellular (EC) metabolite concentrations that were measured are shown as dots in the panels, while lines indicate model predictions. Metabolites that were included in the model, but for which there was no available data, are represented as dashed lines.

this work, we have implemented a training framework, inspired by techniques from neural ODEs, tailored to systems biology models. Due to the *JAX*-based simulation using

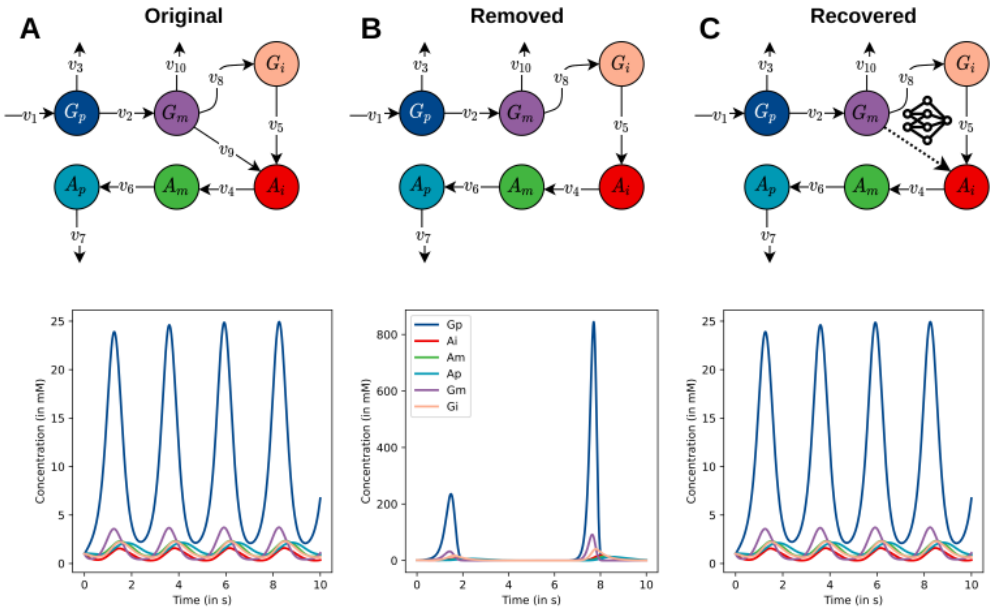


Figure 4.4: ***Jaxkineticmodel* allows for hybridizing kinetic models with neural networks.** Due to automatic differentiation capabilities of JAX/Diffrax, integrating neural networks with kinetic models becomes straightforward. As an example, we show how kinetic mechanisms can be replaced with a neural network. A) A schematic representation (upper) and time-series plot (lower) of a minimal model of metabolic oscillations in a biofilm [219]. B) Schematic and time-series plot of the same model without v_9. C) The hybrid kinetic model, where v_9 is replaced with a neural network. The time-series plot shows the dynamics of the hybrid model after training.

Diffrax, the training is relatively fast, which paves the way to large-scale kinetic model training [206, 209]. Furthermore, due to support for Neural ODEs by *Diffrax*, flexibility in using mechanistic and neural network approaches offers a useful approach to modeling biological systems.

Our default implementation of the *jaxkineticmodel* training framework mitigates parameter fitting challenges caused by characteristics of biological systems [187–189], as well as solutions from the field of neural ODEs [191, 201, 213]. As fitting kinetic models often requires a multiple starting approach, we note that the current approach is easy to parallelize, and will be a future direction. The implementation is publicly available as a Python package at `https://github.com/AbeelLab/jaxkineticmodel`. All data, source code, and results are available on the 4TU repository (`https://data.4tu.nl/datasets/3662eca5-7077-4ca3-8f66-d051e2c79cbe`).

We showcase the methodology on a large collection of SBML models of varying sizes [187, 218] and perform a principled analysis on the effect of parameter priors on the performance. We furthermore show that *jaxkineticmodel* is useful for parameterizing large kinetic models in real-world applications by fitting a glucose pulse dataset from a feast/famine experiment to a previously established glycolysis model [214, 246]. Finally, we show that due to automatic differentation capabilities in *JAX*, kinetic models can be hybridized with neural network components when a systems is only partially understood [204, 205]. This feature makes *jaxkineticmodel* different from other parameterization methods. Future work will aim to make these hybrid modeling capabilities more accessible to the user through tutorials and documentation.

4.5. Supporting information

Additional data, results, and supplementary material referred to in the main text can be found in the original publication (see QR-code).

- **S1 Text**: Supporting information on neural ordinary differential equations and *jaxkineticmodel*. This contains figures **A-F** and tables **A-D**.
- **Fig. A**: Schematic of different methods of gradient computations. In this work, the discretize-then-optimize method is used due to stabler properties.
- **Fig. B**: Cubic Spline interpolation. Used to model the external glucose pulse time-series.
- **Fig. C**: Loss during gradient descent. The loss for three independent datasets, as well as models trained on multiple datasets simultaneously, is shown.
- **Fig. D**: Small subset of parameters explain dynamic behavior in systems biology models. Visualization of the parameter space before and after training.
- **Fig. E**: Additional visualizations of parameter space before and after training.
- **Fig. F**: Timeseries plots and training loss for all reactions of the masking experiment.
- **Table A**: Table with kinetic mechanisms implemented in *jaxkineticmodel*. This table consists of general mechanisms that can be reused when building models.
- **Table B**: Time comparison forward simulation *Diffrax* versus AMICI.
- **Table C**: Time comparison adjoint state methods *Diffrax* versus AMICI.
- **Table D**: SBML-compatibility report. Fraction of 500 models that are supported by *jaxkineticmodel* compared to libroadrunner.

5

Hybrid modeling of metabolic pathways across data modalities: challenges and potential solutions

Paul VAN LENT, Bálint MAGYAR, Rianne VAN DER HOEK, Joep SCHMITZ, Thomas ABEEL

Motivation: *Metabolic kinetic models are essential for analyzing and predicting metabolic phenotypes, but constructing accurate and reliable models remains challenging. In contrast, machine learning approaches prioritize predictive accuracy, often at the expense of mechanistic interpretability. Hybrid modeling combines mechanistic descriptions of well-understood components with neural network representations of uncertain parts. This offers a promising balance between flexibility and insight, but their application in metabolic modeling has not been widespread.*

Results: *We developed and evaluated hybrid modeling approaches for two common tasks in metabolic engineering. First, we addressed time-series prediction in metabolic pathways with partially unknown kinetics by introducing three hybrid architectures, each incorporating different inductive biases. We show that best performance is achieved by a model that enforces mass-balance constraints through the stoichiometric matrix. Second, we modeled the influence of strain genotype on metabolism by linking genetic perturbations to metabolic fluxes using a neural-network component. This approach achieves standard supervised machine-learning performance with better interpretability, but the initial model failed to reproduce the yeast diauxic shift, limiting its generalizability. Refining the mechanistic module to capture this transition significantly improved*

This chapter is under preparation for submission

predictions of biomass and p-Coumaric acid production. Sensitivity analysis suggests that numerous genes may affect p-Coumaric acid biosynthesis. Overall, these results highlight the promise of hybrid models for metabolic engineering, while also underscoring practical challenges in their design and deployment.

Availability and implementation: *The implementation of hybrid models can be found at* `https://github.com/AbeelLab/hybrid_kinetic_modeling`. *Data, models, and results that were used in this study can be found on the 4TU repository* `10.4121/9efb93ec-6b25-4784-945a-c0f17be786ea`.

5.1. Introduction

Systems metabolic engineering is an interdisciplinary field that integrates systems and synthetic biology with metabolic engineering, aiming to design microbial cell factories [249]. These cell factories are key to a more sustainable chemical industry by replacing petrochemical-based chemicals with bio-based alternatives [10]. Engineering microorganisms is a complicated and expensive process, often requiring years of research and development [250], primarily because a quantitative understanding of metabolism remains incomplete. Therefore, modeling tools that can manage partially understood systems can be useful for improving the production of strains [251].

Engineering living systems through model-guided design is often approached using two modeling paradigms. On the one hand, there are mechanistic models, such as constraint-based genome-scale models and metabolic kinetic models [252, 253]. Kinetic models, which are ordinary differential equations (ODEs), provide a more detailed description of metabolism compared to genome-scale models, enabling the dynamic interrogation of metabolic behavior [254]. However, their complex formulation, construction, and parameterization make it difficult to create accurate predictive kinetic models that also resemble physiological and metabolic processes [195]. Conversely, supervised machine learning models are designed to learn the desired output behavior from training data [255]. These machine learning methods have demonstrated impressive predictive abilities across numerous tasks that are important for strain optimization [38, 49, 256, 257]. Nonetheless, these models function as black boxes, limiting the interpretability and explainability of predictions [46].

Universal differential equations (UDEs) have shown significant potential in combining mechanistic models with artificial neural networks through an additive framework [76, 258]. These hybrid approaches aim to combine the best characteristics of two modeling paradigms: strong predictive performance (machine learning models) with increased generalizability and interpretability (mechanistic models) [204, 259]. In metabolic modeling, the mechanistic model is described by the stoichiometric matrix S, kinetic fluxes $\vec{v}$, and an artificial neural network with parameters θ_{NN} (eq. 5.1).

$$\frac{dm(t)}{dt} = S\,\vec{v}(t,m,\theta) + NN(t,m,\theta_{nn}) \tag{5.1}$$

The UDE framework has several benefits over purely mechanistic or purely machine-learning methods. It improves the predictive performance of mechanistic models when the system is not fully understood while simultaneously offering a more interpretable model than machine learning methods. They also potentially result in increased generalizability when predicting outside the training data distribution [204, 259]. However, the flexibility of UDEs also necessitates careful design choices in hybrid modeling architectures to include relevant inductive biases related to the specific application domain. While it is possible for many omics data modalities to generate time-series data (e.g., metabolomics), this is challenging for microbial strain genotypes [260]. Specifically for metabolic engineering, this is relevant; pathway optimization datasets are often cross-sectional in nature [30].

This work investigates hybrid modeling strategies within the context of metabolic engineering. We first concentrate on designing and assessing hybrid architectures for time-series modeling of metabolic systems in which the underlying reactions are not fully specified. Three architectures are introduced, substituting distinct components of the metabolic network, such as stoichiometric relationships or flux rate expressions. These substitutions impose different inductive biases during training. Several techniques are applied to stabilize the training of these types of models [78, 261]. Building on the insights gained from time-series modeling, we then examine how genetic perturbations in a strain influence metabolic system behavior. To this end, a hybrid kinetic model is constructed that uses a neural network to relate genetic perturbations in the strain to the resulting fluxes. Training is performed on combinatorial pathway optimization data obtained from a p-coumaric acid–producing yeast strain [137]. P-Coumaric acid (pCA) is an aromatic amino acid-derived intermediate for the biosynthesis of numerous bioactive compounds [145]. The hybrid kinetic model is evaluated both in terms of its predictive performance and the interpretability of the resulting model.

5.2. Materials and Methods

5.2.1. Data

Synthetic data for time-series modeling

We evaluated hybrid modeling frameworks for time-series analysis using five SBML models that capture different biological dynamics, ranging from simple monotonic changes to complex oscillations [228, 262–265]. Each model included simulations with incomplete knowledge by omitting one or more reactions from the mechanistic component. The hybrid models were trained on sparse, noisy time-series data derived from the SBML models to recover the true system dynamics, using 100 samples with 5% noise based on the standard deviation per metabolite.

Cross-sectional data

For modeling cross-sectional data, combinatorial pathway optimization data from two Design-Build-Test-Learn (DBTL) cycles of pCA producing strains were used. Details on experimental procedures and processing steps to construct the gene numeric matrix can be found in the original study [137]. Briefly, the first cycle involved transforming a large library with 19 gene targets and 20 promoters, generating approximately 3000 strains screened via Nuclear Magnetic Resonance (NMR) and over 403 uniquely sequenced DNA strains. The second cycle featured machine learning-guided recommendations for two chassis strains: the original library transformation strain (SHK0066) and the top predicted strain with four promoter-gene-terminator library elements (SHK0073). Genetic inputs were standardized (z-scored) before training the machine learning models (linear model and XGBoost) and the hybrid model.

To investigate model performance for a range of training set sizes, data from the first cycle was subsetted, ranging from 6 to 250 training instances. Data from the second DBTL cycle was treated as an external test dataset. The model was validated using 10-fold cross-validation sets, test data from chassis strain SHK0066 (approximately 50 data points), and test data from chassis SHK0073 (approximately 25 data points).

Optical density data for model validation

Twenty-one strains from a previous combinatorial pathway optimization of p-Coumaric acid were chosen for optical density measurements in Biolector (Beckman Coulter, United States) [137]. Precultures were prepared in 96-well half-deep well plates (HDWP) containing 350 µL YEPhD + Pen/Strep (Invitrogen, Thermofisher Scientific, United States) and incubated at 30°C, 750 rpm, 80% humidity for 48 hours. 20 µL of a 2x diluted pre-culture was re-inoculated to MTP-R48-B FlowerPlate (m2p-labs) containing 1 mL YEPhD + Pen/Strep (Invitrogen, Thermofisher Scientific, United States). The plate was incubated for 48 h in Biolector Pro® (m2p-labs) at 30°C, 800 rpm, 85% humidity. Biomass (em. 620nm/ex.620nm) with 3 filters (gain of 1, 3, and 7), were measured every 15 minutes.

5.2.2. Hybrid modeling architectures

Model architectures for time-series data

Three hybrid neural ordinary differential equation (ODE) models are proposed. These models integrate a mechanistic model with a neural network in various configurations, which are special cases of the more general UDE approach (Fig. 5.1A). The Additive Neural ODE (**ANO**) approach uses a neural network NN_θ to learn the residual dynamics, capturing the discrepancy between the true system dynamics and those represented by the mechanistic model (eq. 5.2). Here, S represents the stoichiometric matrix, $\vec{v}(t,m,\theta)$ represents the vector of reaction rates of the mechanistic model, and $NN_\phi(m,t,\theta_{nn})$ represents a neural network correction term. This structure is flexible, allowing the neural

network to function as a universal correction term, which is useful when the mechanistic model is known to be incomplete or inaccurate in both its stoichiometry and flux rates.

$$\frac{dm(t)}{dt} = S\,\vec{v}(t,m,\theta) + NN_{\phi}(t,m,\theta_{NN}) \tag{5.2}$$

The Mass-conserving Neural ODE (**MCNO**) architecture enforces system-level mass conservation by applying stoichiometric constraints after combining neural and mechanistic rate contributions, $\vec{v}(m,t,\theta)$ and $NN_{\phi}(m,t,\theta_{nn})$, respectively (eq. 5.3). The neural network term $NN_{\phi}(m,t,\theta_{nn})$ learns a correction term for reaction fluxes, imposing a stronger inductive bias from the stoichiometric matrix S. This ensures that predicted changes in concentration strictly adhere to mass balance laws.

$$\frac{dm(t)}{dt} = S\,[\,\vec{v}\,(t,m,\theta) + NN_{\phi}(t,m,\theta_{nn})] \tag{5.3}$$

Finally, the Reaction-specific Neural ODE (**RSNO**) architecture uses a neural network to learn the kinetics for a specific, predefined subset of reactions whose mechanisms are unknown. The mechanistic model provides rates for known reactions, and the two sets of rates are concatenated into a single vector before being multiplied by the stoichiometric matrix. This explicitly separates the learning task for unknown kinetics from known mechanistic rates.

$$\frac{dm}{dt} = S\left[\vec{v}_i \notin \phi(t,m,\theta)\;;\; NN_{\phi}(t,m,\theta_{nn})\right] \tag{5.4}$$

Perturbation architecture

The hybrid architecture framework integrates a mechanistic kinetic model with a neural network component that models the (nonlinear) mapping from strain genetics to fluxes (eq. 5.5). Here, $dm(t)/dt$ denotes the changes in metabolite concentrations over time, $\vec{v}$ represents kinetic fluxes, y is the metabolic state, and θ denotes the kinetic parameters. Fluxes are modified by a nonnegative vector $\vec{v}_{perturb}$ that depends on the strain genotype (eq. 5.6). An overview of all the computational components of the architecture can be seen in figure 5.2C.

$$\frac{dm(t)}{dt} = S\,[\vec{v}(t,y,\theta) \odot \vec{v}_{perturb}] \tag{5.5}$$

$$\vec{v}_{perturb} = NN(strain_i, w, b) \tag{5.6}$$

The kinetic model of a batch bioprocess consuming glucose was constructed using the Python package *jaxkineticmodel* and exported in SBML format (Fig. 5.2A) [178, 266]. The model describes a batch bioprocess that feeds on glucose, with a simple metabolic pathway consisting of lumped reactions of glycolysis, the pentose-phosphate pathway, and the shikimate pathway [162]. In total, the model includes eight species, ten reactions, and twenty-three kinetic parameters. All reaction fluxes are described using Michaelis-Menten rate laws [267]. The baseline strain (SHK0066) was calibrated so that

the simulated production of pCA matches experimentally measured titers. A detailed description of the model structure, parameter values, and assumptions is provided in Supporting Information 5.6.1.

Strain-specific perturbations to the metabolic fluxes, $\vec{v}_{\text{perturb}}$, were represented by a shallow neural network with two hidden layers (depth = 2, width = 24). A softplus activation function was used in all hidden and output layers to enforce the non-negativity of the predicted flux perturbations [268]. The neural network takes as input the genetic configuration of each strain and outputs flux adjustments that are additively combined with the baseline kinetic fluxes to yield strain-specific flux profiles.

Modeling diauxic growth shift using differentiable logical operators

The diauxic shift from growth on glucose to growth on ethanol in *S. cerevisiae* [269] was modeled as a improved alternative to the glucose-only model. Training using gradient descent requires that the ODEs are differentiable. To include logical switching from glucose to ethanol, a logical statement is replaced with a sigmoid function. Suppose a statement exists that checks whether glucose is almost depleted (s_{gluc}). Then, the sigmoid function (eq. 5.7) maps glucose concentrations above the glucose reference close to one (True) and those below the reference glucose concentration close to zero (False). The parameter k describes the steepness of the transition. In the special case that $k \to \infty$, the sigmoid statement coincides with a boolean statement. Logical operations (AND and OR) are analogous to equations 5.10-5.9. These operations are used to define a differentiable switch from growth on glucose μ_{gluc} to growth on ethanol $\mu_{ethanol}$ (eq. 5.11).

$$s_{gluc} = \frac{1}{1 + e^{-k([gluc]-[gluc_{ref}])}} \tag{5.7}$$

$$s_{eth} = \frac{1}{1 + e^{-k([eth]-[eth_{ref}])}} \tag{5.8}$$

$$s_{gluc} \wedge s_{eth} = s_{gluc} \cdot s_{eth} \tag{5.9}$$

$$s_{gluc} \vee s_{eth} = s_{gluc} + s_{eth} - (s_{gluc} \cdot s_{eth}) \tag{5.10}$$

$$\mu = s_{gluc} \cdot \mu_{gluc} + (1 - s_{gluc}) s_{ref} \cdot \mu_{eth} \tag{5.11}$$

To capture the lag phase associated with proteome rewiring during the diauxic shift [269, 270], an additional state variable R that represents the adaptation of the protein machinery from glucose to ethanol utilization is included (eq. 5.12. This value is zero whenever glucose is abundant, but switches to an active mode when glucose is depleted. The fraction R_{frac} therefore delays ethanol growth (eq.5.13-5.14. Further details are reported in the Supporting Information SI A: Detailed kinetic modeling description.

$$d[R]/dt = \alpha \cdot ((1 - s_{gluc})s_{eth}) - \beta R \tag{5.12}$$

$$R_{frac} = \frac{R}{(K_R + R)} \tag{5.13}$$

$$\mu_{eth} = R_{frac} * \mu_{ethanol} \tag{5.14}$$

5.2.3. Training

Gradient descent specifics

The Adabelief optimizer was used for gradient descent with a learning rate of 3e-3 [271]. The global norm of the parameters in the neural network modeling the perturbation was clipped to prevent exploding gradients ($\hat{g} = 5$) [272]. Gradient descent was performed for the neural network that computes $\vec{v}_{perturb}$. During training, a batch size of four to twelve was chosen, depending on the number of samples in the training data. As the dataset is relatively small, only 300 gradient descent steps were deemed necessary. The loss function was chosen to be the mean squared error between the predicted and measured pCA at the time-point t_{end}, where the measured pCA concentration was sampled at $t_{end} = 50$.

Numerical solver specifics

Numerical simulations required to calculate the loss function were performed using the Tsit5 ODE solver from the Diffrax package [77, 273], with relative and absolute error tolerances of 10e-4 and 10e-6, respectively. The initial time-step dt_0 was set to 10e-10. The model was simulated in the range $t = [0, 60]$.

Stabilization techniques for hybrid architectures

Several methods were utilized to stabilize training. This includes an early stopping condition that halts the solution when unrealistic values emerge, preventing divergence. The loss was calculated only up to the point of divergence. Additionally, two regularization strategies for neural network weights were evaluated: i) initializing weights near zero and ii) applying L2 regularization. Lastly, physiological regularization was tested by penalizing negative or excessively high concentrations. The results presented on the time-series data were obtained using a mix of L2 regularization, near-zero weight initialization, and the early stopping condition.

Other supervised models

The hybrid kinetic model was compared with XGBoost and a linear model for predictive performance [274]. These models do not integrate any dynamics into the solution but predict pCA concentration directly from promoter strengths. For XGBoost, Bayesian hyperparameter optimization was performed using *scikit-optimize version 0.10.2* on

different training set sizes [275]. Model performance was compared on the test sets described in the section Cross-sectional data above. Predictive performance was measured using the Spearman rank correlation coefficient.

5.2.4. Validation

Masking experiments

The performance of three hybrid architectures, outlined in Hybrid modeling architectures, was assessed through a masking experiment using five SBML models. In this experiment, each model had all of its reactions masked once, and the objective for the hybrid architecture was to accurately recover the original dynamics. In the ANO architecture, this means thatsolving the learning problem needs to recover information from both the stoichiometric matrix and the flux function. For the MCNO and RSNO architectures, the focus is solely on learning the flux function since stoichiometry is presumed known. However, a key distinction is that MCNO can adjust corrections for other fluxes, whereas RSNO cannot. The effectiveness was measured by comparing the lag cross-correlation between the original model's true time-series data and the hybrid model's predicted time-series data.

Sensitivity analysis

The sensitivity of V fluxes to G genetic inputs of a strain can be estimated by summing the numerical Jacobian of fluxes $J(t)_{flux} \in \mathbf{R}^{V \times G}$ over simulated time-points for the perturbation vector. The rows represent the effect of an infinitesimal change of a promoter-gene value on the output flux. A similar approach can be taken for the M metabolites, where $J(t)_{species} \in \mathbf{R}^{M \times G}$ is summed over the simulated time-points. The Jacobian was computed using *JAX* [206]. For a set of strains, each individual Jacobian matrix is averaged to obtain the overall sensitivity behavior of the system.

5.2.5. Data and code availability

Datasets and code to generate results can be found at `https://github.com/AbeelLab/hybrid_kinetic_modeling` and the 4tu repository (doi:)

5.3. Results

5.3.1. Mass-Conserving Neural ODEs outperforms other hybrid model architectures for time-series modeling

Metabolic kinetic modeling is important for quantitatively understanding metabolic processes and is extensively applied in bioprocess development [45]. With incomplete insights into metabolic mechanisms, Universal Differential Equations provide a versatile approach to describing partially known systems [76]. The integration of neural networks into an ODE model can be performed in various ways. Here, we focus on three

architectures that impose distinct constraints on the hybrid model, which may decrease data requirements and generalizability (Fig. 5.1A). The Additive Neural ODE (ANO) structure corrects the dynamics with a neural network through an additive structure (eq. 5.2). The Mass-Conserving Neural ODE (MCNO) architecture enforces mass conservation by applying stoichiometric constraints (eq. 5.3). In this case, the neural network corrects flux rates instead of metabolite dynamics. Finally, we investigate a Reaction-Specific Neural ODE (RSNO) (eq. 5.4). In this case, we only model a specific reaction using a neural network while keeping all other reactions purely mechanistic (see Materials and Methods).

A major challenge in training hybrid models (and neural ODEs in general) is numerical instability. Random initialization of the neural network can introduce large perturbations that destabilize the dynamics of the system, causing the ODE solver to fail during the forward and adjoint computation (Fig. 5.1B) [272]. To overcome this, several techniques are employed: near-zero initialization of neural network weights to ensure that the mechanistic model dominates in the early training phase; early stopping of the ODE solver if concentrations exceed plausible bounds; and an adaptive solver selection strategy to balance speed and stability based on the system's numerical stiffness. These methods raised the proportion of successful training runs from almost 0% to over 85% across the chosen set of test models (Fig. 5.1C).

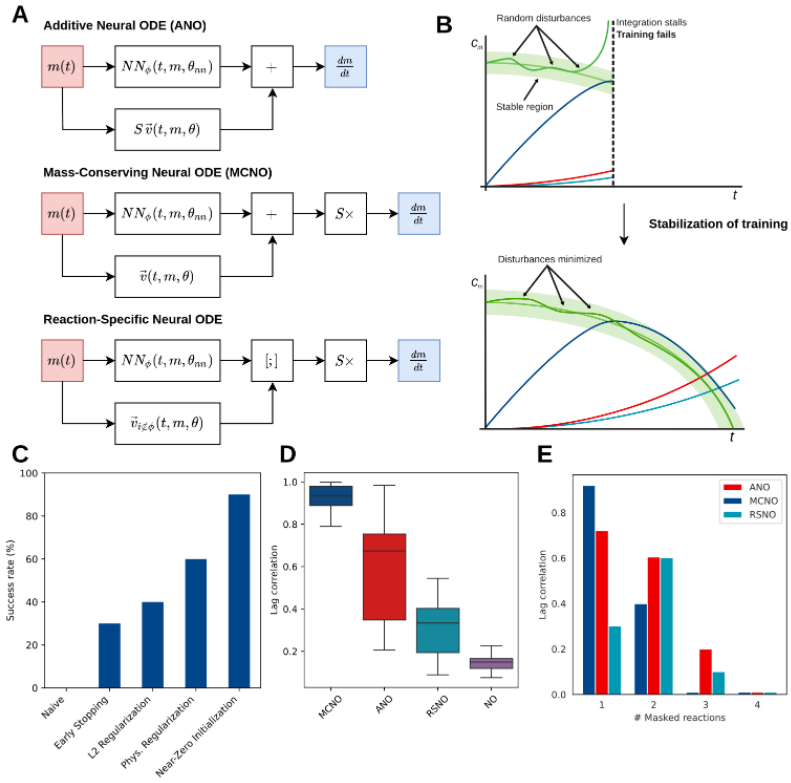


Figure 5.1: **Benchmarking of hybrid architectures for time-series data. A)** Flowcharts illustrating the three architectures: Additive Neural ODE (top), Mass-conserving Neural ODE (MCNO) (middle), and Reaction-Specific Neural ODE (RSNO) (bottom). **B)** Example of numerical instability during training caused by a randomly initialized neural network (top) compared to a stable trajectory using near-zero initialization (bottom). **C)** Training success rate improves dramatically in terms of percentage of finished training runs with the implementation of stabilization techniques. **D)** Comparison of predictive accuracy (lag correlation) for the different architectures when one reaction is masked. **E)** Performance degradation as the number of masked reactions increases. Only two models were used due to computational cost).

Once the training process is stabilized, the three architectures can be assessed on five SBML models. The task is to restore time-series data after omitting reaction(s) from the model (see Validation). The lag correlation was used as a metric that measures the similarity between the predicted and true time-series while accounting for phase shifts. Dynamic delays can lead to low mean squared error estimates despite accurate qualitative dynamics. We first investigate the performance of masking one reaction. When a single reaction was masked from the mechanistic model, the Mass-conserving Neural ODE (MCNO) demonstrated superior and more consistent performance across

all benchmark models (Fig. 5.1D). Its enforcement of stoichiometric constraints leads to more physically realistic and stable learning dynamics. The Additive (ANO) model showed intermediate performance, with a large variation in performance between models and masked reactions. The Reaction-specific (RSNO) model performed poorly, likely because of its rigid structure. It cannot compensate for inaccuracies in other parts of the model, in contrast to the ANO architecture. A pure neural ODE without any mechanistic knowledge performed the worst, highlighting the value of the hybrid approach. Next, we tested the effect of larger knowledge gaps by increasing the number of masked reactions (Fig. 5.1E). All architectures showed performance degradation as more mechanistic information was removed (Fig. 5.1F). This degradation in performance is evident in both predictive performance and training compute time, along with decreased stability. However, the extent of the decrease in performance differed between architectures. MCNO seems to degrade more in performance than the other structures, whereas the ANO architecture may be more stable in terms of performance (see Discussion).

These findings indicate that regularization techniques effectively stabilize the training of all hybrid architectures, a trend noted previously with neural ODEs [276] and UDEs [204]. While imposing stoichiometric constraints (MCNO) enhances performance compared to other architectures, its effectiveness is significantly reduced when applied to multiple masked reactions.

5.3.2. Hybrid kinetic modeling offers increased interpretability

In metabolic engineering, a central challenge is to model how strain genotypes shape the metabolic behavior of microbial cell factories. The genetic makeup of a strain governs the synthesis of metabolites, and the resulting data are usually cross-sectional. Numerous approaches for incorporating omics data into mechanistic models have been developed [277–279], as such integration can enhance both interpretability and explainability [46].

To investigate how genetic variation across engineered strains influences metabolic performance, a hybrid kinetic modeling framework was developed that integrates strain-specific genetic context through a neural-network-based component. This framework is tested with two kinetic models with different levels of detail. The baseline kinetic model employs a simplified representation of the metabolic pathway, comprising five lumped reactions corresponding to glycolysis, the pentose phosphate pathway, DHAP synthase, the shikimate pathway, and the conversion to p-coumaric acid (pCA) (Fig. 5.2A). The refined kinetic model includes a diauxic growth shift from glucose to ethanol and will be discussed in the next section. Combinatorial pathway-optimization data from a recent study included nineteen genes associated with these lumped reactions (shown in red in Fig. 5.2A) [137]. This data is used to train the neural network that modifies the baseline fluxes ($\vec{v}$) from the kinetic model by a non-negative flux perturbation vector ($\vec{v}_{perturb}$). The stoichiometric matrix S ensures mass-balance of the ODE system, similar to the MCNO approach above (Fig. 5.2C).

Across all evaluated training set sizes and test sets, the hybrid kinetic–machine learning model achieved predictive performance for pCA concentrations comparable to that of a linear model and XGBoost (Fig. 5.1D–F), as assessed by Spearman rank correlation between predicted and measured pCA levels (see Methods Cross-sectional data). The Spearman rank correlation was chosen to determine which model has the best strain performance ranking capabilities. All three approaches yielded similar correlations, indicating that pCA production can be reliably predicted by both purely data-driven models and the hybrid framework. For the test set lying outside the training data distribution, the hybrid model did not outperform the supervised machine learning baselines (Fig. 5.1C).

In contrast, the hybrid approach offered distinct advantages in interpretability and mechanistic insight. The model enabled dynamic interrogation of the metabolic response to specific genetic profiles (Fig. 5.4G), and its behavior could be further analyzed using systems biology tools such as flux control coefficients and metabolic control analysis (see Sensitivity analysis). These analyses indicated that genes promoting pCA production, including ARO4, PHA2, PAL1, and C4H, are predicted to reduce biomass formation (Fig. 5.2H,I). This is consistent with their previously reported strong impact on pCA production in *S. cerevisiae* [137]. Conversely, ARO8, CPR1, SOL4, and SOL3 were predicted to enhance biomass formation, in agreement with earlier reports on their roles in yeast metabolism [280–283].

In summary, in terms of predictive performance, we do not observe any improvement over conventional machine learning models. However, these findings do show improved interpretability and explainability, as tools developed for kinetic models can similarly be applied to this architecture [284]. To determine if the model's explanations are reasonable, further validation of these hypotheses is necessary.

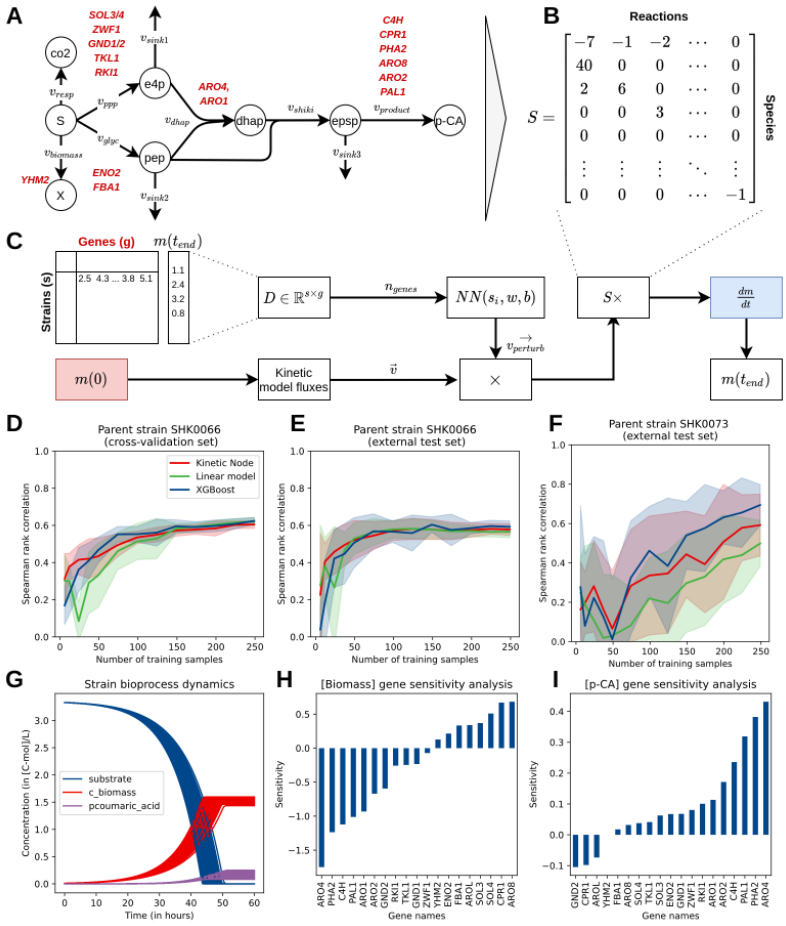


Figure 5.2: **A perturbation kinetic model for modeling combinatorial pathway optimization data. A)** A lumped mechanistic description of the pCA pathway within a simple bioprocess model with carbon dioxide and biomass production. The model consists of seven pathway reactions and three sink reactions on precursors. Genes that were perturbed using different promoters are shown in red. **B)** The stoichiometric matrix of the metabolic network. **C)** The hybrid architecture. Fluxes $\vec{v}$ are multiplied element-wise by the nonnegative output of a neural network that takes as input the promoter strengths per strain. The stoichiometric matrix S constrains the hybrid network to ensure mass-balances. Metabolite concentrations at t_{end} are used to train the hybrid kinetic model. **D)** The Spearman rank correlation coefficient for models trained on different training set sizes. The test data was based on holdout test sets from the same dataset. **E)** A similar plot for the same parent strain, but with a external test set (DBTL cycle two). **F)** Designs with another parent strain SHK0073 that was predicted to be the best performing strains, also from DBTL cycle two. **G)** Bioprocess dynamics of different strains can be retrieved from the hybrid kinetic modeling. **H)** Sensitivity profile of biomass for the genes. **I**) Similar profile for pCA. ARO4, PHA2, PAL1, C4H seem to have the strongest impact.

5.3.3. Extension of the baseline model to capture diauxic shift behavior

Where the baseline kinetic model used in Figure 5.2 was able to provide interpretability and good predictive performance for pCA, it was implemented as a glucose-only batch bioprocess model. Therefore, the model was not able to capture diauxic behavior when we analyzed growth dynamics in a panel of 21 representative strains using online optical density measurements in a Biolector system (Fig. 5.3A). All experimental trajectories showed a clear diauxic shift in the test strains, characterized by an initial glucose-supported growth phase, a subsequent lag phase associated with proteome rewiring, and a second growth phase on ethanol [269]. As a consequence, the initial model systematically failed to reproduce the observed lag phase and yielded inaccurate growth rate predictions (see SI B: Specific growth rate predictions and gene sensitivity analysis of the glucose-only versus diauxic shift bioprocess model).

Therefore, the extended kinetic model that includes diauxic shift behavior has a slightly modified structure compared to the baseline model. First, it includes two additional reactions: (i) conversion of glucose to ethanol and (ii) conversion of ethanol to biomass, assuming zero ethanol at the initial time point and ethanol accumulation during fermentation. Second, it consists of substrate switching mechanisms using a differentiable sigmoid-based switching function (see Modeling diauxic growth shift using differentiable logical operators), which activates ethanol consumption upon glucose depletion. Differentiability is necessary to ensure the application of gradient-based training methods. Third, we incorporated an additional state variable, R, to represent the reprogramming of the cellular protein machinery from glucose- to ethanol-based growth [269]. This variable modulates the ethanol-based growth rate and generates an effective lag phase in the simulations.

Following the calibration of the extended model on the parental strain SHK0066 used in the combinatorial pathway optimization study [50, 137], the model accurately reproduces the experimentally observed optical density trajectories, including the timing and duration of the diauxic shift (Fig. 5.3B). This demonstrates a more suitable mechanistic model for training.

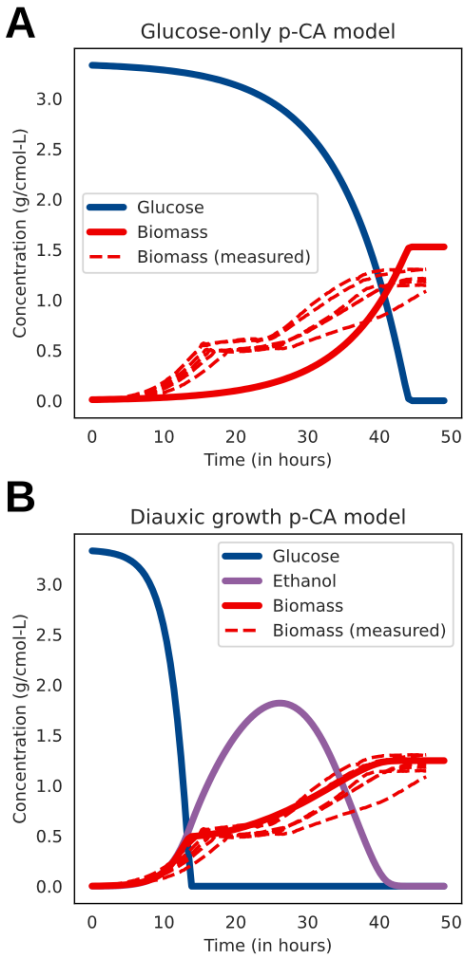


Figure 5.3: **Improved mechanistic modeling of diauxic shift A)** Time-series of the glucose-only model. **B)** Time-series of the improved mechanistic model that incorporates growth on glucose and ethanol. The lag-phase is modeled by a variable R that switches state during depletion of glucose and transiently surpresses growth on ethanol.

5.3.4. Extended model results in improved predictive performance and increased interpretability

Using the refined mechanistic (hybrid kinetic) model, we jointly trained on endpoint pCA measurements and biomass time-series data from the training set. Because the dataset is imbalanced between biomass and pCA, we down-sampled each biomass trajectory to fifty time points and up-weighted the pCA error in the loss function ($\lambda = 25$). For biomass predictions on unseen strains, the overall growth dynamics are captured well, except for

one strain (in purple) (Fig. 5.4A). For pCA levels, the refined mechanistic model achieved predictive performance comparable to both XGBoost and a linear baseline (Fig. 5.4B). These results demonstrate that hybrid kinetic models can match the predictive accuracy of standard supervised learning methods while providing direct access to the underlying process dynamics.

Finally, we examine how the sensitivity profiles obtained from the kinetic model compare to the permutation importance scores of the other machine learning models (Fig. 5.4C). Across all models, PHA2 emerges as highly important, consistent with experimental findings [159]. YHM2, for which the importance is only indicated by the hybrid approach, has also been shown to have a positive effect on pCA [137]. For ARO8, the literature reports conflicting results regarding its role in pCA production [159]. Note that an apparent positive or negative influence may arise indirectly through the gene's impact on biomass formation, which in turn affects pCA levels. Interestingly, C4H is considered important by the machine learning models, while its influence on pCA production has been reported to be minor [137]. Together, these findings underscore the potential of hybrid kinetic modeling to provide increased mechanistic interpretability. This motivates further experimental work to disentangle individual gene contributions and assess whether model-derived explanations are biologically meaningful.

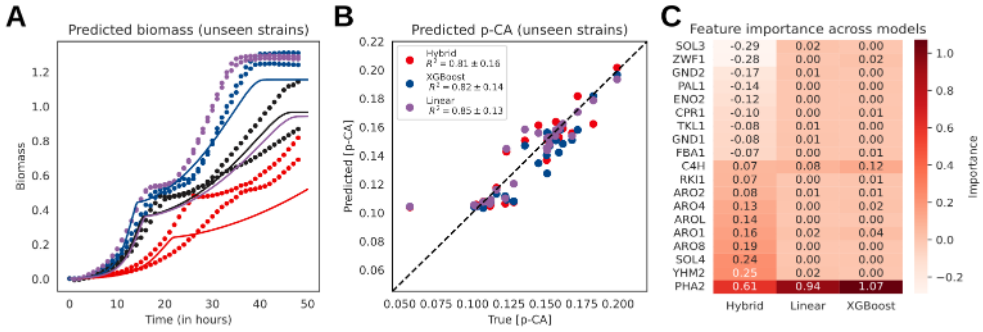


Figure 5.4: **Improved mechanistic modeling and validation of predictions and interpretability. A**) Predicted and measured biomass curves for four unseen strains (trained on 17 strain designs with two replicates). **B)** Predicted versus measured pCA for the hybrid model, XGBoost, and a linear model. **C)** Feature importance scores for the three models. For the ML models, feature relevance was evaluated using permutation-based importance. In the hybrid model, feature importances correspond to the sensitivities of genes with respect to pCA.

5.4. Discussion

This study examines hybrid kinetic–neural models for common metabolic engineering tasks. Building on recent advances in UDEs [76, 258], we compare three hybrid architectures with distinct inductive biases on synthetic time-series data and find that the MCNO architecture, which enforces mass conservation via the stoichiometric matrix, consistently outperforms the alternatives. However, performance deteriorates for all models as more reactions are masked. We then consider modeling cross-sectional genetic data from a combinatorial pathway optimization experiment for pCA production [137]. When the mechanistic model qualitatively captures the system dynamic, i.e. the diauxic shift, its predictive accuracy is comparable to supervised machine learning approaches, while additionally providing mechanistic interpretability that purely data-driven models lack. Although we focus on genomic data, the approach is readily extensible to other omics data types.

A key challenge in interpretable and explainable machine learning is the possible discrepancy between model-predicted behavior and true cellular physiology [46]. For instance, although the glucose-only model suggests that ARO4 and PHA2 negatively affect biomass formation, this influence is reduced when using the refined diauxic growth model (see Fig. SI5.5). This illustrates that when a mechanistic model fails to accurately capture the underlying biology, the resulting explanations are misleading. Nevertheless, even when being incorrect for well-understood reasons can be highly informative, as it enables systematic refinement of the model architecture through targeted, model-driven hypothesis testing.

For the prediction task, the diauxic growth model accurately reproduced the biomass trajectories and also yielded reliable predictions of pCA production. This indicates that the hybrid modeling framework can achieve performance comparable to that of supervised machine learning approaches. Nonetheless, the similar performance of the linear model and XGBoost suggests that the genotype–phenotype mapping within this dataset is relatively simple. Indeed, permutation analysis revealed that only two genes, PHA2 and C4H, substantially influenced pCA predictions in the machine learning models (Fig. 5.4F). Notably, the sensitivity profile inferred by the hybrid model appears more consistent with earlier studies on the metabolic engineering of *S. cerevisiae* for pCA production [137, 159, 285]. However, further targeted experiments are needed to rigorously validate these observations.

These results show that hybrid models are a promising approach for modeling metabolic systems that are only partially understood. By replacing aspects that are not mechanistically understood with neural networks, such as the reaction mechanisms in Figure 5.1 or gene-to-flux mappings in Figures 5.2-5.4, models can be constructed that predict on par with pure machine learning models while remaining explainable. Furthermore, this approach allows for a deeper understanding of model behavior through numerical simulations and well-developed tools from systems biology [249]. On the other hand, several challenges remain that may prevent the widespread adoption of

UDE-based hybrid models. First, the construction of the mechanistic part still requires a substantial time-investment and biological expertise, albeit less than pure kinetic models. Second, the training of hybrid architectures can be quite slow or may fail completely due to the dependency on numerically solving the system of ODEs, which can especially fail in stiff ODE systems during adjoint computation [258]. Regularization and near-zero initialization can help stabilize training and potentially alternative methods could be considered to improve training (Fig. 5.1C) [78, 261]. Finally, when increasing parts of the model are based on a neural network, dataset size and quality become more important [204]. Therefore, standardized databases with well-documented metadata are of high importance for neural network training [286].

5.5. Author contributions statement

Bálint Magyar: Conceptualization, Methodology, Software, Writing - Review and Editing, Visualization. Paul van Lent: Conceptualization, Methodology, Software, Writing - Original Draft, Visualization. Joep Schmitz: Conceptualization, Writing - Review and Editing, Supervision. Thomas Abeel: Conceptualization, Writing - Review and Editing, Supervision.

5.6. Supporting Information

5.6.1. SI A: Detailed kinetic modeling description

The ODE system for the glucose-only batch model is shown in equations 5.15-5.22. For the kinetic descriptions, we used irreversible Michaelis-Menten kinetics. Further details on the mechanistic model are reported on GitHub.

$$\frac{d[\mathrm{S}]}{dt} = -7\upsilon_{\mathrm{biomass}} - \upsilon_{\mathrm{respiration}} - 2\upsilon_{\mathrm{ppp}} - \upsilon_{\mathrm{glyc}} \tag{5.15}$$

$$\frac{d[\mathrm{BIOMASS}]}{dt} = 40\upsilon_{\mathrm{biomass}} \tag{5.16}$$

$$\frac{d[\mathrm{CO_2}]}{dt} = +2\upsilon_{\mathrm{biomass}} + 6\upsilon_{\mathrm{respiration}} \tag{5.17}$$

$$\frac{d[\mathrm{E4P}]}{dt} = 3\upsilon_{\mathrm{ppp}} - \upsilon_{\mathrm{shiki\_step1}} - \upsilon_{\mathrm{sink\_e4p}} \tag{5.18}$$

$$\frac{d[\mathrm{PEP}]}{dt} = 2\upsilon_{\mathrm{glyc}} - \upsilon_{\mathrm{shiki\_step1}} - \upsilon_{\mathrm{shiki\_step2}} - \upsilon_{\mathrm{sink\_pep}} \tag{5.19}$$

$$\frac{d[\mathrm{DHAP}]}{dt} = \upsilon_{\mathrm{shiki\_step1}} - \upsilon_{\mathrm{shiki\_step2}} \tag{5.20}$$

$$\frac{d[\mathrm{EPSP}]}{dt} = \upsilon_{\mathrm{shiki\_step2}} - \upsilon_{\mathrm{product}} - \upsilon_{\mathrm{sink\_epsp}} \tag{5.21}$$

$$\frac{d[\mathrm{PRODUCT}]}{dt} = \upsilon_{\mathrm{product}} \tag{5.22}$$

For the bioprocess model including the diauxic shift, the ODE system shown above is updated for the following rate laws (eq. 5.23-5.27). The reservoir variable ensures that growth on ethanol ($v_{biomass_{eth}}$) is delayed once glucose is depleted. This introduces a lag-phase behavior that is observed in time-series data.

$$\frac{d[\mathrm{S}]}{dt} = -7v_{\mathrm{biomass}} - v_{\mathrm{respiration}} - 2v_{\mathrm{ppp}} - v_{\mathrm{glyc}} - 2v_{eth} \tag{5.23}$$

$$\frac{d[R]}{dt} = v_{reservoir} \tag{5.24}$$

$$\frac{d[\mathrm{ETH}]}{dt} = +2v_{eth} - 20v_{biomass_{eth}} \tag{5.25}$$

$$\frac{d[\mathrm{BIOMASS}]}{dt} = 40v_{\mathrm{biomass}} + 40v_{biomass_{eth}} \tag{5.26}$$

$$\tag{5.27}$$

5.6.2. SI B: Specific growth rate predictions and gene sensitivity analysis of the glucose-only versus diauxic shift bioprocess model

To evaluate how well the model predictions matched the maximum growth rates derived from Biolector measurements, the time-series data were automatically segmented into a glucose-growth phase and an ethanol-growth phase. Maximum growth rates were subsequently determined exclusively for the glucose-growth phase. The μ_{max} was obtained by converting optical density values to logarithmic space and fitting a linear model over the interval in which biomass accumulation exhibited the steepest increase.

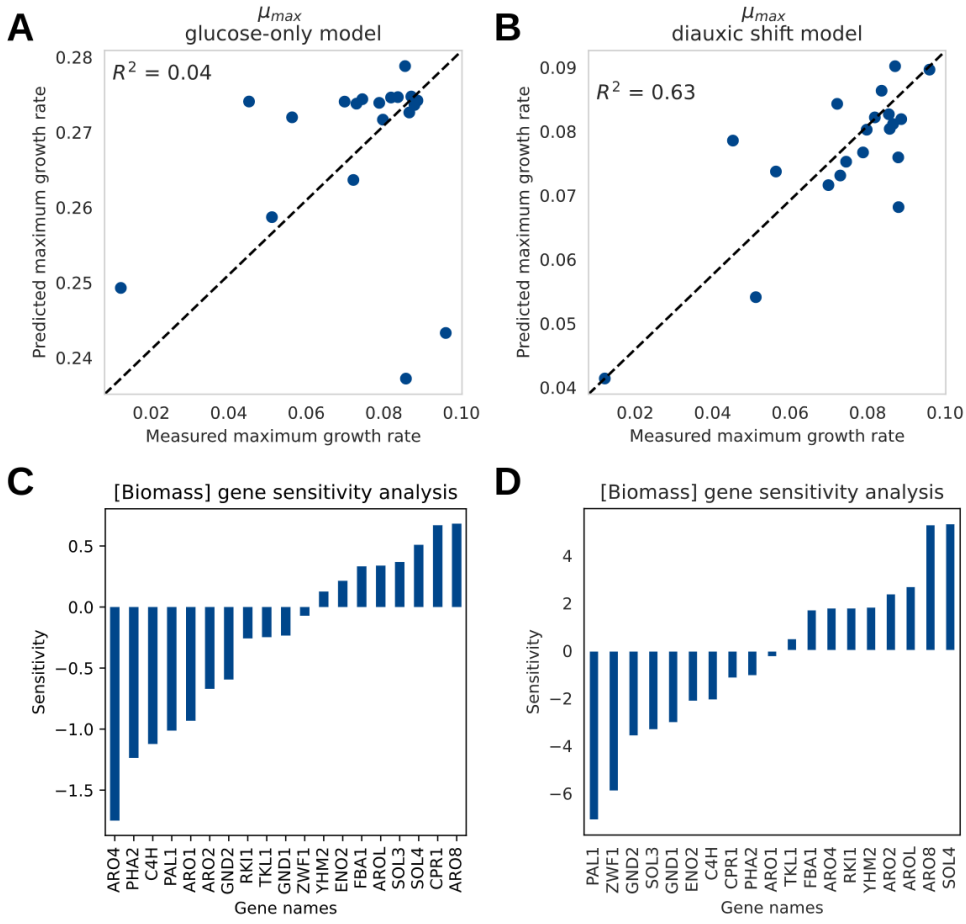


Figure 5.5: **Predicted growth rates by the two models**. **A)** Maximum growth rate prediction for the glucose-only model. **B)** Maximum growth rate prediction by the diauxic shift model. **C)** Gene sensitivity profile of the glucose-only model. **D)** Gene sensitivity profile of the diauxic shift model.

6

A comparative simulation study of machine learning-assisted design-build-test-learn cycles

Paul VAN LENT, Sara MORENO PAZ, Joep SCHMITZ, Thomas ABEEL

Design-Build-Test-Learn (DBTL) cycles are a widely employed engineering framework in metabolic engineering. Nonetheless, their performance depends on a wide range of experimental and algorithmic design choices, whose combined effects on the successful optimization of microbial strains remain an open question. In this study, we performed in-silico DBTL cycles based on metabolic kinetic models to quantitatively assess how key process parameters affect strain optimization outcomes across four distinct metabolic pathway models. This includes parameters governing DNA library design, experimental budget limitations, and machine learning configuration. The results show that screening capacity is a dominant driver of optimization success, whereas DNA sequencing capacity has surprisingly little impact, despite its importance for model training. Selecting top-producing strains for sequencing consistently outperforms stratified sampling, highlighting a trade-off between predictive accuracy and optimization efficiency. DNA library structure strongly affects performance: increasing the number of editable positions generally improves outcomes, while expanding the set of gene targets can hinder optimization due to increased dimensionality or sparse sampling. Together, these findings offer actionable guidance for designing more effective DBTL workflows and underscore the value of simulation frameworks for exploring metabolic engineering strategies prior to experimental implementation.

This chapter has been accepted in Frontiers in Bioengineering and Biotechnology [287]

6.1. Introduction

Metabolic engineering aims to improve microorganisms by genetically modifying their metabolic pathways to boost the synthesis of desired products [288]. This can be used to develop microbial cell factories that facilitate the synthesis of bulk and specialty chemicals using potentially sustainable substrates [289]. However, this development from a proof-of-principle strain to a microbial cell factory is a complicated process, often taking many years of development and significant financial investment [290]. Reducing strain development times is essential for the widespread adoption of bioprocesses for manufacturing.

Advances in high-throughput screening and DNA sequencing technologies have allowed for the targeting of multiple metabolic pathway elements simultaneously for optimization [33]. This approach is known as combinatorial pathway optimization [30]. By targeting many elements in parallel, the likelihood of finding an optimal configuration for the production of a desired compound increases compared to the sequential de-bottlenecking of metabolic pathways. However, as the number of pathway elements targeted increases, the number of theoretical designs combinatorially explodes. To navigate this large landscape of possible designs, the Design-Build-Test-Learn (DBTL) cycle is widely used in metabolic engineering. In the Design phase, a set of designs is chosen based on the selection of targets and experimental design. This is then followed by the Build and Test phase, during which the strains are constructed and screened for their performance. The results from the screening phase are used to guide new designs in the subsequent DBTL cycle (Learn phase). This iterative approach has been shown to be successful in many applications [39, 291]. However, due to the flexibility of the DBTL cycle framework, many open questions remain regarding how DBTL cycle process parameters affect strain optimization performance.

Machine learning (ML) has been increasingly used to propose new strain designs for subsequent DBTL cycles [37, 74, 292]. These ML-assisted approaches are promising for automated strain engineering, as they effectively close the loop between DBTL cycles in a data-driven manner [144]. However, the success of the recommendation strategy is closely linked to the process parameters related to the DBTL cycle. This includes experimental design choices related to DNA sequencing and screening (phenotyping) budget, DNA library design choices [142], and exploration-exploitation tradeoffs for optimization. Often, these design choices are a consequence of experimental budget constraints, lab capabilities, and experience. Consequently, elucidating how ML-assisted optimization performance is affected by these experimental design choices is of high importance to strain engineering.

In silico experiments serve as a versatile tool for evaluating metabolic engineering scenarios that would not be practically feasible in an experimental setting [35, 81, 293]. Previously, we developed simulated DBTL cycles, a framework based on kinetic models to consistently evaluate ML methods across different metabolic engineering scenarios [81]. This benchmarking framework provided insights into supervised ML

methods, data requirements, and budgetary constraints. Whereas numerous studies have examined the machine-learning components of recommendation strategies [52, 74, 144], the current work instead focuses on how the process parameters of the DBTL cycle influence the optimization performance of an ML-assisted strategy. Four hypothetical metabolic pathways of diverse topologies and complexities are implemented in a batch bioprocess kinetic model of *Saccharomyces cerevisiae*. These different pathway models are used as a test suite for comparing various metabolic engineering scenarios. The aim is to quantitatively assess the role of process parameters and their influence on strain optimization, thereby informing experimental design of DBTL cycles in industrial metabolic engineering. We offer code and workflows for user-specific simulated DBTL cycle scenarios at `https://github.com/AbeelLab/BenchmarkProjectDSMF`.

6.2. Materials and methods

6.2.1. Kinetic models

Kinetic models: implementation properties

Four bioprocess models of varying complexity and scale were constructed using the *jaxkineticmodel* Python package [178] and subsequently exported to SBML format [294] for standardized representation and downstream analysis. Each model describes a hypothetical metabolic pathway, integrated in a bioprocess model of *Saccharomyces cerevisiae*, that is designed to satisfy carbon balance across all reactions. Biomass formation reactions were parameterized to yield physiologically reasonable biomass yields of approximately 0.5 gram biomass per gram glucose[295].

The pathway topologies used in the kinetic models were derived from structures observed in real metabolic networks (see Table 6.1). Simplified representations of these network topologies are provided in Fig. Supporting Information [296]. Pathway A is based on the p-coumaric acid pathway [162], while Pathway B is loosely modeled after the glutamate pathway, incorporating a cyclic structure reminiscent of the Krebs cycle [297]. Pathway C draws inspiration from the ornithine pathway and includes acetate formation as a byproduct [298]. Finally, Pathway D is loosely based on the cocaine biosynthetic pathway [299]. It should be noted that not all these pathways are native to *S. cerevisiae*.

To account for the metabolic burden associated with genetic modifications, a protein load term was incorporated [300]. Metabolic burden was represented by an increase in the maximum rate of a non-growth-associated maintenance reaction that depletes ATP required for biomass synthesis. Specifically, the parameter $v_{max}^{maintenance}$ in the maintenance ATP sink reaction (eq. 6.1) was scaled relative to a wild-type reference strain. In the reference condition, the maximum flux is given by v_{max}^{ref}, where all pathway enzyme concentrations $[E_i]$ are normalized to one. A constant C was introduced to ensure that the resulting fraction evaluates to unity under the reference enzyme concentration profile, thereby preserving $v_{max}^{maintenance} = v_{max}^{ref}$ for the wild-type model while enabling systematic variation of metabolic burden in perturbed strains.

$$v_{max}^{maintenance} = v_{max}^{ref} \cdot \frac{(C + \sum_{i=1}^{N} [E_i])}{100} \tag{6.1}$$

Kinetic models: summary statistics

Table 6.1 summarizes the properties of all pathway models, including the numbers of species, reactions, parameters, and optimization targets. The topological complexity was quantified using the approach described in [301] (Fig. Supporting Information).

Table 6.1: **Key statistics of the four pathway models used in this study.** The network complexity is a random walk measure for the stoichiometric matrix *S*. The topology characteristic gives a brief description of the features in the model.

Model	Species	Reactions	Parameters	Enzyme targets	Network complexity	Topology characteristic
Pathway A	23	23	86	20	0.30	linear
Pathway B	18	16	59	12	0.32	cycle
Pathway C	25	21	82	17	0.22	cycle + linear
Pathway D	29	25	98	21	0.61	cycle + linear

6.2.2. Metabolic engineering process

Kinetic models described in the previous section are used as a surrogate model for the simulation of metabolic engineering scenarios. Below, more details are provided on how these scenarios are simulated.

Simulated Design-Build-Test-Learn cycles

The simulation of DBTL cycles was introduced in [81] and subsequently re-implemented using JAX [178, 206]. The number of enzyme targets (F) and promoters (X) for a set of library positions (P) is used to construct a DNA library. This DNA library is represented as a matrix with $F \times X$ rows by P columns, where each element corresponds to a promoter-gene pair. A probability is assigned to each element such that the positions sum to one. Strains are constructed based on random assembly, according to these probabilities, so that a single strain includes P promoter-gene pairs. This approach mirrors the method used in many combinatorial pathway optimization methods involving library transformation [30, 50]. In the first DBTL cycle, all promoter-gene pairs are equiprobable. In subsequent cycles, probabilities are adjusted using an ML-assisted approach (see next section). For all strains that are constructed (S) during the build phase, the designs are numerically simulated using Diffrax with the Kvaerno5 numerical solver [77, 302]. In the test phase, the built strains are simulated as a batch bioprocess without additional glucose feeding. Then, 10% homoscedastic noise is introduced. Finally, these simulated strains serve as training data for a ML model, which will be used in the next DBTL cycle.

Recommending new strains: an ML-assisted approach

To automate the recommendation of new strains for comparing metabolic engineering scenarios across multiple DBTL cycles, we use an ML-assisted recommendation method that outputs a probability distribution for the DNA library that will be used in the next DBTL cycle.

Machine learning model XGBoost is trained on the input strain design data and the simulated values of the product of interest [274]. Default hyperparameters were used, with the exception of number of boosting rounds and early stopping conditions (*num_boosting_rounds*=20, *early_stopping_rounds*=40). Ten-fold cross-validation was performed to estimate model performance using the Pearson correlation coefficient.

Recommending strains using the machine learning model The DNA library was randomly sampled during each DBTL cycle to generate a large candidate set (600,000 variants). This down-sampling is necessary, as the combinatorial design space can be prohibitively large. Samples were then evaluated using the trained XGBoost model. The predicted strain performances were expressed relative to the parent strain and subsequently converted into a probability distribution using the softmax function (eq.

6.2) [53]. In this formulation, the parameter β controls the trade-off between exploration of the DNA library design space and exploitation of the model predictions.

$$p(y_{pred}) = \frac{e^{\beta(y_{pred} - y_{parent})}}{\sum_{i=1}^{N} e^{\beta(y_{pred} - y_{parent})}} \tag{6.2}$$

Strains predicted to outperform the parent strain are assigned higher probabilities than those predicted to underperform. To obtain probabilities for each DNA element in the library, we summed the probabilities from equation 6.2 over all strains containing that element. This procedure yields element-wise (column-wise) probability distributions across the DNA library, as determined by the ML model.

Dynamic balancing exploration-exploitation To automatically balance exploration and exploitation, we selected the softmax temperature parameter β using an entropy-based heuristic. For a given β, we computed the entropy of the softmax distribution (Equation 6.2) according to

$$H(p(y))_\beta = \sum_{i=1}^{N} p(y)_\beta \log(p(y)_\beta), \tag{6.3}$$

where $p(y)_\beta$ denotes the softmax probabilities over the N candidates at temperature β. We evaluated $H(p(y))_\beta$ over a range of temperatures ($\beta \in [0, 100]$) and identified the value of β at which $\mathrm{d}H/\mathrm{d}\beta = 0$. This point corresponds to the steepest decrease in entropy, which can be interpreted as the onset of a transition from a high-entropy (more exploratory) to a low-entropy (more exploitative) regime. The β value at this point was then used to compute the updated DNA library sampling probabilities in each iteration. This procedure was inspired by the need to avoid premature convergence to local optima in Bayesian optimization and reinforcement learning settings [53, 303].

6.2.3. Metabolic engineering scenarios

To understand how the success of the DBTL cycle is impacted by design choices, the key DBTL cycle process parameters that were identified are reported in Table 6.2. Typical ranges for these parameters are shown, but they may vary depending on the specific pathway model (see code availability). For each model, we rank the enzyme importance using sensitivity analysis on the kinetic model[304]. This serves as the ground truth ranking. Ten percent homoscedastic noise is added to the simulated production values.

Two rounds of computational experiments were conducted. In the first round, each individual process variable listed in Table 6.2 was evaluated for pathway model A to identify those that significantly influenced the optimization process. This is performed on one pathway model to reduce the computational load for more elaborate

Table 6.2: Description of process parameters

Symbol	Description	Model A	Model B	Model C	Model D
F	Number of enzyme targets	6-19	6-12	6-17	5-21
X	Number of distinct promoters	2-6	4	4	4
P	Number of positions in library	4-10	4-10	4-10	4-10
S	Number of screened strains	50-1200	50-1200	50-1200	50-1200
N	Number of DNA sequenced strains	50-200	50-200	50-200	50-200
Sampling approach	DNA sequencing selection method	stratified or best sampling	stratified or best sampling	stratified or best sampling	stratified or best sampling
β	Exploration-exploitation parameter	0-100	0-100	0-100	0-100

follow-up experiments. In the second round, informed by these results, a large combinatorial experiment was designed using the subset of variables deemed most likely to affect metabolic flux optimization performance. The variables included the number of enzyme targets (F), the number of screened strains (S), the number of sequenced strains (N), and the number of engineerable positions in the DNA library (P).

For each pathway model, all feasible combinations of these variables were generated, resulting in between 144 and 320 distinct process configurations, depending on the model. For each configuration, five simulated DBTL cycles were executed. Performance was quantified by computing the area under the curve (AUC) of the production of the best built strain over five DBTL cycles and (ii) the median production across all built strains within each cycle. Each configuration was repeated for 10 independent simulation runs to account for stochastic variability. For certain pathway models, specific variable combinations could not be simulated (e.g., due to numerical stiffness or runtime errors); these configurations were excluded from subsequent analysis.

6.2.4. Code availability

Code and results are available on GitHub at *https://github.com/AbeelLab/BenchmarkProjectDSMF*.

6.3. Results

6.3.1. Quantitative comparison of optimization processes reveals key DBTL cycle process parameters

Design-Build-Test-Learn (DBTL) cycles are widely used in metabolic engineering, helping to manage the extensive combinatorial design options of potential strain designs [30]. Despite its utility, the DBTL cycle involves many parameters that could affect the optimization of microbial strains. These include factors related to the experimental budget, experimental design, and ML-related factors. Given the substantial time and resources necessary for metabolic engineering experiments, employing simulated DBTL cycles offers a cost-efficient alternative to real-world experimentation for assessing process parameters in various scenarios [293]. In this research, we employ a previously developed kinetic model-based framework to assess the impact of process parameters on four distinct metabolic pathway optimization tasks [81].

As an example of how simulated DBTL cycles are used to evaluate metabolic engineering strategies, we compared an ML-assisted recommendation approach with a random DNA library transformation baseline [74]. Five rounds of optimization were chosen, as this is a a number of DBTL cycles that could be encountered in real-world settings. In the baseline approach, each DBTL cycle involves constructing a DNA library through random integration of genetic elements into the host strain, after which the best-performing strain is selected and used as the parent for the next iteration. In contrast, the ML-assisted strategy employs a gradient bandit algorithm to iteratively update strain-selection preferences based on predicted productivity [53, 137] (see 3.4), thereby increasing the likelihood of identifying higher-performing parent strains for subsequent DBTL cycles. As expected, the ML-assisted method outperforms the baseline in terms of the average production across the set of constructed strains (Fig. 6.1A). A similar, albeit less pronounced, improvement is observed when considering the maximum-producing strain (Fig. 6.1B). Together, these results show that ML-assisted recommendations can enhance strain performance relative to random DNA library transformation.

Relating to DBTL cycle parameters, the impacts of screening capacity (S), DNA sequencing capacity (N), and screening-to-sequencing strain selection are evaluated in terms of the area under the curve (Fig. 6.1C). Increasing screening capacity consistently improved optimization performance (Fig. 6.1D). This improvement likely occurs because larger screens increase the probability of including top-producing strains in the training data. However, the improvement between screening 200 strains and 300 strains was relatively minor, suggesting that for this pathway, the sampling density with 200 samples was enough to find high-performance strains. Interestingly, increasing DNA sequencing capacity had little effect on optimization performance (Fig. 6.1E), despite the expectation that providing more data to the ML model would enhance predictive accuracy and consequentially a better optimization performance.

We further compared two strain-selection strategies for sequencing: (i) selecting the top producers identified in screening (best sampling) and (ii) selecting a stratified set that spans the observed production range (stratified sampling). These strategies reflect a trade-off when choosing strains for further DNA sequencing: choosing the best screened strains is desirable from a metabolic flux optimization perspective, whereas a stratified approach is more desirable when aiming to reduce training data bias (and thereby predictive performance). From an optimization perspective, it is clear that the best sampling approach led to a notably better optimization performance compared to stratified sampling (Fig. 6.1F). In the case of stratified sampling, there is a chance that the best strain is not used as the new parent strain, which may result in worse optimization performance.

Finally, we considered parameters associated with the DNA library, including the number of positions (P) and the number of gene targets (F). Findings suggest a significant effect of the DNA library positions on optimization performance by enhancing the exploration of the combinatorial design space (Fig. 6.1G). Although increasing the number of library positions might currently be technically challenging for strain engineering, these results underscore the potential benefits of developing synthetic biology tools that enable the

integration of more genetic elements. Regarding the choice of the number of gene targets considered in the DNA library, reducing the number of targets significantly improves the optimization efficiency of ML-assisted recommendation (6.1H). However, this is only the case when the gene targets are chosen carefully, as selecting the wrong gene targets can result in worse performance than if no selection is performed (shown in gray). Consequently, feature selection methods, whether data-driven or grounded in mechanistic understanding, are crucial for improving metabolic engineering [47, 305].

In summary, the results on one metabolic pathway indicate that optimization performance is impacted by screening capacity (S), the strain selection strategy for further DNA sequencing selection, the number of library positions (P), and the number of gene targets (F), whereas the number of sequenced strains (N) does not appear to impact performance. Effects of other process parameters, including promoter strengths, exploration-exploitation trade-offs in ML-assisted recommendation, and the number of promoters included per gene, are reported in Supporting Information.

6.3.2. Screening capacity, not sequencing capacity, is a key driver of optimization performance

Although the first section identified some important process parameters, this was only performed on one metabolic pathway optimization problem. We therefore further substantiate our analysis of these identified parameters to include three additional metabolic pathways with varying complexities (see Methods).

Results found on pathway model A showed that increasing screening capacity greatly improves optimization, whereas increasing DNA sequencing capacity does not. To validate this finding, a grid search is performed for these two parameters. Figure 6.2 presents four heatmaps for the different metabolic pathway models. The screening ratio indicates the down-sampling factor after screening. For instance, in a scenario involving 200 DNA-sequenced strains, a screening ratio of six leads to 1200 simulated strains. Across all models, an increase in the screening ratio correlates with enhanced AUC performance. This reinforces the result that the number of DNA-sequenced strains appears to have minimal effect on performance (Fig. 6.1E). While the overall performance differs between metabolic pathways, the independence of performance from DNA sequencing capabilities is generally observed. This is somewhat unexpected since the ML model is exclusively trained on this selection. It is often assumed that a larger training dataset would yield a superior model. In terms of optimization, the emphasis lies in identifying high-producing strains, which becomes more feasible with enhanced screening capabilities. In industrial metabolic engineering, the screening throughput of workflows may therefore represent a more significant bottleneck than DNA sequencing capacity.

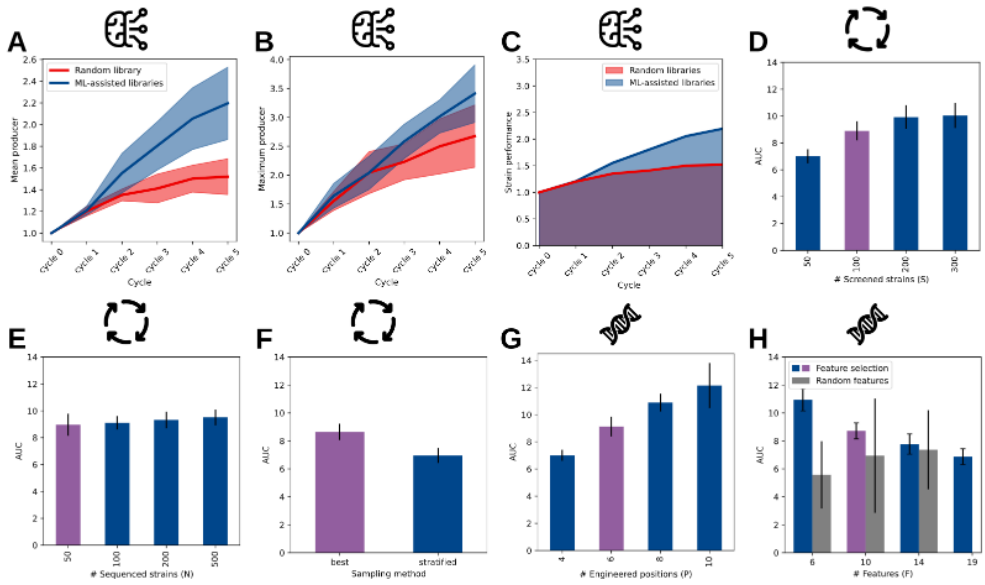


Figure 6.1: **Evaluating process variables for pathway model A.** In plots D-H, the purple bar represents the baseline scenario ($S = 100, N = 50, F = 10, P = 6, X = 4, \beta = 10$). Three categories of process parameters are illustrated with pictograms: parameters related to ML, those related to the DBTL cycle, and parameters concerning library design. **A)** A comparison between a random library optimization approach and a strategy utilizing ML-assisted recommendations. The y-axis indicates the average production of the recommended strain set relative to the parent strain's reference production. **B)** A similar graph, but depicting the top producer among the recommended strains. **C)** To assess scenarios, optimization performance is measured using the area under the curve (AUC). **D)** A comparison of approaches for selecting strains from screening results for DNA sequencing. The purple bar denotes the baseline scenario maintained across all plots. **E)** Optimization performance with varying DNA sequencing capacities. **F)** Sample selection strategies for screening: choosing the top producers (best sampling) versus a representative set (stratified sampling). **G)** The number of DNA library positions that can be integrated into the host strain's genome. **H)** The impact of feature selection on optimization performance, with grey bars showing random feature selection.

6.3.3. Sampling top producers for screenings outperforms stratified approaches for metabolic flux optimization

Regression models in ML are susceptible to data imbalances stemming from selection biases, as noted in studies [151, 306, 307]. In the field of metabolic engineering, these biases are often introduced when selecting which strains to sequence for further

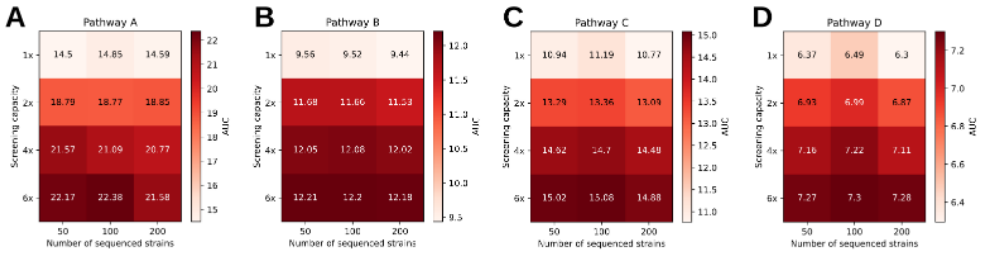


Figure 6.2: **Screening capacity has a strong impact on ML-assisted optimization performance. A-D)** Performance heatmaps that compare the effect of screening capacity (y-axis) and number of DNA sequence strains (x-axis). For pathway model A-C, six gene targets were considered ($F = 6$), whereas pathway model D consisted of five targeted genes ($F = 5$).

study. Ideally, from a metabolic flux optimization standpoint, one would select the highest-performing strains, even though they may not represent the underlying genotypic distribution of the DNA library well. To assess the implications of this selection, we compared the best sampling strategy (selecting top producers) with a stratified sampling method (see Methods).

Across all pathway models, the selection of top producers consistently yields a higher AUC compared to using the stratified method (Fig. 6.3A-D). Notably, in metabolic pathway model D, after completing two cycles, the performance of the strain actually diminishes (Fig. 6.3D). This decline is attributable to the protein load effect accounted for in the kinetic models; as additional enzyme copies are included, biomass flux decreases due to increased maintenance (Fig. Supporting Information). Nevertheless, selecting top producers in all DBTL cycles proves more effective than the stratified approach. Conversely, the performance of the ML model shows a slightly different trend. In pathway models A and B, the Pearson correlation coefficient on cross-validation sets is actually lower when using the best sampling method compared to the stratified method (Fig. 6.3E,F), which aligns with expectations for a biased distribution [151]. In contrast, for pathway models C and D, no differences are observed between the two selection methods (Fig. 6.3G,H), potentially because the chosen strains more accurately reflect the underlying distribution.

To summarize, these results highlight a common trade-off in metabolic engineering. While a stratified selection approach improves the predictive accuracy of the XGBoost model, a bias towards selecting high-producing strains can result in better overall performance in strain optimization. In terms of metabolic flux optimization, sampling the best strains outperforms the stratified approach and should be preferred. However, alternative selection strategies that consider both aspects may help to mitigate this trade-off.

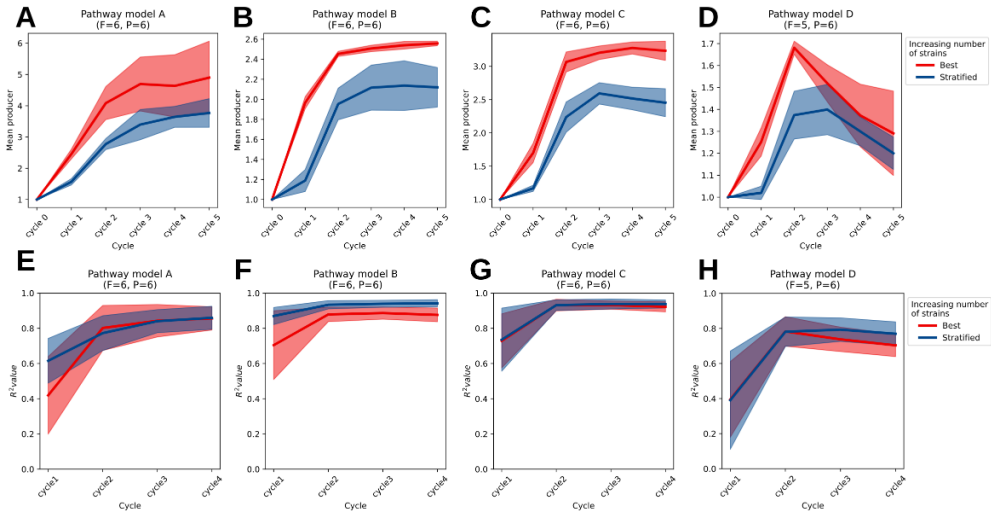


Figure 6.3: **Selection method of strains for DNA sequencing A-D)** The product optimization result for pathway models A-D. The stratified approach selects strains uniformly across the range of production values, while the best sampling approach selects the top producers (see Methods). In general, product optimization favors a best sampling approach. **E-H)** The Pearson correlation coefficient R^2 of the XGBoost model for the same optimization process. A higher performance is typically observed for the stratified sampling approach.

6.3.4. DNA library parameters have mixed impact on strain optimization performance

In figure 6.1G and H, it was noted that parameters concerning the DNA library design have a considerable impact on performance. For a DNA library composed of a set number of positions (P), each capable of accommodating different promoter-gene-terminator combinations ($X \times F$), the theoretical space of combinatorial designs is represented by $(F \times X)^P$ strains. Although enlarging the DNA library size can enhance genetic variation among strains, it also results in a combinatorial explosion of the design space [30]. In this context, we aim to understand the effect of the number of included gene targets (F) and the number of positions (P) for the four pathway models.

We first evaluated how the number of gene targets in the library influenced performance across the four pathway models (Fig. 6.4A). Increasing the number of gene targets negatively affected performance for models A and C, whereas models B and D were mostly unaffected. This mixed effect might be explained by two reasons. First, as the dimensionality of the design space increases, the associated optimization problem becomes more challenging. Problems defined over larger gene sets are inherently more

difficult to solve, as can be observed in models A and C [155, 308]. Second, because each strain contains only six genes ($P = 6$), adding more candidate gene targets beyond this limit leads to sparser coverage of the combinatorial design space. This results in a less informative training dataset.

The effect of increasing the number of library positions is shown in Figure 6.4B. For pathway models A and C, enhancing the number of library positions leads to a notable improvement in the AUC. Conversely, for pathway model B, such an increase does not result in any noticeable effect, indicating that only a limited number of genes may be crucial for performance enhancement. Interestingly, for pathway model D, the increased number of positions has a negative effect. This is likely due to the more prominent effect of protein load in pathway model D compared to the other pathway models (see Fig. Supporting Information). Introducing more gene copies reduces biomass production, thus hindering performance. This shows that while an expansion of engineering capabilities is potentially beneficial, it may depend on the specific pathway to be optimized.

The design of libraries is crucial in combinatorial pathway optimization [30]. Proposed strategies for DNA library design aim to balance genetic variation while preventing combinatorial explosion [142]. Here, we demonstrate that choosing appropriate genes and limiting the number of genes considered can enhance performance. Furthermore, the expansion of the number of DNA positions affects the area under the specific problem being addressed.

6.4. Discussion

The DBTL cycle is a flexible and widely adopted framework for navigating the landscape of strain design options in metabolic flux optimization. In this study, we investigate how optimization performance depends on parameters associated with DNA library design and the DBTL cycle when using ML-assisted recommendation methods [50, 51, 74]. To accomplish this, we employed simulated DBTL cycles, which offer a cost-effective alternative to real-world experimentation [81].

Our analysis began by identifying seven key parameters related to the DBTL cycle (see 6.2, specifically in the Design phase (number of features and positions), the Build and Test phase (screening capacity, sequencing capacity, and sampling approach), and the Learn phase (the exploration-exploitation trade-off). These were tested on one pathway model to determine which parameters had a significant effect on optimization performance and should be further investigated (see Fig. 6.1 and Fig. Supporting Information). This showed that screening capacity, the sampling approach for DNA sequencing, the number of positions in the library, and the number of features had the most prominent effect on performance.

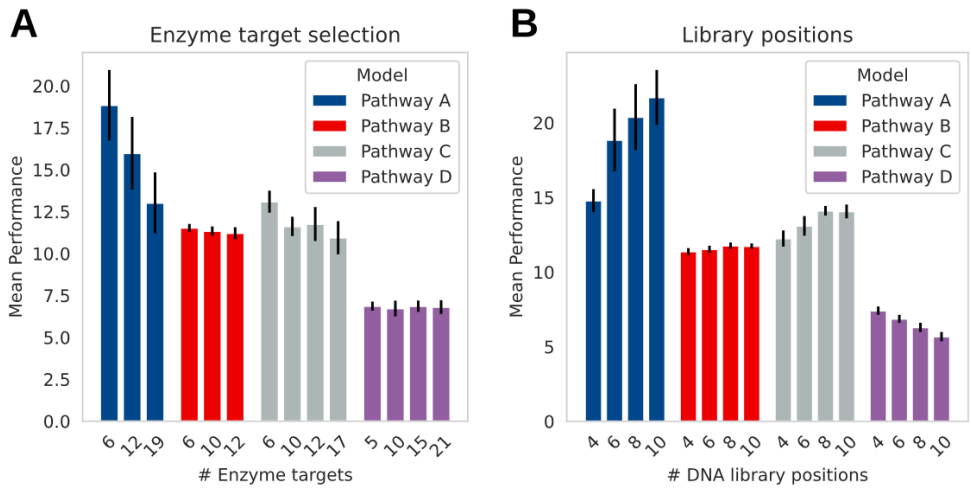


Figure 6.4: **Library design: effect of feature selection and number of positions on optimization performance. A)** Number of enzyme targets utilized for optimization in each pathway model. Enzyme targets were selected based on sensitivity analysis of the kinetic model [304]. In general, fewer enzyme targets are associated with enhanced optimization performance. **B)** The number of DNA library positions that can be incorporated into a single strain. Typically, increasing the number of positions leads to improved performance, although it depends on the specific optimization issue (see Discussion).

An interesting finding when expanding the analysis to multiple pathways is that screening capacity, but not DNA sequencing capacity, was a major driver of optimization performance (Fig. 6.2). Although increasing the number of samples would be expected to improve predictive accuracy, this does not necessarily translate into improved recommendations. Here, we distinguish between predictive performance, defined as the model's ability to accurately estimate outcomes across the design space, and recommendation performance, which reflects the model's ability to identify and prioritize high-performing candidates. We hypothesize that this discrepancy arises because increased screening capacity raises the likelihood of observing high-performing strains. As a result, the training data become enriched in regions of the design space most relevant for optimization, enabling the model to better distinguish and prioritize top-performing designs, even if overall predictive accuracy improves only modestly. This may also explain why sampling the best strains for DNA sequencing is a better strategy than the stratified approach, as the ML model observes more high-performing strains in this scenario. In this regard, expanding screening capacity can enhance the predictive performance of ML models, as it enlarges the domain and range of data on which the model is trained.

For gene target selection, enzyme sensitivities with respect to the product were employed to rank and select the most important enzymes (see Methods). In reality, performing gene target selection can be difficult, as it is not *a priori* known which genes affect production. This can have substantial consequences on metabolic engineering success (Fig. 6.1H). To address this, genome-scale modeling approaches could be used to choose gene targets [44, 305]. Expanding this analysis to other pathways indeed verifies that performance is positively impacted by feature selection, although the impact may differ between products (Fig. 6.4A).

For the number of library positions, we found that for pathway model D, the optimization performance decreases, in contrast to the other models (Fig. 6.4B). This is likely an attribute of the network topology of pathway model D, which had a byproduct route that might be favored over the product route (see Supporting Information). Furthermore, it could also be a consequence of performing library transformation. As more gene copies are added, the protein load increasingly affects biomass formation, resulting in decreased production. This can also be observed in Figure 6.3D, where production drops after three DBTL cycles.

In the present study, our optimization process focused on the addition of individual gene copies to a host strain. While this approach is straightforward and experimentally tractable, it does not capture the full range of possible metabolic engineering interventions. In particular, strategies involving gene down-regulation or knockouts could provide valuable alternatives, as demonstrated by approaches such as OptKnock [55], especially in systems where metabolic burden or metabolic regulation principles are a limiting factor. Incorporating such interventions would allow for a more comprehensive exploration of the design space. However, implementing down-regulation typically requires the replacement or modification of native promoters, which introduces additional experimental complexity compared to gene over-expression. For this reason, these strategies were not included in the present study. Furthermore, although our work focused on yeast, other host organisms may exhibit polycistronic gene organization, which could influence optimization outcomes. Finally, although the implemented pathways were topologically diverse, metabolic regulation principles such as product inhibition were not included. While these scenarios fall outside the scope of this study, they could be addressed in future work using similar modeling approaches.

Balancing exploration and exploitation in machine-learning-driven recommendations within the DBTL cycle is known to be important [38, 137, 309]. In Figure 6.1, we fixed β to a single value to enable consistent comparisons between scenarios. In practice, the preferred exploration–exploitation balance may shift over the course of a combinatorial pathway optimization experiment. For instance, one might prioritize exploration in the early DBTL cycles and then gradually increase exploitation in later cycles. To avoid searching for this exploration-exploitation schedule, we applied an entropy-based approach to adaptively manage this trade-off (see Methods). We observed that this strategy resulted only in a minor improvement over a static approach (Fig. 6.4B). Nonetheless, alternative strategies for controlling exploration versus exploitation are

conceivable and may be worth exploring in future work.

Overall, our results highlight practical considerations that can be used to improve the DBTL cycle for real-world metabolic engineering. As the variety of metabolic engineering scenarios that can be considered is abundant, workable templates for running alternative scenarios of the DBTL cycle are provided.

Supporting Information

Additional data, results, and supplementary material referred to in the main text can be found in the publication (see QR-code).

- **SI 1**: This text contains pathway visualizations made with Fluxer [296].
- **SI 2**: This text contains validation plots and pathway descriptions of the four models used in this study.
- **SI 3**: This text contains plots of process parameters from Table 6.2 that are not reported in the main results. These include the results of the exploration-exploitation parameter β, the number of promoters X.
- **SI 4**: This contains results on the network complexity of a set of metabolic networks.

Figure 6.5:

7

Discussion

Metabolic engineering has shown great potential to transform microorganisms into efficient microbial cell factories, expanding the range of chemicals that can be produced in a bio-based manner [249]. Advances in synthetic biology tools, such as strain construction, high-throughput screening, and DNA sequencing, are rapidly expanding the scale and complexity of available datasets. Machine learning methods offer an attractive and flexible solution for dealing with the opportunities and challenges of big data. These advantages include the superior predictive performance of supervised methods and machine learning-assisted recommendation strategies that navigate large combinatorial design spaces [37]. This may result in automating the Design-Build-Test-Learn cycle used in iterative metabolic engineering [99], thus reducing the development times for the construction of microbial cell factories.

This thesis examined the application of machine learning in two key aspects of the DBTL cycle: predicting strain design outcomes and recommending new designs for follow-up cycles. We developed a kinetic model-based framework (*simulated*) to compare metabolic engineering scenarios and machine learning algorithms. Based on this framework, it was shown that ensemble methods such as Random Forest and XGBoost outperform other models in low-data regimes typical for metabolic engineering. Furthermore, under a fixed experimental budget, a strategy involving one large initial round of strain construction followed by a smaller second round yielded a performance similar to that of performing multiple DBTL cycles (chapter 2). Based on insights from chapter 2, we hypothesized that larger combinatorial design spaces could be explored than were typically encountered in the metabolic engineering literature (e.g., six genes and a few promoters). We designed a large DNA library (19 genes, 20 promoters; 170 million configurations) to optimize p-Coumaric acid production. With a well-balanced exploration–exploitation strategy, machine learning-based recommendations outperformed a greedy approach, increasing p-Coumaric acid production 2.37 times over the parent strain (chapter 3). To improve interpretability while preserving predictive accuracy, we then integrated mechanistic models with machine learning, developing the systems biology tool *jaxkineticmodel* (chapter 4) and hybrid neural–mechanistic architectures (chapter 5). Incorporating mass-conservation via the stoichiometric matrix S improved generalization relative to

Neural ODEs. When the mechanistic model qualitatively captured system dynamics, the hybrid model demonstrated performance on par with traditional machine learning models while increasing model explainability. Finally, simulated DBTL cycles were used for a large comparative study of the effect of the DBTL cycle parameters on the performance of machine learning-assisted recommendation (chapter 6). This revealed that the screening capacity and gene target selection positively impact DBTL optimization performance.

The results of the chapters show that machine learning can be successfully applied and is likely to decrease the strain development time. However, this research also opens up new challenges and questions that will be discussed in the following sections.

7.1. What to expect from a metabolic model?

Metabolic models are mathematical descriptions that help predict and quantify the behavior of metabolism. Models that capture increasingly complete descriptions of metabolism are promising [73], but the focus on completeness could distract from the effective modeling of core phenomenological behavior. Perhaps paradoxically, this exactitude potentially results in reduced interpretability and explainability [310]. For example, while the rewiring of the protein machinery during a diauxic shift from glucose to ethanol growth in *Saccharomyces cerevisiae* is highly complicated [269], this can be effectively modeled using a few equations [270]. The natural question that arises in this context is: what are the minimal requirements for a metabolic model in the context of strain optimization that captures the essential behavior necessary for practical use while remaining as simple as possible?

In metabolic engineering, there are roughly two model types that are used for different purposes. Kinetic models are used to gain a quantitative and mechanistic understanding of bioprocesses, which have a variety of applications in strain development. Applications include modeling metabolic pathway dynamics [311], optimizing bioprocess feeding strategies [43], and simulating metabolic engineering scenarios [35]. Supervised machine learning models are used for prediction tasks. This includes predicting metabolite production [38], machine learning-assisted protein evolution [75, 312], and enzyme function prediction [313]. In this context, some remarks should be made about what is important in terms of predictive capability, generalizability, explainability, and data requirements.

7.1.1. Predictive performance

In metabolic engineering, two types of prediction tasks can be distinguished. First, there are strain predictions within a combinatorial design space that can be accessed from a given parent strain. We refer to these as *interpolation* scenarios because the target strains lie within the distribution of the training data used to fit the model. However, it is also very common to switch to a different chassis strain for further optimization. Strain

designs constructed on top of this new chassis may then fall outside the original training distribution, which constitutes an *extrapolation* scenario. This distinction is relevant, as machine learning models are often more accurate in *interpolation* settings but less so in *extrapolation* scenarios [314].

In chapter 3, we reported findings on predictive performance for different parent strains and scenarios. We found that although the Pearson correlation decreased when the original parent strain was changed (*interpolation* domain) to a different parent strain (*extrapolation* domain), the relative ranking of strains by production level remained unchanged. Furthermore, we observed that with increasing exploitative-ness of the scenario, the predictive performance of the model decreased. Because the strains showed reduced variability in p-Coumaric acid production in exploitation scenarios, the trained model achieved better predictive performance in settings where the exploration of the pathway designs was balanced with the exploitation of the model predictions. Based on these results, we emphasize two points. First, ranking strains based on performance might be just as valuable as absolute predictions for automated recommendation purposes. Second, by balancing exploration and exploitation for new parent strains that effectively result in predictions in the *extrapolation* domain, supervised machine learning models can still be valuable for recommendation, despite not being trained within that domain.

7.1.2. Generalizability in the fermentation context

Fermentation processes come in many shapes and forms, encompassing various operation modes (batch, fed-batch, continuous) [19], media compositions [21], feed-stocks [15], and bioreactor geometries, among other factors. In this work, we only considered the internal metabolism of microbial strains. Models that can incorporate differing conditions are highly useful in bioprocess development. Although supervised machine learning models could, in theory, incorporate fermentation modes directly as input features, their predictions will likely not transfer well to operating conditions not seen during training. For instance, a model fitted only on strains from batch and continuous fermentations will probably perform poorly when asked to predict fed-batch behavior. Kinetic models are generally better suited to represent such process conditions, but they are often difficult to develop because they require extensive and detailed mechanistic understanding. Hybrid approaches, where known parts and bioprocess context are modeled using mechanistic rate equations and unknown parts are modeled using neural networks, can be a promising middle-ground [76, 77]. The idea is that this hybrid approach increases generalization to unseen conditions while maintaining strong predictive capabilities. In chapter 5, we describe the implementation of several hybrid modeling architectures in the context of fitting time-series data. These results showed that combining neural networks with kinetic models can indeed improve time-series prediction over a full Neural ODE approach [315]. Although challenges in training these models remain (e.g., ODE stiffness, exploding/vanishing gradients, and numerical instability) [261, 272], we expect that these hybrid approaches are a promising way of modeling partially understood systems where the fermentation operation context

matters.

7.1.3. Interpretability and explainability

Interpretability and explainability are two important topics in artificial intelligence, with major consequences for the adoption of machine learning models [46, 316]. While the interpretability of models focuses on whether the internal workings of a model are transparent, explainability focuses on providing justification for the output of complicated models. This essentially means that interpretability focuses on technical aspects, while explainability is focused on the human experience. Black-box machine learning models can therefore be interpretable but not explainable, despite the presence of explainable AI tools [176, 317]. An explanation that suffices depends not only on the explanatory mechanism but also on the user.

In the context of metabolic engineering, the models used can provide explanations by showing which genes are important for increased production. These results are tangible and can be experimentally verified.

The other type of explainability can be offered in terms of integrating kinetic modeling and machine learning, as this not only provides a prediction for a target variable but also gives insight into how the dynamic behavior of (intermediate) metabolites is altered for the genetically engineered strain. This can be used to verify whether the dynamic behavior of engineered strains is physiologically plausible. Models that do not describe realistic metabolic behavior can be improved iteratively using additional experiments. Therefore, hybrid models can generate a broader range of hypotheses than purely supervised machine learning approaches. This can be important in the context of guiding the data acquisition for follow-up experiments, as well as potentially allowing for transfer learning from one optimization process to another [318].

7.1.4. Data requirements

In chapter 3, we generated a combinatorial pathway optimization dataset, with processed DNA sequencing information of strains as the features and the screened p-Coumaric acid production as the target variable. This genomic dataset is inherently cross-sectional since a strain's genetic makeup is stable over time. Many other omics modalities are likewise collected as cross-sectional data, but they can also be measured as time-series. In chapter 4, we focused on developing software (*jaxkineticmodel*) for parameterizing kinetic models using time-series data. In chapter 5, we follow up on this by testing several hybrid model architectures on time-series data. Since many datasets are cross-sectional, we also investigated how this data type can be included in hybrid architectures. This shows that hybrid modeling architectures are versatile tools for the integration of distinct data modalities.

Even though we have not widely used publicly accessible databases for most of the work in this thesis, the effective integration of publicly available metabolic engineering data is an interesting practical problem. Addressing this problem could substantially improve the development of (hybrid) kinetic models and machine learning models. Although there are many databases available [165, 286, 319, 320], integrating distinct data modalities with varying contexts is an issue that cannot easily be resolved. Experimental conditions are not always well-reported, and there is a large discrepancy between *in vivo* and *in vitro* measurements for the kinetic parameters [321]. Reconciling these practical challenges in data integration is an interesting open problem. High-quality data for biotechnology could result in interesting scientific breakthroughs using machine learning [322].

7.2. Automated recommendation for strain engineering

Automated strain recommendation combined with autonomous labs may allow for fully-automated DBTL cycles, reducing development times and costs [99]. In chapter 3, we developed an XGBoost model-based gradient bandit recommendation algorithm that can be used to recommend strain designs by balancing exploration-exploitation tradeoffs [53]. This allows for the recommendation of specific designs, but it could also be used in library design (chapter 6). In the latter case, the algorithm is highly similar to active learning-assisted directed evolution that has been proposed in protein engineering [75, 292]. However, there are also alternative methods for strain recommendation that were not compared to the method proposed in this work. Future work in the recommendation of strains could focus on benchmarking different methods. In the following, we briefly discuss the types of algorithms that could be interesting to benchmark rigorously.

7.2.1. Bayesian optimization

Bayesian optimization refers to the optimization of a black-box function and is widely used for optimization problems where there is a well-defined search space [155]. In typical Bayesian optimization problems, the objective function is considered to be independent of the state of the system. This means the parent strain in which the designs are integrated does not change during optimization. However, Bayesian optimization approaches that take into account state dependence could be considered for comparison [323].

7.2.2. Reinforcement learning

Reinforcement learning algorithms aim to find a policy that makes decisions to maximize the cumulative reward. The theory of reinforcement learning revolves around Markov decision processes, which were briefly introduced in the introduction and chapter 3 [53]. Although the Markov decision process is a natural mathematical framework for formally describing strain engineering, reinforcement learning algorithms have not been extensively used to date [52]. This is because reinforcement learning requires that

rewards are cheap to evaluate. This is not the case in metabolic engineering, where experiments can take several weeks. Furthermore, reinforcement learning is more suitable for maximizing rewards in a sequential decision making approach. Due to the length of a typical fermentation, this is not ideal when strain development times remain a major bottleneck. However, a recent paper suggests a reinforcement learning algorithm that overcomes these two main limitations [74].

7.3. An automated scientist for metabolic engineering?

The chapters discussed in this thesis study the use of machine learning for iterative metabolic engineering.We show that, for several aspects related to the DBTL cycle, machine learning methods can be used to extract knowledge from large datasets and to recommend new strains. In principle, this closes the DBTL cycle and paves the way for automated strain optimization [99]. This heavily relates to biofoundries, which are specialized facilities that aim to streamline the development of biological systems using high-throughput and automation technologies [324–326]. Through automation and the increasing throughput of building, screening, and DNA sequencing of strains, larger design spaces can be considered. It has been suggested that microbial cell factory development could be completely automated, removing the scientist entirely from the process [327]. Although more and more experimental pipelines are expected to be automated, it is unlikely that scientists will be replaced by robots in the near future.

7.3.1. Feature selection

In chapter 6, we observed that the selection of gene targets has a prominent impact on optimization performance by reducing the dimensionality of the combinatorial design space. Furthermore, it may be necessary to consider alternative genes during optimization, as the current design space is not expressive enough to achieve microbial cell factory status. In most of the current biofoundry setups, the initial design space is specified, and then the experiments are performed under these conditions. For the choice of alternative genes, it is conceivable that artificial intelligence can provide insights into the selection of new gene targets. However, the scientist still has an important influence on the experimental prerequisites and on any further modifications.

7.3.2. Sanity checking of algorithmic choices

In chapter 3, after building a machine learning model that predicts the production of p-Coumaric acid, we used this model to design a set of scenarios with increasing exploitation of model predictions. For the last scenario, we initially set the exploitation level so high that the predicted strains were all very similar. The experimentalist responsible for building strains noted that if one chooses this option, there is a high chance of failing to build all strain designs or of constructing a poorly performing strain if the model was not as reliable as validated. Therefore, we changed the

exploration-exploitation scenario to incorporate slightly more exploration than initially intended. This is a simple example of the importance of experienced scientists being involved in the decision-making loop, as this would likely not have been prevented in a fully automated setting.

Reward functions and outside-the-box experimentation

Automated systems define success through a reward function that captures the goals of the problem at hand. In strain optimization, this typically includes the titer, rate, and yield values of the product [20]. It therefore requires that the aspect to be optimized be captured in this reward function. Sometimes, it might be interesting to perform experiments on the biological system being studied that are not directly related to this goal but are valuable for exploring fundamental behavior. These types of experiments require experimental designs that are unlikely to be proposed by any automated technology. Therefore, the role of the scientist will involve not only correcting automated workflows but also learning more about the strain to be optimized, driven by a fundamental interest in the inner workings of the biological system.

7.4. Ethical considerations

Since its beginnings, public and political debate has strongly shaped discussions on biotechnology and synthetic biology, particularly concerning Genetically Modified Organisms (GMOs) [4]. With the emergence of CRISPR-Cas gene-editing technologies [26], these debates have gained urgency, as the technique allows for precise modifications of DNA. In parallel, concerns about Artificial Intelligence (AI) are receiving increasing attention as its negative impacts become more apparent [328–330]. Since this work explores the use of machine learning in metabolic engineering, it is essential to briefly consider the related ethical questions, public perceptions, and political aspects.

Assessing the ethical implications of implementing a particular technology is not straightforward and depends on the specific context. One must consider the *intended* purpose. For instance, biotechnology can be used to reduce malnutrition and improve climate resilience through GMOs [331], but it can also be used for dubious practices akin to eugenics [332]. The *consequences* of new biotechnology must also be weighed [333]. Within the scope of this thesis, the applied machine learning methods target a well-established procedure—metabolic flux optimization—that has been used for many years. The question of whether synthetic biology as a whole is ethical is left as an exercise for the reader [334]. If one concludes that it is ethical, then using machine learning to accelerate this process is not expected to be intrinsically problematic.

Ethical considerations are also reflected in public opinion. Some objections are superstitious, based on misinformation, or due to a lack of education. There are also reasonable concerns: the erosion of boundaries between what is *natural* and *artificial*, worries about ecological impact and safety, mistrust of large corporations, and emotional

appeals [4]. Because many of these concerns can be reasonable, a transparent public debate about which applications of biotechnology and machine learning are desirable is necessary [334]. In this context, science (and scientists) can inform the public regarding what is factually true and technically feasible, but it cannot offer insights into what is ethically justifiable [335].

7.5. Final remarks

Machine learning already has a broad impact on the field of biotechnology, particularly in the design and optimization of microbial cell factories. As this will decrease development times, machine learning will become an enabling technology to partially substitute traditional petrochemical-based production methods with bio-based alternatives derived from renewable feedstock. This is one of the prerequisites for a transformation of the chemical industry towards sustainability [10, 336]. Although this thesis used a specific experimental application and several kinetic model-based simulation studies, we expect the presented results to hold for other metabolic engineering optimization targets.

Bibliography

[1] L. Pasteur. *Studies on fermentation: the diseases of beer, their causes, and the means of preventing them*. Macmillan & Company, 1879.

[2] A. van Leeuwenhoek. *Alle de brieven. Deel 1: 1673-1676, Anthoni van Leeuwenhoek - DBNL — dbnl.org*. https://www.dbnl.org/tekst/leeu027alle01_01/index.php. [Accessed 03-09-2025].

[3] *Bier, Geschiedenis van de techniek in Nederland. De wording van een moderne samenleving 1800-1890. Deel I - DBNL — dbnl.org*. https://www.dbnl.org/tekst/lint011gesc01_01/lint011gesc01_01_0009.php#551T. [Accessed 13-10-2025].

[4] R. Bud. *The uses of life: a history of biotechnology*. Cambridge University Press, 1994.

[5] S. Y. Lee, J. H. Park, S. H. Jang, L. K. Nielsen, J. Kim and K. S. Jung. 'Fermentative butanol production by Clostridia'. In: *Biotechnology and bioengineering* 101.2 (2008), pp. 209–228.

[6] A. Fleming. 'On the antibacterial action of cultures of a penicillium, with special reference to their use in the isolation of B. influenzae'. In: *British journal of experimental pathology* 10.3 (1929), p. 226.

[7] J. D. Watson and F. H. Crick. 'The structure of DNA'. In: *Cold Spring Harbor symposia on quantitative biology*. Vol. 18. Cold Spring Harbor Laboratory Press. 1953, pp. 123–131.

[8] J. E. Bailey. 'Toward a science of metabolic engineering'. In: *Science* 252.5013 (1991), pp. 1668–1675.

[9] J. Spekreijse, T. Lammens, C. Parisi, T. Ronzon, M. Vis *et al.* 'Insights into the European market for bio-based chemicals'. In: *Analysis based on ten key product categories* 1831-9424 (2019).

[10] J. A. Tickner, K. Geiser and S. Baima. 'Transitioning the chemical industry: elements of a roadmap toward sustainable chemicals and materials'. In: *Environment: Science and Policy for Sustainable Development* 64.2 (2022), pp. 22–36.

[11] E. Foote. 'Circumstances affecting the heat of the sun's rays.[Paper read on her behalf.]' In: *American Association for the Advancement of Science, August* 23 (1856).

[12] *Climate Change 2022: Impacts, Adaptation and Vulnerability — ipcc.ch*. https://www.ipcc.ch/report/ar6/wg2/. [Accessed 16-07-2025].

[13] *unfccc.int*. https://unfccc.int/process-and-meetings/the-paris-agreement. [Accessed 16-07-2025]. 2015.

[14] D. Petrides *et al.* 'Bioprocess design and economics'. In: *Bioseparations science and engineering* (2000), pp. 1–83.

[15] J. A. Lee, H. U. Kim, J.-G. Na, Y.-S. Ko, J. S. Cho and S. Y. Lee. 'Factors affecting the competitiveness of bacterial fermentation'. In: *Trends in Biotechnology* 41.6 (2023), pp. 798–816.

[16] B. Karlson, C. Bellavitis and N. France. 'Commercializing LanzaTech, from waste to fuel: An effectuation case'. In: *Journal of Management & Organization* 27.1 (2021), pp. 175–196.

[17] H. Narayanan, J. A. Hinckley, R. Barry, B. Dang, L. A. Wolffe, A. Atari, Y.-Y. Tseng and J. C. Love. 'Accelerating cell culture media development using Bayesian optimization-based iterative experimental design'. In: *Nature Communications* 16.1 (2025), p. 6055.

[18] E. Franco-Lara and D. Weuster-Botz. 'Estimation of optimal feeding strategies for fed-batch bioprocesses'. In: *Bioprocess and Biosystems Engineering* 27.4 (2005), pp. 255–262.

[19] Y. Yang and M. Sha. 'A beginner's guide to bioprocess modes–batch, fed-batch, and continuous fermentation'. In: *Enfield, CT: Eppendorf Inc* 408 (2019), pp. 1–16.

[20] O. Konzock and J. Nielsen. 'TRYing to evaluate production costs in microbial biotechnology'. In: *Trends in biotechnology* 42.11 (2024), pp. 1339–1347.

[21] S. Moreno-Paz, R. van Der Hoek, E. Eliana, V. A. Martins dos Santos, J. Schmitz and M. Suarez-Diez. 'Combinatorial optimization of pathway, process and media for the production of p-coumaric acid by Saccharomyces cerevisiae'. In: *Microbial Biotechnology* 17.3 (2024), e14424.

[22] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts and P. Walter. *Molecular Biology of the Cell*. 6th. New York: W. W. Norton & Company, 2017. Chap. 2.

[23] D. De Martino, A. Mc Andersson, T. Bergmiller, C. C. Guet and G. Tkačik. 'Statistical mechanics for metabolic networks during steady state growth'. In: *Nature communications* 9.1 (2018), p. 2988.

[24] S. Gude, E. Pinçe, K. M. Taute, A.-B. Seinen, T. S. Shimizu and S. J. Tans. 'Bacterial coexistence driven by motility and spatial competition'. In: *Nature* 578.7796 (2020), pp. 588–592.

[25] S. R. Hackett, V. R. Zanotelli, W. Xu, J. Goya, J. O. Park, D. H. Perlman, P. A. Gibney, D. Botstein, J. D. Storey and J. D. Rabinowitz. 'Systems-level analysis of mechanisms regulating yeast metabolic flux'. In: *Science* 354.6311 (2016), aaf2786.

[26] M. Jinek, K. Chylinski, I. Fonfara, M. Hauer, J. A. Doudna and E. Charpentier. 'A programmable dual-RNA–guided DNA endonuclease in adaptive bacterial immunity'. In: *science* 337.6096 (2012), pp. 816–821.

[27] A. Pickar-Oliver and C. A. Gersbach. 'The next generation of CRISPR–Cas technologies and applications'. In: *Nature reviews Molecular cell biology* 20.8 (2019), pp. 490–507.

[28] G. Stephanopoulos, A. A. Aristidou and J. Nielsen. 'Metabolic engineering: principles and methodologies'. In: (1998).

[29] J. M. Berg, J. L. Tymoczko and L. Stryer. *Biochemistry (loose-leaf)*. Macmillan, 2007.

[30] M. Jeschek, D. Gerngross and S. Panke. 'Combinatorial pathway optimization for streamlined metabolic engineering'. In: *Current Opinion in Biotechnology* 47 (2017), pp. 142–151.

[31] Y. Cui, D. Liu, H. Xue, M. Li, W. Guo, C. Huang, X. Zheng, J. Yang, H. Liu, H. Yin *et al.* 'Metabolic Engineering of Yarrowia lipolytica with Massive Gene Assembly and Genomic Integration'. In: *ACS Synthetic Biology* (2025).

[32] E. M. Young, Z. Zhao, B. E. Gielesen, L. Wu, D. B. Gordon, J. A. Roubos and C. A. Voigt. 'Iterative algorithm-guided design of massive strain libraries, applied to itaconic acid production in yeast'. In: *Metabolic Engineering* 48 (2018), pp. 33–43.

[33] J. A. Dietrich, A. E. McKee and J. D. Keasling. 'High-throughput metabolic engineering: advances in small-molecule screening and selection'. In: *Annual review of biochemistry* 79.1 (2010), pp. 563–590.

[34] N. Gurdo, D. C. Volke, D. McCloskey and P. I. Nikel. 'Automating the design-build-test-learn cycle towards next-generation bacterial cell factories'. In: *New Biotechnology* 74 (2023), pp. 1–15.

[35] S. Moreno-Paz, J. Schmitz and M. Suarez-Diez. 'In silico analysis of design of experiment methods for metabolic pathway optimization'. In: *Computational and Structural Biotechnology Journal* 23 (2024), pp. 1959–1967.

[36] X. Liao, H. Ma and Y. J. Tang. 'Artificial intelligence: a solution to involution of design–build–test–learn cycle'. In: *Current opinion in biotechnology* 75 (2022), p. 102712.

[37] T. Radivojević, Z. Costello, K. Workman and H. Garcia Martin. 'A machine learning Automated Recommendation Tool for synthetic biology'. In: *Nature communications* 11.1 (2020), p. 4879.

[38] J. Zhang, S. D. Petersen, T. Radivojevic, A. Ramirez, A. Pérez-Manríquez, E. Abeliuk, B. J. Sánchez, Z. Costello, Y. Chen, M. J. Fero *et al.* 'Combining mechanistic and machine learning models for predictive engineering and optimization of tryptophan metabolism'. In: *Nature communications* 11.1 (2020), p. 4880.

[39] P. Opgenorth, Z. Costello, T. Okada, G. Goyal, Y. Chen, J. Gin, V. Benites, M. de Raad, T. R. Northen, K. Deng *et al.* 'Lessons from two design–build–test–learn cycles of dodecanol production in Escherichia coli aided by machine learning'. In: *ACS synthetic biology* 8.6 (2019), pp. 1337–1351.

[40] P. Carbonell, A. J. Jervis, C. J. Robinson, C. Yan, M. Dunstan, N. Swainston, M. Vinaixa, K. A. Hollywood, A. Currin, N. J. Rattray *et al.* 'An automated Design-Build-Test-Learn pipeline for enhanced microbial production of fine chemicals'. In: *Communications biology* 1.1 (2018), p. 66.

[41] C. Engler, R. Kandzia and S. Marillonnet. 'A one pot, one step, precision cloning method with high throughput capability'. In: *PloS one* 3.11 (2008), e3647.

[42] M. J. Smanski, S. Bhatia, D. Zhao, Y. Park, L. BA Woodruff, G. Giannoukos, D. Ciulla, M. Busby, J. Calderon, R. Nicol *et al.* 'Functional optimization of gene clusters by combinatorial design and assembly'. In: *Nature biotechnology* 32.12 (2014), pp. 1241–1249.

[43] Z. Wang, C. Wang and G. Chen. 'Kinetic modeling: a tool for temperature shift and feeding optimization in cell culture process development'. In: *Protein Expression and Purification* 198 (2022), p. 106130.

[44] I. Domenzain, Y. Lu, H. Wang, J. Shi, H. Lu and J. Nielsen. 'Computational biology predicts metabolic engineering targets for increased production of 103 valuable chemicals in yeast'. In: *Proceedings of the National Academy of Sciences* 122.9 (2025), e2417322122.

[45] J. Almquist, M. Cvijovic, V. Hatzimanikatis, J. Nielsen and M. Jirstrand. 'Kinetic models in industrial biotechnology–improving cell factory performance'. In: *Metabolic engineering* 24 (2014), pp. 38–60.

[46] S. Ali, T. Abuhmed, S. El-Sappagh, K. Muhammad, J. M. Alonso-Moral, R. Confalonieri, R. Guidotti, J. Del Ser, N. Díaz-Rodríguez and F. Herrera. 'Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence'. In: *Information fusion* 99 (2023), p. 101805.

[47] J. Alonso-Gutierrez, E.-M. Kim, T. S. Batth, N. Cho, Q. Hu, L. J. G. Chan, C. J. Petzold, N. J. Hillson, P. D. Adams, J. D. Keasling *et al.* 'Principal component analysis of proteomics (PCAP) as a tool to direct metabolic engineering'. In: *Metabolic engineering* 28 (2015), pp. 123–133.

[48] D. Choe, C. A. Olson, R. Szubin, H. Yang, J. Sung, A. M. Feist and B. O. Palsson. 'Advancing the scale of synthetic biology via cross-species transfer of cellular functions enabled by iModulon engraftment'. In: *Nature Communications* 15.1 (2024), p. 2356.

[49] Z. Costello and H. G. Martin. 'A machine learning approach to predict metabolic pathway dynamics from time-series multiomics data'. In: *NPJ systems biology and applications* 4.1 (2018), pp. 1–14.

[50] S. Moreno-Paz, R. Van der Hoek, E. Eliana, P. Zwartjens, S. Gosiewska, V. A. Martins dos Santos, J. Schmitz and M. Suarez-Diez. 'Machine learning-guided optimization of p-coumaric acid production in yeast'. In: *ACS synthetic biology* 13.4 (2024), pp. 1312–1322.

[51] A. Pandi, C. Diehl, A. Yazdizadeh Kharrazi, S. A. Scholz, E. Bobkova, L. Faure, M. Nattermann, D. Adam, N. Chapin, Y. Foroughijabbari *et al.* 'A versatile active learning workflow for optimization of genetic and metabolic networks'. In: *Nature Communications* 13.1 (2022), p. 3876.

[52] M. Sabzevari, S. Szedmak, M. Penttilä, P. Jouhten and J. Rousu. 'Strain design optimization using reinforcement learning'. In: *PLoS computational biology* 18.6 (2022), e1010177.

[53] R. S. Sutton, A. G. Barto *et al. Reinforcement learning: An introduction*. Vol. 1. 1. MIT press Cambridge, 1998.

[54] C. Gu, G. B. Kim, W. J. Kim, H. U. Kim and S. Y. Lee. 'Current status and applications of genome-scale metabolic models'. In: *Genome biology* 20.1 (2019), p. 121.

[55] A. P. Burgard, P. Pharkya and C. D. Maranas. 'Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization'. In: *Biotechnology and bioengineering* 84.6 (2003), pp. 647–657.

[56] G. I. Guzmán, J. Utrilla, S. Nurk, E. Brunk, J. M. Monk, A. Ebrahim, B. O. Palsson and A. M. Feist. 'Model-driven discovery of underground metabolic functions in Escherichia coli'. In: *Proceedings of the National Academy of Sciences* 112.3 (2015), pp. 929–934.

[57] R. A. Khandelwal, B. G. Olivier, W. F. Röling, B. Teusink and F. J. Bruggeman. 'Community flux balance analysis for microbial consortia at balanced growth'. In: *PloS one* 8.5 (2013), e64567.

[58] X. Fang, C. J. Lloyd and B. O. Palsson. 'Reconstructing organisms in silico: genome-scale models and their emerging applications'. In: *Nature Reviews Microbiology* 18.12 (2020), pp. 731–743.

[59] J. D. Orth, I. Thiele and B. Ø. Palsson. 'What is flux balance analysis?' In: *Nature biotechnology* 28.3 (2010), pp. 245–248.

[60] C. S. Henry, L. J. Broadbelt and V. Hatzimanikatis. 'Thermodynamics-based metabolic flux analysis'. In: *Biophysical journal* 92.5 (2007), pp. 1792–1805.

[61] S. van den Bogaard, P. A. Saa and T. B. Alter. 'Sensitivities in protein allocation models reveal distribution of metabolic capacity and flux control'. In: *Bioinformatics* 40.12 (2024), btae691.

[62] R. Mahadevan, J. S. Edwards and F. J. Doyle. 'Dynamic flux balance analysis of diauxic growth in Escherichia coli'. In: *Biophysical journal* 83.3 (2002), pp. 1331–1340.

[63] D. A. Beard and H. Qian. *Chemical biophysics*. Cambridge University Press, 2008. Chap. 1.

[64] U. Von Stockar, T. Maskow, J. Liu, I. W. Marison and R. Patino. 'Thermodynamics of microbial growth and metabolism: an analysis of the current situation'. In: *Journal of Biotechnology* 121.4 (2006), pp. 517–533.

[65] P. Salvy, G. Fengos, M. Ataman, T. Pathier, K. C. Soh and V. Hatzimanikatis. 'pyTFA and matTFA: a Python package and a Matlab toolbox for Thermodynamics-based Flux Analysis'. In: *Bioinformatics* 35.1 (2019), pp. 167–169.

[66] A. Wachtel, R. Rao and M. Esposito. 'Thermodynamically consistent coarse graining of biocatalysts beyond Michaelis–Menten'. In: *New Journal of Physics* 20.4 (2018), p. 042002.

[67] P. A. Saa and L. K. Nielsen. 'Formulation, construction and analysis of kinetic models of metabolism: a review of modelling frameworks'. In: *Biotechnology advances* 35.8 (2017), pp. 981–1003.

[68] Y.-S. Cho and H.-S. Lim. 'Comparison of various estimation methods for the parameters of Michaelis-Menten equation based on in vitro elimination kinetic simulation data'. In: *Translational and clinical pharmacology* 26.1 (2018), p. 39.

[69] L. Michaelis, M. L. Menten *et al.* 'Die kinetik der invertinwirkung'. In: *Biochem. z* 49.333-369 (1913), p. 352.

[70] C. R. Bartman, D. R. Weilandt, Y. Shen, W. D. Lee, Y. Han, T. TeSlaa, C. S. Jankowski, L. Samarah, N. R. Park, V. da Silva-Diz *et al.* 'Slow TCA flux and ATP production in primary solid tumours but not metastases'. In: *Nature* 614.7947 (2023), pp. 349–357.

[71] L. Canini and A. S. Perelson. 'Viral kinetic modeling: state of the art'. In: *Journal of pharmacokinetics and pharmacodynamics* 41.5 (2014), pp. 431–443.

[72] B. Narayanan, W. Jiang, S. Wang, J. Sáez-Sáez, D. Weilandt, M. M. Barcon, V. Hesselberg-Thomsen, I. Borodina, V. Hatzimanikatis and L. Miskovic. 'Kinetic-model-guided engineering of multiple S. cerevisiae strains improves p-coumaric acid production'. In: *Metabolic engineering* (2025).

[73] I. Toumpe, S. Choudhury, V. Hatzimanikatis and L. Miskovic. 'The Dawn of High-Throughput and Genome-Scale Kinetic Modeling: Recent Advances and Future Directions'. In: *ACS Synthetic Biology* (2025).

[74] Y. Wei, B. Peng, R. Xie, Y. Chen, Y. Qin, P. Wen, S. Bauer, P.-Y. Tung and D. Raabe. 'Deep active optimization for complex systems'. In: *Nature Computational Science* 5.9 (2025), pp. 1–12.

[75] K. Ding, M. Chin, Y. Zhao, W. Huang, B. K. Mai, H. Wang, P. Liu, Y. Yang and Y. Luo. 'Machine learning-guided co-optimization of fitness and diversity facilitates combinatorial library design in enzyme engineering'. In: *Nature Communications* 15.1 (2024), p. 6392.

[76] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, A. Ramadhan and A. Edelman. 'Universal differential equations for scientific machine learning'. In: *arXiv preprint arXiv:2001.04385* (2020).

[77] P. Kidger. 'On neural differential equations'. In: *arXiv preprint arXiv:2202.02435* (2022).

[78] M. de Rooij, B. Erdős, N. A. van Riel and S. D. O'Donovan. 'Physiology-informed regularisation enables training of universal differential equation systems for biological applications'. In: *PLOS Computational Biology* 21.1 (2025), e1012198.

[79] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne and Q. Zhang. *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. 2018. URL: http://github.com/jax-ml/jax.

[80] A. Martins, D. Camacho, J. Shuman, W. Sha, P. Mendes and V. Shulaev. 'A systems biology study of two distinct growth phases of Saccharomyces cerevisiae cultures'. In: *Current Genomics* 5.8 (2004), pp. 649–663.

[81] P. van Lent, J. Schmitz and T. Abeel. 'Simulated design–build–test–learn cycles for consistent comparison of machine learning methods in metabolic engineering'. In: *ACS Synthetic Biology* 12.9 (2023), pp. 2588–2599.

[82] G. B. Kim, W. J. Kim, H. U. Kim and S. Y. Lee. 'Machine learning applications in systems metabolic engineering'. In: *Current Opinion in Biotechnology* 64 (Aug. 2020), pp. 1–9. ISSN: 0958-1669. DOI: 10.1016/J.COPBIO.2019.08.010.

[83] K. R. Choi, W. D. Jang, D. Yang, J. S. Cho, D. Park and S. Y. Lee. 'Systems Metabolic Engineering Strategies: Integrating Systems and Synthetic Biology with Metabolic Engineering'. In: *Trends in Biotechnology* 37.8 (Aug. 2019), pp. 817–837. ISSN: 0167-7799. DOI: 10.1016/J.TIBTECH.2019.01.003.

[84] G. Staphanopoulos, AA. Aristidou and J Nielsen. *Metabolic Engineering: principles and methodologies*. 1998, pp. 205–273.

[85] J. E. Bailey, A. Sburlati, V. Hatzimanikatis, K. Lee, W. A. Renner and P. S. Tsai. 'Inverse metabolic engineering: a strategy for directed genetic engineering of useful phenotypes'. In: *Wiley Online Library* 79.5 (2002), pp. 568–579. DOI: 10.1002/bit.10441. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/bit.10441.

[86] J. C. Way, J. J. Collins, J. D. Keasling and P. A. Silver. 'Integrating Biological Redesign: Where Synthetic Biology Came From and Where It Needs to Go'. In: *Cell* 157.1 (Mar. 2014), pp. 151–161. ISSN: 0092-8674. DOI: 10.1016/J.CELL.2014.02.039.

[87] J. Mampel, J. M. Buescher, G. Meurer and J. Eck. 'Coping with complexity in metabolic engineering'. In: *Trends in Biotechnology* 31.1 (Jan. 2013), pp. 52–60. ISSN: 0167-7799. DOI: 10.1016/J.TIBTECH.2012.10.010.

[88] D. J. Pitera, C. J. Paddon, J. D. Newman and J. D. Keasling. 'Balancing a heterologous mevalonate pathway for improved isoprenoid production in Escherichia coli'. In: *Metabolic Engineering* 9.2 (2007). ISSN: 10967176. DOI: 10.1016/j.ymben.2006.11.002.

[89] M. Jeschek, D. Gerngross and S. Panke. 'Combinatorial pathway optimization for streamlined metabolic engineering'. In: *Current Opinion in Biotechnology* 47 (Oct. 2017), pp. 142–151. ISSN: 0958-1669. DOI: 10.1016/J.COPBIO.2017.06.014.

[90] M. J. Smanski, S. Bhatia, D. Zhao, Y. J. Park, L. B. Woodruff, G. Giannoukos, D. Ciulla, M. Busby, J. Calderon, R. Nicol, D. B. Gordon, D. Densmore and C. A. Voigt. 'Functional optimization of gene clusters by combinatorial design and assembly'. In: *Nature Biotechnology* (2014). ISSN: 15461696. DOI: 10.1038/nbt.3063.

[91] E. M. Young, Z. Zhao, B. E. Gielesen, L. Wu, D. Benjamin Gordon, J. A. Roubos and C. A. Voigt. 'Iterative algorithm-guided design of massive strain libraries, applied to itaconic acid production in yeast'. In: *Metabolic Engineering* 48 (July 2018), pp. 33–43. ISSN: 1096-7176. DOI: 10.1016/J.YMBEN.2018.05.002.

[92] M. Jeschek, D. Gerngross and S. Panke. 'Rationally reduced libraries for combinatorial pathway optimization minimizing experimental effort'. In: *Nature Communications* 7 (2016). ISSN: 20411723. DOI: 10.1038/ncomms11163.

[93] B. Kim, J. Du, D. T. Eriksen and H. Zhao. 'Combinatorial design of a highly efficient xylose-utilizing pathway in Saccharomyces cerevisiae for the production of cellulosic biofuels'. In: *Applied and Environmental Microbiology* 79.3 (Feb. 2013), pp. 931–941. ISSN: 00992240. DOI: 10.1128/AEM.02736-12. URL: https://journals.asm.org/doi/10.1128/AEM.02736-12.

[94] J. Yuan and C. B. Ching. 'Combinatorial engineering of mevalonate pathway for improved amorpha-4,11-diene production in budding yeast'. In: *Biotechnology and Bioengineering* 111.3 (Mar. 2014), pp. 608–617. ISSN: 1097-0290. DOI: 10.1002/BIT.25123. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/bit.25123%20https://onlinelibrary.wiley.com/doi/abs/10.1002/bit.25123%20https://onlinelibrary.wiley.com/doi/10.1002/bit.25123.

[95] C. Engler and S. Marillonnet. 'Golden Gate cloning'. In: *Methods in Molecular Biology* 1116 (2014), pp. 119–131. ISSN: 10643745. DOI: 10.1007/978-1-62703-764-8{_}9/FIGURES/3. URL: https://link.springer.com/protocol/10.1007/978-1-62703-764-8_9.

[96] X. Liao, H. Ma and Y. J. Tang. 'Artificial intelligence: a solution to involution of design–build–test–learn cycle'. In: *Current Opinion in Biotechnology* 75 (June 2022), p. 102712. ISSN: 0958-1669. DOI: 10.1016/J.COPBIO.2022.102712.

[97] J. Alonso-Gutierrez, E. M. Kim, T. S. Batth, N. Cho, Q. Hu, L. J. G. Chan, C. J. Petzold, N. J. Hillson, P. D. Adams, J. D. Keasling, H. Garcia Martin and T. S. Lee. 'Principal component analysis of proteomics (PCAP) as a tool to direct metabolic engineering'. In: *Metabolic Engineering* 28 (Mar. 2015), pp. 123–133. ISSN: 1096-7176. DOI: 10.1016/J.YMBEN.2014.11.011.

[98] A. Zelezniak, J. Vowinckel, B. Klaus, M. A. Keller and M. R. Correspondence. 'Machine Learning Predicts the Yeast Metabolome from the Quantitative Proteome of Kinase Knockouts'. In: *Cell Systems* 7 (2018), pp. 269–283. DOI: 10.1016/j.cels.2018.08.001. URL: https://doi.org/10.1016/j.cels.2018.08.001.

[99] M. HamediRad, R. Chao, S. Weisberg, J. Lian, S. Sinha and H. Zhao. 'Towards a fully automated algorithm driven platform for biosystems design'. In: *Nature Communications* (2019). ISSN: 20411723. DOI: 10.1038/s41467-019-13189-z.

[100] M. Sabzevariid, S. Szedmakid, M. Penttilä, P. Jouhten and J. Rousuid. 'Strain design optimization using reinforcement learning'. In: *PLOS Computational Biology* 18.6 (June 2022). Ed. by P. Mendes, e1010177. ISSN: 1553-7358. DOI: 10.1371/JOURNAL.PCBI.1010177. URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1010177.

[101] T. Radivojević, Z. Costello, K. Workman and H. Garcia Martin. 'A machine learning Automated Recommendation Tool for synthetic biology'. In: *Nature Communications* (2020). ISSN: 20411723. DOI: 10.1038/s41467-020-18008-4.

[102] A. Pandi, C. Diehl, A. Yazdizadeh Kharrazi, S. A. Scholz, E. Bobkova, L. Faure, M. Nattermann, D. Adam, N. Chapin, Y. Foroughijabbari, C. Moritz, N. Paczia, N. S. Cortina, J. L. Faulon and T. J. Erb. 'A versatile active learning workflow for optimization of genetic and metabolic networks'. In: *Nature Communications* 13.1 (2022). ISSN: 20411723. DOI: 10.1038/s41467-022-31245-z.

[103] J. Zhang, S. D. Petersen, T. Radivojevic, A. Ramirez, A. Pérez-Manríquez, E. Abeliuk, B. J. Sánchez, Z. Costello, Y. Chen, M. J. Fero, H. G. Martin, J. Nielsen, J. D. Keasling and M. K. Jensen. 'Combining mechanistic and machine learning models for predictive engineering and optimization of tryptophan metabolism'. In: *Nature Communications* (2020). ISSN: 20411723. DOI: 10.1038/s41467-020-17910-1.

[104] A. Khodayari and C. D. Maranas. 'A genome-scale Escherichia coli kinetic metabolic model k-ecoli457 satisfying flux data for multiple mutant strains'. In: *Nature Communications 2016 7:1* 7.1 (Dec. 2016), pp. 1–12. ISSN: 2041-1723. DOI: 10.1038/ncomms13806. URL: https://www.nature.com/articles/ncomms13806.

[105] A. Khodayari, A. Chowdhury and C. D. Maranas. 'Succinate Overproduction: A Case Study of Computational Strain Design Using a Comprehensive Escherichia coli Kinetic Model'. In: *Frontiers in Bioengineering and Biotechnology* 2 (Jan. 2015), p. 76. ISSN: 22964185. DOI: 10.3389/FBIOE.2014.00076/BIBTEX.

[106] D. N. Macklin, T. A. Ahn-Horst, H. Choi, N. A. Ruggero, J. Carrera, J. C. Mason, G. Sun, E. Agmon, M. M. DeFelice, I. Maayan, K. Lane, R. K. Spangler, T. E. Gillies, M. L. Paull, S. Akhter, S. R. Bray, D. S. Weaver, I. M. Keseler, P. D. Karp, J. H. Morrison and M. W. Covert. 'Simultaneous cross-evaluation of heterogeneous E. coli datasets via mechanistic simulation'. In: *Science* 369.6502 (July 2020). ISSN: 10959203. DOI: 10.1126/SCIENCE.AAV3751/SUPPL{_}FILE/AAV3751-MACKLIN-SM.PDF. URL: https://dx.doi..

[107] F. Avanzini, G. Falasco, M. Esposito, N. Freitas, A. Wachtel and R. Rao. 'Thermodynamically consistent coarse graining of biocatalysts beyond Michaelis–Menten'. In: *New Journal of Physics* 20.4 (Apr. 2018), p. 042002. ISSN: 1367-2630. DOI: 10.1088/1367-2630/AAB5C9. URL: https://iopscience.iop.org/article/10.1088/1367-2630/aab5c9%20https://iopscience.iop.org/article/10.1088/1367-2630/aab5c9/meta.

[108] S. Raman, J. K. Rogers, N. D. Taylor and G. M. Church. 'Evolution-guided optimization of biosynthetic pathways'. In: *Proceedings of the National Academy of Sciences of the United States of America* 111.50 (2014). ISSN: 10916490. DOI: 10.1073/pnas.1409523111.

[109] A. Varma and B. O. Palsson. 'Metabolic Capabilities of Escherichia coli: I. Synthesis of Biosynthetic Precursors and Cofactors'. In: *Journal of Theoretical Biology* 165.4 (Dec. 1993), pp. 477–502. ISSN: 0022-5193. DOI: 10.1006/JTBI.1993.1202.

[110] D. R. Weilandt, P. Salvy, M. Masid, G. Fengos, R. Denhardt-Erikson, Z. Hosseini, V. Hatzimanikatis and O. Vitek. 'Symbolic kinetic models in python (SKiMpy): intuitive modeling of large-scale biological kinetic models'. In: *Bioinformatics* 39.1 (Jan. 2023). Ed. by O. Vitek. ISSN: 1367-4803. DOI:

10.1093/BIOINFORMATICS/BTAC787. URL: https://academic.oup.com/bioinformatics/article/39/1/btac787/6887139.

[111] S. R. Hackett, V. R. T. Zanotelli, W. Xu, J. Goya, J. O. Park, D. H. Perlman, P. A. Gibney, D. Botstein, J. D. Storey and J. D. Rabinowitz. 'Systems-level analysis of mechanisms regulating yeast metabolic flux'. In: (). DOI: 10.1126/science.aaf2786. URL: http://dx.doi..

[112] J. Lee, S. Y. Lee, S. Park and A. P. Middelberg. 'Control of fed-batch fermentations'. In: *Biotechnology Advances* 17.1 (Apr. 1999), pp. 29–48. ISSN: 0734-9750. DOI: 10.1016/S0734-9750(98)00015-9.

[113] B. Narayanan, D. Weilandt, M. Masid, L. Miskovic and V. Hatzimanikatis. 'Rational strain design with minimal phenotype perturbation'. In: *bioRxiv* (2022).

[114] M. E. Lee, A. Aswani, A. S. Han, C. J. Tomlin and J. E. Dueber. 'Expression-level optimization of a multi-enzyme pathway in the absence of a high-throughput assay'. In: *Nucleic Acids Research* 41.22 (2013). ISSN: 03051048. DOI: 10.1093/nar/gkt809.

[115] H. Alper, C. Fischer, E. N. .-. P. o. t. … and u. 2005. 'Tuning genetic control through promoter engineering'. In: *National Acad Sciences* (2005). URL: https://www.pnas.org/doi/abs/10.1073/pnas.0504604102.

[116] X. Cui, X. Ma, K. L. Prather and K. Zhou. 'Controlling protein expression by using intron-aided promoters in Saccharomyces cerevisiae'. In: *Biochemical Engineering Journal* 176 (Dec. 2021), p. 108197. ISSN: 1369-703X. DOI: 10.1016/J.BEJ.2021.108197.

[117] Y. Wang, H. Wang, L. Wei, S. Li, L. L. .-. N. A. … and u. 2020. 'Synthetic promoter design in Escherichia coli based on a deep generative network'. In: *academic.oup.com* 48.12 (2020), pp. 6403–6412. DOI: 10.1093/nar/gkaa325. URL: https://academic.oup.com/nar/article-abstract/48/12/6403/5837049.

[118] H. Meng, J. Wang, Z. Xiong, F. Xu, G. Zhao and Y. Wang. 'Quantitative Design of Regulatory Elements Based on High-Precision Strength Prediction Using Artificial Neural Network'. In: *PLOS ONE* 8.4 (Apr. 2013), e60288. ISSN: 1932-6203. DOI: 10.1371/JOURNAL.PONE.0060288. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0060288.

[119] M. Henrique, A. Cassiano and R. Silva-Rocha. 'Benchmarking Bacterial Promoter Prediction Tools: Potentialities and Limitations'. In: *mSystems* (2020). DOI: 10.1128/MSYSTEMS.00439-20. URL: https://journals.asm.org/doi/10.1128/mSystems.00439-20.

[120] D. Na and D. Lee. 'RBSDesigner: software for designing synthetic ribosome binding sites that yields a desired level of protein expression'. In: *Bioinformatics* 26.20 (2010), pp. 2633–2634.

[121] R. Young, M. Haines, M. Storch and P. S. Freemont. 'Combinatorial metabolic pathway assembly approaches and toolkits for modular assembly'. In: *Metabolic Engineering* 63 (Jan. 2021), pp. 81–101. ISSN: 1096-7176. DOI: 10.1016/J.YMBEN.2020.12.001.

[122] N. Gunantara. 'A review of multi-objective optimization: Methods and its applications'. In: *http://www.editorialmanager.com/cogenteng* 5.1 (Jan. 2018), pp. 1–16. ISSN: 23311916. DOI: 10.1080/23311916.2018.1502242. URL: https://www.tandfonline.com/doi/abs/10.1080/23311916.2018.1502242.

[123] M. Babaei, G. M. Borja Zamfir, X. Chen, H. B. Christensen, M. Kristensen, J. Nielsen and I. Borodina. 'Metabolic Engineering of Saccharomyces cerevisiae for Rosmarinic Acid Production'. In: *ACS Synthetic Biology* 9.8 (2020). ISSN: 21615063.

[124] G. S. Hossain, H. Yuan, J. Li, H. d. Shin, M. Wang, G. Du, J. Chen and L. Liu. 'Metabolic engineering of Raoultella ornithinolytica BF60 for production of 2,5-furandicarboxylic acid from 5-hydroxymethylfurfural'. In: *Applied and Environmental Microbiology* 83.1 (2017). ISSN: 10985336.

[125] I. Surowiec, L. Vikström, G. Hector, E. Johansson, C. Vikström and J. Trygg. 'Generalized Subset Designs in Analytical Chemistry'. In: *Analytical Chemistry* 89.12 (June 2017), pp. 6491–6497. ISSN: 15206882. DOI: 10.1021/ACS.ANALCHEM.7B00506/ASSET/IMAGES/LARGE/AC-2017-00506Q{_}0006.JPEG. URL: https://pubs.acs.org/doi/full/10.1021/acs.analchem.7b00506.

[126] B. E. Slatko, A. F. Gardner and F. M. Ausubel. 'Overview of Next-Generation Sequencing Technologies'. In: *Current Protocols in Molecular Biology* 122.1 (Apr. 2018), e59. ISSN: 1934-3647. DOI: 10.1002/CPMB.59. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/cpmb.59%20https://onlinelibrary.wiley.com/doi/abs/10.1002/cpmb.59%20https://currentprotocols.onlinelibrary.wiley.com/doi/10.1002/cpmb.59.

[127] B. Moseley and W. Liebermeister. 'Structural Thermokinetic Modelling'. In: *Metabolites 2022, Vol. 12, Page 434* 12.5 (May 2022), p. 434. ISSN: 2218-1989. DOI: 10.3390/METABO12050434. URL: https://www.mdpi.com/2218-1989/12/5/434/htm%20https://www.mdpi.com/2218-1989/12/5/434.

[128] A. Ebrahim, J. A. Lerman, B. O. Palsson and D. R. Hyduke. 'COBRApy: COnstraints-Based Reconstruction and Analysis for Python'. In: *BMC Systems Biology* 7.1 (Aug. 2013), pp. 1–6. ISSN: 17520509. DOI: 10.1186/1752-0509-7-74/FIGURES/2. URL: https://link.springer.com/articles/10.1186/1752-0509-7-74%20https://link.springer.com/article/10.1186/1752-0509-7-74.

[129] M. Yasemi and M. Jolicoeur. 'Modelling cell metabolism: A review on constraint-based steady-state and kinetic approaches'. In: *Processes* 9.2 (2021). ISSN: 22279717. DOI: 10.3390/pr9020322.

[130] P. Salvy, G. Fengos, M. Ataman, T. Pathier, K. C. Soh and V. Hatzimanikatis. 'pyTFA and matTFA: a Python package and a Matlab toolbox for Thermodynamics-based Flux Analysis'. In: *Bioinformatics* 35.1 (Jan. 2019), pp. 167–169. ISSN: 1367-4803. DOI: 10.1093/BIOINFORMATICS/BTY499. URL: https://academic.oup.com/bioinformatics/article/35/1/167/5047753.

[131] C. S. Henry, L. J. Broadbelt and V. Hatzimanikatis. 'Thermodynamics-Based Metabolic Flux Analysis'. In: *Biophysical Journal* 92.5 (Mar. 2007), pp. 1792–1805. ISSN: 0006-3495. DOI: 10.1529/BIOPHYSJ.106.093138.

[132] L. Miskovic and V. Hatzimanikatis. 'Production of biofuels and biochemicals: In need of an ORACLE'. In: *Trends in Biotechnology* 28.8 (Aug. 2010), pp. 391–397. ISSN: 01677799. DOI: 10.1016/j.tibtech.2010.05.003.

[133] N. J. Averesch and J. O. Krömer. 'Metabolic engineering of the shikimate pathway for production of aromatics and derived compounds-Present and future strain construction strategies'. In: *Frontiers in Bioengineering and Biotechnology* 6.MAR (Mar. 2018), p. 32. ISSN: 22964185. DOI: 10.3389/FBIOE.2018.00032/BIBTEX.

[134] B. S. Laursen, H. P. Sørensen, K. K. Mortensen and H. U. Sperling-Petersen. 'Initiation of Protein Synthesis in Bacteria'. In: *Microbiology and Molecular Biology Reviews* 69.1 (2005). ISSN: 1092-2172. DOI: 10.1128/mmbr.69.1.101-123.2005.

[135] F. Pedregosa, V. Michel, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, J. Vanderplas, D. Cournapeau, F. Pedregosa, G. Varoquaux, A. Gramfort, B. Thirion, O. Grisel, V. Dubourg, A. Passos, M. Brucher, M. Perrot andÉdouardand, a. Duchesnay and F. Duchesnay EDOUARDDUCHESNAY. 'Scikit-learn: Machine Learning in Python'. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830. URL: http://scikit-learn.sourceforge.net..

[136] T. Head, M. Kumar, H. Nahrstaedt, G. Louppe and I. Shcherbatyi. 'scikit-optimize/scikit-optimize'. In: (Oct. 2021). DOI: 10.5281/ZENODO.5565057. URL: https://zenodo.org/record/5565057.

[137] P. van Lent, R. van der Hoek, S. M. Paz, I. Kooi, M. Jonkers, P. Zwartjens, J. Schmitz and T. Abeel. 'Machine Learning-Assisted Pathway Optimization in Large Combinatorial Design Spaces: a p-Coumaric Acid Case Study'. In: *bioRxiv* (2025), pp. 2025–06.

[138] *EUR-Lex - 52024DC0137 - EN - EUR-Lex*. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:52024DC0137.

[139] G. B. Kim, S. Y. Choi, I. J. Cho, D.-H. Ahn and S. Y. Lee. 'Metabolic engineering for sustainability and health'. In: *Trends in Biotechnology* 41.3 (2023), pp. 425–451.

[140] M. Gustavsson and S. Y. Lee. 'Prospects of microbial cell factories developed through systems metabolic engineering'. In: *Microbial Biotechnology* 9.5 (2016), pp. 610–617.

[141] S. Y. Lee and H. U. Kim. 'Systems strategies for developing industrial microbial strains'. In: *Nature biotechnology* 33.10 (2015), pp. 1061–1072.

[142] M. Jeschek, D. Gerngross and S. Panke. 'Rationally reduced libraries for combinatorial pathway optimization minimizing experimental effort'. In: *Nature communications* 7.1 (2016), p. 11163.

[143] R. S. Sutton. 'Reinforcement learning: An introduction'. In: *A Bradford Book* (2018).

[144] M. HamediRad, R. Chao, S. Weisberg, J. Lian, S. Sinha and H. Zhao. 'Towards a fully automated algorithm driven platform for biosystems design'. In: *Nature communications* 10.1 (2019), p. 5150.

[145] J. Kaur and R. Kaur. 'p-Coumaric acid: A naturally occurring chemical with potential therapeutic applications'. In: *Current Organic Chemistry* 26.14 (2022), pp. 1333–1349.

[146] T. Chen and C. Guestrin. 'Xgboost: A scalable tree boosting system'. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[147] C. K. Kang, J. Shin, Y. Cha, M. S. Kim, M. S. Choi, T. Kim, Y.-K. Park and Y. J. Choi. 'Machine learning-guided prediction of potential engineering targets for microbial production of lycopene'. In: *Bioresource Technology* 369 (2023), p. 128455.

[148] H. Lu, F. Li, B. J. Sánchez, Z. Zhu, G. Li, I. Domenzain, S. Marcišauskas, P. M. Anton, D. Lappa, C. Lieven *et al.* 'A consensus S. cerevisiae metabolic model Yeast8 and its ecosystem for comprehensively probing cellular metabolism'. In: *Nature communications* 10.1 (2019), p. 3586.

[149] N. E. Lewis, K. K. Hixson, T. M. Conrad, J. A. Lerman, P. Charusanti, A. D. Polpitiya, J. N. Adkins, G. Schramm, S. O. Purvine, D. Lopez-Ferrer *et al.* 'Omic data from evolved E. coli are consistent with computed optimal growth from genome-scale models'. In: *Molecular systems biology* 6.1 (2010), p. 390.

[150] F.-Z. Li, J. Yang, K. E. Johnston, E. Gürsoy, Y. Yue and F. H. Arnold. 'Evaluation of machine learning-assisted directed evolution across diverse combinatorial landscapes'. In: *bioRxiv* (2024), pp. 2024–10.

[151] M. Steininger, K. Kobs, P. Davidson, A. Krause and A. Hotho. 'Density-based weighting for imbalanced regression'. In: *Machine Learning* 110.8 (2021), pp. 2187–2211.

[152] S. Nitzan and A. Rubinstein. 'A further characterization of Borda ranking method'. In: *Public choice* (1981), pp. 153–158.

[153] J. Mao, M. T. Mohedano, J. Fu, X. Li, Q. Liu, J. Nielsen, V. Siewers and Y. Chen. 'Fine-tuning of p-coumaric acid synthesis to increase (2S)-naringenin production in yeast'. In: *Metabolic Engineering* 79 (2023), pp. 192–202.

[154] C. Jeong, S. H. Han, C. G. Lim, S. C. Kim and K. J. Jeong. 'Metabolic engineering of Escherichia coli for enhanced production of p-coumaric acid via L-phenylalanine biosynthesis pathway'. In: *Bioprocess and Biosystems Engineering* (2025), pp. 1–12.

[155] P. I. Frazier. 'A tutorial on Bayesian optimization'. In: *arXiv preprint arXiv:1807.02811* (2018).

[156] P. Linardatos, V. Papastefanopoulos and S. Kotsiantis. 'Explainable ai: A review of machine learning interpretability methods'. In: *Entropy* 23.1 (2020), p. 18.

[157] Q. Liu, T. Yu, K. Campbell, J. Nielsen and Y. Chen. 'Modular pathway rewiring of yeast for amino acid production'. In: *Methods in Enzymology*. Vol. 608. Elsevier, 2018, pp. 417–439.

[158] O. Borkowski, M. Koch, A. Zettor, A. Pandi, A. C. Batista, P. Soudier and J.-L. Faulon. 'Large scale active-learning-guided exploration for in vitro protein production optimization'. In: *Nature communications* 11.1 (2020), p. 1872.

[159] Q. Liu, T. Yu, X. Li, Y. Chen, K. Campbell, J. Nielsen and Y. Chen. 'Rewiring carbon metabolism in yeast for high level production of aromatic chemicals'. In: *Nature communications* 10.1 (2019), p. 4976.

[160] D. Lao-Martil, J. P. Schmitz, B. Teusink and N. A. van Riel. 'Elucidating yeast glycolytic dynamics at steady state growth and glucose pulses through kinetic metabolic modeling'. In: *Metabolic Engineering* 77 (2023), pp. 128–142.

[161] M. Maftahi, J.-M. Nicaud, H. Levesque and C. Gaillardin. 'Sequencing analysis of a 24.7 kb fragment of yeast chromosome XIV identifies six known genes, a new member of the hexose transporter family and ten new open reading frames.' In: *Yeast (Chichester, England)* 11.11 (1995), pp. 1077–1085.

[162] A. Rodriguez, K. R. Kildegaard, M. Li, I. Borodina and J. Nielsen. 'Establishment of a yeast platform strain for production of p-coumaric acid through metabolic engineering of aromatic amino acid biosynthesis'. In: *Metabolic engineering* 31 (2015), pp. 181–188.

[163] D. Szklarczyk, R. Kirsch, M. Koutrouli, K. Nastou, F. Mehryary, R. Hachilif, A. L. Gable, T. Fang, N. T. Doncheva, S. Pyysalo *et al.* 'The STRING database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest'. In: *Nucleic acids research* 51.D1 (2023), pp. D638–D646.

[164] E. Noor, H. S. Haraldsdóttir, R. Milo and R. M. Fleming. 'Consistent estimation of Gibbs energy using component contributions'. In: *PLoS computational biology* 9.7 (2013), e1003098.

[165] I. Schomburg, L. Jeske, M. Ulbrich, S. Placzek, A. Chang and D. Schomburg. 'The BRENDA enzyme information system–From a database to an expert system'. In: *Journal of biotechnology* 261 (2017), pp. 194–206.

[166] J. A. Roubos and N. N. M. E. Van Peij. *Method for achieving improved polypeptide expression*. US Patent 8,812,247. Aug. 2014.

[167] R. Verwaal, N. Buiting-Wiessenhaan, S. Dalhuijsen and J. A. Roubos. 'CRISPR/Cpf1 enables fast and simple genome editing of Saccharomyces cerevisiae'. In: *Yeast* 35.2 (2018), pp. 201–211.

[168] R. D. Gietz, R. H. Schiestl, A. R. Willems and R. A. Woods. 'Studies on the transformation of intact yeast cells by the LiAc/SS-DNA/PEG procedure'. In: *Yeast* 11.4 (1995), pp. 355–360.

[169] W. J. Dekker, S. J. Wiersma, J. Bouwknegt, C. Mooiman and J. T. Pronk. 'Anaerobic growth of Saccharomyces cerevisiae CEN. PK113-7D does not depend on synthesis or supplementation of unsaturated fatty acids'. In: *FEMS yeast research* 19.6 (2019), foz060.

[170] R. Wick and J. Volkening. 'Porechop: adapter trimmer for Oxford Nanopore reads'. In: *Available online: github com/rrwick/Porechop* (2018).

[171] S. Koren, B. P. Walenz, K. Berlin, J. R. Miller, N. H. Bergman and A. M. Phillippy. 'Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation'. In: *Genome research* 27.5 (2017), pp. 722–736.

[172] R. Vaser, I. Sović, N. Nagarajan and M. Šikić. 'Fast and accurate de novo genome assembly from long uncorrected reads'. In: *Genome research* 27.5 (2017), pp. 737–746.

[173] C. Wright and M. Wykes. 'Medaka: sequence correction provided by ONT research'. In: *Medaka: sequence correction provided by ONT research* (2023).

[174] J. Ren, M. Zhang, C. Yu and Z. Liu. 'Balanced mse for imbalanced visual regression'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 7926–7935.

[175] T. Head, G. L. MechCoder, I. Shcherbatyi *et al.* 'scikit-optimize/scikit-optimize: v0. 5.2'. In: *Version v0* 5 (2018).

[176] S. M. Lundberg and S.-I. Lee. 'A unified approach to interpreting model predictions'. In: *Advances in neural information processing systems* 30 (2017).

[177] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay. 'Scikit-learn: Machine Learning in Python'. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[178] P. van Lent, O. Bunkova, B. Magyar, L. Planken, J. Schmitz and T. Abeel. 'Jaxkineticmodel: Neural ordinary differential equations inspired parameterization of kinetic models'. In: *PLOS Computational Biology* 21.7 (2025), e1012733.

[179] C. R. Bartman, D. R. Weilandt, Y. Shen, W. D. Lee, Y. Han, T. TeSlaa, C. S. Jankowski, L. Samarah, N. R. Park, V. da Silva-Diz, M. Aleksandrova, Y. Gultekin, A. Marishta, L. Wang, L. Yang, A. Roichman, V. Bhatt, T. Lan, Z. Hu, X. Xing, W. Lu, S. Davidson, M. Wühr, M. G. Vander Heiden, D. Herranz, J. Y. Guo, Y. Kang and J. D. Rabinowitz. 'Slow TCA flux and ATP production in primary solid tumours but not metastases'. In: *Nature* 614.7947 (2023). ISSN: 14764687. DOI: 10.1038/s41586-022-05661-6.

[180] R. Moreno-Sánchez, E. Saavedra, S. Rodríguez-Enríquez and V. Olín-Sandoval. 'Metabolic control analysis: a tool for designing strategies to manipulate metabolic pathways'. In: *BioMed Research International* 2008.1 (2008), p. 597913.

[181] S. Moreno-Paz, J. Schmitz and M. Suarez-Diez. 'In silico analysis of design of experiment methods for metabolic pathway optimization'. In: *Computational and Structural Biotechnology Journal* 23 (Dec. 2024), pp. 1959–1967. ISSN: 2001-0370. DOI: 10.1016/J.CSBJ.2024.04.062.

[182] P. van Lent, J. Schmitz and T. Abeel. 'Simulated Design-Build-Test-Learn Cycles for Consistent Comparison of Machine Learning Methods in Metabolic Engineering'. In: *ACS Synthetic Biology* 12.9 (2023). ISSN: 21615063. DOI: 10.1021/acssynbio.3c00186.

[183] Z. Wang, C. Wang and G. Chen. 'Kinetic modeling: A tool for temperature shift and feeding optimization in cell culture process development'. In: *Protein Expression and Purification* 198 (2022). ISSN: 10960279. DOI: 10.1016/j.pep.2022.106130.

[184] H. Lu, F. Li, B. J. Sánchez, Z. Zhu, G. Li, I. Domenzain, S. Marcišauskas, P. M. Anton, D. Lappa, C. Lieven, M. E. Beber, N. Sonnenschein, E. J. Kerkhoven and J. Nielsen. 'A consensus S. cerevisiae metabolic model Yeast8 and its ecosystem for comprehensively probing cellular metabolism'. In: *Nature Communications 2019 10:1* 10.1 (Aug. 2019), pp. 1–13. ISSN: 2041-1723. DOI: 10.1038/s41467-019-11581-3. URL: https://www.nature.com/articles/s41467-019-11581-3.

[185] A. Sahin, D. R. Weilandt and V. Hatzimanikatis. 'Optimal enzyme utilization suggests that concentrations and thermodynamics determine binding mechanisms and enzyme saturations'. In: *Nature Communications* 14.1 (2023). ISSN: 20411723. DOI: 10.1038/s41467-023-38159-4.

[186] P. A. Saa and L. K. Nielsen. 'Formulation, construction and analysis of kinetic models of metabolism: A review of modelling frameworks'. In: *Biotechnology Advances* 35.8 (Dec. 2017), pp. 981–1003. ISSN: 0734-9750. DOI: 10.1016/J.BIOTECHADV.2017.09.005.

[187] H. Hass, C. Loos, E. Raimúndez-Álvarez, J. Timmer, J. Hasenauer and C. Kreutz. 'Benchmark problems for dynamic modeling of intracellular processes'. In: *Bioinformatics* 35.17 (2019). ISSN: 14602059. DOI: 10.1093/bioinformatics/btz020.

[188] M. K. Transtrum, B. B. MacHta and J. P. Sethna. 'Geometry of nonlinear least squares with applications to sloppy models and optimization'. In: *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* 83.3 (Mar. 2011), p. 036701. ISSN: 15393755. DOI: 10.1103/PHYSREVE.83.036701/FIGURES/29/MEDIUM.

[189] O. T. Chis, J. R. Banga and E. Balsa-Canto. 'Structural identifiability of systems biology models: A critical comparison of methods'. In: *PLoS ONE* 6.11 (2011). ISSN: 19326203. DOI: 10.1371/journal.pone.0027755.

[190] P. Städter, Y. Schälte, L. Schmiester, J. Hasenauer and P. L. Stapor. 'Benchmarking of numerical integration methods for ODE models of biological systems'. In: *Scientific Reports* 11.1 (2021). ISSN: 20452322. DOI: 10.1038/s41598-021-82196-2.

[191] S. Kim, W. Ji, S. Deng, Y. Ma and C. Rackauckas. 'Stiff neural ordinary differential equations'. In: *Chaos* 31.9 (2021). ISSN: 10897682. DOI: 10.1063/5.0060697.

[192] S. Gopalakrishnan, S. Dash and C. Maranas. 'K-FIT: An accelerated kinetic parameterization algorithm using steady-state fluxomic data'. In: *Metabolic engineering* 61 (2020), pp. 197–205.

[193] M. Hu, P. F. Suthers and C. D. Maranas. 'KETCHUP: Parameterizing of large-scale kinetic models using multiple datasets with different reference states'. In: *Metabolic engineering* 82 (2024), pp. 123–133.

[194] S. Choudhury, M. Moret, P. Salvy, D. Weilandt, V. Hatzimanikatis and L. Miskovic. 'Reconstructing Kinetic Models for Dynamical Studies of Metabolism using Generative Adversarial Networks'. In: *Nature Machine Intelligence 2022 4:8* 4.8 (Aug. 2022), pp. 710–719. ISSN: 2522-5839. DOI: 10.1038/s42256-022-00519-y.

[195] S. Choudhury, B. Narayanan, M. Moret, V. Hatzimanikatis and L. Miskovic. 'Generative machine learning produces kinetic models that accurately characterize intracellular metabolic states'. In: *Nature Catalysis* 7.10 (2024), pp. 1086–1098.

[196] T. Groves, N. L. Cowie and L. K. Nielsen. 'Bayesian Regression Facilitates Quantitative Modeling of Cell Metabolism'. In: *ACS Synthetic Biology* 13.4 (2024), pp. 1205–1214.

[197] Y. Schälte, F. Fröhlich, P. J. Jost, J. Vanhoefer, D. Pathirana, P. Stapor, P. Lakrisenko, D. Wang, E. Raimúndez, S. Merkt, L. Schmiester, P. Städter, S. Grein, E. Dudkin, D. Doresic, D. Weindl and J. Hasenauer. 'pyPESTO: a modular and scalable tool for parameter estimation for dynamic models'. In: *Bioinformatics* 39.11 (2023). ISSN: 13674811. DOI: 10.1093/bioinformatics/btad711.

[198] F. Fröhlich, D. Weindl, Y. Schälte, D. Pathirana, Ł. Paszkowski, G. T. Lines, P. Stapor and J. Hasenauer. 'AMICI: high-performance sensitivity analysis for large ordinary differential equation models'. In: *Bioinformatics* 37.20 (2021), pp. 3676–3677.

[199] M. Muselli. 'A theoretical approach to restart in global optimization'. In: *Journal of Global Optimization* 10.1 (1997), pp. 1–16.

[200] A. F. Villaverde, F. Fröhlich, D. Weindl, J. Hasenauer and J. R. Banga. 'Benchmarking optimization methods for parameter estimation in large kinetic models'. In: *Bioinformatics* 35.5 (2019), pp. 830–838.

[201] R. T. Q. Chen, Y. Rubanova, J. Bettencourt and D. K. Duvenaud. 'Neural Ordinary Differential Equations'. In: *Advances in Neural Information Processing Systems* 31 (2018).

[202] Q. Wu, T. Avanesian, X. Qu and H. Van Dam. 'PolyODENet: Deriving mass-action rate equations from incomplete transient kinetics data'. In: *Journal of Chemical Physics* 157.16 (2022). ISSN: 10897690. DOI: 10.1063/5.0110313.

[203] T. Wang, X.-W. Wang, K. A. Lee-Sarwar, A. A. Litonjua, S. T. Weiss, Y. Sun, S. Maslov and Y.-Y. Liu. 'Predicting metabolomic profiles from microbial composition through neural ordinary differential equations'. In: *Nature machine intelligence* 5.3 (2023), pp. 284–293.

[204] M. Philipps, A. Körner, J. Vanhoefer, D. Pathirana and J. Hasenauer. 'Non-Negative Universal Differential Equations With Applications in Systems Biology'. In: *arXiv preprint arXiv:2406.14246* 58.23 (2024), pp. 25–30.

[205] B. Noordijk, M. L. Garcia Gomez, K. H. ten Tusscher, D. de Ridder, A. D. van Dijk and R. W. Smith. 'The rise of scientific machine learning: a perspective on combining mechanistic modelling with machine learning for systems biology'. In: *Frontiers in Systems Biology* 4 (Aug. 2024), p. 1407994. ISSN: 26740702. DOI: 10.3389/FSYSB.2024.1407994/BIBTEX.

[206] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne *et al.* 'JAX: composable transformations of Python+ NumPy programs'. In: *https://jax.readthedocs.io/en/latest/* (2018).

[207] C. Rackauckas and Q. Nie. 'Differentialequations. jl–a performant and feature-rich ecosystem for solving differential equations in julia'. In: *Journal of open research software* 5.1 (2017), pp. 15–15.

[208] M. Etcheverry, M. Levin, C. Moulin-Frier and P.-Y. Oudeyer. 'SBMLtoODEjax: Efficient Simulation and Optimization of Biological Network Models in JAX'. In: *arXiv preprint arXiv:2307.08452* (2023).

[209] P. Kidger. 'On Neural Differential Equations'. In: *PhD thesis* (2022).

[210] DeepMind, I. Babuschkin, K. Baumli, A. Bell, S. Bhupatiraju, J. Bruce, P. Buchlovsky, D. Budden, T. Cai, A. Clark, I. Danihelka, A. Dedieu, C. Fantacci, J. Godwin, C. Jones, R. Hemsley, T. Hennigan, M. Hessel, S. Hou, S. Kapturowski, T. Keck, I. Kemaev, M. King, M. Kunesch, L. Martens, H. Merzic, V. Mikulik, T. Norman, G. Papamakarios, J. Quan, R. Ring, F. Ruiz, A. Sanchez, L. Sartran, R. Schneider, E. Sezener, S. Spencer, S. Srinivasan, M. Stanojević, W. Stokowiec, L. Wang, G. Zhou and F. Viola. *The DeepMind JAX Ecosystem*. 2020. URL: http://github.com/google-deepmind.

[211] B. J. Bornstein, S. M. Keating, A. Jouraku and M. Hucka. 'LibSBML: An API library for SBML'. In: *Bioinformatics* 24.6 (2008). ISSN: 13674811. DOI: 10.1093/bioinformatics/btn051.

[212] A. Kværnø. 'Singly diagonally implicit runge-kutta methods with an explicit first stage'. In: *BIT Numerical Mathematics* 44.3 (2004). ISSN: 00063835. DOI: 10.1023/B:BITN.0000046811.70614.38.

[213] R. Pascanu, T. Mikolov and Y. Bengio. 'On the difficulty of training recurrent neural networks'. In: *30th International Conference on Machine Learning, ICML 2013*. PART 3. 2013.

[214] C. Suarez-Mendez, A. Sousa, J. Heijnen and A. Wahl. 'Fast "Feast/Famine" Cycles for Studying Microbial Physiology Under Dynamic Conditions: A Case Study with Saccharomyces cerevisiae'. In: *Metabolites* 4.2 (2014). DOI: 10.3390/metabo4020347.

[215] J. Zhuang, T. Tang, Y. Ding, ; Sekhar Tatikonda, N. Dvornek, X. Papademetris and J. S. Duncan. 'AdaBelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients'. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 18795–18806. URL: https://github.com/juntang-zhuang/Adabelief-Optimizer.

[216] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh *et al.* 'SymPy: symbolic computing in Python'. In: *PeerJ Computer Science* 3 (2017), e103.

[217] A. Raue, M. Schilling, J. Bachmann, A. Matteson, M. Schelke, D. Kaschek, S. Hug, C. Kreutz, B. D. Harms, F. J. Theis, U. Klingmüller and J. Timmer. 'Lessons Learned from Quantitative Dynamical Modeling in Systems Biology'. In: *PLoS ONE* 8.9 (2013). ISSN: 19326203. DOI: 10.1371/journal.pone.0074335.

[218] R. S. Malik-Sheriff, M. Glont, T. V. Nguyen, K. Tiwari, M. G. Roberts, A. Xavier, M. T. Vu, J. Men, M. Maire, S. Kananathan, E. L. Fairbanks, J. P. Meyer, C. Arankalle, T. M. Varusai, V. Knight-Schrijver, L. Li, C. Dueñas-Roca, G. Dass, S. M. Keating, Y. M. Park, N. Buso, N. Rodriguez, M. Hucka and H. Hermjakob. 'BioModels–15 years of sharing computational models in life science'. In: *Nucleic Acids Research* 48.D1 (Jan. 2020), pp. D407–D415. ISSN: 0305-1048. DOI: 10.1093/NAR/GKZ1055. URL: https://dx.doi.org/10.1093/nar/gkz1055.

[219] R. Garde, B. Ibrahim and S. Schuster. 'Extending the minimal model of metabolic oscillations in Bacillus subtilis biofilms'. In: *Scientific Reports 2020 10:1* 10.1 (Mar. 2020), pp. 1–11. ISSN: 2045-2322. DOI: 10.1038/s41598-020-62526-6.

[220] K. Smallbone and N. J. Stanford. 'Kinetic modeling of metabolic pathways: Application to serine biosynthesis'. In: *Methods in Molecular Biology* 985 (2013), pp. 113–121. ISSN: 10643745. DOI: 10.1007/978-1-62703-299-5{_}7/TABLES/7.

[221] M. Bruno, J. Koschmieder, F. Wuest, P. Schaub, M. Fehling-Kaschek, J. Timmer, P. Beyer and S. Al-Babili. 'Enzymatic study on AtCCD4 and AtCCD7 and their potential to form acyclic regulatory metabolites'. In: *Journal of Experimental Botany* 67.21 (Nov. 2016), pp. 5993–6005. ISSN: 0022-0957. DOI: 10.1093/JXB/ERW356.

[222] R. Beer, K. Herbst, N. Ignatiadis, I. Kats, L. Adlung, H. Meyer, D. Niopek, T. Christiansen, F. Georgi, N. Kurzawa, J. Meichsner, S. Rabe, A. Riedel, J. Sachs, J. Schessner, F. Schmidt, P. Walch, K. Niopek, T. Heinemann, R. Eils and B. Di Ventura. 'Creating functional engineered variants of the single-module non-ribosomal peptide synthetase IndC by T domain exchange'. In: *Molecular BioSystems* 10.7 (2014). ISSN: 17422051. DOI: 10.1039/c3mb70594c.

[223] N. Patil, Z. Mirveis and H. J. Byrne. 'Kinetic modelling of the cellular metabolic responses underpinning in vitro glycolysis assays'. In: *FEBS Open Bio* 14.3 (Mar. 2024), pp. 466–486. ISSN: 2211-5463. DOI: 10.1002/2211-5463.13765.

[224] S. Palani and C. A. Sarkar. 'Synthetic conversion of a graded receptor signal into a tunable, reversible switch'. In: *Molecular Systems Biology* 7 (Mar. 2011), p. 480. ISSN: 17444292. DOI: 10.1038/MSB.2011.13/SUPPL{_}FILE/MSB201113-SUP-0001-SOURCEDATA-S1.ZIP.

[225] F. Crauste, J. Mafille, L. Boucinha, S. Djebali, O. Gandrillon, J. Marvel and C. Arpin. 'Identification of Nascent Memory CD8 T Cells and Modeling of Their Ontogeny'. In: *Cell Systems* 4.3 (Mar. 2017), pp. 306–317. ISSN: 24054720. DOI: 10.1016/j.cels.2017.01.014.

[226] J. Sneyd and J. F. Dufour. 'A dynamic model of the type-2 inositol trisphosphate receptor'. In: *Proceedings of the National Academy of Sciences* 99.4 (Feb. 2002), pp. 2398–2403. ISSN: 00278424. DOI: 10.1073/PNAS.032281999.

[227] V. Becker, M. Schilling, J. Bachmann, U. Baumann, A. Raue, T. Maiwald, J. Timmer and U. Klingmüller. 'Covering a broad dynamic range: Information processing at the erythropoietin receptor'. In: *Science* 328.5984 (2010). ISSN: 00368075. DOI: 10.1126/science.1184913.

[228] C. Brännmark, R. Palmér, S. T. Glad, G. Cedersund and P. Strålfors. 'Mass and information feedbacks through receptor endocytosis govern insulin signaling as revealed using a parameter-free modeling framework'. In: *Journal of Biological Chemistry* 285.26 (June 2010), pp. 20171–20179. ISSN: 00219258. DOI: 10.1074/jbc.M110.106849. (Visited on 30/06/2025).

[229] D. Ray, Y. Su and P. Ye. 'Dynamic modeling of yeast meiotic initiation'. In: *BMC Systems Biology* 7.1 (May 2013), pp. 1–14. ISSN: 17520509. DOI: 10.1186/1752-0509-7-37/FIGURES/9.

[230] M. B. Elowitz and S. Leibier. 'A synthetic oscillatory network of transcriptional regulators'. In: *Nature 2000 403:6767* 403.6767 (Jan. 2000), pp. 335–338. ISSN: 1476-4687. DOI: 10.1038/35002125.

[231] A. Fiedler, S. Raeth, F. J. Theis, A. Hausser and J. Hasenauer. 'Tailored parameter optimization methods for ordinary differential equation models with steady-state constraints'. In: *BMC Systems Biology* 10.1 (Aug. 2016), pp. 1–19. ISSN: 17520509. DOI: 10.1186/S12918-016-0319-7/FIGURES/8.

[232] J. A. Borghans, G. Dupont and A. Goldbeter. 'Complex intracellular calcium oscillations. A theoretical exploration of possible mechanisms'. In: *Biophysical Chemistry* 66.1 (1997). ISSN: 03014622. DOI: 10.1016/S0301-4622(97)00010-0.

[233] K. A. Fujita, Y. Toyoshima, S. Uda, Y. I. Ozaki, H. Kubota and S. Kuroda. 'Decoupling of receptor and downstream signals in the Akt pathway by its low-pass filter characteristics'. In: *Science Signaling* 3.132 (2010). ISSN: 19450877. DOI: 10.1126/scisignal.2000810.

[234] A. L. Bertozzi, E. Franco, G. Mohler, M. B. Short and D. Sledge. 'The challenges of modeling and forecasting the spread of COVID-19'. In: *Proceedings of the National Academy of Sciences* 117.29 (July 2020), pp. 16732–16738. ISSN: 10916490. DOI: 10.1073/PNAS.2006520117.

[235] H. Hass, F. Kipkeew, A. Gauhar, E. Bouché, P. May, J. Timmer and H. H. Bock. 'Mathematical model of early Reelin-induced Src family kinase-mediated signaling'. In: *PLoS ONE* 12.10 (2017). ISSN: 19326203. DOI: 10.1371/journal.pone.0186927.

[236] V. Raia, M. Schilling, M. Böhm, B. Hahn, A. Kowarsch, A. Raue, C. Sticht, S. Bohl, M. Saile, P. Möller, N. Gretz, J. Timmer, F. Theis, W. D. Lehmann, P. Lichter and U. Klingmüller. 'Dynamic mathematical modeling of IL13-induced signaling in Hodgkin and primary mediastinal B-cell lymphoma allows prediction of therapeutic targets'. In: *Cancer Research* 71.3 (Feb. 2011), pp. 693–704. ISSN: 00085472.

[237] K. Smallbone, N. Malys, H. L. Messiha, J. A. Wishart and E. Simeonidis. 'Building a Kinetic Model of Trehalose Biosynthesis in Saccharomyces cerevisiae'. In: *Methods in Enzymology* 500 (Jan. 2011), pp. 355–370. ISSN: 0076-6879. DOI: 10.1016/B978-0-12-385118-5.00018-9.

[238] P. Weber, M. Hornjik, M. A. Olayioye, A. Hausser and N. E. Radde. 'A computational model of PKD and CERT interactions at the trans-Golgi network of mammalian cells'. In: *BMC Systems Biology* 9.1 (Feb. 2015), pp. 1–14. ISSN: 17520509. DOI: 10.1186/S12918-015-0147-1/FIGURES/6.

[239] Y. Zheng, S. M. Sweet, R. Popovic, E. Martinez-Garcia, J. D. Tipton, P. M. Thomas, J. D. Licht and N. L. Kelleher. 'Total kinetic analysis reveals how combinatorial methylation patterns are established on lysines 27 and 36 of histone H3'. In: *Proceedings of the National Academy of Sciences of the United States of America* 109.34 (2012). ISSN: 00278424. DOI: 10.1073/pnas.1205707109.

[240] J. Isensee, M. Kaufholz, M. J. Knape, J. Hasenauer, H. Hammerich, H. Gonczarowska-Jorge, R. P. Zahedi, F. Schwede, F. W. Herberg and T. Hucho. 'PKA-RII subunit phosphorylation precedes activation by cAMP and regulates activity termination'. In: *Journal of Cell Biology* 217.6 (2018). ISSN: 15408140. DOI: 10.1083/jcb.201708053.

[241] C. Chassagnole, N. Noisommit-Rizzi, J. W. Schmid, K. Mauch and M. Reuss. 'Dynamic modeling of the central carbon metabolism of Escherichia coli'. In: *Biotechnology and Bioengineering* 79.1 (July 2002), pp. 53–73. ISSN: 1097-0290. DOI: 10.1002/BIT.10288.

[242] H. L. Messiha, E. Kent, N. Malys, K. M. Carroll, N. Swainston, P. Mendes and K. Smallbone. 'Enzyme characterisation and kinetic modelling of the pentose phosphate pathway in yeast'. In: *PeerJ PrePrints* 2 (2014). ISSN: 2167-9843.

[243] M. Mosbacher, S. S. Lee, G. Yaakov, M. Nadal-Ribelles, E. de Nadal, F. van Drogen, F. Posas, M. Peter and M. Claassen. 'Positive feedback induces switch between distributive and processive phosphorylation of Hog1'. In: *Nature Communications* 14.1 (2023). ISSN: 20411723. DOI: 10.1038/s41467-023-37430-y.

[244] B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. y Arcas. 'Communication-efficient learning of deep networks from decentralized data'. In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.

[245] M. D. McKay, R. J. Beckman and W. J. Conover. 'A comparison of three methods for selecting values of input variables in the analysis of output from a computer code'. In: *Technometrics* 42.1 (2000), pp. 55–61. ISSN: 15372723. DOI: 10.1080/00401706.2000.10485979.

[246] D. Lao-Martil, J. P. Schmitz, B. Teusink and N. A. van Riel. 'Elucidating yeast glycolytic dynamics at steady state growth and glucose pulses through kinetic metabolic modeling'. In: *Metabolic Engineering* 77 (2023). ISSN: 10967184. DOI: 10.1016/j.ymben.2023.03.005.

[247] I. Schomburg, L. Jeske, M. Ulbrich, S. Placzek, A. Chang and D. Schomburg. *The BRENDA enzyme information system–From a database to an expert system.* 2017. DOI: 10.1016/j.jbiotec.2017.04.020.

[248] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker and C. S. Woodward. 'SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers'. In: *ACM Transactions on Mathematical Software (TOMS)* 31.3 (2005), pp. 363–396.

[249] K. R. Choi, W. D. Jang, D. Yang, J. S. Cho, D. Park and S. Y. Lee. 'Systems metabolic engineering strategies: integrating systems and synthetic biology with metabolic engineering'. In: *Trends in biotechnology* 37.8 (2019), pp. 817–837.

[250] E. D. Carlson, R. Gan, C. E. Hodgman and M. C. Jewett. 'Cell-free protein synthesis: applications come of age'. In: *Biotechnology advances* 30.5 (2012), pp. 1185–1194.

[251] S. M. Paz. 'Model-guided strain engineering: bridges between design and learning in bioprocess development'. PhD thesis. Wageningen University and Research, 2024.

[252] E. J. O'Brien, J. M. Monk and B. O. Palsson. 'Using genome-scale models to predict biological capabilities'. In: *Cell* 161.5 (2015), pp. 971–987.

[253] D. Lao-Martil, K. J. Verhagen, J. P. Schmitz, B. Teusink, S. A. Wahl and N. A. van Riel. 'Kinetic modeling of Saccharomyces cerevisiae central carbon metabolism: achievements, limitations, and opportunities'. In: *Metabolites* 12.1 (2022), p. 74.

[254] M. Yasemi and M. Jolicoeur. 'Modelling cell metabolism: a review on constraint-based steady-state and kinetic approaches'. In: *Processes* 9.2 (2021), p. 322.

[255] C. E. Lawson, J. M. Martí, T. Radivojevic, S. V. R. Jonnalagadda, R. Gentz, N. J. Hillson, S. Peisert, J. Kim, B. A. Simmons, C. J. Petzold *et al.* 'Machine learning for metabolic engineering: A review'. In: *Metabolic Engineering* 63 (2021), pp. 34–60.

[256] C. Culley, S. Vijayakumar, G. Zampieri and C. Angione. 'A mechanism-aware and multiomic machine-learning pipeline characterizes yeast cell growth'. In: *Proceedings of the National Academy of Sciences* 117.31 (2020), pp. 18869–18879.

[257] G. Li, K. S. Rabe, J. Nielsen and M. K. Engqvist. 'Machine learning applied to predicting microorganism growth temperatures and enzyme catalytic optima'. In: *ACS synthetic biology* 8.6 (2019), pp. 1411–1420.

[258] M. Philipps, N. Schmid and J. Hasenauer. 'Current state and open problems in universal differential equations for systems biology'. In: *npj Systems Biology and Applications* 11.1 (2025), p. 101.

[259] C. Plate, C. J. Martensen and S. Sager. 'Optimal experimental design for universal differential equations'. In: *IEEE Transactions on Automatic Control* (2025).

[260] H. Jo, S. W. Cho and H. J. Hwang. 'Estimating the distribution of parameters in differential equations with repeated cross-sectional data'. In: *PLOS Computational Biology* 20.12 (2024), e1012696.

[261] S. Kim, W. Ji, S. Deng, Y. Ma and C. Rackauckas. 'Stiff neural ordinary differential equations'. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31.9 (2021).

[262] M. E. Boehm, L. Adlung, M. Schilling, S. Roth, U. Klingmüller and W. D. Lehmann. 'Identification of isoform-specific dynamics in phosphorylation-dependent STAT5 dimerization by quantitative mass spectrometry and mathematical modeling'. en. In: *J. Proteome Res.* 13.12 (Dec. 2014), pp. 5685–5694.

[263] R. Garde, B. Ibrahim and S. Schuster. 'Extending the minimal model of metabolic oscillations in Bacillus subtilis biofilms'. en. In: *Sci. Rep.* 10.1 (Mar. 2020), p. 5579.

[264] J. M. Borghans, G. Dupont and A. Goldbeter. 'Complex intracellular calcium oscillations. A theoretical exploration of possible mechanisms'. en. In: *Biophys. Chem.* 66.1 (May 1997), pp. 25–41.

[265] A. Fiedler, S. Raeth, F. J. Theis, A. Hausser and J. Hasenauer. 'Tailored parameter optimization methods for ordinary differential equation models with steady-state constraints'. en. In: *BMC Syst. Biol.* 10.1 (Aug. 2016), p. 80.

[266] M. Hucka, F. T. Bergmann, C. Chaouiya, A. Dräger, S. Hoops, S. M. Keating, M. König, N. L. Novère, C. J. Myers, B. G. Olivier *et al.* 'The systems biology markup language (SBML): language specification for level 3 version 2 core release 2'. In: *Journal of integrative bioinformatics* 16.2 (2019), p. 20190021.

[267] A. Cornish-Bowden. 'One hundred years of Michaelis–Menten kinetics'. In: *Perspectives in Science* 4 (2015), pp. 3–9.

[268] X. Glorot, A. Bordes and Y. Bengio. 'Deep sparse rectifier neural networks'. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 315–323.

[269] D. Schlossarek, M. Luzarowski, E. M. Sokołowska, V. P. Thirumalaikumar, L. Dengler, L. Willmitzer, J. C. Ewald and A. Skirycz. 'Rewiring of the protein–protein–metabolite interactome during the diauxic shift in yeast'. In: *Cellular and Molecular Life Sciences* 79.11 (2022), p. 550.

[270] A. Kremling, J. Geiselmann, D. Ropers and H. de Jong. 'An ensemble of mathematical models showing diauxic growth behaviour'. In: *BMC systems biology* 12.1 (2018), p. 82.

[271] J. Zhuang, T. Tang, Y. Ding, S. C. Tatikonda, N. Dvornek, X. Papademetris and J. Duncan. 'Adabelief optimizer: Adapting stepsizes by the belief in observed gradients'. In: *Advances in neural information processing systems* 33 (2020), pp. 18795–18806.

[272] R. Pascanu, T. Mikolov and Y. Bengio. 'On the difficulty of training recurrent neural networks'. In: *International conference on machine learning*. Pmlr. 2013, pp. 1310–1318.

[273] C. Tsitouras. 'Runge–Kutta pairs of order 5 (4) satisfying only the first column simplifying assumption'. In: *Computers & mathematics with applications* 62.2 (2011), pp. 770–775.

[274] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou *et al.* 'Xgboost: extreme gradient boosting'. In: *R package version 0.4-2* 1.4 (2015), pp. 1–4.

[275] *scikit-optimize: sequential model-based optimization in Python 2014; scikit-optimize 0.8.1 documentation — scikit-optimize.github.io*. `https://scikit-optimize.github.io/stable/`. [Accessed 04-09-2025].

[276] C. Finlay, J.-H. Jacobsen, L. Nurbekyan and A. Oberman. 'How to train your neural ode: the world of jacobian and kinetic regularization'. In: *International conference on machine learning*. PMLR. 2020, pp. 3154–3164.

[277] D. Dorešić, S. Grein and J. Hasenauer. 'Efficient parameter estimation for ODE models of cellular processes using semi-quantitative data'. In: *Bioinformatics* 40.Supplement_1 (2024), pp. i558–i566.

[278] I. Domenzain, B. Sánchez, M. Anton, E. J. Kerkhoven, A. Millán-Oropeza, C. Henry, V. Siewers, J. P. Morrissey, N. Sonnenschein and J. Nielsen. 'Reconstruction of a catalogue of genome-scale metabolic models with enzymatic constraints using GECKO 2.0'. In: *Nature communications* 13.1 (2022), p. 3766.

[279] L. Bezjak, V. Erklavec Zajec, Š. Baebler, T. Stare, K. Gruden, A. Pohar, U. Novak and B. Likozar. 'Incorporating RNA-Seq transcriptomics into glycosylation-integrating metabolic network modelling kinetics: Multiomic Chinese hamster ovary (CHO) cell bioreactors'. In: *Biotechnology and Bioengineering* 118.4 (2021), pp. 1476–1490.

[280] D. Zhang, F. Wang, Y. Yu, S. Ding, T. Chen, W. Sun, C. Liang, B. Yu, H. Ying, D. Liu *et al.* 'Effect of quorum-sensing molecule 2-phenylethanol and ARO genes on Saccharomyces cerevisiae biofilm'. In: *Applied Microbiology and Biotechnology* 105.9 (2021), pp. 3635–3648.

[281] I.-S. Kim, H.-Y. Kim, S.-Y. Shin, Y.-S. Kim, D. H. Lee, K. M. Park and H.-S. Yoon. 'A cyclophilin A CPR1 overexpression enhances stress acquisition in Saccharomyces cerevisiae'. In: *Molecules and cells* 29.6 (2010), pp. 567–574.

[282] J. Nocon, M. Steiger, T. Mairinger, J. Hohlweg, H. Rußmayer, S. Hann, B. Gasser and D. Mattanovich. 'Increasing pentose phosphate pathway flux enhances recombinant protein production in Pichia pastoris'. In: *Applied Microbiology and Biotechnology* 100.13 (2016), pp. 5955–5963.

[283] L. Phégnon, J. Pérochon, S. Uttenweiler-Joseph, E. Cahoreau, P. Millard and F. Létisse. '6-Phosphogluconolactonase is critical for the efficient functioning of the Pentose phosphate pathway'. In: *The FEBS Journal* 291.20 (2024), pp. 4459–4472.

[284] L. Wang, I. Birol and V. Hatzimanikatis. 'Metabolic control analysis under uncertainty: framework development and case studies'. In: *Biophysical journal* 87.6 (2004), pp. 3750–3763.

[285] H. Li, W. Ma, W. Wang, S. Gao, X. Shan and J. Zhou. 'Synergetic engineering of multiple pathways for de novo (2S)-naringenin biosynthesis in Saccharomyces cerevisiae'. In: *ACS Sustainable Chemistry & Engineering* 12.1 (2023), pp. 59–71.

[286] A. Radev, J. Casado Gómez-Pallete, S. Monsalve Duarte, K. Faust and H. Zafeiropoulos. 'muGrowthDB: querying, visualizing, and sharing microbial growth curve data'. In: *bioRxiv* (2025), pp. 2025–10.

[287] P. van Lent, S. M. Paz, J. Schmitz and T. Abeel. 'Comparing metabolic engineering scenarios using simulated design-build-test-learn-cycles'. In: *Frontiers in Bioengineering and Biotechnology* (2026), p. 1802948.

[288] J. S. Cho, G. B. Kim, H. Eun, C. W. Moon and S. Y. Lee. 'Designing microbial cell factories for the production of chemicals'. In: *Jacs Au* 2.8 (2022), pp. 1781–1799.

[289] J. Tickner, K. Geiser and S. Baima. 'Transitioning the chemical industry: the case for addressing the climate, toxics, and plastics crises'. In: *Environment: Science and Policy for Sustainable Development* 63.6 (2021), pp. 4–15.

[290] R. F. Beall, T. J. Hwang and A. S. Kesselheim. 'Pre-market development times for biologic versus small-molecule drugs'. In: *Nature Biotechnology* 37.7 (2019), pp. 708–711.

[291] L. Hägele, N. Trachtmann and R. Takors. 'The knowledge driven DBTL cycle provides mechanistic insights while optimising dopamine production in Escherichia coli'. In: *Microbial Cell Factories* 24.1 (2025), p. 111.

[292] J. Yang, R. G. Lal, J. C. Bowden, R. Astudillo, M. A. Hameedi, S. Kaur, M. Hill, Y. Yue and F. H. Arnold. 'Active learning-assisted directed evolution'. In: *Nature Communications* 16.1 (2025), p. 714.

[293] S. Roy, T. Radivojevic, M. Forrer, J. M. Marti, V. Jonnalagadda, T. Backman, W. Morrell, H. Plahar, J. Kim, N. Hillson *et al.* 'Multiomics data collection, visualization, and utilization for guiding metabolic engineering'. In: *Frontiers in bioengineering and biotechnology* 9 (2021), p. 612893.

[294] B. J. Bornstein, S. M. Keating, A. Jouraku and M. Hucka. 'LibSBML: an API library for SBML'. In: *Bioinformatics* 24.6 (2008), pp. 880–881.

[295] P. Van Hoek, J. P. Van Dijken and J. T. Pronk. 'Effect of specific growth rate on fermentative capacity of baker's yeast'. In: *Applied and environmental microbiology* 64.11 (1998), pp. 4226–4233.

[296] A. Hari and D. Lobo. 'Fluxer: a web application to compute, analyze and visualize genome-scale metabolic flux networks'. In: *Nucleic acids research* 48.W1 (2020), W427–W435.

[297] K. S. Vuoristo, A. E. Mars, J. V. Sangra, J. Springer, G. Eggink, J. P. Sanders and R. A. Weusthuis. 'Metabolic engineering of the mixed-acid fermentation pathway of Escherichia coli for anaerobic production of glutamate and itaconate'. In: *Amb Express* 5.1 (2015), p. 61.

[298] J. Qin, Y. J. Zhou, A. Krivoruchko, M. Huang, L. Liu, S. Khoomrung, V. Siewers, B. Jiang and J. Nielsen. 'Modular pathway rewiring of Saccharomyces cerevisiae enables high-level production of L-ornithine'. In: *Nature Communications* 6.1 (2015), p. 8224.

[299] Y.-J. Wang, J.-P. Huang, T. Tian, Y. Yan, Y. Chen, J. Yang, J. Chen, Y.-C. Gu and S.-X. Huang. 'Discovery and engineering of the cocaine biosynthetic pathway'. In: *Journal of the American Chemical Society* 144.48 (2022), pp. 22000–22007.

[300] G. Wu, Q. Yan, J. A. Jones, Y. J. Tang, S. S. Fong and M. A. Koffas. 'Metabolic burden: cornerstones in synthetic biology and metabolic engineering applications'. In: *Trends in biotechnology* 34.8 (2016), pp. 652–664.

[301] A. Ghavasieh and M. De Domenico. 'Diversity of information pathways drives sparsity in real-world networks'. In: *Nature Physics* 20.3 (2024), pp. 512–519.

[302] A. Kværnø. 'Singly diagonally implicit Runge–Kutta methods with an explicit first stage'. In: *BIT Numerical Mathematics* 44.3 (2004), pp. 489–502.

[303] O. Berger-Tal, J. Nathan, E. Meron and D. Saltz. 'The exploration-exploitation dilemma: a multidisciplinary framework'. In: *PloS one* 9.4 (2014), e95693.

[304] Z. Zi. 'Sensitivity analysis approaches applied to systems biology models'. In: *IET systems biology* 5.6 (2011), pp. 336–346.

[305] R. Van Rosmalen, S. Moreno-Paz, Z. Duman-Özdamar and M. Suarez-Diez. 'CFSA: Comparative flux sampling analysis as a guide for strain design'. In: *Metabolic Engineering Communications* 19 (2024), e00244.

[306] Y. Yang, K. Zha, Y. Chen, H. Wang and D. Katabi. 'Delving into deep imbalanced regression'. In: *International conference on machine learning*. PMLR. 2021, pp. 11842–11851.

[307] Y. I. Tepeli, M. de Wolf and J. P. Gonçalves. 'Metric-DST: Mitigating Selection Bias Through Diversity-Guided Semi-Supervised Metric Learning'. In: *arXiv preprint arXiv:2411.18442* (2024).

[308] M. Del Giudice. 'Effective dimensionality: A tutorial'. In: *Multivariate behavioral research* 56.3 (2021), pp. 527–542.

[309] J. McInerney, B. Lacker, S. Hansen, K. Higley, H. Bouchard, A. Gruson and R. Mehrotra. 'Explore, exploit, and explain: personalizing explainable recommendations with bandits'. In: *Proceedings of the 12th ACM conference on recommender systems*. 2018, pp. 31–39.

[310] J. L. Borges. *On Exactitude in Science. Borges JL Collected Fictions (trans. Andrew Hurley)*. 1999.

[311] A. Khodayari and C. D. Maranas. 'A genome-scale Escherichia coli kinetic metabolic model k-ecoli457 satisfying flux data for multiple mutant strains'. In: *Nature communications* 7.1 (2016), p. 13806.

[312] Z. Wu, S. J. Kan, R. D. Lewis, B. J. Wittmann and F. H. Arnold. 'Machine learning-assisted directed protein evolution with combinatorial libraries'. In: *Proceedings of the National Academy of Sciences* 116.18 (2019), pp. 8852–8858.

[313] T. Yu, H. Cui, J. C. Li, Y. Luo, G. Jiang and H. Zhao. 'Enzyme function prediction using contrastive learning'. In: *Science* 379.6639 (2023), pp. 1358–1363.

[314] E. S. Muckley, J. E. Saal, B. Meredig, C. S. Roper and J. H. Martin. 'Interpretable models for extrapolation in scientific machine learning'. In: *Digital Discovery* 2.5 (2023), pp. 1425–1435.

[315] R. T. Chen, Y. Rubanova, J. Bettencourt and D. K. Duvenaud. 'Neural ordinary differential equations'. In: *Advances in neural information processing systems* 31 (2018).

[316] *EU Artificial Intelligence Act | Up-to-date developments and analyses of the EU AI Act – – artificialintelligenceact.eu*. https://artificialintelligenceact.eu/. [Accessed 22-10-2025].

[317] M. T. Ribeiro, S. Singh and C. Guestrin. '" Why should i trust you?" Explaining the predictions of any classifier'. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144.

[318] S. Choudhury, M. Moret, P. Salvy, D. Weilandt, V. Hatzimanikatis and L. Miskovic. 'Reconstructing kinetic models for dynamical studies of metabolism using generative adversarial networks'. In: *Nature Machine Intelligence* 4.8 (2022), pp. 710–719.

[319] H. Mochão, P. Barahona and R. S. Costa. 'KiMoSys 2.0: an upgraded database for submitting, storing and accessing experimental data for kinetic modeling'. In: *Database* 2020 (2020), baaa093.

[320] M. E. Beber, M. G. Gollub, D. Mozaffari, K. M. Shebek, A. I. Flamholz, R. Milo and E. Noor. 'eQuilibrator 3.0: a database solution for thermodynamic constant estimation'. In: *Nucleic acids research* 50.D1 (2022), pp. D603–D609.

[321] J. J. Heijnen and P. J. Verheijen. 'Parameter identification of in vivo kinetic models: Limitations and challenges'. In: *Biotechnology journal* 8.7 (2013), pp. 768–775.

[322] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko *et al.* 'Highly accurate protein structure prediction with AlphaFold'. In: *nature* 596.7873 (2021), pp. 583–589.

[323] J. P. Folch, C. Tsay, R. Lee, B. Shafei, W. Ormaniec, A. Krause, M. van der Wilk, R. Misener and M. Mutny. 'Transition constrained Bayesian optimization via Markov decision processes'. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 88194–88235.

[324] T. M. Rosch, J. Tenhaef, T. Stoltmann, T. Redeker, D. Kosters, N. Hollmann, K. Krumbach, W. Wiechert, M. Bott, S. Matamouros *et al.* 'AutoBioTech: a versatile biofoundry for automated strain engineering'. In: *ACS Synthetic Biology* 13.7 (2024), pp. 2227–2237.

[325] A. Casini, F.-Y. Chang, R. Eluere, A. M. King, E. M. Young, Q. M. Dudley, A. Karim, K. Pratt, C. Bristol, A. Forget *et al.* 'A pressure test to make 10 molecules in 90 days: external evaluation of methods to engineer biology'. In: *Journal of the American Chemical Society* 140.12 (2018), pp. 4302–4316.

[326] C. J. Robinson, P. Carbonell, A. J. Jervis, C. Yan, K. A. Hollywood, M. S. Dunstan, A. Currin, N. Swainston, R. Spiess, S. Taylor *et al.* 'Rapid prototyping of microbial production strains for the biomanufacture of potential materials monomers'. In: *Metabolic Engineering* 60 (2020), pp. 168–182.

[327] A. H. Singh, B. B. Kaufmann-Malaga, J. A. Lerman, D. P. Dougherty, Y. Zhang, A. L. Kilbo, E. H. Wilson, C. Y. Ng, O. Erbilgin, K. A. Curran *et al.* 'An automated scientist to design and optimize microbial strains for the industrial production of small molecules'. In: *BioRxiv* (2023), pp. 2023–01.

[328] P. van Ammelrooy. *Ouders klagen ChatGPT aan vanwege zelfmoord van hun zoon — volkskrant.nl*. https://www.volkskrant.nl/ai/ouders-klagen-chatgpt-aan-vanwege-zelfmoord-van-hun-zoon~bf4d50d0/. [Accessed 25-11-2025].

[329] *Ruim 40 BN'ers en artsen slachtoffer van deepfakes over supplementen — rtl.nl*. https://www.rtl.nl/nieuws/binnenland/artikel/5539833/ruim-40-bners-en-artsen-slachtoffer-van-deepfakes-supplementen. [Accessed 25-11-2025].

[330] Telegraaf. https://www.telegraaf.nl/binnenland/media-willen-aandacht-voor-gevaar-van-dominante-rol-techbedrijven/106342969.html. [Accessed 25-11-2025].

[331] A. KhokharVoytas, M. Shahbaz, M. F. Maqsood, U. Zulfiqar, N. Naz, U. Z. Iqbal, M. Sara, M. Aqeel, N. Khalid, A. Noman *et al.* 'Genetic modification strategies for enhancing plant resilience to abiotic stresses in the context of climate change'. In: *Functional & integrative genomics* 23.3 (2023), p. 283.

[332] *China jails 'gene-edited babies' scientist for three years — bbc.com*. https://www.bbc.com/news/world-asia-china-50944461. [Accessed 25-11-2025].

[333] *Africa And The Monopolization Of GM Technology - The Yale Review Of International Studies — yris.yira.org*. https://yris.yira.org/column/africa-and-the-monopolization-of-gm-technology/. [Accessed 25-11-2025].

[334] L. Asveld, P. Osseweijer and J. A. Posada. 'Societal and ethical issues in industrial biotechnology'. In: *Sustainability and life cycle assessment in industrial biotechnology* (2019), pp. 121–141.

[335] M. Stokhof. *Wittgensteins Betekenis*. Athenaeum - Polak and van Gennep, Amsterdam, 2025, pp. 147–155.

[336] P. Anastas and N. Eghbali. 'Green chemistry: principles and practice'. In: *Chemical society reviews* 39.1 (2010), pp. 301–312.

[337] B.-M. Cosma, R. Shirali Hossein Zade, E. N. Jordan, P. van Lent, C. Peng, S. Pillay and T. Abeel. 'Evaluating long-read de novo assembly tools for eukaryotic genomes: insights and considerations'. In: *GigaScience* 12 (2023), giad100.

[338] S. Pillay, R. S. H. Zade, P. van Lent, D. Calderón-Franco and T. Abeel. 'Metagenomic analysis of antibiotic resistance across the wastewater process'. In: *Heliyon* 11.5 (2025).

[339] S. Pillay, Y. Tepeli, P. van Lent and T. Abeel. 'A Metagenomic Study of Antibiotic Resistance Across Diverse Soil Types and Geographical Locations'. In: *bioRxiv* (2024), pp. 2024–09.

[340] E. N. Jordan, R. Shirali Hossein Zade, S. Pillay, P. van Lent, T. Abeel and O. Kayser. 'Integrated omics of Saccharomyces cerevisiae CENPK2-1C reveals pleiotropic drug resistance and lipidomic adaptations to cannabidiol'. In: *npj Systems Biology and Applications* 10.1 (2024), p. 63.

Acknowledgments

Vier jaar onderzoek is op zichzelf mooi om te doen, maar het zou toch een legere ervaring zijn zonder alle vriendelijke en behulpzame mensen die je tijdens die vier jaar tegenkomt. Ik ben vele mensen dankbaar die direct of indirect hebben bijgedragen aan de totstandkoming van dit proefschrift.

First, I want to thank my promoters, **Thomas** and **Marcel**, and my supervisor **Joep**. **Thomas**, as my promoter and daily supervisor, thank you so much for being a great manager of the PhD process, the very pleasant and amusing update meetings, the professional advice, and of course the nice Christmas dinner parties at your house. It was always a lot of fun to share food with each other and I hope you continue with this tradition. **Marcel**, although we did not have too many meetings together, I appreciate all the scientific input during the AI4b.io meetings as well as other moments and had a lot of fun discussing with you. **Joep**, I am extremely grateful for all the interesting discussions we have had on metabolic engineering, general career advice, and the art of beer brewing. I appreciate the time you took to explain and help me during the research and your willingness to bet with me on progress promises I made during update meetings. Unfortunately for me, you were typically more realistic than I was.

I would like to specifically acknowledge some of the people who have also contributed to this thesis. **Rianne**, I am very grateful for all the work you have performed in the lab, as well as the discussions on experimental designs and follow-up experiments. Without you and **Moniek**, this thesis would have been just a large computational exercise without grounding in experiment. **Léon**, thank you for the weekly meetings that helped me to professionalize the code. I have learned a lot about writing (somewhat) acceptable code thanks to you. To the bachelor students that have contributed to this thesis, **Bálint** and **Olga**, thank you for putting in the effort to make figures and text. I have learned a lot from you both and wish you all the best in your future career and life. Sara, thank you for the brainstorming sessions on the benchmarking project. Finally, I would like to also thank the members of the thesis committee for their feedback on the thesis to improve the work.

For my DBL colleagues, I appreciate the openness and supportiveness that you have fostered and I am happy to have spent four years with you. If I forget to put your name below and you feel like it should be there, please confront me aggressively after the defense. For the members of the Abeellab, thank you for all your nice talks and awkward group discussions during the Monday meetings; **Chengyao**, I have enjoyed sharing the

office with you and talking with you, especially when this meant being invited to eat food from Lychee or some illegal kitchen; **Ramin**, thank you for all your help during the initial stages of my PhD. You have a great sense of humor; **Stephanie**, I always really enjoyed hanging out in your office. I feel like you and Bianca were sort of bullying me, but I probably deserved it; **Bianca**, please get rid of this steamer, it can not be good for your health. Other than that, I wish you all the best; **Erin**, thank you for giving your interesting insights from Dortmund during all the group meetings. Whenever I did not feel like performing any work, I could always go to the Joana office to bother people; **Ivan**, thank you for your interesting scientific insights and for challenging my unfunded opinions on Jazz; **Sara**, please choose to feel safe. I still owe you a pack of _ ; **Roy**, thanks for the fun conversations during the Thursday borrels; **Yasin**, I appreciate your positive role in the DBL group and willingness to organise events to make everyone feel comfortable; **Sander**, we somewhat missed each other at the end of my contract, but it is always very nice to talk to you and I wish you all the best for the end of your PhD; **Colm**, I really enjoyed talking with you about martial arts and general nonsense. I am inspired by how you dealt with all the adversity you had to go through during your PhD and am very happy that you are my paranymph. Stop calling me a racist though; **Gabriel**, It is always a lot of fun to hang out with you and especially to hear about your very cool research. I am very happy that you accepted to be a paranymph for my defense and hope that we keep in contact afterwards; **Swier**, I enjoyed all the conversations with you that were on a superficial level deeply philosophical. I still think we should build a physical neural network. I wish you all the best and hope you defend soon too; **Jasper**, I love your enthusiasm about math and found your way of approaching problems very interesting. It was nice to have someone in the group that challenges AI bullshit consistently; **Gerard**, thank you for all the nice conversations and for sharing your work with me. **Kirti**, thank you for the chats and your kindness; **Lorenzo**, thank you very much for always being so extremely friendly all the time and for your good sense of humor. I am not being hyperbolic if I say that you are probably one of the most positive people I have ever met; **Daan**, it was really nice that you were there for a while in the DBL group, since we already know each other from the Master. I hope that we can keep this borrel thing alive for a while; **Bram**, you are a fun person and I appreciate that you always try to improve the group for everyone; **Amelia**, thank you for all the coffee talks we had together after five, especially when talking about future careers. They always felt reassuring and I wish you all the best in your new job; **Madaleon**, thank you again for the gladiolen during the Vierdaagse; **Niek**, I wish you all the best and hopefully you can run a sub three hour marathon at some point (without seeing ghosts); **Alex**: we did not speak a lot, but wish you all the best with your research. **Timo**, I wish you all the best with your research and your band. I still need to go to one of your concerts; **Inez**, it was really nice to have a train buddy at the end. I wish you had started a few years earlier. **Lieke**, thank you for the “gezelligheid” when you were still in the office. I hope you are enjoying your time in New York; **Marunka**, thank you for all your help during the PhD and with the organization of the AI4b.io symposium; **Ruud**, thank you for all the help whenever I accidentally ruined my computer in unforeseeable ways. I hope that you enjoy your retirement; **Azza**, thank you for helping me with all the (stupid) questions I had about the HPC.

I also want to express my gratitude to all the AI4b.io colleagues that we have had many scientific interactions with over the years. **Kim**, **Mahdi**, **Joery** and **Chengyao**, it was really nice to hear all of your interesting science during these meetings and the chance to borrel together afterwards at De Gist; **Renger**, thank you very much for the great organization and for all the sea socks that I have been able to gather over the years thanks to you. Jana, as the lab manager, thanks for all the work you have put in to making this a success and for all the interesting, constructive and fun conversations. I hope you will not be too critical during my defense, because I cannot take these words back after printing.

Dan wil ik nog graag vrienden en familie bedanken. Lieve ouders, bedankt voor het leven, dat jullie me naar school hebben gestuurd en natuurlijk de steun in de afgelopen vier jaar. **Rik**, aangezien je als huisgenoot het meest hebt moeten lijden onder mijn PhD bestaan, hierbij bedankt dat we later konden eten vanwege de late treinen die ik moest nemen om de spits te vermijden en voor het uithouden van mijn gezwets over wetenschap. **Rebecca**, bedankt voor al je steun, zeker in de laatste maanden.

Curriculum Vitæ

Paul Harrie VAN LENT

01-04-1997	Born in Beneden Leeuwen, the Netherlands.

Education

2009-2015	Voorbereidend Wetenschappelijk Onderwijs Dominicus College Nijmegen, the Netherlands
2015-2019	BSc Biology Utrecht University Utrecht, the Netherlands
2019-2021	MSc Bioinformatics and Systems Biology Vrije Universiteit/Universiteit van Amsterdam, Amsterdam, the Netherlands
2021-2025	Doctoral candidate Technische Universiteit Delft *Thesis*: Machine learning for iterative strain optimization *Promoters*: prof. dr. ir. Marcel Reinders & dr. Thomas Abeel *Supervisors*: dr. Joep Schmitz & dr. Thomas Abeel Delft, the Netherlands

Additional courses

2022	Microbial Physiology and Fermentation Technology BiotechDelft Delft, the Netherlands
2024	Oxford Machine Learning Summer School Oxford University Oxford, England, United Kingdom

List of Publications

Publications pertaining to this thesis

1. P. van Lent, J. Schmitz and T. Abeel. 'Simulated design–build–test–learn cycles for consistent comparison of machine learning methods in metabolic engineering'. In: *ACS Synthetic Biology* 12.9 (2023), pp. 2588–2599
2. P. van Lent, O. Bunkova, B. Magyar, L. Planken, J. Schmitz and T. Abeel. 'Jaxkineticmodel: Neural ordinary differential equations inspired parameterization of kinetic models'. In: *PLOS Computational Biology* 21.7 (2025), e1012733
3. P. van Lent, R. van der Hoek, S. M. Paz, I. Kooi, M. Jonkers, P. Zwartjens, J. Schmitz and T. Abeel. 'Machine Learning-Assisted Pathway Optimization in Large Combinatorial Design Spaces: a p-Coumaric Acid Case Study'. In: *bioRxiv* (2025), pp. 2025–06
4. P. van Lent, S. M. Paz, J. Schmitz and T. Abeel. 'Comparing metabolic engineering scenarios using simulated design-build-test-learn-cycles'. In: *Frontiers in Bioengineering and Biotechnology* (2026), p. 1802948

Other publications

1. B.-M. Cosma, R. Shirali Hossein Zade, E. N. Jordan, P. van Lent, C. Peng, S. Pillay and T. Abeel. 'Evaluating long-read de novo assembly tools for eukaryotic genomes: insights and considerations'. In: *GigaScience* 12 (2023), giad100
2. S. Pillay, R. S. H. Zade, P. van Lent, D. Calderón-Franco and T. Abeel. 'Metagenomic analysis of antibiotic resistance across the wastewater process'. In: *Heliyon* 11.5 (2025)
3. S. Pillay, Y. Tepeli, P. van Lent and T. Abeel. 'A Metagenomic Study of Antibiotic Resistance Across Diverse Soil Types and Geographical Locations'. In: *bioRxiv* (2024), pp. 2024–09
4. E. N. Jordan, R. Shirali Hossein Zade, S. Pillay, P. van Lent, T. Abeel and O. Kayser. 'Integrated omics of Saccharomyces cerevisiae CENPK2-1C reveals pleiotropic drug resistance and lipidomic adaptations to cannabidiol'. In: *npj Systems Biology and Applications* 10.1 (2024), p. 63